Springer Protocols

Methods in Molecular Biology 696

Data Mining in Proteomics

From Standards to Applications

Edited by
Michael Hamacher
Martin Eisenacher
Christian Stephan

Humana Press

METHODS IN MOLECULAR BIOLOGY™

Series Editor
John M. Walker
School of Life Sciences
University of Hertfordshire
Hatfield, Hertfordshire, AL10 9AB, UK

For other titles published in this series, go to
www.springer.com/series/7651

Data Mining in Proteomics

From Standards to Applications

Edited by

Michael Hamacher

Lead Discovery Center GmbH, Dortmund, Germany

Martin Eisenacher
and
Christian Stephan

Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany

Humana Press

Editors
Dr. Michael Hamacher
Lead Discovery Center GmbH
Dortmund
Germany
hamacher@lead-discovery.de

Dr. Martin Eisenacher
Medizinisches Proteom-Center
Ruhr-Universität Bochum
Bochum
Germany
martin.eisenacher@rub.de

Dr. Christian Stephan
Medizinisches Proteom-Center
Ruhr-Universität Bochum
Bochum
Germany
christian.stephan@rub.de

ISSN 1064-3745 e-ISSN 1940-6029
ISBN 978-1-60761-986-4 e-ISBN 978-1-60761-987-1
DOI 10.1007/978-1-60761-987-1
Springer New York Dordrecht Heidelberg London

Humana Press is part of Springer Science+Business Media (www.springer.com)

Preface

Inspired by the enormous impact of Genomics and the hopes that came along with it, biochemistry and its methods slowly evolved into what is now widely known as Proteomics. Scientists dedicated to mass spectrometry and gel-based technologies became aware of the powerful tools they hold in hand, dreaming of the quantitative analyses of proteins in cells, tissues, and diseases. Thus, Proteomics soon went from a shooting-star in the life science field to a must-have in each larger wet-lab group.

Methods and technology developed rapidly, often much faster than the awareness of the special needs of the tools in use and even faster than standard protocols and standard formats could mature. Soon proteomics techniques created more and more data, while meaningful approaches for data handling, interpretation, and exchange sometimes were clearly behind, resulting in misinterpreted studies and frustrated colleagues from time to time.

However, the know-how generated and experiences made especially in the last several years caused a rethinking of strategy design and data interpretation. Moreover, the elaboration of standards by such voluntarily driven groups as Proteomics Standards Initiative within the Human Proteome Organisation or the US institutions, Institute of Systems Biology (ISB), and National Institute of Standards and Technology (NIST), ushered in a new era of understanding and quality, proving how powerful Proteomics is when the technology can be controlled through data generation, handling, and mining.

This book reflects these new insights within the Proteomics community, taking the historical evolution as well as the most important international standardization projects into account so that the reader gets a feeling for the dynamism and openness in this field. Basic and sophisticated overviews are given in regard to proteomics technologies, standard data formats, and databases – both local laboratory databases and public repositories. There are chapters dealing with detailed information concerning data interpretation strategies, including statistics, spectra interpretation, and analysis environments. Other chapters describe the HUPO initiatives or are about more specialized tasks, such as data annotation, peak picking, phosphoproteomics, spectrum libraries, LC/MS imaging, and splice isoforms. This volume also includes in-depth description of tools for data mining and visualization of Proteomics data, leading to modeling and Systems Biology approaches. To look beyond the Proteomics tasks and challenges, some chapters present insights into protein interaction network evolution, text mining, and random matrix approaches.

All in all, we believe that this book is a well-balanced compendium for beginners and experts, offering a broad scope of data mining topics but always focusing on the current state-of-the-art and beyond. Enjoy!

Dortmund, Germany — *Michael Hamacher*
Bochum, Germany — *Martin Eisenacher*
Bochum, Germany — *Christian Stephan*

Contents

Contributors

Philip C. Andrews • *Departments of Biological Chemistry, Bioinformatics and Chemistry, University of Michigan, Ann Arbor, MI, USA*

Hubert Bahl • *Division of Microbiology, Institute of Biological Sciences, University of Rostock, Rostock, Germany*

Chris Bauer • *MicroDiscovery GmbH, Berlin, Germany*

Andreas Bertsch • *Division for Simulation of Biological Systems, WSI/ZBIT, Eberhard-Karls-Universität Tübingen, Tübingen, Germany*

Martin Blüggel • *Protagen AG, Dortmund, Germany*

Erich Bornberg-Bauer • *Bioinformatics Division, Institute for Evolution and Biodiversity, School of Biological Sciences, University of Muenster, Münster, Germany*

Jean-Pierre Both • *Commissariat à l'Énergie Atomique, Saclay, France*

Frederic Brosseron • *Department of Functional Proteomics, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*

Daniel Chamrad • *Protagen AG, Dortmund, Germany*

Piotr Chartowski • *Department of Functional Proteomics, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*

Richard Côté • *European Molecular Biology Laboratory, European Bioinformatics Institute, Cambridge, UK*

Rainer Cramer • *The BioCentre and Department of Chemistry, The University of Reading, Whiteknights, Reading, UK*

Eric W. Deutsch • *Institute for Systems Biology, Seattle, WA, USA*

Martin Eisenacher • *Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*

Ralf-Jörg Fischer • *Division of Microbiology, Institute of Biological Sciences, University of Rostock, Rostock, Germany*

Daniel Fulger • *Department of Chemistry and WZMW, Computer Simulation Group, Philipps-University Marburg, Marburg, Germany*
Complex Systems Lagrange Lab, Institute for Scientific Interchange, Torino, Italy

Simon J. Gaskell • *Michael Barber Centre for Mass Spectrometry, School of Chemistry, Manchester Interdisciplinary Biocentre, University of Manchester, Manchester, UK*

Mark A. Gjukich • *Departments of Biological Chemistry, Bioinformatics and Chemistry, University of Michigan, Ann Arbor, MI, USA*

Clemens Gröpl • *Division for Simulation of Biological Systems, WSI/ZBIT, Eberhard-Karls-Universität Tübingen, Tübingen, Germany*

Bernd Gröttrup • *Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*

Joel B. Hagen • *Department of Biology, Radford University, Radford, VA, USA*

Jari Häkkinen • *Department of Oncology, Clinical Sciences, Lund University, Lund, Sweden*
Michael Hamacher • *Lead Discovery Center GmbH, Dortmund, Germany*
Sarah R. Hart • *Michael Barber Centre for Mass Spectrometry, School of Chemistry, Manchester Interdisciplinary Biocentre, University of Manchester, Manchester, UK Institute for Science and Technology in Medicine/School of Medicine, Keele University, Staffordshire, UK*
Ron M. A. Heeren • *FOM Institute for Atomic and Molecular Physics, Amsterdam, The Netherlands*
Henning Hermjakob • *European Molecular Biology Laboratory, European Bioinformatics Institute, Cambridge, UK*
Alfons Hester • *Justus Liebig University, Giessen, Germany*
James A. Hill • *Departments of Biological Chemistry, Bioinformatics and Chemistry, University of Michigan, Ann Arbor, MI, USA*
Simon J. Hubbard • *Faculty of Life Sciences, University of Manchester, Manchester, UK*
Philip Jones • *European Molecular Biology Laboratory, European Bioinformatics Institute, Cambridge, UK*
Klaus Jung • *Department of Medical Statistics, Georg-August-University Göttingen, Göttingen, Germany*
Ivo Klinkert • *FOM Institute for Atomic and Molecular Physics, Amsterdam, The Netherlands*
Michael Kohl • *Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*
Oliver Kohlbacher • *Division for Simulation of Biological Systems, WSI/ZBIT, Eberhard-Karls-Universität Tübingen, Tübingen, Germany*
Gerhard Körting • *Protagen AG, Dortmund, Germany*
Axel Kowald • *Protagen AG, Dortmund, Germany*
Kyung-Hoon Kwon • *Korea Basic Science Institute, Deajeon, Republic of Korea*
King Wai Lau • *Faculty of Life Sciences, Michael Barber Centre for Mass Spectrometry, School of Chemistry, Manchester Interdisciplinary Biocentre, University of Manchester, Manchester, UK*
Fredrik Levander • *Department of Immunotechnology and CREATE Health Strategic Centre for Translational Cancer Research, Lund University, Lund, Sweden*
Frédérique Lisacek • *Proteome Informatics Group, Swiss Institute of Bioinformatics, Geneva, Switzerland*
Katrin Marcus • *Department of Functional Proteomics, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*
Lennart Martens • *European Molecular Biology Laboratory, European Bioinformatics Institute, Cambridge, UK*
Caroline May • *Department of Functional Proteomics, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*
Rajasree Menon • *Center for Computational Medicine and Biology and National Center for Integrative Biomedical Informatics, University of Michigan,*

Ann Arbor, MI, USA
Helmut E. Meyer • *Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*
Thomas Millat • *Systems Biology & Bioinformatics, Institute of Computer Science, University of Rostock, Rostock, Germany*
Markus Müller • *Proteome Informatics Group, Swiss Institute of Bioinformatics, Geneva, Switzerland*
Gilbert S. Omenn • *Departments of Medicine and Genetics, Center for Computational Medicine and Bioinformatics, Medical School and School of Public Health, University of Michigan, Ann Arbor, MI, USA*
Sandra Orchard • *European Molecular Biology Laboratory, European Bioinformatics Institute, Cambridge, UK*
Patricia M. Palagi • *Proteome Informatics Group, Swiss Institute of Bioinformatics, Geneva, Switzerland*
Young Mok Park • *Korea Basic Science Institute, Daejeon, Republic of Korea*
Knut Reinert • *Division for Simulation of Biological Systems, WSI/ZBIT, Eberhard-Karls-Universität Tübingen, Tübingen, Germany*
Florian Reisinger • *European Molecular Biology Laboratory, European Bioinformatics Institute, Cambridge, UK*
Andreas Römpp • *Justus Liebig University, Giessen, Germany*
Enrico Scalas • *Department of Advanced Sciences and Technology, Laboratory on Complex Systems, University of East Piedmont Amedeo Avogadro, Alessandria, Italy*
Christoph Schaab • *Kinaxo Biotechnologies GmbH, Martinsried, Germany*
Max Planck Institute of Biochemistry, Martinsried, Germany
Sebastian Schmeier • *South African National Bioinformatics Institute, University of the Western Cape, Bellville, South Africa*
Bodo Schoenebeck • *Department of Functional Proteomics, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*
Thorsten Schramm • *Justus Liebig University, Giessen, Germany*
Johannes Schuchhardt • *MicroDiscovery GmbH, Berlin, Germany*
Andreas Schüler • *Bioinformatics Division, School of Biological Sciences, Institute for Evolution and Biodiversity, University of Muenster, Münster, Germany*
Cornelia Schumbrutzki • *Department of Functional Proteomics, Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*
Bryan E. Smith • *Departments of Biological Chemistry, Bioinformatics and Chemistry, University of Michigan, Ann Arbor, MI, USA*
Bernhard Spengler • *Justus Liebig University, Giessen, Germany*
Christian Stephan • *Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*
Markus Stöckli • *Novartis Institutes for BioMedical Research, Basel, Switzerland*
Michael Turewicz • *Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*
Juan Antonio Vizcaíno • *European Molecular Biology Laboratory, European Bioinformatics Institute, Cambridge, UK*
Daniel Walther • *Proteome Informatics Group, Swiss Institute of Bioinformatics,*

Geneva, Switzerland
Bettina Warscheid • *Clinical & Cellular Proteomics, Medical Faculty and Center for Medical Biotechnology, Duisburg-Essen University, Essen, Germany*
Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany
Sebastian Wiese • *Medizinisches Proteom-Center, Ruhr-Universität Bochum, Bochum, Germany*
Olaf Wolkenhauer • *Systems Biology & Bioinformatics, Institute of Computer Science, University of Rostock, Rostock, Germany*

Part I

Data Generation and Result Finding

Chapter 1

Instruments and Methods in Proteomics

Caroline May, Frederic Brosseron, Piotr Chartowski, Cornelia Schumbrutzki, Bodo Schoenebeck, and Katrin Marcus

Abstract

In the past decade, major developments in instrumentation and methodology have been achieved in proteomics. For proteome investigations of complex biological samples derived from cell cultures, tissues, or whole organisms, several techniques are state of the art. Especially, many improvements have been undertaken to quantify differences in protein expression between samples from, e.g., treated vs. untreated cells and healthy vs. control patients. In this review, we give a brief insight into the main techniques, including gel-based protein separation techniques, and the growing field of mass spectrometry.

1. Introduction

The proteome describes the quantitative expression of genes within, e.g., a cell, a tissue, or body fluid at specific time points and under defined circumstances (1). In contrast to the genome, the proteome is highly dynamic and the protein expression pattern of cells in an organism varies depending on the physiological functions, differentiation status, and environmental factors. In addition, alternative splicing of mRNAs and a broad range of posttranslational modifications (e.g., phosphorylation, glycosylation, and ubiquitination) increase proteome complexity (2, 3). Transcription analysis also does not allow insight into degradation and transport phenomena, alternative splicing, or posttranslational modifications. Furthermore, mRNA and protein levels often do not correlate (4, 5). All these influences are unconsidered in genome analysis and underline the importance of proteome analysis to obtain deeper insights into cellular functions.

In general, proteome analysis provides a snap-shot of proteins expressed in a cell or tissue at a defined time point (1). Indeed, not only qualitative analysis resulting in a defined "protein inventory"

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_1,

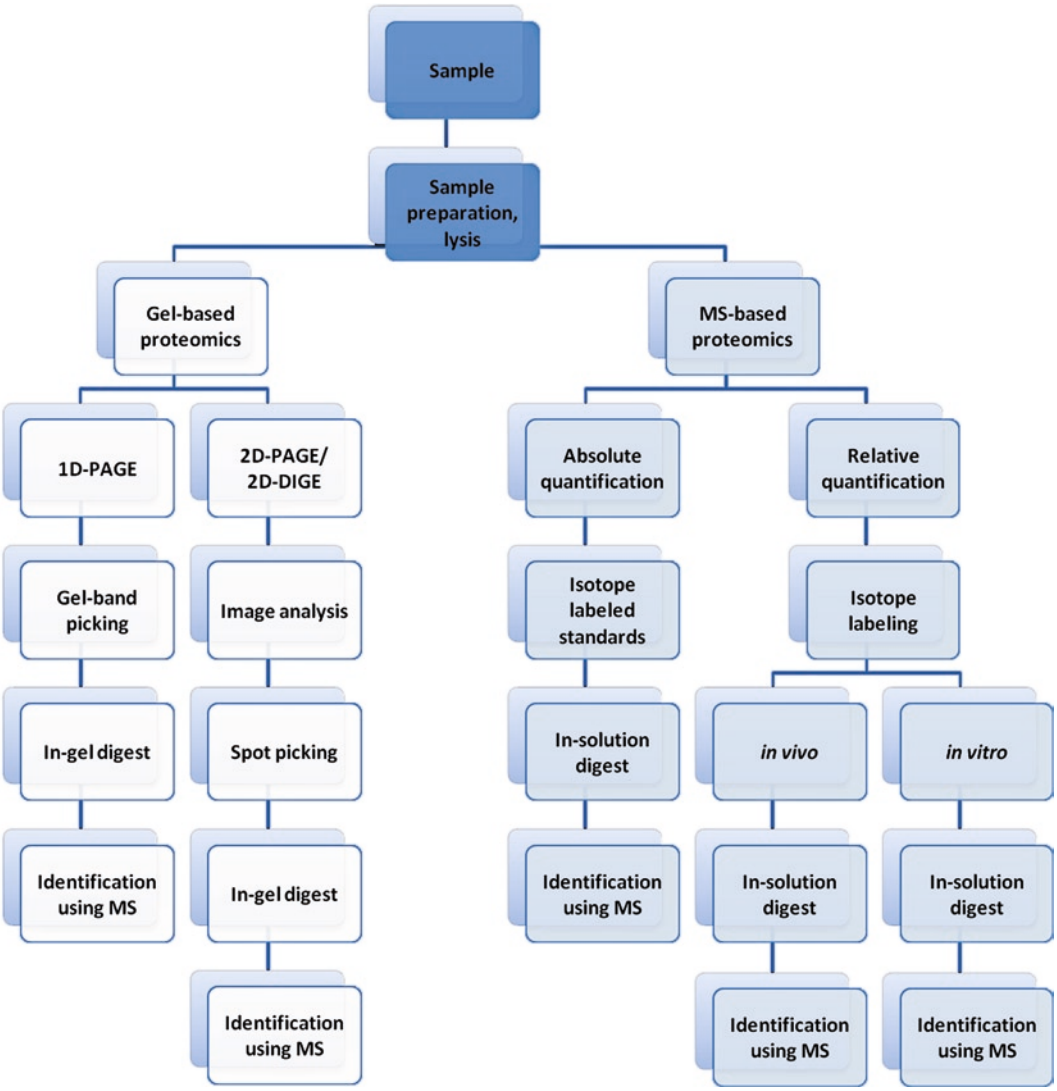


Fig. 1. *General workflow for proteomics.* Several different methods and technologies exist today which can be combined in order to achieve best results for a given scientific question. Most commonly used techniques and strategies are presented in the following chapters. *MS* mass spectrometry; *1D-PAGE* one-dimensional protein separation; *2D-PAGE* two-dimensional protein separation; *2D-DIGE* two-dimensional difference in gel electrophoresis.

can be obtained, but differential proteome analysis also allows for the detection of distinct differences in protein expression. This is of implicit interest, e.g., in the fields of fundamental and clinical research in order to understand main cellular functions and physiological/pathophysiological processes. For proteome investigation of complex biological samples derived from cell cultures, tissues, or whole organisms, several techniques were developed over the last decade, the most important of which are reviewed in the following paragraphs. Figure 1 gives a general overview of different workflows in proteomics.

2. Gel-Based Protein Separation Techniques and Applications

Gel-based approaches belong to the most frequently used assays in proteomics to separate proteins and to analyze them qualitatively and quantitatively. For simple pre-separation of complex protein mixtures before mass spectrometric analysis, one-dimensional polyacrylamide gel electrophoresis (1D-PAGE) is often used. Additionally, two-dimensional approaches such as two-dimensional polyacrylamide gel electrophoresis (2D-PAGE) allow for the separation of up to 10,000 protein species (6), providing the potential for global differential proteome analysis. Different gel-based methods especially differing in their respective resolution and application in proteomics are summarized in the following sections.

2.1. One-Dimensional Protein Separation: 1D-PAGE

One-dimensional polyacrylamide gel electrophoresis, according to Lämmli, with s*odium dodecyl sulfate* (SDS) as negative-charge detergent (7) is widely used for the separation of proteins according to their electrophoretic mobility. Due to SDS binding, the proteins are denaturated showing identical charge per unit protein mass which after the application of an electric field results in fractionation by size (see Fig. 2). High mass proteins will be retained longer by the polyacrylamide network than smaller proteins. After visualization by one of several existing staining methods, protein identification can easily be performed by mass spectrometry (MS) (see Subheading 3). The resolution of 1D-PAGE in contrast to that of 2D-PAGE is (see Subheading 2.2) rather low since the proteins are separated only according to their molecular mass. Nevertheless, 1D-PAGE is often used to achieve a pre-separation prior to MS or for the detection of proteins by subsequent Western blotting.

2.2. Two-Dimensional Protein Separation: 2D-PAGE

Two-dimensional polyacrylamide gel electrophoresis was developed in order to obtain higher resolved protein patterns than obtained using 1D-PAGE, offering a huge potential to give a comprehensive overview of the proteins present in the examined system. 2D-PAGE is a combination of two orthogonal separation techniques: in the first dimension, the proteins are separated according to their isoelectric point (Isoelectric Focusing: IEF), followed by a conventional SDS-PAGE in the second dimension. For IEF, two different techniques are described, namely, the carrier-ampholyte (CA)-based (8, 9) and immobilized pH gradient (IPG) system (10, 11). The spot pattern can be visualized with several protein staining methods, which differ in sensitivity and dynamic range. For differential proteome analysis, spot patterns of related gels are compared with each other and protein species can be relatively quantified automatically using one of several

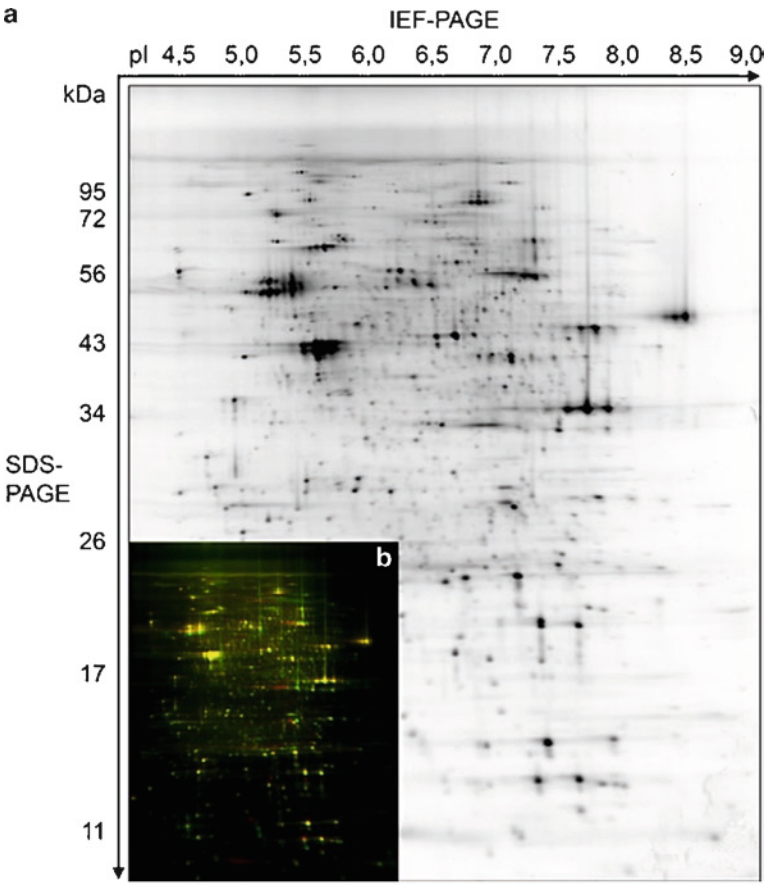


Fig. 2. *2D-IEF/SDS-PAGE of SH-SY5Ycells*. The proteins of an SH-SY5Y cell lysate were separated according to their isoelectric point in the first dimension (isoelectric focusing) and to their electrophoretic mobility in the second dimension (SDS PAGE). After 2D-PAGE, protein spots were visualized with silver staining.

available image analysis software tools (12). Differentially expressed proteins of interest are subsequently identified by MS (see Subheading 3). One drawback of 2D-PAGE is the fact that mainly hydrophilic proteins with a molecular weight of 5–150 kDa in a pH range of 3.5–10 can be analyzed. Especially hydrophobic/membrane proteins are underrepresented and must be analyzed with alternative gel-based methods such as 2D-benzyldimethyl-n-hexadecylammonium chloride (BAC)/SDS (13), 2D-cetyltrimethylammonium bromide (CTAB)/SDS (14, 15), SDS/SDS (16), and BlueNative-PAGE (17), or MS-based strategies (see below). Nevertheless, in combination with image analysis and MS, 2D-PAGE is still the method of choice to analyze complex protein samples. For more detailed description of 2D-PAGE, see Marcus et al. (18) and Rabilloud et al. (19).

2.2.1. 2D-DIGE: A Sophisticated Application

The invention of *two-dimensional difference in-gel electrophoresis* (2D-DIGE) in 1997 drastically improved the technical reproducibility of 2D-PAGE and the accurate quantification of different proteins in samples with high statistical significance (18, 20). Proteins of different samples are covalently labeled with spectrally resolvable fluorescent dyes (CyDyes™, GE Healthcare Europe GmbH) and afterwards separated simultaneously on the same gel. The application of an internal standard, optimally consisting of a mixture of all samples included in the study, allows accurate matching and normalization of protein spots in all gels, and with this highly accurate quantification (21). Two methodologies can be distinguished: CyDye™ minimal labeling and CyDye™ saturation labeling. For minimal labeling, dyes react with the ε-amino group of lysine residues. Three to five percent of all proteins and only one lysine per protein on average are labeled. Three different dyes are available: Cy™2, Cy™3, and Cy™5. Saturation labeling allows for the analysis of scarce protein samples down to an amount of 3 μg per gel (15, 22). The label reacts with thiol groups of cysteine residues. All cysteine residues of all proteins are labeled. In this technique, two different dyes are available, Cy™3 and Cy™5.

Protein patterns are digitalized using confocal fluorescent imagers, resulting in a gel image at a specific wavelength for each dye without any crosstalk. Appropriate analysis software allows for automated detection, background subtraction, quantification, normalization, and inter-gel matching.

3. Mass Spectrometry-Based Techniques and Applications

Similar to gel-based protein separation, MS is one of the most popular techniques in proteomics (23–25). In MS, the chemical compounds of a sample are ionized and the resulting charged molecules (ions) are analyzed according to their mass-to-charge (m/z) ratios. In proteomics, the molecules of interest are either proteins or peptides obtained from enzymatic digestion of proteins. MS can be used for the identification of either the peptides or the proteins, as well as for the quantification of the measured ion species. Up to date, several different MS setups and assays have been developed for use in proteome studies. Each of them has its own advantages and disadvantages, and is used for characteristic purposes, comprising identification of proteins from 2D-gel spots, description of peptides with chemical modifications, and quantitative MS assays (18, 26–29). The following chapters illustrate the most important aspects of MS in proteomics and their characteristic applications.

3.1. Setup of a Mass Spectrometer

In general, a mass spectrometer consists of the following components: ion source, mass analyzer, and detector (30). The ion source is used to create protein or peptide ions usually by transferring positive charged protons (H^+) onto the molecules. The ionization is called "soft" because the chemical structure of the proteins or peptides remains unharmed. One or more mass analyzers are used to separate the ions by their m/z ratio or to fragment the ions for further sequence analysis. At last, the ions are passed to a detector connected to a PC with appropriate software for data analysis. Modern software tools include control programs for all parts of the mass spectrometer setup. Optional to this setup is the use of a chromatography system (widely HPLC) upstream of the ion source to reduce sample complexity (see Fig. 3). All hardware components are described in more detail in the following chapters.

3.1.1. Liquid Chromatography Techniques for Proteome Analysis

Different types of liquid chromatography (LC) are used in proteomics to complement gel-based separation techniques (29). The basic principle of LC is to separate solute analytes (e.g., proteins or peptides) in a fluid that flows over solid particles. The solution is referred to as the mobile phase, while the particles are termed the stationary phase. Depending on their differing chemical and physical properties, different analyte species will interact in different ways with both phases. Usually, the stationary phase is packed into a column through which the mobile phase flow is led. This way, the analytes separate over time until they elute from the column. The time point in which a peptide elutes is called its retention time (RT). The amount of analytes eluting over the time is usually documented as a chromatogram by UV detectors. Different variants of LC systems each make use of special properties of the analytes of interest, e.g., polarity or chemical functional groups. It is common to use LC for protein purification or

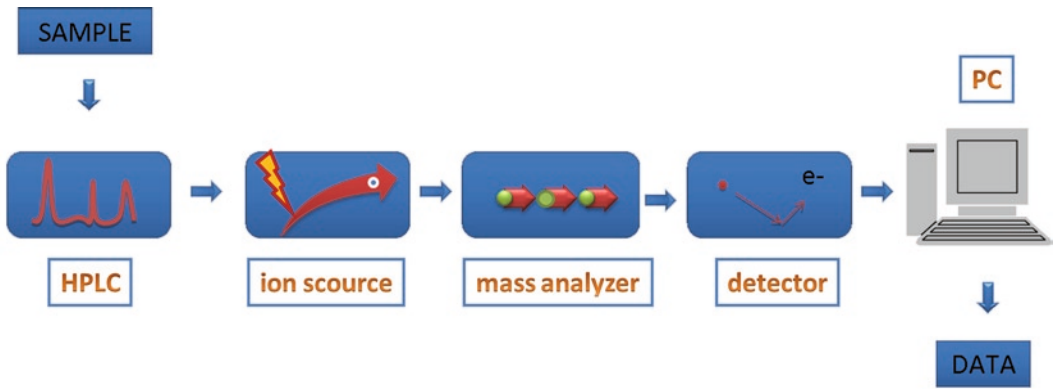


Fig. 3. *Setup of mass spectrometers.* A typical mass spectrometer for proteomic purposes will be set up in the following way: high-performance liquid chromatography (HPLC) (optional), ion source, mass analyzer, detector, and personal computer. See the following chapters for details on hardware configuration.

fractionation as one of the first steps in a proteome study. Nevertheless, peptides are more homogenous in size and polarity than proteins, and are thus better suited for chromatographic separation and analysis. Therefore, LC is a powerful tool to reduce the complexity of peptide samples, e.g., digested protein bands from 1D-gels or whole cell lysates (31). It is also used for the separation of less complex samples, such as 2D-gel spots.

A major advantage of LC is the possibility to automate the separation progress. Modern automated systems can cover the whole separation progress, beginning with the loading of the sample onto the column up to the MS analysis of the eluted analytes (mostly peptides). This combination is referred to as LC–MS. Automation allows complex and elongated gradients of mobile phase composition as well as the combination of several columns with different stationary phases in one analysis. An example for such sophisticated LC systems is the multi-dimensional protein identification technology (MuDPIT) (32). The peptide solution is separated first by strong cation exchange (SCX) with a pH gradient, followed directly by reversed phase (RP) chromatography using hydrophobic C18 material as the stationary phase and a polar solution of water with increasing amount of organic compounds (33). MuDPIT runs can be prolonged to 12 or even more hours to increase their separation power.

Another advantage of LC is the possibility of nano-size applications with increased sensitivity. In nano-high pressure liquid chromatography (nano-HPLC), the mobile phase is pumped through capillary columns (34). The columns contain porous nonpolar particles serving as a hydrophobic solid phase with which the peptides can interact. The mobile phase is a polar fluid consisting mostly of a mixture of water, organic compounds such as acetonitrile, and low amounts of acids. For this reason, this type of HPLC is referred to as RP-HPLC. Usually, the amount of acetonitrile in the mixture is increased over the time of analysis following an automated gradient. As a result, hydrophilic peptides will elute first from the capillary column, followed by other peptides depending on their increasing hydrophobicity. Nano-HPLC is a very common proteomics method because even short runs (between 1 and 3 h) can be used to separate complex samples. Additionally, it is possible to couple the chromatography system either directly ("online") or indirectly ("offline") with a mass spectrometer for subsequent MS analysis of the eluting peptides. In online LC–MS, the nano-HPLC system is connected directly with an electrospray ionization (ESI) ion source (see Subheading 3.1.2). This is possible because ESI requires liquid samples, which means the solution eluting from the nano-HPLC can be led directly into the ion source. Offline LC–MS establishes the connection between nano-HPLC and matrix-assisted laser desorption ionization (MALDI), which is another common

ionization technique that requires samples in solid (crystallized) state (see Subheading 3.1.2). For this purpose, automated fractionators spot small amounts of liquid eluting from the nano-HPLC onto steel plates ("targets") suitable for MALDI ion sources (31).

One drawback of offline nano-LC–MALDI–MS in comparison to online LC–ESI–MS is a longer analysis time. Indeed, spotted samples can be stored for some time, allowing for a re-investigation of the samples (for more details, see (29)).

3.1.2. Ionization Methods

In principle, two main ionization methods are used in proteomics today, MALDI and ESI (23). In *MALDI*, the sample molecules are immobilized by co-crystallization in the presence of organic compounds such as alpha-cyano-4-hydroxycinnamic acid or 2,5-dihydroxybenzoic acid on a metal sample target (35). By administering laser energy to the samples, the matrix ions partially transfer their charge on the analyte molecules, producing mainly single-charged peptide ions. Since the pulsed laser operates rather in "shots" than continuously, MALDI is used primarily in combination with time of flight (TOF) analyzers (36). This combination is termed as MALDI-TOF, which is used in proteomics for analysis of proteins and peptides (37–39).

ESI is another well-suited ionization method for biomolecules such as peptides (23). Like MALDI, ESI is a "soft" method of ionization producing charged peptides in solution (40). ESI requires liquid samples which are delivered either by direct injection with a syringe or "online" coupled with a (nano)-RP-HPLC system. The sample passes a capillary needle on which voltage is applied. As a result, charged droplets are generated at the capillary tip. The solvent partially evaporates, resulting in the reduction of the droplets' diameter and enhanced density of charges. The rising charge density leads to the so-called coulomb explosions which further reduce the diameter of the droplets. Hence, the analytes are dispersed as a fine spray (41, 42). Different mechanisms have been discussed to describe the ESI process, which all end up with the fact that gas-phase ions are generated (43, 44). The ions are subsequently detected by the mass analyzer. One of the major advantages of ESI for proteomics is the possibility to separate highly complex peptide mixtures upstream by nano-HPLC, e.g., resulting from whole cell lysates.

In general, both ionization techniques described above can be combined with different types of mass analyzers. Depending on the application desired, each combination is characterized by typical features such as enhanced mass accuracy, sensitivity, dynamic range, or resolving power. Therefore, for best performance, mass spectrometer setups favorable for, e.g., identification, quantification, high throughput analyses, or detection of modifications should differ from each other (for a comprehensive overview, see Domon and Aebersold (36)).

3.1.3. Types of Mass Analyzers and Hybrid Mass Spectrometers

Independent of the ionization technique, the molecular masses of free ions are measured in mass analyzers after passing them through a vacuum chamber. Different types of analyzers are often combined in a so-called hybrid mass spectrometer (24, 36). After the ions pass the analysis system, the detector measures the m/z ratios of all incoming ions and transfers this information to a computer. Most common in proteomics are TOF analyzers, different types of ion traps, and high-resolution analyzers such as Fourier transform ion cyclotron resonance (FT-ICR) or the latest development, the orbitrap.

In *TOF analyzers*, ions are accelerated by a potential between two electrodes (45). The analyzer itself is merely a vacuum tube. Ions with different masses pass the vacuum chamber with different velocities. By measuring the time the ions need until they reach the detector, the m/z ratio is calculated. TOF analyzers can reach resolutions of up to 15,000 full-width half-height maximum (fwhm) with a mass accuracy of up to 2 ppm (36, 45, 46). In Q-Q-TOF instruments, two quadrupoles (Qs) are combined with a TOF analyzer. In the MS mode, the quadrupole serves as a guide for the ions toward the mass analyzer. In the MS/MS mode, where detailed peptide information is gained, the precursor ions are selected in the first quadrupole and subsequently fragmented in the second quadrupole. This setup results in a high mass accuracy and high resolution of selected precursor ions (36).

In a *quadrupole (Q) analyzer*, ions accelerated by strong electric fields pass a set of stab electrodes arranged in cylindrical constellation (47, 48). Between the stab electrodes, an alternating electric field ensures that only ions of a defined mass can pass. In this way, the quadrupole acts as a mass filter. Furthermore, ions can be trapped in the electric fields for fragmentation. Quadrupoles are most common as parts of hybrid instruments, e.g., for focusing of the ion beam emitted from the ion source on the way to another mass analyzer with better resolution, like an orbitrap (49, 50). In addition, combinations of quadrupoles with TOF analyzers or as parts of FT-ICR mass spectrometers occur. Triple quadrupole (Q-Q-Q) instruments became more and more important in proteomics research. With the arrangement of three quadrupoles or two quadrupoles followed by a linear ion trap (LIT), new scanning methods such as product ion scanning, parent ion scanning (51, 52), neutral loss scanning (53, 54), and multiple reaction monitoring (55) (see Subheading 3.2) became feasible. All these scanning methods commonly use concomitant mass analyzers serving as a combination of mass filters and collision cells to enhance the sensitivity of a subset of ions one aims to analyze.

In "ion trap" (IT) analyzers, ions are trapped and get accumulated over a given time in a physical device. *Nonlinear ITs* were first described by Paul *et al.* (56). The IT itself consists of two adversely arranged hyperbolic electrodes with a ring electrode

between them. This setup is used to establish dynamic electric fields in all three dimensions, which allows focusing of incoming ions in the center of the trap. From this point on, the ions can be selectively ejected and passed to the detector, or can be fragmented. This is usually done by collision-induced dissociation (CID) and/or electron transfer dissociation (ETD) (see Subheading 3.2), combined with the activation of the ions induced by resonance to the changing electric fields (57). A detailed description of theory, instrumentation, and working modes can be found in ref. (58–62).

Linear ion traps function as mass filters and simultaneously act as a storage device for specific ions. Ions that possess a defined m/z range can be trapped and stored before they are further passed through the detector. This is conducted by four electrode rods in a quadrupolar orientation describing a combination of alternating and co-current flows. Ions that reside within the adjusted m/z range oscillate through the drifting channel, whereas all other ions describe unstable flight paths and, therefore, get stopped by collision with the electrodes. During the scanning of the mass field, both co-current (U) and alternating current (V) are simultaneously enhanced. With the change of this U/V ratio during the scan, the mass range of stable oscillation becomes shifted, resulting in a mass separation (49). LITs have the advantage of increased ion storage capacity compared to non-linear ion traps, leading to a higher sensitivity and dynamic range. In general, IT technology is characterized by MS/MS capabilities with unmatched sensitivity and fast data acquisition. Indeed, limited resolution, low-ion trapping capacities, and space-charging effects result in low accuracy of the mass measurements.

Fourier transform ion cyclotron resonance mass spectrometers are ITs with an additional homogeneous magnetic field (63, 64). The magnetic field forces ions into a circular path in which they cycle with high frequency, the so-called cyclotron circle frequency. By adding a changing electric field perpendicular to the magnetic field, a resonance between the ion mass and the cyclotron circle frequency is built up. In this process, energy is consumed from the changing electric field. This energy shift can be measured and transformed into m/z ratios by Fourier transformation. FT-ICR spectrometers reach high-resolution mass accuracy of up to 1.0 ppm (65). Nevertheless, FT-ICR spectrometers are less common than other types because of their high operation expenses.

The last important development in the field of mass analyzers was attained by the *Orbitrap* (66, 67). This type consists of a single, spindle-shaped electrode. In this setup, ions move on circuits around the electrode and oscillate along the axis at the same time. The frequency of this oscillation is dependent on the masses and charges of the respective ions. On this basis, m/z can be calculated by Fourier transformation. Orbitrap analyzers reach

resolutions and accuracies similar to those of FT-ICR analyzers combined with significantly lower operation expenses. For this reason, Orbitrap instruments become increasingly popular in proteome analysis (68).

3.2. Identification of Proteins by Mass Spectrometry: Scanning Methods and Fragmentation Types

Mass spectrometry can be used for whole protein mass and peptide mass determination as well as peptide fragmentation analysis. Peptide fragmentation analysis became the most popular application over the years as it allows obtaining information not only about the mass and charge of a protein or peptide ion, but also on its chemical composition. Different main scanning methods suitable for peptide mass and peptide fragmentation analysis can be distinguished, which are *peptide mass fingerprinting* (PMF) (69), post-source decay (PSD) (70), tandem-MS (also called MS/MS or MS^2), product ion scanning (24, 36, 71), *neutral loss* (NL) *scanning* (53, 54), *precursor ion scanning* (PIS) (52, 72, 73), and *multiple reaction monitoring* (MRM) (36, 55, 74).

Peptide mass fingerprinting or peptide mass mapping is based on the fact that digestion of a protein by enzymes will result in a specific mixture of peptides. When analyzed with a mass spectrometer, the peptide mixture will lead to a characteristic pattern of m/z values, the PMF. By comparing the PMF with databases, it is possible to identify the corresponding protein (75). This makes PMF ideally suitable for the identification of proteins from low complex mixtures, e.g., 2D gel spots using MALDI-TOF MS (24).

If the number of peptides for PMF analysis is not sufficient or the complete genome sequence of the analyzed species is unknown, fragmentation analysis can be performed for a more detailed and specific analysis.

PSD, tandem-MS (MS/MS, MS^2): The fragmentation of the peptide can be induced by metastable decay (PSD) (70), CID (76), or ETD (57). CID is an older, but still common technique that uses neutral gas molecules such as helium, nitrogen, or argon to transfer kinetic energy on the peptide ions, leading to fragmentation. In ETD, this is achieved by using fluoranthene radicals as electron donors that destabilize peptide ions by transferring the electron on them. ETD leads to different fragments than CID (see spectra interpretation). While CID is still the state of the art, especially for sequencing of peptide ions, ETD and combinations of both methods have become important when analyzing posttranslational modifications such as phosphorylation or glycosylation (77–80) PSD analysis is restricted to MALDI-TOF/(TOF) instruments, whereas tandem-MS (MS/MS, MS^2) analysis can be done on different types of instruments such as ITs, Q-Q-Qs, or Orbitraps. During MS fragmentation analysis, peptide ions are automatically selected for fragmentation, resulting in predictable breakdown products. These fragment ions are recorded by the detector and give rise to the so-called PSD or tandem-MS (MS/MS, MS^2) spectra.

To date, the most common applications in proteomics use MS^2 spectra without further fragmentation for protein identification. This is due to the fact that generally samples in proteomics are analyzed after digestion of the proteins to peptides, and the resulting MS^2 spectra are sufficient for identification of the peptides. For detailed analyses of fragment ions, especially detection of posttranslational modifications, further fragmentations can be performed, resulting in MS^n spectrometry (81, 82). Basically, the next described scanning modes are specialized MS/MS applications for Q-Q-Q instruments which are used to enhance the selectivity and sensitivity for the measurement of a subset of ions.

Product ion scanning is the most common method for sequencing peptide ions generally on Q-Q-Q instruments (24, 36, 71). This scan determines, in a single experiment, all peptide (parent) m/z ratios that react to produce a selected product (daughter) ion m/z ratio. In Q-Q-Qs, one peptide of a specified m/z is selected in Q1 as a parent ion. In the next step, the parent ion is fragmented in Q2. All resulting fragment ions are subsequently scanned in Q3. Usually, several parent ions of different m/z ratios are sequentially analyzed by stepwise alteration of the quadrupole field in Q1 in one MS run in this way. New developments in MS instrumentation today allows for product ion scanning with specialized hybrid-TOF such as Q-TOF or TOF-TOF instruments.

Converse to the product ion scan, the *PIS* is a scan that determines, in a single experiment, all the product (daughter) ion m/z ratios that are produced by the reaction of a selected peptide (parent) ion m/z ratio. Parent ions of the whole mass range are transferred through Q1 and fragmented in Q2. Q3 is then fixed on a single fragment ion mass, filtering for pre-specified fragment ions selectively produced by the parent ions (73). This scanning method can be especially useful for the selective detection (and quantification) of posttranslational modifications such as glycosylation or phosphorylation (83, 84).

Another selective scanning mode especially useful for the detection of protein/peptide phosphorylation or glycosylation is *NL scanning* verifying the loss of a neutral particle from a fragmented parent ion (24, 85). Similar to PIS, in NL scanning, parent ions of the whole mass range are transferred through Q1 and fragmented in Q2. Q3 is not fixed on a special fragment mass but operates synchronously to Q1 scanning for a defined mass shift between precursor and fragment ion. In other words, only fragment ions that differ from their parent ion by a characteristic mass difference will reach the detector. Because the charge of the peptide ion does not change, this was designated as a neutral loss. NL scanning and PIS can be combined with product ion scanning for sequencing of the modified peptide ions.

Multiple reaction monitoring is one special application in proteome analysis allowing for the targeted detection (and quantification) of

pre-selected peptides in a complex peptide mixture. MRM analysis can be performed on Q-Q-Qs as well as on Q-hybrids such as Q-Q–LIT instruments (74, 86). In MRM (or single/selected reaction monitoring, SRM), Q1 serves as a mass filter for the selection of ions of a defined m/z ratio (Q1). Selected parent ions are fragmented in Q2 and pre-defined fragment ions are specifically detected in Q3. The combination of pre-defined m/z ratios in Q1 and Q3, representing the precursor and a characteristic fragment ion, is called an MRM transition. Thus, MRM differs from the other scan types in the way that two pre-requisites have to be fulfilled in order to produce a signal in the detector: both ions, precursor and related fragment ion, need to be specifically measured in one scan. This makes the MRM scan highly specific even for low abundant peptide ions in complex mixtures. MRM can be used for all kinds of hypothesis-driven approaches where a specified protein/peptide of interest should be identified or even quantified (relatively or absolutely), e.g., in a complex protein mixture (87).

3.3. MS-Data Interpretation

All kinds of MS and MS/MS analyses result in the generation of the so-called raw data. These raw data containing information about the peptide masses and, in case of MS/MS data, also fragment ion masses and their intensities are transformed to a "peak list." Identification of the peptide/protein is performed by using a search engine such as MASCOT (88) or Sequest (89) to search the peak list against a database of proteins "digested in silico," meaning that the practically obtained MS and MS/MS data are directly compared with theoretically generated data from protein databases. Knowledge about sample preparation and separation conditions, type and mass accuracy of the mass spectrometer, and mode of peptide fragmentation (90) allows for a reliable peptide assignment (88, 89, 91). Typically, the algorithms give a probability value for the correctness of the identification. The peptides assigned should be unique for a protein species in order to annotate the analyzed spectrum clearly to only one protein. This kind of data analysis is possible only in cases where the genome of the investigated organism is sequenced and a database is available. Otherwise, *de novo* sequence analysis needs to be performed entailing manual interpretation and annotation of the MS/MS spectra in order to obtain sufficient information on the peptides' sequence.

4. Quantitative Mass Spectrometry

Due to the described disadvantages of gel-based differential proteome analysis (see Subheading 2.2), over the last years worldwide efforts have led to the development of MS-based

quantification methods. The fundamental idea with this was to shift the separation as well as quantification problem from protein to peptide level as peptides are much easier to handle than proteins due to their physic-chemical characteristics. Today, several MS-based quantification methods, including chemical, metabolic, enzymatic labeling, and label-free approaches ranging from the quantification of single peptides up to the quantification of proteins from whole cell lysates, exist that can be used as an alternative or complementary setup to 2D-PAGE for analyzing complex protein and/or peptide mixtures. They include methods for relative and absolute quantification such as label-free approaches (see Subheading 4.1.1); isotope labeling, e.g., isotope-coded affinity tags (ICAT) (92), isotope-coded protein labeling (ICPL) (93), isobaric tags for relative and absolute quantification (iTRAQ, TMT) (94), enzymatic labeling during protein hydrolysis in the presence of heavy (^{18}O-containing) water (95, 96), and stable isotope labeling with amino acids in cell culture (SILAC) (97); and absolute quantification of proteins (AQUA) (98, 99). For a general overview, see (28, 29, 100, 101). All the listed methods hold their advantages and disadvantages. Global internal standard (GIST) approaches where proteins are digested to peptides prior to labeling hold two major limitations: the high sample complexity results in the detection and quantification of only a limited number of peptides (undersampling of the mass spectrometer), and by protein digestion prior to labeling, all information about the original belonging to the resulting peptide is lost. For protein-based chemical labeling, the main limitation is the incomplete labeling of the proteins resulting in falsified results. Today, the most accurate results are obtained with SILAC; this method is indeed mainly restricted to cells grown in culture and simple organisms.

4.1. Relative Quantification

In the next two chapters, most frequently used methods for MS-based relative protein/peptide quantification are described shortly.

4.1.1. Isotope Labeling

Labeling of proteins or peptides with isotopes or other kinds of reagents distinguishable by MS is the most common strategy for gel-free protein quantification in proteomics. It is a universal approach as labeling is done after protein extraction. Over the years, several strategies have been developed which each suit different needs. Usually, they are used for "shotgun" experiments starting directly on peptide level using LC–MS for separation, quantification, and sometimes even identification in one step. It is to be noted that these parameters depend much on the capabilities of the mass spectrometer used. Disadvantages of isotope labeling include cost expensiveness and the possibility of incomplete labeling. Most of the state-of-the-art labeling chemistries are summarized by Julka and Regnier (100).

4.1.1.1. Chemical Labeling

As the first method using isotopic labels for quantitative MS, the ICAT or cleavable ICAT (cICAT) was invented by Aebersold and co-workers in 1999 (92). The reagent with specificity toward side chains of cysteinyl residues consists of three elements: first, a reactive group toward thiol groups (cysteines); second, a linker containing either ^{12}C (light ICAT) or ^{13}C(heavy ICAT) atoms; and third, a biotin group that can be used for affinity purification before MS analysis. To quantify protein expression levels, e.g., of two different cell states, the protein mixture of the first cell state is labeled with light ICAT and the protein mixture of the second is labeled with the heavy ICAT. After pooling of both samples, they are enzymatically digested to peptides, separated with HPLC, and analyzed via MS. The light or heavy ICAT-modified peptides co-elute in HPLC and can be easily distinguished from each other by a 9-Da mass shift. The relative quantification is determined by the ratio of the peptide pairs (102). The main drawback is that ICAT cannot be used to quantify all proteins due to the fact that the number of proteins containing cysteines is restricted and only limited sequence coverage of the protein can be reached (28). As a result, information about protein isoforms, degradation products, or posttranslational modifications, which are not located in the cysteine-containing peptide, are lost.

The techniques *isobaric tags for relative and absolute quantification* (*iTRAQ*) and tandem mass tagging (TMT) were first introduced by Ross and Thompson, respectively (94, 103). Either protein or peptide labeling can be performed on lysine residues and/or the N-terminus. To date, eight different iTRAQ with eight different isobaric (same mass) mass tags, and six TMT reagents are available, allowing for multiplexing of samples. Isobaric peptides hold the advantage of identical migration properties in the HPLC before MS analysis. Quantification is done after peptide fragmentation by the generation of label-specific low molecular weight reporter ion and signal integration. The different tags can be distinguished after peptide fragmentation as they result in different mass spectra. Therefore, this method allows the simultaneous determination of both identity and relative abundance of the peptide species (104, 105). iTRAQ and TMT can also be used for absolute quantification. Indeed, both methods hold the described limitations of GIST approaches. Additionally, iTRAQ/TMT quantification cannot be obtained on all kinds of mass spectrometers as low molecular mass reporter ion region is not accessible in all instruments.

Isotope-coded protein labeling is based on isotopic labeling of all free amino groups in proteins (93). Proteins from two different samples are extracted, alkylated, and labeled with either the isotope-free ICPL (light) or the isotope ICPL tag (heavy). After labeling, the protein mixtures are combined, optionally separated, e.g., by 1D-PAGE to reduce complexity, enzymatically digested,

and subsequently analyzed by MS (93). The heavy and light peptides differ in mass, and are visible as doublets in the mass spectra. Again, the peak intensities reflect relative quantitative information of the original proteins. The main advantage of this approach is the labeling already on protein level, circumventing all described limitations of the GIST approaches, although it holds the risk of incomplete protein labeling.

Enzymatic labeling with heavy water ($^{16}O/^{18}O$ method) uses the fact that during protein digestion with trypsin, Glu-C or Lys-C up to two O atoms are incorporated into the peptide. Thus, digestion in the presence of $H_2{}^{18}O$ results in a peptide mass shift of 4 Da compared to that in peptides generated during digestion in the presence of normal $H_2{}^{16}O$. In a workflow using the $^{16}O/^{18}O$ method, the samples are independently digested in the presence of either $H_2{}^{16}O$ or $H_2{}^{18}O$, and the samples are pooled and separated by HPLC, followed by peptide quantification and identification. This method is relatively simple; indeed, it holds the risk of back exchange of the O atoms and does not allow for multiplexing.

4.1.1.2. Metabolic Labeling

Stable isotope labeling by amino acids in cell culture (SILAC) is a metabolic labeling based on the in vivo incorporation of specific amino acids into mammalian proteins (106). For example, mammalian cells are grown up in a medium with normal essential amino acids (light label) and concomitantly in a medium with isotopic modified forms of essential amino acids (heavy label). After some proliferation cycles, the isotopic/normal amino acids incorporate completely into the cells. Protein extracts can be pooled, digested, and analyzed by MS. The heavy and light peptides elute as peak pairs separated by a defined mass difference. The ratios of the resulting relative peak intensities reflect the abundances of each measured peptide (107). Mainly, the isotopes ^{13}C, ^{15}N, ^{2}H, and ^{18}O are used for stable isotope labeling. The incorporation of the isotopes in proteins can be performed in cell culture and even in vivo in simple organisms such as *Drosophila melanogaster*, *Caenorhabditis elegans*, or mice (107, 108). For higher organisms, especially humans, this kind of metabolic labeling is technically not feasible or completely impossible due to ethical reasons.

4.1.2. Label-Free Quantification

To overcome the limitations of incomplete labeling, and also to spare costs and to reduce loss of proteins in the cause of sample preparation, label-free MS approaches have been developed (101, 109). One disadvantage of label-free quantification indeed is that this technique does not allow multiplexing, and has a slight lack of sensitivity compared to labeling assays. Nevertheless, label-free approaches offer the opportunity to analyze samples with a

protein amount that would be too low for labeling or 2D-DIGE strategies, since they omit many preparation steps.

In *spectral counting*, the number of mass spectra repeatedly measured for one protein serves as a value for quantitation of this ion (109, 110). It could be shown that this number is proportional to the concentration of a peptide in a sample when analyzed by nano-LC–MS (111). This is due to the fact that the higher the concentrations of a peptide, the longer it will take to elute from the HPLC system. Modern mass spectrometers can produce several MS^2 spectra in the time interval the peptides need to completely elute and be ionized by ESI. Disadvantages of spectral counting rise from the complexity of biological samples: Even with the best available LC system, co-eluting of peptides will still occur when analyzing complex mixtures such as cell lysates. Mass spectrometers will not be able to identify all co-eluting peptides at once. As a consequence, several replicated LC–MS runs will be needed to reach maximum identification results from one sample (111). This also leads to the second disadvantage of spectral counting that quantitative information can be obtained only from the peptides chosen as precursors, while information on less intensive or unselected peptides will be lost. Nevertheless, spectral counting is a cost-sparing alternative to labeling assays taking into account that this approach seems to be accurate, especially for high abundance proteins, but is highly sensitive to run-to-run variations (normalization is mandatory!).

One of the latest quantitative MS methods that is still under development is *comparative or differential LC–MS* (112). This method utilizes the ability of mass spectrometers to record not only m/z and the intensity of the MS signal, but also RT. Softwares use these data to build contour plots in the form of heat maps, in which RT and m/z span up a plane, and MS intensity will be displayed in a color code (101). Quantitative information is obtained by integration of the volume of the m/z–RT intensity peaks. Software calculates the features which are the sum of all peaks generated by one peptide as quantitative factors. Special algorithms are used for normalization between the LC–MS runs. The advantage of this method is that it does not need any MS^2 spectra for quantitation, with the result that all signals recorded in one LC–MS run will be quantified. This could become the main advantage of comparative LC–MS, as the quantitative information should be more extensive than in spectral counting. Indeed, spectral counting still has advantages in sensitivity and reproducibility (109). A major disadvantage of comparative LC–MS is that it allows no multiplexing and thus is more sensitive for run-to-run variations than labeling methods. Nevertheless, some studies report successful use of comparative LC–MS methods (example given by Johansson (113)).

Intensive effort is spent currently to improve label-free quantification approaches, especially with respect to reproducibility, data analysis, and statistics.

4.2. Absolute Quantification

Over the last years, proteome research is more and more focused on the *Absolute quantification of proteins (AQUA)*. *AQUA* permits the direct quantification of differences in proteins and posttranslational modified protein expression levels (98). Therefore, chemically synthesized isotope peptides, which are unique for the proteins of interest, are used as internal standards by adding a known quantity to the analytical sample (114, 115). The ratio of synthetic to endogenous peptide is measured and the absolute level of the endogenous peptide can be precisely and quantitatively calculated and consequently the absolute levels of proteins and posttranslational modified proteins are known (98).

Although there are efforts to use MALDI, factors such as variable crystallization and laser ablation may lead to poor reproducibility, and thus generally ESI is the method of choice for AQUA (114). Before starting the AQUA approach, one has to adjust the peptide retention by RP chromatography, ionization efficiency, fragmentation via CID, and the amount of added standards to fit with the dynamic detection range of the mass spectrometer (see Gerber et al. for detailed information (98)). In a rather complex sample, the detection of the desired peptide likely competes with the detection of other isobaric peptides in the sample. This can be overcome by the combination of AQUA with MRM, allowing for a selective absolute quantification of the target protein (115). This technique is of considerable benefit for, e.g., the absolute quantification of known biomarkers. Other available approaches for absolute quantification based on internal standards are *QConCat* (116) and *protein standard for absolute quantification* (*PSAQ*) (117).

5. Summary

In the past decade, major developments in instrumentation and methodology have been achieved in proteomics. Powerful techniques have been established to identify and differentially quantify protein species of complex biological samples. Many proteomic laboratories are investigating new techniques to overcome consistent obstacles. Beyond alterations of the genome, the increasing advances in proteomics hold great promise for a comprehensive description of protein isoforms or even posttranslational modifications. With the ongoing improvement of sample preparation techniques and mass spectrometer sensitivities, the resolution of quantifiable compounds will be further improved in proteomics

research allowing for the identification and especially reliable quantification of, e.g., physiologically relevant biomarkers indicating specific disease states.

6. Notes

1. For the electrophoretic separation of membrane proteins, conventional 2D-PAGE is not suitable. For this purpose, the application of specialized gel-based gel techniques such as CTAB- or BAC-SDS-PAGE, or MS-based methods is highly recommended (15, 118, 119).
2. Whenever a labeling approach is chosen for quantitative proteomics, labeling limitations have to be considered. For example, a saturation DIGE approach in 2D-DIGE will enhance the sensitivity but only cysteine residues will be labeled. Since cysteines are not found in all proteins, information about these proteins is lost. Moreover, peptide labeling might be more efficient than protein labeling.
3. In order to rule out labeling preferences, a dye swap should be included in 2D-DIGE experiments. This can be performed by switching the labeling dyes of samples A and B in two consecutive experiments.
4. Protein differences between samples which have been found to be statistically valid in one technique need to be further validated by an independent method.
5. One has to consider that gel-based and MS-based techniques generally do not result in identical protein lists. Rather, both approaches complement each other. For a detailed and broad description of proteins within a sample, one may think about combining both approaches.

Acknowledgments

FB, PC, CS, BS, and KM are funded by the BMBF (grant 01 GS 08143). CM is supported by the Alma-Vogelsang Foundation.

References

1. Wilkins MR, Sanchez JC, Gooley AA, Appel RD, Humphery-Smith I, Hochstrasser DF, Williams KL (1996) Progress with proteome projects: why all proteins expressed by a genome should be identified and how to do it. Biotechnol Genet Eng Rev 13:19–50
2. Patterson SD, Aebersold RH (2003) Proteomics: the first decade and beyond. Nat Genet 33(Suppl):311–323
3. Pandey A, Mann M (2000) Proteomics to study genes and genomes. Nature 405:837–846

4. Gygi SP, Rochon Y, Franza BR, Aebersold R (1999) Correlation between protein and mRNA abundance in yeast. Mol Cell Biol 19:1720–1730
5. Anderson NL, Anderson NG (1998) Proteome and proteomics: new technologies, new concepts, and new words. Electrophoresis 19:1853–1861
6. Klose J, Kobalz U (1995) Two-dimensional electrophoresis of proteins: an updated protocol and implications for a functional analysis of the genome. Electrophoresis 16:1034–1059
7. Laemmli UK (1970) Cleavage of structural proteins during the assembly of the head of bacteriophage T4. Nature 227:680–685
8. Klose J (1975) Protein mapping by combined isoelectric focusing and electrophoresis of mouse tissues. A novel approach to testing for induced point mutations in mammals. Humangenetik 26:231–243
9. O'Farrell PH (1975) High resolution two-dimensional electrophoresis of proteins. J Biol Chem 250:4007–4021
10. Bjellqvist B, Ek K, Righetti PG, Gianazza E, Görg A, Westermeier R, Postel W (1982) Isoelectric focusing in immobilized pH gradients: principle, methodology and some applications. J Biochem Biophys Methods 6:317–339
11. Görg A, Postel W, Gunther S (1988) The current state of two-dimensional electrophoresis with immobilized pH gradients. Electrophoresis 9:531–546
12. Luhn S, Berth M, Hecker M, Bernhardt J (2003) Using standard positions and image fusion to create proteome maps from collections of two-dimensional gel electrophoresis images. Proteomics 3:1117–1127
13. MacFarlane DE (1989) Two dimensional benzyldimethyl-n-hexadecylammonium chloride – sodium dodecyl sulfate preparative polyacrylamide gel electrophoresis: a high capacity high resolution technique for the purification of proteins from complex mixtures. Anal Biochem 176:457–463
14. Eley MH, Burns PC, Kannapell CC, Campbell PS (1979) Cetyltrimethylammonium bromide polyacrylamide gel electrophoresis: estimation of protein subunit molecular weights using cationic detergents. Anal Biochem 92:411–419
15. Helling S, Schmitt E, Joppich C, Schulenborg T, Mullner S, Felske-Muller S, Wiebringhaus T, Becker G, Linsenmann G, Sitek B, Lutter P, Meyer HE, Marcus K (2006) 2-D differential membrane proteome analysis of scarce protein samples. Proteomics 6:4506–4513
16. Rais I, Karas M, Schägger H (2004) Two-dimensional electrophoresis for the isolation of integral membrane proteins and mass spectrometric identification. Proteomics 4:2567–2571
17. Schägger H, von Jagow G (1991) Blue native electrophoresis for isolation of membrane protein complexes in enzymatically active form. Anal Biochem 199:223–231
18. Marcus K, Joppich C, May C, Pfeiffer K, Sitek B, Meyer H, Stuehler K (2009) High-resolution 2DE. Methods Mol Biol 519:221–240
19. Rabilloud T, Vaezzadeh AR, Potier N, Lelong C, Leize-Wagner E, Chevallet M (2009) Power and limitations of electrophoretic separations in proteomics strategies. Mass Spectrom Rev 28:816–843
20. Unlu M, Morgan ME, Minden JS (1997) Difference gel electrophoresis: a single gel method for detecting changes in protein extracts. Electrophoresis 18:2071–2077
21. Alban A, David SO, Bjorkesten L, Andersson C, Sloge E, Lewis S, Currie I (2003) A novel experimental design for comparative two-dimensional gel analysis: two-dimensional difference gel electrophoresis incorporating a pooled internal standard. Proteomics 3:36–44
22. Sitek B, Luttges J, Marcus K, Kloppel G, Schmiegel W, Meyer HE, Hahn SA, Stuhler K (2005) Application of fluorescence difference gel electrophoresis saturation labelling for the analysis of microdissected precursor lesions of pancreatic ductal adenocarcinoma. Proteomics 5:2665–2679
23. Nyman TA (2001) The role of mass spectrometry in proteome studies. Biomol Eng 18:221–227
24. Aebersold R, Mann M (2003) Mass spectrometry-based proteomics. Nature 422: 198–207
25. Steen H, Mann M (2004) The ABC's (and XYZ's) of peptide sequencing. Nat Rev Mol Cell Biol 5:699–711
26. Wuhrer M, Deelder AM, Hokke CH (2005) Protein glycosylation analysis by liquid chromatography-mass spectrometry. J Chromatogr B Analyt Technol Biomed Life Sci 825: 124–133
27. Boersema PJ, Mohammed S, Heck AJ (2009) Phosphopeptide fragmentation and analysis by mass spectrometry, *J.* Mass Spectrom 44:861–878
28. Bantscheff M, Schirle M, Sweetman G, Rick J, Kuster B (2007) Quantitative mass spectrometry in proteomics: a critical review. Anal Bioanal Chem 389:1017–1031

29. Urlaub H, Gronborg M, Richter F, Veenstra TD, Müller T, Tribl F, Meyer HE, Marcus K (2008) Common methods in proteomics. In: Nothwang HG, Pfeiffer SE (eds) Proteomics of the nervous system, 1st edn. Weinheim, Wiley-VCH
30. Glish GL, Vachet RW (2003) The basics of mass spectrometry in the twenty-first century. Nat Rev Drug Discov 2:140–150
31. Mitulovic G, Mechtler K (2006) HPLC techniques for proteomics analysis - a short overview of latest developments. Brief Funct Genomic Proteomic 5:249–260
32. Washburn MP, Wolters D, Yates JR III (2001) Large-scale analysis of the yeast proteome by multidimensional protein identification technology. Nat Biotechnol 19:242–247
33. Nägele E, Vollmer M, Horth P, Vad C (2004) 2D-LC/MS techniques for the identification of proteins in highly complex mixtures. Expert Rev Proteomics 1:37–46
34. Chervet JP, Ursem M, Salzmann JP (1996) Instrumental requirements for nanoscale liqid chromatography. Anal Chem 68:1507–1512
35. Zaluzec EJ, Gage DA, Watson JT (1995) Matrix-assisted laser desorption ionization mass spectrometry: applications in peptide and protein characterization, Protein Expr Purif 6:109–123
36. Domon B, Aebersold R (2006) Mass spectrometry and protein analysis. Science 312:212–217
37. Karas M, Hillenkamp F (1988) Laser desorption ionization of proteins with molecular masses exceeding 10, 000 daltons. Anal Chem 60:2299–2301
38. Nordhoff E, Egelhofer V, Giavalisco P, Eickhoff H, Horn M, Przewieslik T, Theiss D, Schneider U, Lehrach H, Gobom J (2001) Large-gel two-dimensional electrophoresis-matrix assisted laser desorption/ionization-time of flight-mass spectrometry: an analytical challenge for studying complex protein mixtures. Electrophoresis 22: 2844–2855
39. Stuhler K, Meyer HE (2004) MALDI: more than peptide mass fingerprints. Curr Opin Mol Ther 6:239–248
40. Fenn JB, Mann M, Meng CK, Wong SF, Whitehouse CM (1989) Electrospray ionization for mass spectrometry of large biomolecules. Science 246:64–71
41. Loo JA, Udseth HR, Smith RD (1989) Peptide and protein analysis by electrospray ionization-mass spectrometry and capillary electrophoresis-mass spectrometry. Anal Biochem 179:404–412
42. Cech NB, Enke CG (2001) Practical implications of some recent studies in electrospray ionization fundamentals. Mass Spectrom Rev 20:362–387
43. Iribarne JV, Thomson BA (1976) On the evaporation of small ions from charged droplets. J Chem Phys 64:2287–2294
44. Dole M, Dole M, Mack LL, Mack LL, Hines RL, Hines RL, Mobley RC, Mobley RC, Ferguson LD, Ferguson LD, Alice MB, Alice MB (1968) Molecular Beams of Macroions. J Chem Phys 49:2240–2249
45. Wollnik H (1993) Time-of-flight mass analyzers. Mass Spectrom Rev 12:89–114
46. Balogh MP (2004) Debating resolution and mass accuracy in mass spectrometry. Spectroscopy 19:34–40
47. Schwartz JC, Senko MW, Syka JE (2002) A two-dimensional quadrupole ion trap mass spectrometer. J Am Soc Mass Spectrom 13:659–669
48. March RE (2000) Quadrupole ion mass spectrometry: a view at the turn of the century. Int J Mass Spectrom 200:285–312
49. Douglas DJ, Frank AJ, Mao D (2005) Linear ion traps in mass spectrometry. Mass Spectrom Rev 24:1–29
50. Hager JW (2002) A new linear mass spectromter. Rapid Commun Mass Spectrom 16:512–526
51. Wilm M, Neubauer G, Mann M (1996) Parent ion scans of unseparated peptide mixtures. Anal Chem 68:527–533
52. Steen H, Kuster B, Fernandez M, Pandey A, Mann M (2001) Detection of tyrosine phosphorylated peptides by precursor ion scanning quadrupole TOF mass spectrometry in positive ion mode. Anal Chem 73: 1440–1448
53. HunterAP,GamesDE(1994)Chromatographic and mass spectrometric methods for the identification of phosphorylation sites in phosphoproteins. Rapid Commun Mass Spectrom 8:559–570
54. Schlosser A, Pipkorn R, Bossemeyer D, Lehmann WD (2001) Analysis of protein phosphorylation by a combination of elastase digestion and neutral loss tandem mass spectrometry. Anal Chem 73:170–176
55. Yocum AK, Chinnaiyan AM (2009) Current affairs in quantitative targeted proteomics: multiple reaction monitoring-mass spectrometry. Brief Funct Genomic Proteomic 8:145–157
56. Busch FV, Paul W (1961) Isotopentrennung mit dem elektrischen. Massenfilter Zeitschrift für Physik 164:581–587

57. Mikesh LM, Ueberheide B, Chi A, Coon JJ, Syka JE, Shabanowitz J, Hunt DF (2006) The utility of ETD mass spectrometry in proteomic analysis. Biochim Biophys Acta 1764:1811–1822
58. Wang Y, Franzen J (1992) The non-linear resonance QUISTOR Part1: Potential distribution in hyperboloidal QUISTORs. Int J Mass Spectrom Ion Processes 112:167–178
59. Wang Y, Franzen J, Wanczek KP (2009) The non-linear resonance ion trap. Part 2. A general theoretical analysis. Int J Mass Spectrom Ion Processes 124:125–144
60. Wang Y, Franzen J (1994) The non-linear ion trap. Part 3. Multipole components in three types of practical ion trap. Int J Mass Spectrom Ion Processes 132:155–172
61. Franzen J (1993) The non-linear ion trap: Part 4. Mass selcetive instability scan with multipole superposition. Int J Mass Spectrom Ion Processes 125:165–170
62. Franzen J (1994) The non-linear ion trap. Part 5. Nature of non-linear resonances and resonant ion ejection. Int J Mass Spectrom Ion Processes 130:15–40
63. Marshall AG, Hendrickson CL, Jackson GS (1998) Fourier transform ion cyclotron resonance mass spectrometry: a primer. Mass Spectrom Rev 17:1–35
64. Comisarow MB, Marshall AG (1974) Fourier transform ion cyclotron resonance spectroscopy. Chem Phys Lett 25:282–283
65. Goodlett DR, Bruce JE, Anderson GA, Rist B, Pasa-Tolic L, Fiehn O, Smith RD, Aebersold R (2000) Protein identification with a single accurate mass of a cysteine-containing peptide and constrained database searches. Anal Chem 72:1112–1118
66. Hu Q, Noll RJ, Li H, Makarov A, Hardman M, Graham CR (2005) The Orbitrap: a new mass spectrometer, *J.* Mass Spectrom 40:430–443
67. Perry RH, Cooks RG, Noll RJ (2008) Orbitrap mass spectrometry: instrumentation, ion motion and applications. Mass Spectrom Rev 27:661–699
68. Scigelova M, Makarov A (2006) Orbitrap mass analyzer–overview and applications in proteomics. Proteomics 6(Suppl 2):16–21
69. Aebersold R, Goodlett DR (2001) Mass spectrometry in proteomics. Chem Rev 101:269–295
70. Spengler B, Kirsch D, Kaufmann R, Jaeger E (1992) Peptide sequencing by matrix-assisted laser-desorption mass spectrometry. Rapid Commun Mass Spectrom 6:105–108
71. de Hoffmann E (1996) Tandem mass spectrometry: a primer. J Mass Spectrom 31: 129–137
72. Steen H, Kuster B, Mann M (2001) Quadrupole time-of-flight versus triple-quadrupole mass spectrometry for the determination of phosphopeptides by precursor ion scanning. J Mass Spectrom 36:782–790
73. Aldini G, Regazzoni L, Orioli M, Rimoldi I, Facino RM, Carini M (2008) A tandem MS precursor-ion scan approach to identify variable covalent modification of albumin Cys34: a new tool for studying vascular carbonylation. J Mass Spectrom 43:1470–1481
74. Hopfgartner G, Varesio E, Tschappat V, Grivet C, Bourgogne E, Leuthold LA (2004) Triple quadrupole linear ion trap mass spectrometer for the analysis of small molecules and macromolecules. J Mass Spectrom 39:845–855
75. Yates JR III, Speicher S, Griffin PR, Hunkapiller T (1993) Peptide mass maps: a highly informative approach to protein identification. Anal Biochem 214:397–408
76. Johnson RS, Martin SA, Biemann K, Stults JT, Watson JT (1987) Novel fragmentation process of peptides by collision-induced decomposition in a tandem mass spectrometer: differentiation of leucine and isoleucine. Anal Chem 59:2621–2625
77. Chi A, Huttenhower C, Geer LY, Coon JJ, Syka JE, Bai DL, Shabanowitz J, Burke DJ, Troyanskaya OG, Hunt DF (2007) Analysis of phosphorylation sites on proteins from Saccharomyces cerevisiae by electron transfer dissociation (ETD) mass spectrometry. Proc Natl Acad Sci USA 104:2193–2198
78. Perdivara I, Petrovich R, Allinquant B, Deterding LJ, Tomer KB, Przybylski M (2009) Elucidation of O-glycosylation structures of the beta-amyloid precursor protein by liquid chromatography-mass spectrometry using electron transfer dissociation and collision induced dissociation. J Proteome Res 8:631–642
79. Alley WR Jr, Mechref Y, Novotny MV (2009) Characterization of glycopeptides by combining collision-induced dissociation and electron-transfer dissociation mass spectrometry data. Rapid Commun Mass Spectrom 23:161–170
80. Wiesner J, Premsler T, Sickmann A (2008) Application of electron transfer dissociation (ETD) for the analysis of posttranslational modifications. Proteomics 8:4466–4483
81. Carr SA, Huddleston MJ, Annan RS (1996) Selective detection and sequencing of phosphopeptides at the femtomole level by mass spectrometry. Anal Biochem 239:180–192
82. Huddleston MJ, Bean MF, Carr SA (1993) Collisional Fragmentation of Glycopeptides by Electrospary Ionization LC/MS and LC/

MS/MS: Methods for selective detection of glycopeptides in protein digests. Anal Chem 65:877–884

83. Annan RS, Carr SA (1997) The essential role of mass spectrometry in characterizing protein structure: mapping posttranslational modifications. J Protein Chem 16:391–402
84. Williamson BL, Marchese J, Morrice NA (2006) Automated identification and quantification of protein phosphorylation sites by LC/MS on a hybrid triple quadrupole linear ion trap mass spectrometer. Mol Cell Proteomics 5:337–346
85. Gadgil HS, Bondarenko PV, Treuheit MJ, Ren D (2007) Screening and sequencing of glycated proteins by neutral loss scan LC/MS/MS method. Anal Chem 79: 5991–5999
86. Langenfeld E, Zanger UM, Jung K, Meyer HE, Marcus K (2009) Mass spectrometry-based absolute quantification of microsomal cytochrome P450 2D6 in human liver. Proteomics 9:2313–2323
87. Unwin RD, Griffiths JR, Leverentz MK, Grallert A, Hagan IM, Whetton AD (2005) Multiple reaction monitoring to identify sites of protein phosphorylation with high sensitivity. Mol Cell Proteomics 4:1134–1144
88. Perkins DN, Pappin DJ, Creasy DM, Cottrell JS (1999) Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis 20:3551–3567
89. Eng JK, McCormack AL, Yates JR 3rd (1994) An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. J Am Soc Mass Spectrom 5:976–989
90. Biemann K (1990) Appendix 5. Nomenclature for peptide fragment ions (positive ions). Methods Enzymol 193:886–887
91. Zhang W, Chait BT (2000) ProFound: an expert system for protein identification using mass spectrometric peptide mapping information. Anal Chem 72:2482–2489
92. Gygi SP, Rist B, Gerber SA, Turecek F, Gelb MH, Aebersold R (1999) Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. Nat Biotechnol 17:994–999
93. Schmidt A, Kellermann J, Lottspeich F (2005) A novel strategy for quantitative proteomics using isotope-coded protein labels. Proteomics 5:4–15
94. Ross PL, Huang YN, Marchese JN, Williamson B, Parker K, Hattan S, Khainovski N, Pillai S, Dey S, Daniels S, Purkayastha S, Juhasz P, Martin S, Bartlet-Jones M, He F, Jacobson A, Pappin DJ (2004) Multiplexed protein quantitation in Saccharomyces cerevisiae using amine-reactive isobaric tagging reagents. Mol Cell Proteomics 3:1154–1169
95. Yao X, Freas A, Ramirez J, Demirev PA, Fenselau C (2001) Proteolytic ^{18}O labeling for comparative proteomics: model studies with two serotypes of adenovirus. Anal Chem 73:2836–2842
96. Staes A, Demol H, Van DJ, Martens L, Vandekerckhove J, Gevaert K (2004) Global differential non-gel proteomics by quantitative and stable labeling of tryptic peptides with oxygen-18. J Proteome Res 3:786–791
97. Ong SE, Blagoev B, Kratchmarova I, Kristensen DB, Steen H, Pandey A, Mann M (2002) Stable isotope labeling by amino acids in cell culture, SILAC, as a simple and accurate approach to expression proteomics. Mol Cell Proteomics 1:376–386
98. Gerber SA, Rush J, Stemman O, Kirschner MW, Gygi SP (2003) Absolute quantification of proteins and phosphoproteins from cell lysates by tandem MS. Proc Natl Acad Sci USA 100:6940–6945
99. Kito K, Ito T (2008) Mass spectrometry-based approaches toward absolute quantitative proteomics. Curr Genomics 9:263–274
100. Julka S, Regnier FE (2005) Recent advancements in differential proteomics based on stable isotope coding. Brief Funct Genomic Proteomic 4:158–177
101. Mueller LN, Brusniak MY, Mani DR, Aebersold R (2008) An assessment of software solutions for the analysis of mass spectrometry based quantitative proteomics data. J Proteome Res 7:51–61
102. Shiio Y, Aebersold R (2006) Quantitative proteome analysis using isotope-coded affinity tags and mass spectrometry. Nat Protoc 1:139–145
103. Thompson A, Schafer J, Kuhn K, Kienle S, Schwarz J, Schmidt G, Neumann T, Johnstone R, Mohammed AK, Hamon C (2003) Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. Anal Chem 75:1895–1904
104. Aggarwal K, Choe LH, Lee KH (2006) Shotgun proteomics using the iTRAQ isobaric tags. Brief Funct Genomic Proteomic 5:112–120
105. Bantscheff M, Boesche M, Eberhard D, Matthieson T, Sweetman G, Kuster B (2008) Robust and sensitive iTRAQ quantification on an LTQ Orbitrap mass spectrometer. Mol Cell Proteomics 7:1702–1713

106. Ong SE, Mann M (2005) Mass spectrometry-based proteomics turns quantitative. Nat Chem Biol 1:252–262
107. Kruger M, Moser M, Ussar S, Thievessen I, Luber CA, Forner F, Schmidt S, Zanivan S, Fassler R, Mann M (2008) SILAC mouse for quantitative proteomics uncovers kindlin-3 as an essential factor for red blood cell function. Cell 134:353–364
108. Krijgsveld J, Ketting RF, Mahmoudi T, Johansen J, Rtal-Sanz M, Verrijzer CP, Plasterk RH, Heck AJ (2003) Metabolic labeling of C. elegans and D. melanogaster for quantitative proteomics. Nat Biotechnol 21:927–931
109. Old WM, Meyer-Arendt K, Veline-Wolf L, Pierce KG, Mendoza A, Sevinsky JR, Resing KA, Ahn NG (2005) Comparison of label-free methods for quantifying human proteins by shotgun proteomics. Mol Cell Proteomics 4:1487–1502
110. Carvalho PC, Hewel J, Barbosa VC, Yates JR III (2008) Identifying differences in protein expression levels by spectral counting and feature selection. Genet Mol Res 7:342–356
111. Liu H, Sadygov RG, Yates JR III (2004) A model for random sampling and estimation of relative protein abundance in shotgun proteomics. Anal Chem 76:4193–4201
112. America AH, Cordewener JH (2008) Comparative LC-MS: a landscape of peaks and valleys. Proteomics 8:731–749
113. Johansson C, Samskog J, Sundstrom L, Wadensten H, Bjorkesten L, Flensburg J (2006) Differential expression analysis of Escherichia coli proteins using a novel software for relative quantitation of LC-MS/MS data. Proteomics 6:4475–4485
114. Lill J (2003) Proteomic tools for quantitation by mass spectrometry. Mass Spectrom Rev 22:182–194
115. Langenfeld E, Meyer HE, Marcus K (2008) Quantitative analysis of highly homologous proteins: the challenge of assaying the "CYP-ome" by mass spectrometry. Anal Bioanal Chem 392:1123–1134
116. Rivers J, Simpson DM, Robertson DH, Gaskell SJ, Beynon RJ (2007) Absolute multiplexed quantitative analysis of protein expression during muscle development using QconCAT. Mol Cell Proteomics 6:1416–1427
117. Brun V, Dupuis A, Adrait A, Marcellin M, Thomas D, Court M, Vandenesch F, Garin J (2007) Isotope-labeled protein standards: toward absolute quantitative proteomics. Mol Cell Proteomics 6:2139–2149
118. Basch JJ, Farrell HM Jr (1979) Charge separation of proteins complexed with sodium dodecyl sulfate by acid gel electrophoresis in the presence of cetyltrimethylammonium bromide. Biochim Biophys Acta 577:125–131
119. Akins RE, Tuan RS (1994) Separation of proteins using cetyltrimethylammonium bromide discontinuous gel electrophoresis. Mol Biotechnol 1:211–228

Chapter 2

In-Depth Protein Characterization by Mass Spectrometry

Daniel Chamrad, Gerhard Körting, and Martin Blüggel

Abstract

Within this chapter, various techniques and instructions for characterizing primary structure of proteins are presented, whereas the focus lies on obtaining as much complete sequence information of single proteins as possible. Especially, in the area of protein production, mass spectrometry-based detailed protein characterization plays an increasing important role for quality control. In comparison to typical proteomics applications, wherein it is mostly sufficient to identify proteins by few peptides, several complementary techniques have to be applied to maximize primary structure information and analysis steps have to be specifically adopted. Starting from sample preparation down to mass spectrometry analysis and finally to data analysis, some of the techniques typically applied are outlined here in a summarizing and introductory manner.

1. Introduction

The field of Proteomics has been very successful in identifying the quantification of large sets of proteins (protein mixtures), for example, from whole organelles or cell lysates. Nowadays, hundreds of proteins within a complex sample can be easily identified by mass spectrometry, whereas only few peptides per protein are usually detected (1). This allows elucidating the name of the protein via searching protein sequence databases. In addition to analyzing complex protein mixtures, at least equally challenging is the art of in-depth characterization of individual proteins, or in other words, gaining as much primary structure information (including posttranslational modifications) as possible from a protein of interest.

In-depth protein characterization is of great importance, as it increases the chance to detect posttranslational modification (PTM), which modulates the activity of most eukaryote proteins. Also validating and distinguishing protein isoforms within a sample

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_2,

demands detailed elucidation of the protein sequence. Especially, therapeutic protein products require thorough characterization, for example, during protein engineering, protein production, and for first in men studies throughout routine testing.

Mass spectrometry (MS) is an excellent tool for this purpose as it allows deducing the primary structure of proteins, including PTM by measuring mass per charge ratios (m/z) of peptide ions and corresponding peptide fragment ions in a high-throughput manner (2). Especially, the technology advances in recent years, including the increase in accuracy (today at ppm for peptides and peptide fragments), sensitivity (femtomol) and acquisition speed (more than 10,000 spectra/h) has turned MS into the most valuable analysis tool for detailed characterization of complex molecules like proteins.

While high-throughput protein identification from peptide fragmentation (MS/MS) has become a standard in modern MS-based protein analytics, complete primary structure elucidation, including PTM is still a challenge due to various reasons:

(a) Masses measured by MS are generally not unique, i.e., different amino acid sequences, including PTM may have identical or similar mass values, making them hard to distinguish.

(b) Protein and peptide modifications can be induced by sample preparation and these must therefore be carefully distinguished from original in vivo PTM.

(c) Some protein sequence segments may be hard to monitor by MS, e.g., some peptides are hard to ionize or show poor fragmentation.

(d) Protein modifications may not be homogenous, and due to numerous gene products caused by alternative splicing and combinations of modifications the protein mixture can be very complex.

(e) Sample preparation methods have to be individually developed as low protein concentration and interfering small molecules like salt, detergent, and stabilizers in formulation are limiting or even preventing mass spectrometric analysis.

In this chapter, we explore various current methods for complementary primary structure elucidation via mass spectrometry. We also focus on sample preparation as this is an essential prerequisite to enable and improve primary structure discovery.

2. Methods

2.1. Sample Preparation

Sample preparation methods for in-depth protein characterization by MS have to be developed to fulfill two aspects. On the one hand, sample preparation has to be performed to enable mass

spectrometric analysis. On the other hand, it has to be designed in a way to minimize the risk of primary structure change due to the sample preparation.

2.1.1. Enabling Mass Spectrometric Analysis

Adjuvants and contaminants, such as salt, detergent, or stabilizers, have the potential to prevent or reduce the results of mass spectrometric analysis. In case of liquid chromatography coupled to electrospray ionization mass spectrometry, salts in millimolar concentrations and even low detergent concentrations can be removed online within the HPLC setup (e.g., guard column or dedicated trapping column). For higher concentrations and for MALDI-MS applications, spinning columns (e.g., 3.5-kDa cutoff), dialysis (also available as microdialysis) or precipitation are the methods which are mostly applied. Additionally, separation techniques with high resolving power, such as reverse phase-HPLC or the combination of SDS-PAGE (1D or 2D) with protein digestion, are also well suited to move to an MS compatible buffer, with salts like ammonia carbonate, solvents like water, acetonitril, methanol, and acids like formic or triflouracetic acid.

2.1.2. Minimizing Risk of Primary Structure Change

Oxidation of, for example, Methionine, deamidation of Asparagine, or truncation may occur under conditions of sample preparation. Additionally existing modifications (e.g., phosphorylation) may be removed (e.g., by contact to iron in not inert HPLC systems).

Therefore, the sample preparation steps have to be limited to the minimum steps needed. Harsh conditions have to be avoided (e.g., 4 h, 37°C protein digestion method instead of 24 h, 37°C to avoid deamidation).

There are no universal protocols as the methods have to be adopted and altered to meet several aspects:

(a) Aim of analysis and intended MS technique.

(b) Starting protein concentration and nature of buffer content.

(c) Final protein amount and concentration needed.

Additionally, protein specific aspects like hydrophobicity, tertiary structure, or modification often result in a need for protein-specific method development.

Some general rules provide a guideline to method development:

(a) Avoid any unnecessary step (e.g., multiple concentration, buffer changes).

(b) Work at high protein concentrations so that only a minor fraction of the analyzed proteins is lost due to unspecific adsorption and reduce unfavorable adjuvant to protein ratios.

(c) Minimize harsh stress conditions like high temperature or RT for longer time, freeze/thaw cycle, extreme pH, lyophilization steps; oxidative stress.

(d) Do not introduce any adjuvants where not needed.

2.2. Primary Structure Elucidation by Mass Spectrometry

The primary structure of a biological molecule is the exact specification of its atomic composition and the chemical bonds connecting those atoms. For a high molecular weight protein like an antibody with approximately 20,000 atoms, the information of its primary structure is very complex. Fortunately, a good portion of this information can be reduced to the amino acid sequence.

However, for proteins the primary structure is not only covering the exact amino acid sequence, but also cross-links like disulfide bridges and modifications. Microheterogeneity will add another level of complexity into sample characterization as it is present in many highly purified recombinant proteins as well.

During the last 20 years, a huge number of mass spectrometric methods were developed to analyze the primary structure in detail. A full molecular weight determination by MS can provide a good insight for the verification of primary sequence and detection of modification. MALDI-TOF-MS is robust in sample preparation and salt concentration and can give you accuracy with as low as a few Daltons for midsized proteins. With this accuracy, information on N-/C-terminal truncation or modifications like glycosylation or phosphorylation can be obtained. However, for modifications like deamidation, disulfid linkage, or even oxidation a higher accuracy may be needed. The ability of Electro Spray Ionization to measure the molecular weight of multiple highly charged ions in parallel results in a much better accuracy. For ESI-FT-MS measurement, these molecular weight determination can be in a sub-Dalton range.

For a more detailed primary characterization, the protein has to be cleaved into subunits or peptides which are then measured by mass spectrometry.

The "MALDI In Source Decay" method fragments a full intact protein within the mass spectrometer and enables here a direct sequencing of the N- and C-terminal sequence area.

A sample preparation with a highly specific enzymatic digestion (e.g., Trypsin, Glu-C, Asp-N, etc.) will result into peptides which can be measured in a mixture (e.g., by MALDI-MS) or separated and analyzed by online LC-ESI-MS. With today's instruments, these peptides can be measured with high sensitivity (fmol) and with highest mass accuracy (low to even sub-ppm level). In the same experiment, these peptides can be fragmented within the mass spectrometer and the resulting peptide fragment pattern will be recorded also with highest mass accuracy and sensitivity.

With this ability and lab automation, it is possible to resolve also very complex primary structures and microheterogeneity of low abundant sequence variants.

However, data analysis becomes increasingly important to unravel the full potential and latest improvements of mass spectrometry.

2.3. Signal Extraction

Signal extraction and calibration are the most common first steps in the MS data interpretation process. Most software tools for MS-based protein analysis accept so-called peak lists, which are a collection of signals of a mass spectrum. Peak extraction is a complex task due to signal resolution, noise, signal overlapping, and the need for deisotoping.

In case of ESI-MS, peptides and proteins are typically detected in various charge states (z), e.g., with $z = 1–4$ for peptides, $z = 5–100$ for proteins and complexes). In order to determine the exact molecular weight of a peptide or protein, the spectrum has to be deconvoluted (calculate M or MH + from m/z values). The information of the charge state can be derived directly from the given isotopic m/z signal pattern using software tools (3, 4). However, one should be aware that the applied software may fail to assign the correct charge state. In case of proteins, molecular mass is derived from m/z mass peaks of multiple charge states of the same protein. In case of time of flight (TOF) measurements calibration of the spectra is essential to obtain sufficient mass accuracy. Calibration can be done internally (e.g., using theoretical m/z values of known peptides within the dataset, or by injecting substances in the MS instrument with each spectrum ("lock mass")), or externally (using the calibration constants of an earlier run, which contains spectra of a known substance).

After calibration, modern MS instruments can achieve a mass accuracy of few ppm.

2.4. Peptide Fragmentation Fingerprinting

Fragmentation mass spectra of peptides can be correlated to protein sequences in a database in an automatic manner (5, 6). This can be done by dedicated protein sequence database search software (see Table 1). It is advantageous that this method does not require any a-priori knowledge about the analyzed proteins, and therefore it is often used as an initial step to identify all major protein components in a sample.

Table 1
Overview on commonly used peptide fragmentation fingerprinting software

Mascot	http://www.matrixscience.com/
MS-Seq	http://prospector.ucsf.edu/
Phenyx	http://www.genebio.com/products/phenyx/
Popitam	http://www.expasy.org/tools/popitam/
SEQUEST	http://fields.scripps.edu/sequest/
SpectrumMill	http://www.home.agilent.com/
X! Tandem	http://prowl.rockefeller.edu/prowl/

Initially, the user has to define various input parameters carefully, such as the specificity of the applied proteolysis enzyme, maximum allowed mass errors for peptide parent ion and fragment masses and the protein sequence database to be searched. Then, the software generates theoretical spectra by theoretical fragmentation of peptides obtained from in silico digestion of the searched database proteins. The obtained theoretical spectra are compared to the measured spectra and the result is a list of matching peptides and proteins. Commonly, the reported proteins and peptides are sorted by a specific search score that relates to the significance of the found database match.

Protein and peptide modifications can be elucidated with this approach to some extent as typical database search engines that allow searching up to three different variable modifications (each amino acid in question is tested whether it is modified or not) and also fixed modifications (every amino acids is treated to be modified). Also regarding enzyme nonspecificity, missed cleavage sites and even peak picking errors (e.g., failure to detect the correct monoisotopic peptide signal from overlapping isotopic distributions) can be searched but generally applying these search strategies may lead to a drop in sensitivity. Therefore, it is advisable regarding only experimentally induced modifications (e.g., methionine-oxidation) and a maximum of one or two missed cleavages and no unspecific cleavage. In case of in-depth protein characterization, primary structure elucidation beyond this scope should be addressed by dedicated second round search engines (see below).

Mass accuracy is crucial to obtain unambiguous results. The maximum allowed mass error parameters within the search should be set to at least two standard deviations (assuming a normal distribution, about 95% of the measurement errors fall in two times standard deviation). The standard deviation for mass measurements can be determined within routine MS-instrument calibration.

Peptide masses determined by MS are generally not unique and each measured mass can randomly match a peptide from a sequence database. Therefore, a certain risk to obtain false positive results remains. Assessing the correctness of a possible identification is a challenging task. In fact, the probability that the match in question is correct cannot be calculated; however, most reported search scores relate to the probability that the observed peptide match is a pure random event (7, 8). In case of in-depth protein characterization, evaluation of sequence database search results is frequently not done automatically, but remains the task of an expert who manually inspects spectra matching to the protein of interest.

Usually, the primary structure detectable by a single database search is limited and must be extended by further experiments such as using a different cleavage enzyme, or using dedicated second round search engines.

2.5. Second Round Searches

Standard database searches which can be seen as "first round" searches are limited in the elucidation of posttranslational modifications, unspecific, and missed cleavages products, sequence errors, amino acid substitutions, and unsuspected mass shifts. For example, taking more than 200 described posttranslational modifications for all protein sequences of an organism into account would lead to an amount of peptides to be tested that impedes a brute force approach. Apart from the huge time exposure, simply the huge number of possible combinations leads to randomly matching sequences. To overcome this problem, second round searches have been developed, which work similar to peptide fragmentation fingerprinting described above but instead of searching a complete protein sequence database, only few selected protein sequences are regarded (9).

Typically, protein identification is done in the first step using standard search algorithms. Second round searches are then used in the second step to elucidate previously unexplained spectra. In case of the software tools Mascot and Phenyx, the second round search feature is directly integrated, and can be triggered after the first round search. There is also a dedicated second round search tool named Modiro™ (http://www.modiro.com) available. In case of Modiro™, the user can enter own protein sequences, which is of, for example, special interest in case of therapeutic protein products from biotechnology. During the second round, search batches of unidentified spectra (e.g., whole LC-MS/MS runs) are screened in a sequential manner for various different posttranslational modifications, unknown mass shifts, unspecific cleavages, and sequence errors in one single step. A typical search result obtained by using Modiro™ is shown in Fig. 1.

2.6. De Novo Sequencing

As genome sequencing capabilities have increased dramatically during the last decades, many organisms are sequenced today and sequences are available to the public community. However, genome sequence information is still lacking for many organisms at the same time while some of them are of interest in industrial or biochemical research.

As MS/MS spectra of peptides are generated by fragmentation within the backbone of the peptide, the mass difference between two fragment ions directly provide information on the amino acid at a given peptide position. As a result, de novo sequencing is feasible for a peptide and partly also for proteins. However, each fragmentation is highly sequence dependent, and the intensity of the different ions differs a lot for each fragment ion. Therefore, some positions may not be resolved. Additionally, a mass difference may be explained by more than one amino acid combination leading to inconclusive sequences. As additional fragmentation (e.g., from internal fragments, side cleavage, doubly charged ions) may occure and overlay the ion series, the manual interpretation is quite laborious.

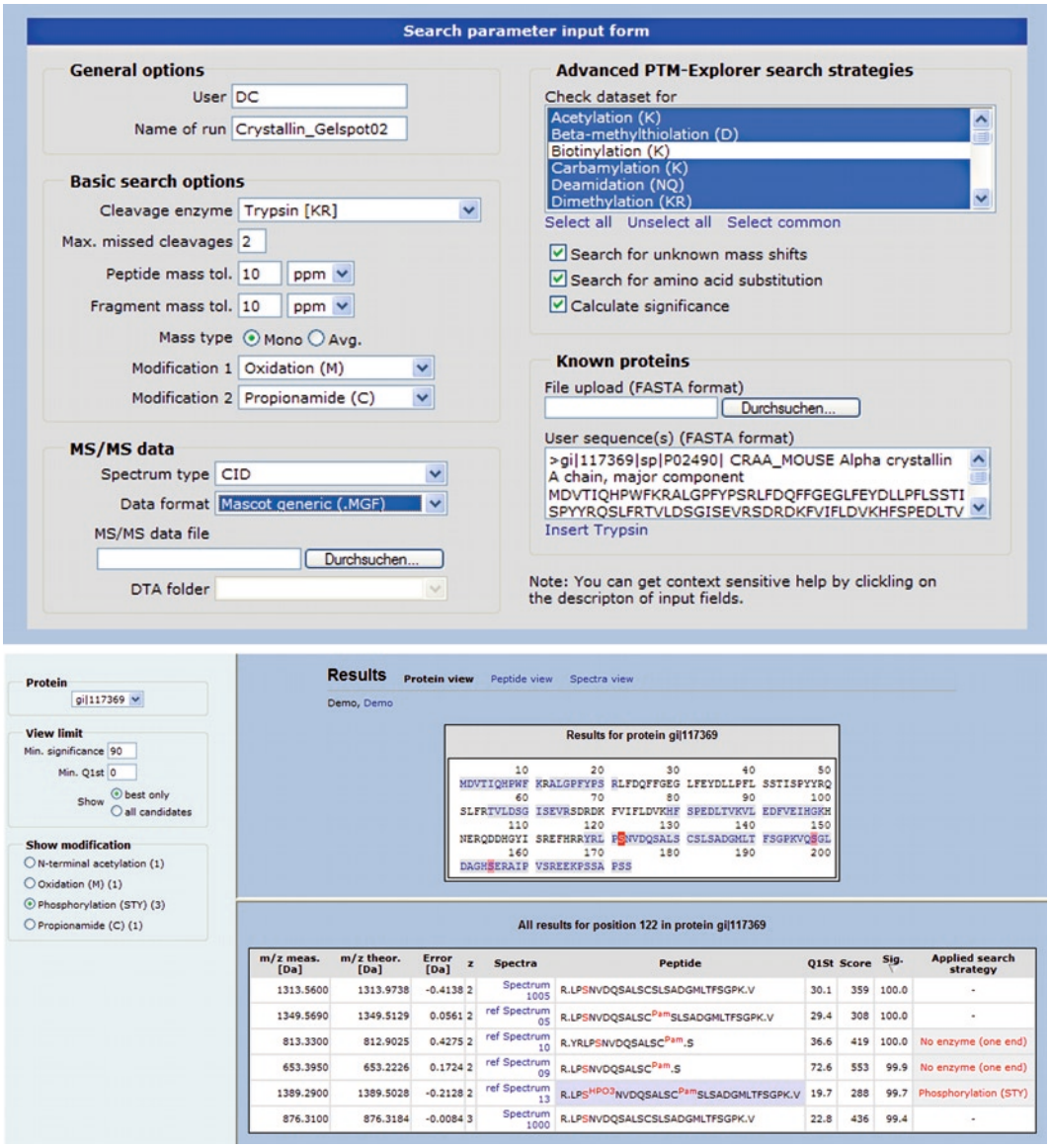


Fig. 1. Screenshots of the Modiro™ Software showing search parameter input and the obtained result page, including detected protein modifications in MS/MS datasets.

Several software solutions were developed to perform an automated de novo sequencing (e.g., PEAKS (10), PepNovo (11), Lutefisk (12)). They provide the best guess of the sequence, at least a sequence tag. The accuracy of this prediction highly depends on the quality of the fragmentation spectra. Resulting peptide candidates can be easily searched for homology against sequence databases. MS-BLAST (13) is a dedicated alignment tool for this purpose.

Additionally, MS instrument providers deliver software packages where either a full de novo algorithm is incorporated or sequence tag generation is supported by interactive annotation of a resulting MS/MS spectrum (e.g., BioTools, Bruker Daltonik GmbH).

Although knowing that a given protein is derived from a non-sequenced organism, its MS/MS data should be analyzed in the first round by a search engine (see Subheading 2.4) with no or broad taxonomy restriction. For some peptides, the homology might be sufficient to pick up the homolog protein from another already sequenced organism, which reduces the workload for de novo sequencing.

For isolated unknown proteins from an unsequenced organism internal protein sequence parts are needed, in order to construct nucleotidic degenerative primers for PCR and subsequent DNA sequencing. For this purpose, high quality sequence information ideally form the C-terminal region and long (minimum 7, best 15 amino acids) stretches are best suited.

2.7. Combination of Results (see Fig. 2)

In-depth characterization of protein requires the identification of the complete protein sequence. Usually, within a single MS analysis, some sequence areas are not identified or confirmed, as some peptides are outside the mass range detectable with a specific MS instrument, or have poor fragmentation. Therefore, it is advisable to make several MS runs, using different enzymes (or enzyme combinations) for proteolysis, or to apply other sample preparation techniques. Ideally, missing sequence areas will be different for the different runs and applied techniques, yielding more complete sequence coverage after the combination of the found

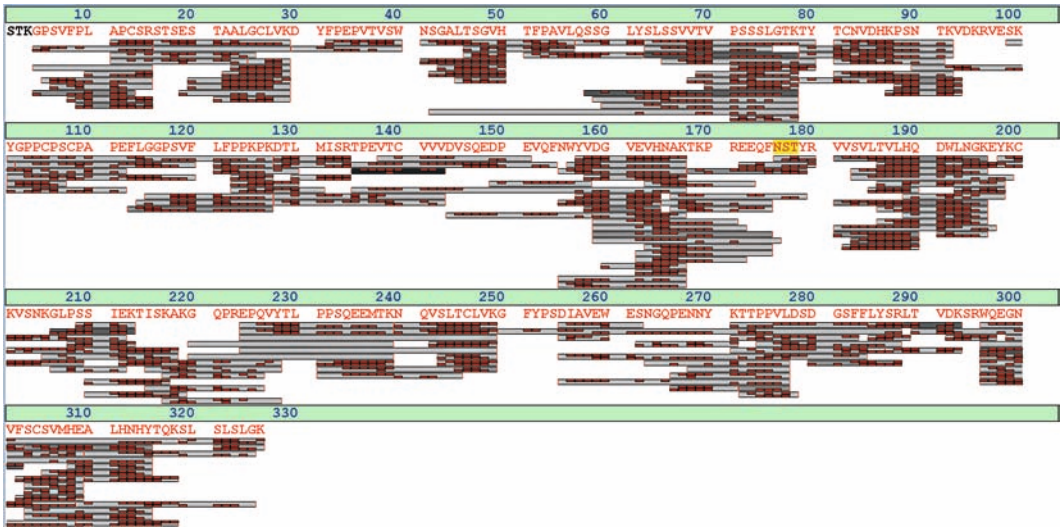


Fig. 2. Combining search results of MS/MS runs with several cleavage enzymes to get nearly complete sequence coverage.

peptides. Equally, analyzing the sample with differing MS instrumentation (e.g., MALDI-MS and LC-ESI-MS/MS) will give a complementary dataset.

Dedicated software is required to combine the outcome of the database searches, as a combined search with, e.g., different cleavage rules or mass spectrometric methods is not possible using currently available sequence database search software. In ProteinScape (Bruker Daltonik GmbH and Protagen AG), which is a Proteomics Bioinformatics Platform (14, 15), an algorithm for this task is integrated. Within ProteinScape, a new protein list is built, combining all peptides from all searches. Additionally, only the best matching sequence for each spectrum is annotated.

2.8. Differential EIC

For complete protein characterization of therapeutic proteins, it is necessary to show that the amino acid sequence, including modifications such as glycosylation meets the expected patterns. Second round searches with tools like Modiro™ can help to analyze existing modifications.

In case of LC-ESI data, the level of a specific modification can be validated by the visualization of Extracted Ion Chromatograms (EIC) of the modified and unmodified peptide. An EIC shows the mass spectrometric signal intensity of a specific m/z value over the retention time. With an overlay of two EICs, showing the m/z of the modified and the unmodified peptide, the level of modification can be detected (Figs. 3 and 4). If both signals are visible, there should be a retention time shift between them.

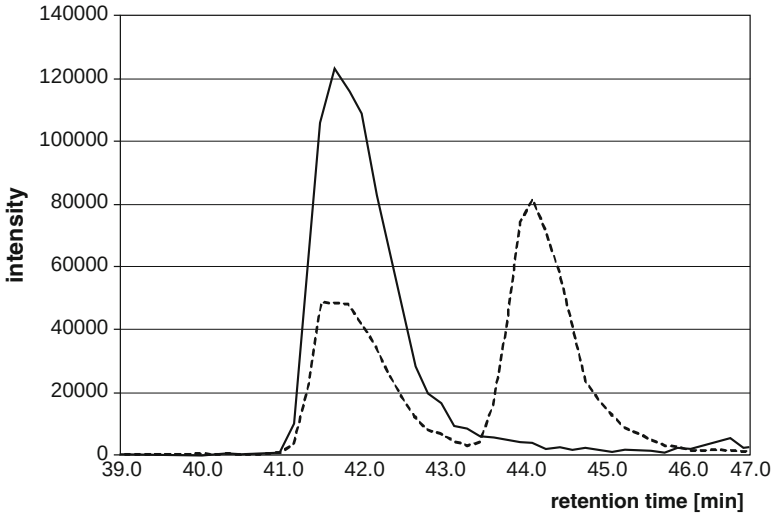


Fig. 3. Overlay of the extracted ion chromatograms of the unmodified and deamidated peptide W.LNGKEY.K. The peak at 42.0 min is the unmodified peptide (m/z = 723.3672), the peak at 44.5 min the deamidated peptide (m/z = 724.3512). By comparing the peak intensities or areas a medium deamidation can be estimated. The lower signal at 42.0 min is the second isotope of the unmodified peptide which has nearly the same m/z as the deamidated peptide.

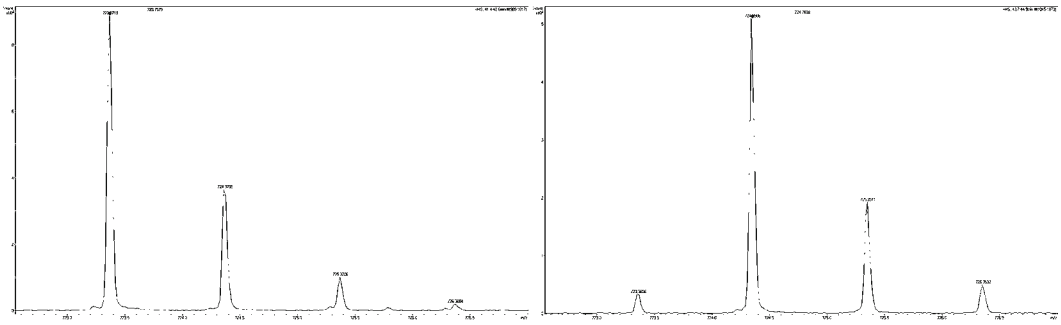


Fig. 4. MS spectra of the deamidation of Fig. 3. The first spectrum is the unmodified peptide at 42.0 min, the second spectrum the deamidated peptide at 44.5 min.

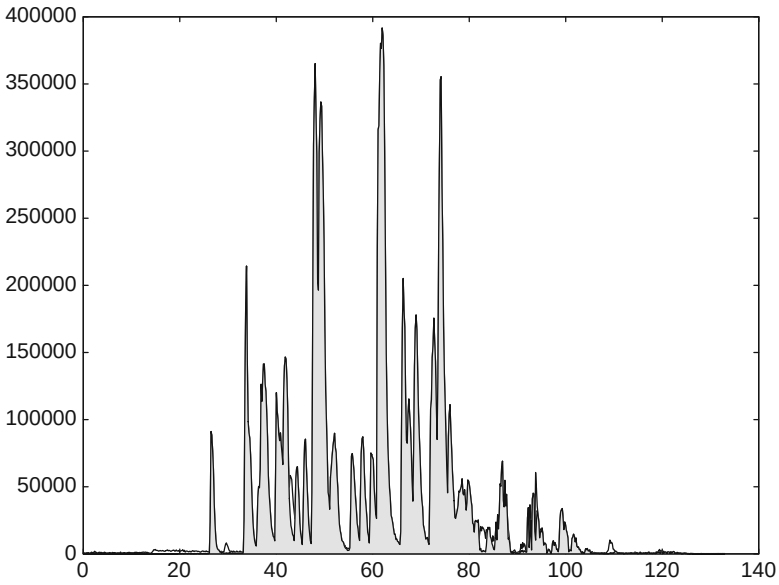


Fig. 5. Base peak chromatogram with identified peaks colored. Most of the MS run is explained. The remaining peak at 42 min was assigned to a peptide containing glycan, but the MS/MS fragmentation was not sufficient for identification.

Another way of assuring that there are no major signals left unexplained can be done by coloring identified peptides in a base peak chromatogram (Fig. 5). Ideally, there should be no peaks left unexplained. If major signals are still unexplained, the corresponding MS and MS/MS spectra must be analyzed further.

3. Conclusions

In-depth protein characterization by MS is significantly different from the task to identify proteins from simple or complex mixtures. The whole analysis process from sample preparation to MS acquisition

and data interpretation has to be specifically adopted to the analyzed protein samples in order to increase the amount of elucidatable primary structure information. Therefore, in-depth protein characterization is not standardized and remains an expert task. The major keys to successful primary structure characterization are firstly, sample preparation for the isolation and enrichment of the proteins to be analyzed, and secondly, the combination of several analysis methods to maximize the protein sequence coverage.

Significantly, more material is needed compared to protein identification approaches which require mapping of only a few peptides of each identified protein. The focus lies more on the enrichment or isolation of structural variants, including product impurities which are in low concentration. Chromatographic, electrophoretic separations or immunoaffinity purification are usable methods to isolate suitable amounts of the protein to be analyzed. Often milligrams of proteins are isolated to enable in-depth protein characterization.

Applying different complementary analysis methods is required. An example is increasing sequence coverage by using different proteolysis enzymes or combinations to make more protein sequence segments accessible to the MS measurement. Also the combination of various software tools for analyzing mass spectrometric data maximizes the primary structure yield contained in the acquired data.

As much as possible MS data has to be collected and must be evaluated in a combinatorial approach. However, MS data interpretation can only be partly automated by software. Laborious manual evaluation of mass spectra and primary structure assignment is still required.

4. Notes

1. Due to computational reasons, MS spectrum identification via software (Peptide Fragmentation Fingerprinting, De Novo Sequencing) works on peak lists rather than the originally acquired raw spectra. The preceding automatic peak picking procedures are not flawless and not lossless. Deconvolution and deisotoping is not always correct. Additionally, signals with low signal to noise ratio may be missed. For that purpose, it can be very helpful to validate a critical peptide match in question manually, using raw spectra. MS instrument providers usually deliver suitable software for manual raw spectrum annotation.
2. Especially, in the area of quality control for protein production, it is very important to elucidate sample preparation and MS-induced artifacts which are not related to the production process itself. Examples are Na^+ adducts, nonspecific proteolysis,

skimmer nozzle fragmentation, keratin contaminations, pyroglutamate formation from N-Term of internal peptides, etc. As long as these spectra remain unexplained, one cannot be sure about the purity of the product. Here, second round search engines are very helpful as they allow screening a wealth of possible modification in parallel, including also artificially induced ones.

3. Characterizing a protein via peptide fragmentation fingerprinting relies on the correctness of the protein sequence which is matched to the spectra. In case of, for example, sequencing errors, elucidation of corresponding fragmentation spectra fails. Sequence errors from single amino acid exchanges can be elucidated by second round searches. Other sequence errors must be elucidated via de novo sequencing.
4. In case of analyzing a specific peptide via an EIC, there may be unrelated signals and other peptides visible in the chromatogram with nearly the same *m/z*. Therefore, corresponding MS and MS/MS spectra have to be checked, too. The MS spectrum must show that the signal is a monoisotopic peak, and has the correct charge state, the MS/MS spectra must match to the peptide sequence.
5. To cover the whole amino acid sequence of a protein, LC-MS/MS runs from digests with several enzymes and enzyme combinations are necessary. To minimize the laboratory work, use software tools and theoretical digests to predict which enzymes or enzyme combinations are optimal to get peptides within the *m/z* acquisition range of a mass spectrometer.

References

1. Yates JR 3rd, McCormack AL, Schieltz D, Carmack E, Link A (1997) Direct analysis of protein mixtures by tandem mass spectrometry. J Protein Chem 16:495–497
2. Yates JR 3rd (1998) Mass spectrometry and the age of the proteome. J Mass Spectrom 33:1–19
3. Coombes KR, Fritsche HA Jr, Clarke C, Chen JN, Baggerly KA, Morris JS, Xiao LC, Hung MC, Kuerer HM (2003) Quality control and peak finding for proteomics data collected from nipple aspirate fluid by surface-enhanced laser desorption and ionization. Clin Chem 49:1615–1623
4. Gras R, Müller M, Gasteiger E, Gay S, Binz PA, Bienvenut W, Hoogland C, Sanchez JC, Bairoch A, Hochstrasser DF, Appel RD (1999) Improving protein identification from peptide mass fingerprinting through a parameterized multi-level scoring algorithm and an optimized peak detection. Electrophoresis 20:3535–3550
5. Mann M, Wilm M (1994) Error-tolerant identification of peptides in sequence databases by peptide sequence tags. Anal Chem 66:4390–4399
6. Eng JK, McCormack AL, Yates JR III (1994) An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. J Am Soc Mass Spectrom 5:976–989
7. Elias JE, Gygi SP (2007) Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. Nat Methods 4:207–214
8. Keller A, Nesvizhskii AI, Kolker E, Aebersold R (2002) Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. Anal Chem 74:5383–5392

9. Chamrad DC, Körting G, Schäfer H, Stephan C, Thiele H, Apweiler R, Meyer HE, Marcus K, Blüggel M (2006) Gaining knowledge from previously unexplained spectra-application of the PTM-Explorer software to detect PTM in HUPO BPP MS/MS data. Proteomics 6:5048–5058
10. Ma B, Zhang K, Hendrie C, Liang C, Li M, Doherty-Kirby A, Lajoie G (2003) PEAKS: powerful software for peptide de novo sequencing by tandem mass spectrometry. Rapid Commun Mass Spectrom 17:2337–2342
11. Frank A, Pevzner P (2005) PepNovo: de novo peptide sequencing via probabilistic network modeling. Anal Chem 77:964–973
12. Taylor JA, Johnson RS (1997) Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. Rapid Commun Mass Spectrom 11:1067–1075
13. Shevchenko A, Sunyaev S, Loboda A, Shevchenko A, Bork P, Ens W, Standing KG (2001) Charting the proteomes of organisms with unsequenced genomes by MALDI-quadrupole time-of-flight mass spectrometry and BLAST homology searching. Anal Chem 73:1917–1926
14. Blueggel M, Chamrad D, Meyer HE (2004) Bioinformatics in proteomics. Curr Pharm Biotechnol 5:79–88
15. Thiele H, Glandorf J, Hufnagel P, Körting G, Blüggel M (2008) Managing proteomics data: from generation and data warehousing to central data repository. J Proteomics Bioinform 1:485–507

Chapter 3

Analysis of Phosphoproteomics Data

Christoph Schaab

Abstract

Regulation of protein phosphorylation plays an important role in many cellular processes, particularly in signal transduction. Diseases such as cancer and inflammation are often linked to aberrant signaling pathways. Mass spectrometry-based methods allow monitoring the phosphorylation status in an unbiased and quantitative manner. The analysis of this data requires the application of advanced statistical methods, some of which can be borrowed from the gene expression analysis field. Nevertheless, these methods have to be enhanced or complemented by new methods. After reviewing the key concepts of phosphoproteomics and some major data analysis methods, these tools are applied to a real-world data set.

1. Introduction

Protein phosphorylation plays an important role in regulating many cellular processes. The phosphorylation status of a protein can influence its ability to interact with other proteins, its subcellular localization, and, in case of enzymes, its activity (1, 2). Phosphorylation events play a particularly prominent role in signal transduction pathways, which transmit signals caused by external stimuli from the cell membrane to the nucleus. Here, external stimuli activate receptors at the cell membrane that in turn activate a cascade of phosphorylation events in the cell membrane and the cytoplasm. Finally, the signal arrives at the nucleus and regulates gene expression by activating or inhibiting transcription factors. Although signal transduction pathways are often visualized as a series of steps, the reality is likely to be more complex: many of the pathways in general run parallel, they are cross-connected, and contain positive and negative feed-back loops.

Discovering signal transduction pathways is particularly important for understanding the mechanisms of certain diseases, such as cancer, inflammation, and diabetes (3, 4). Knowing the

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_3,

participant signaling pathways and the effect of a drug on these pathways will help in understanding the mode of action of the drug, to further optimize the drug or its use in combination with other drugs and to predict the responsiveness of patients to the drug (5–7).

Understanding these mechanisms requires, on the one hand, advanced statistical and mathematical tools (computational systems biology), and on the other hand, experimental data that can be modeled by these tools. Experimental methods such as western-blotting or ELISA-based assays allow monitoring the phosphorylation of tens or hundreds of sites. Both methods require phospho-specific antibodies and are therefore limited to the detection of known phosphorylation sites. Recent advances in mass spectrometry, methods for the enrichment of phosphorylated proteins or peptides, and software for analyzing the data enable the application of mass spectrometry-based proteomics to monitor the phosphorylation events in a global and unbiased manner. These methods become sufficiently sensitive and robust to localize the phosphorylation sites within the peptide sequence and to quantify them (8). However, the sheer amount of data generated by these methods makes computer-based processing of the data and sound statistical methods indispensable. A typical mass spectrometry-based experiment generates ~300 GB of raw data and can identify ~20,000 phosphorylation sites.

Before further delving into the bioinformatics tools available for the analysis of phosphoproteomics data, this chapter briefly explains the key aspects of sample preparation, enrichment methods, and quantification, followed by a review of some of the current software tools for processing the raw data generated by mass spectrometry-based phosphoproteomics experiments. The core of this chapter deals with the major methods that can be applied to the downstream analysis of phosphoproteomics data. Although most of the methods can also be applied to data obtained by ELISA assays, the focus here is on data generated with LC-MS/MS. These methods are then applied to a case study in which phosphoproteomics facilitated the generation of hypotheses about the mode of action of sorafenib (Nexavar®, Bayer), a drug approved for the treatment of kidney and liver cancer.

2. Methods

2.1. Sample Preparation and Detection of Phosphorylations

Before analyzing phosphoproteomics data, it is helpful to review the key aspects of how the samples are prepared, how phosphorylation sites are identified, and how they are quantified. The various state-of-the-art methods differ in these aspects, and it is important to account for these differences when analyzing the data. Nevertheless,

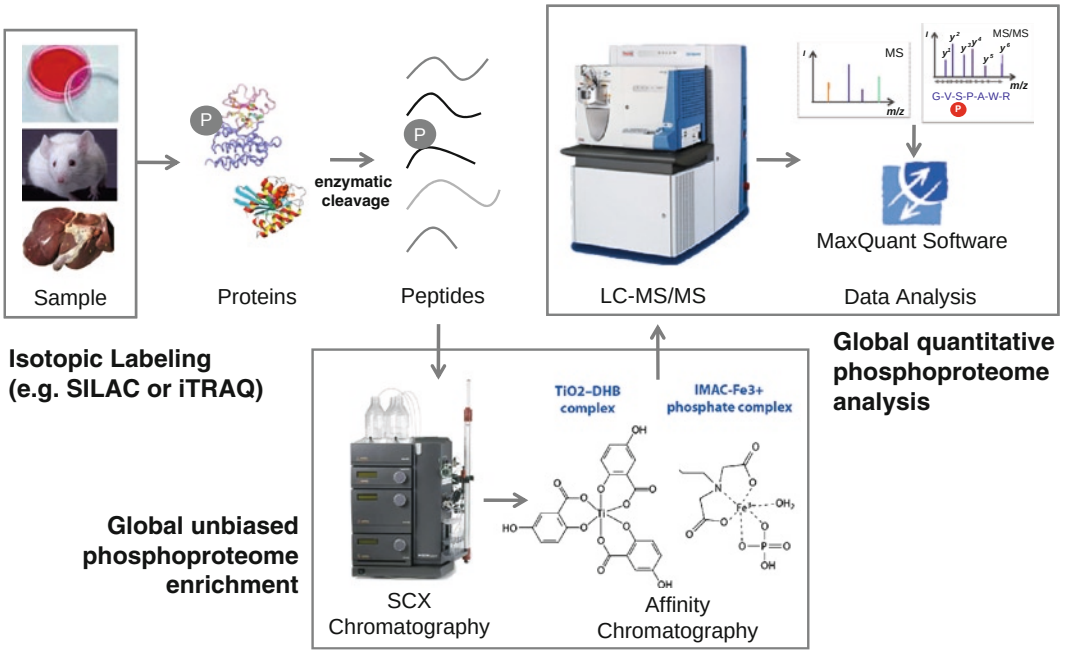


Fig. 1. Global quantitative phosphoproteomics workflow (5). Cells are lysed, proteins extracted and enzymatically cleaved. The peptides are enriched for phosphorylated peptides using TiO_2 or IMAC combined with strong cation chromatography (SCX). Finally, the peptides are identified and quantified by LC-MS/MS, and the raw data is processed with MaxQuant.

for the sake of conserving space only a limited overview of the methods is presented (see (9) for a comprehensive review).

Figure 1 shows the global quantitative phosphoproteomics workflow that was used in (5). After the cells from the samples to be processed are lysed, the protein extracts are digested (e.g., by trypsin), the resulting peptides are optionally labeled (*see* below) and then enriched. The phosphorylation is stabilized by adding phosphatase (e.g., Orthovanadate) and kinase inhibitors (e.g., EDTA). The proportion of phosphorylated peptides (phosphopeptides) is relatively small. Since the likelihood that a specific peptide is identified in the LC-MS/MS decreases with increasing complexity of the sample, enriching the sample for phosphopeptides is essential. The available enrichment methods differ in the type of phosphopeptides that is enriched. Metal affinity-based methods, such as IMAC and TiO_2, utilize the affinity of phosphates to certain metal ions. These methods have the advantage that they are not specific to a certain phosphorylated amino acid and thus can be used for a global analysis of the phosphoproteome.

The alternative enrichment method using immunopurification with immobilized antiphosphotyrosine antibodies, on the other hand, can only enrich peptides with phosphorylated tyrosine (10). Tyrosine phosphorylation represents only a small fraction (<1%) of the whole phosphoproteome. Nevertheless, it plays an

important role in signal transduction cascades, such as the ones starting with receptor tyrosine kinases, and is of special interest in cancer cells in which these cascades are often mutated. When selecting an enrichment method the reduced complexity obtained with antiphosphotyrosine antibodies must be balanced with the more complete coverage achieved with affinity-based methods.

The obtained phosphopeptides are usually analyzed by LC-MS/MS, i.e., by a nano-liquid chromatography (nanoLC) combined with a tandem mass spectrometer (for overview please also see Schönebeck et al. in this issue). The nanoLC is necessary to further reduce the enormous complexity of the sample. The first-stage MS detects the ionized peptides and selects them for fragmentation. The fragmented peptides are analyzed in the second-stage MS. The fragment spectrum allows a unique identification of the peptide and, in many cases, a localization of one or more phosphorylation sites within the peptide sequence.

However, in most studies the phosphorylation sites have to be not only identified, but also quantified and compared between different samples (see Notes 1 and 2). Because the peptide ion counts can vary significantly from sample to sample and from LC-MS/MS run to LC-MS/MS run, the label-free quantification using precursor ion counts is usually not sufficiently accurate. This is particularly true for phosphoproteomics experiments, since the quantification is based on single peptides only. A strategy to circumvent these issues is to label the samples such that two or more samples can be measured in the same LC-MS/MS run. The two most frequently used labeling methods are the metabolic method SILAC (11) and the chemical method iTRAQ™ (12).

A common aspect of all mass spectrometry-based methods is that the selection of the peptides for fragmentation is to a certain extent random. The likelihood for selection depends not only on the ion count of the respective peptide, but also on the ion counts of all other peptides in the same retention time window. Thus, the resulting data will contain missing values, and if a peptide is missing in a certain sample, it cannot necessarily be concluded that its concentration in this sample was low. This is different in microarray-based methods for gene expression analysis and has to be taken into account later. Please see Note 3 for the possibilities to improve the coverage.

2.2. Raw Data Processing

After analyzing the samples by LC-MS/MS, the resulting MS and MS/MS spectra have to be processed in order to identify the phosphopeptides that best match the measured MS/MS spectra and, depending on the labeling method used, the corresponding quantification information has to be read from the MS or MS/MS spectra. Both tasks are far from trivial and until recently involved a lot of manual work. The identification is usually done by searching the measured spectra against theoretical spectra

computed from protein sequence databases, such as IPI, UniProt, or Entrez. Standard search engines, such as Mascot or Sequest, perform well on unmodified peptides. However, the exact localization of the phosphorylated amino acids requires the comparison of the measured spectra with theoretical spectra where the phosphate group is placed on each possible amino acid. Tools such as MSQuant (http://msquant.sourceforge.net), SuperHirn (13), or MaxQuant (14) implement sophisticated algorithms to improve the identification of the peptides and the localization of the phosphorylation sites. MSQuant and MaxQuant additionally allow the quantification of SILAC-labeled peptides, whereas SuperHirn can quantify nonlabeled peptides.

In the following, we illustrate the raw data processing steps by using MaxQuant. A detailed description of the steps can be found in (15). MaxQuant currently supports raw files produced by Thermo LTQ-FT-ICR and LTQ-Orbitrap instruments only. After acquisition of the raw data with the vendor's software, it is processed with the "Quant" module in a first step. All LC-MS/MS runs that belong to one experiment should be processed together, since MaxQuant can use the information from all runs to improve the peptide identification. Quant performs a 3D peak and isotope pattern detection for identifying the peaks that belong to isotopic SILAC pairs or triplets and quantifies these. The produced "msm" files containing processed MS/MS spectra are then submitted to the Mascot search engine. It is important that the sequence database also contains the reverse protein sequences of all database entries, since these are used by the scoring algorithm for estimating the false discovery rate. Finally, the raw files, the intermediate files from Quant, and the Mascot results are processed by the "Identify" module. Identify integrates all the information, assigns the identified peptides to proteins, aggregates the quantitative information, and performs statistical validation. In particular, it uses the identified phosphorylated peptides to deduct quantitative information for all identified phosphorylation sites. The generated file "Phospho (STY)Sites.txt" is used for further analysis. Besides information on the identified phosphorylation sites and their quantification in the performed experiments, it contains various scores that allow judging the reliability of the identification and localization of the phosphorylation sites (please see Note 4). Figures 2 and 3 show two examples of MS/MS spectra that allowed a localization of the phosphorylation site with high confidence (Fig. 2) and with low confidence (Fig. 3).

2.3. Downstream Analysis

A typical phosphoproteomics experiment yields thousands of phosphorylation sites (e.g., Olsen et al. identified 6,600 sites (8)). This high dimensionality makes further analysis of the data using sound statistical methods indispensable. Many of the methods developed for the analysis of microarray data can be applied here too.

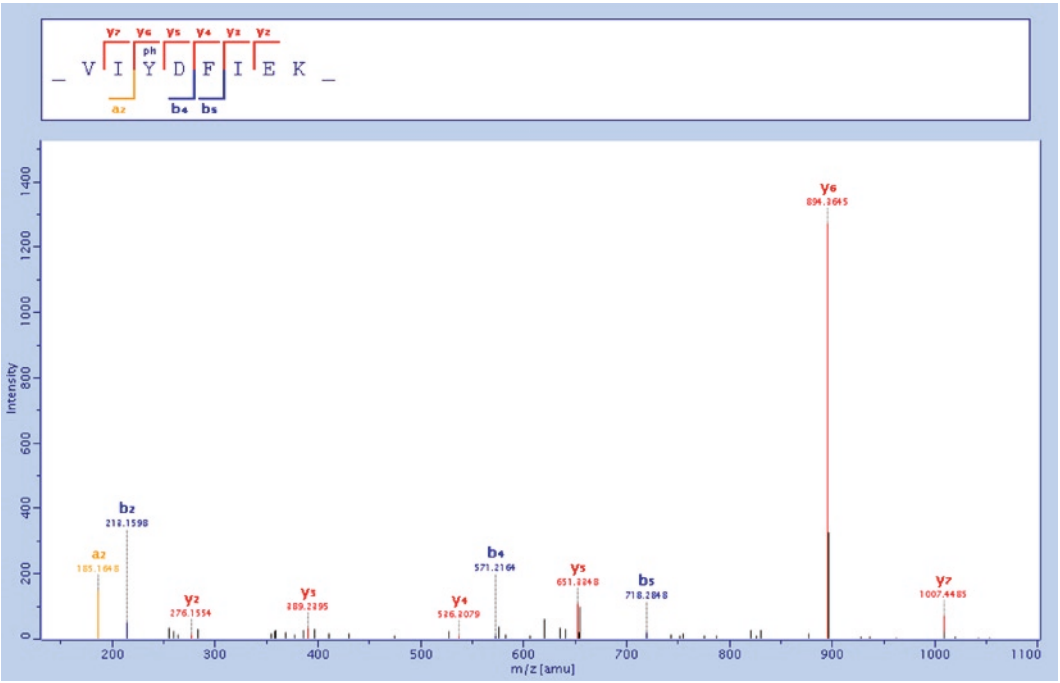


Fig. 2. Example of an MS/MS spectrum that allows the localization of the phosphorylation site with high confidence. Because of the identification of both the y5- and the y6-ions, the phosphate group can uniquely be assigned to the tyrosine at position 3.

There is no single method that should be applied to all phosphoproteomics experiments; rather the questions to be addressed by the experiments and the chosen experimental design determine which analysis methods are appropriate. Below some of the available methods are reviewed. How these methods are applied in concrete case studies is shown in the next section. All discussed procedures are available in Bioconductor (12) within the R platform or in the Statistics toolbox of Matlab (The Mathworks) or can easily be implemented therein.

2.3.1. Identification of Differential Phosphorylations

In many experimental setups, two or more conditions are compared and the first question arising is which of the phosphorylation sites are differentially phosphorylated. For example, the conditions could be untreated cells and cells treated with a small molecule or cells stimulated by growth factors. Or the experiment might compare wild type cells with cells carrying a certain mutation. Unless label-free methods are used, the ratio between the phosphorylation degree under condition 2 compared to its degree under condition 1 are measured for each phosphorylation site. In order to reliably identify differential phosphorylations the experiment has to be repeated several times. The null hypothesis then is that the ratio is 1, or 0 if log-ratios are considered. In general,

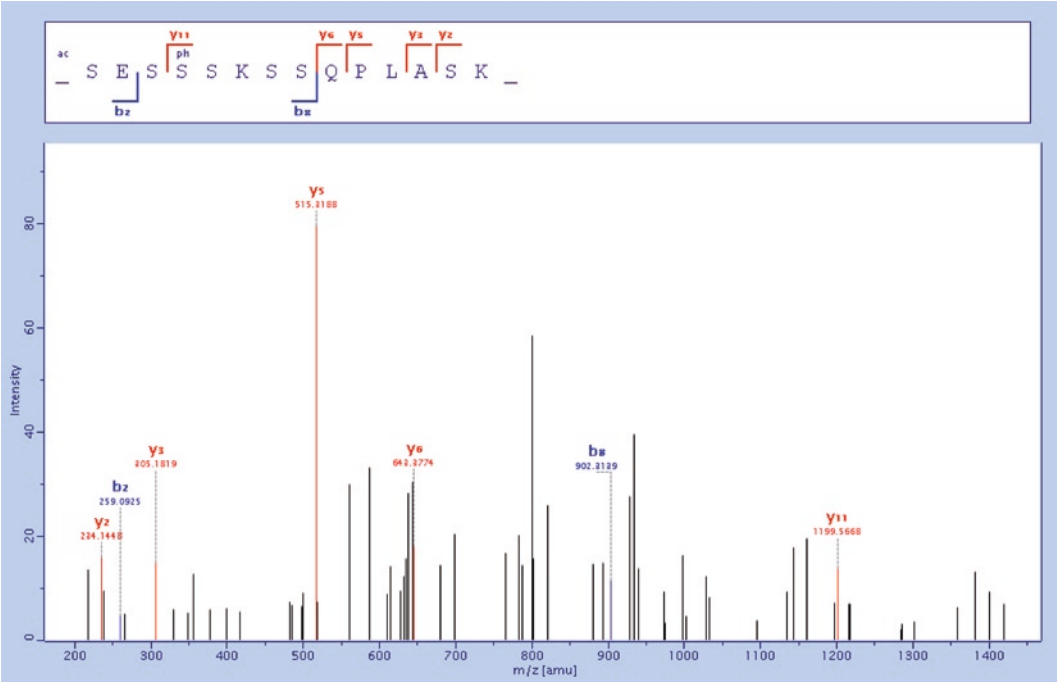


Fig. 3. Example of an MS/MS spectrum that does not allow the localization of the phosphorylation site. Since no further y-ions could be detected between y6 and y11, the phosphate group could not be assigned to any specific amino acid. The four serines between positions 4 and 8 have the same localization probability $p = 0.235$. With probability $p = 0.06$, one of the other serines is phosphorylated.

log-transforming the ratios is preferred since the distribution of the transformed ratios is closer to normal.

Many different statistics can be applied to test this null hypothesis. If the data is normally distributed, a natural choice is the *t*-statistics. As in the early days of microarrays often only a few replicates can be performed. This significantly influences the power of the *t*-test since the variance cannot be reliably estimated. Several modifications have been proposed to circumvent this issue. One of them, SAM (16) adds a "global" standard deviation to the feature-specific standard deviation. The global standard deviation is estimated from the whole data set. If the data is not normally distributed, one can also use rank (e.g., Mann–Whitney–Wilcoxon test) or permutation tests (see also Note 5).

After calculating the test statistics, a cut-off has to be defined above which the null hypothesis has rejected. Usually, the cut-off is defined based on the probability distribution under the null hypothesis. Since the question in performing, such experiments is usually not whether one specific site is differentially phosphorylated but rather which of the many thousand sites is differentially phosphorylated, one has to adjust for multiple testing. Otherwise, too many false positives will be selected. There are two principal

concepts to this. The first concept, the family-wise error rate (FWER), controls the probability that the selected list of differential sites contains at least one false positive. In case of the Bonferroni correction, the cut-off p-value is simply divided by the number of tests (17). Other corrections are less conservative, but still control the FWER (see (18, 19) for example). Often, a few false positives can be accepted as long as the proportion of false positives in the list of all selected differential sites is not too large. This idea is the basis of the second concept, the False Discovery Rate (FDR) (20). The FDR level gives the expected number of false positives in the selected list. The procedure by Benjamini and Hochberg (20) is step-down: the p-values $p_{(i)}$ are sorted in ascending order. If k is the largest i for which $p_{(i)} \leq \frac{i}{n}q$, then all hypotheses for $i = 1,...,k$ are rejected. Here, q is the FDR level and n the number of tests. Modifications of this procedure use different methods for estimating the distribution under the null hypothesis and for estimating the proportion of true positives. Most authors propose permutation-based procedures (21–23).

2.3.2. Enrichment Analysis

Depending on the tested conditions, number of replicates, and test procedure typically a few hundred or a few thousand sites will be identified as differentially phosphorylated. Often, these are too many to be individually validated in experiments. One can instead try to identify groups of proteins that have a known common feature and show similar phosphorylation profiles. Common features could be, for example, common gene ontology (GO) terms, pathways (like KEGG), or protein domains (e.g., PFAM or InterPro). More rigorously speaking, one is interested in sets in which differentially phosphorylated sites are overrepresented. Since the given groups often represent groups of proteins or genes, the phosphorylation sites have to be mapped to proteins. A protein is called differentially phosphorylated if it has at least one differentially phosphorylated site, where the mapping of sites to proteins is not necessarily unique.

A standard method for enrichment analysis is Fisher's exact test (24). The test is implemented in many publicly available tools. Most of them are searching for enriched GO terms (e.g., the standalone application GoMiner (25) or the Cytoscape-plugin BINGO (26)). See Ackermann and Strimmer (27) for a comprehensive overview and Goeman et al. (28) for a critical analysis of the proposed methods.

Because of the nature of phosphoproteomics experiments, the enrichment analysis of two classes of groups is of special interest. Firstly, signal transduction pathways function via the regulation of certain phosphorylation sites of involved kinases and other proteins (see Note 6). Secondly, kinases recognize their substrates

through patterns (motifs) in the amino acid sequence around the phosphorylation sites (see Note 7).

2.3.3. Multiple Conditions

When more than two biological conditions are tested, many additional methods can be applied to the data. Again, most of the methods used for microarray data can be applied to phosphoproteome data as well. The methods can be classified as either supervised or unsupervised methods. Supervised methods use the label information of the experiment, e.g., whether the sample was treated with a compound or not. Unsupervised methods do not use the label. Examples of unsupervised methods are clustering (hierarchical, *k*-means, self-organizing-maps …) and principal component analysis. Examples of supervised methods are ANOVA, partial least square analysis, and classification methods (decision trees, linear discriminant analysis, support vector machines …). The reader is referred to the extensive literature on these.

3. Case Study: Mode of Action Analysis for Sorafenib

Below the methods just discussed are applied to a case study. Details about this study can be found in (5). In this study, phosphoproteomics was used to investigate the mode of action of sorafenib in the prostate cancer cell line PC3. Sorafenib (Nexavar®, Bayer) is approved for the treatment of kidney and liver cancer.

Its most prominent target is b-Raf. However, since PC3 cells are sensitive to sorafenib, but not to other b-Raf inhibitors (29), sorafenib's mode of action remains unclear in these cells. PC3 cells were SILAC-labeled, the Arg^6/Lys^4- and the Arg^{10}/Lys^8-cells were treated with 10 μM sorafenib for 30 and 90 min, respectively, and the Arg^0/Lys^0-cells were used as a control (see Fig. 4). The cells were mixed, lysed, and the extracted proteins were digested with trypsin. The peptides were enriched for phosphorylated peptides using TiO_2 and IMAC combined with strong cation exchange chromatography (SCX). Finally, the peptides were identified and quantified by LC-MS/MS. The experiment was repeated three times.

3.1 Raw Data Processing

All the raw data was processed with MaxQuant version 1.0.12.28 (http://www.maxquant.org), and the fragment spectra were searched against UniProt human database version 57.4 using the Mascot search engine. MaxQuant writes the results to a number of tab-delimited text files in a folder named "combined." The most important file for our purposes is called "Phospho (STY) Sites.txt." This file contains all identified phosphorylation sites together with their scores, localization probabilities, matching proteins, and quantification values.

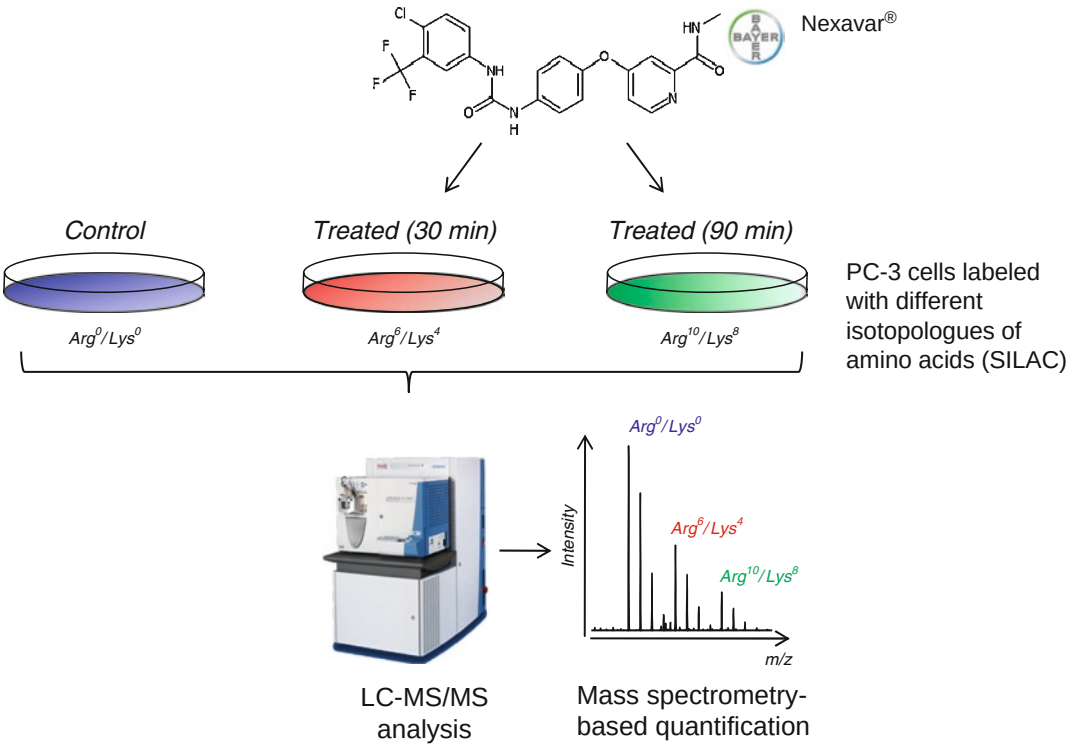


Fig. 4. Global quantitative phosphoproteomics workflow. PC3 cells are SILAC-labeled, the Arg^6/Lys^4- and the Arg^{10}/Lys^8-cells are treated with 1 μM sorafenib for 30 and 90 min, respectively, the Arg^0/Lys^0-cells are used as control. Afterward the global quantitative phosphoproteomics workflow is applied (see Fig. 1).

Table 1 shows the number of identified phosphorylation sites and onto how many distinct proteins they map. In total, ~25,000 phosphorylation sites were identified, of which ~16,000 are class I sites. Class I sites are defined as sites with a localization probability (column "Localization Prob") of at least 75% and a score difference (column "Score Diff") of at least 5. These phosphorylation sites are identified and localized with high confidence. In the following analysis steps, all other sites are ignored.

3.2. Differentially Phosphorylated Sites

The next step is to identify all phosphorylation sites that significantly differ between treated and untreated cells. Since two time points are not sufficient to perform any sensible time-series analysis, we simply take the more extreme average log-ratio for each phospho-site. Here, the average is taken over the three replicates. To be more precise: if $r^{(j)}_{i,M/L}$ ($r^{(j)}_{i,H/L}$) is the log-ratio of phospho-site i in replicate j between cells treated for 30 min (90 min) and untreated cell, and then the average ratio

Table 1
Number of identified phosphorylation sites. Class I sites are sites with a localization probability of at least 75% and a score difference of at least 5. For definition of "regulation," see text

	All proteins	Kinases
No. of detected phosphorylation sites	24,543	1,407
No. of detected phosphorylation sites (class I)	15,825	961
No. of detected proteins with phosphorylations (class I)	3,931	228
No. of regulated sites (class I)	1,012	68
No. of proteins with regulated phosphorylations (class I)	605	40

$$r_i = \begin{cases} \frac{1}{3}\sum_{j=1}^{3} r_{i,M/L}^{(j)} & \text{if } \left|\sum_{j=1}^{3} r_{i,M/L}^{(j)}\right| \leq \left|\sum_{j=1}^{3} r_{i,H/L}^{(j)}\right| \\ \frac{1}{3}\sum_{j=1}^{3} r_{i,H/L}^{(j)} & \text{otherwise} \end{cases}$$

is taken for further analysis.

Because of the low number of replicates, estimating the ratio variance will be very imprecise. Rather than applying a *t*-test or derivations thereof, we apply the global rank test as mentioned in Note 5. We use the nonparametric estimate of the expected number $\alpha^0(T)$ of top-T sites under the null hypothesis (see (30) for more details). The parameter T is determined such that the resulting FDR is below 5%. In total, 1,012 sites are significantly differentially regulated. Table 1 gives more details onto how many proteins and kinases these sites map.

3.3. Pathway Mapping

It would be an overwhelming task to investigate any single differentially phosphorylated site, to review the literature on this protein or this site, and to finally decide whether this differential phosphorylated site is of interest or not. Additionally, one has to keep in mind that many of the differential sites are not directly connected to the mode of action of the substance but rather are due to secondary effects.

If the investigated substance inhibits a certain signal transduction pathway, many of the participating proteins will show differentially phosphorylated sites. Contrariwise, if many of the differential sites belong to members of a certain pathway, it is likely that this pathway is affected by the substance. Thus, it makes

Table 2
KEGG signal transduction pathways for which proteins with differentially phosphorylated sites are significantly overrepresented as determined by Fisher's exact test

KEGG pathway	Proteins with detected P-sites	Proteins with regulated P-sites	*p*-value
Insulin signaling pathway	52	19	0.0002
MAPK signaling pathway	74	21	0.004
mTOR signaling pathway	22	9	0.005
ErbB signaling pathway	40	13	0.007
Axon guidance	34	10	0.04
Prostate cancer	21	7	0.04
Nonsmall cell lung cancer	17	6	0.04

a lot of sense to identify the pathways with overrepresented differential sites. We use the known human signal transduction pathways from KEGG (31) and apply the Fisher's exact test to test whether proteins with differential sites are significantly enriched in any of these pathways. The list of the pathways with a *p*-value below 5% is given in Table 2.

The list contains pathways that are expected (e.g., MAPK signaling), some pathways that are likely due to secondary effects (e.g., axon guidance), and some more surprising pathways, such as the mTOR signaling pathway. The map kinase signaling pathways are expected to be inhibited since many of sorafenib's targets are involved in these pathways, e.g., b-RAF, p38α, MEKK1. Figure 5 shows the KEGG MAPK pathway diagram annotated by the information whether the particular protein has phosphorylation sites that are downregulated after the treatment with sorafenib. If a protein has more than one detected phosphorylation site, the one with the most extreme average ratio is shown. In general, the pathway is covered by many proteins with detected phosphorylations. Furthermore, many of these phosphorylations go down after the treatment with sorafenib, whereas only a few go up. This confirms the expected inhibition of map signaling pathways.

Obviously, for many proteins more than one phosphorylation site is detected and often these sites behave differently. For example, eight class I phosphorylation sites were identified for the

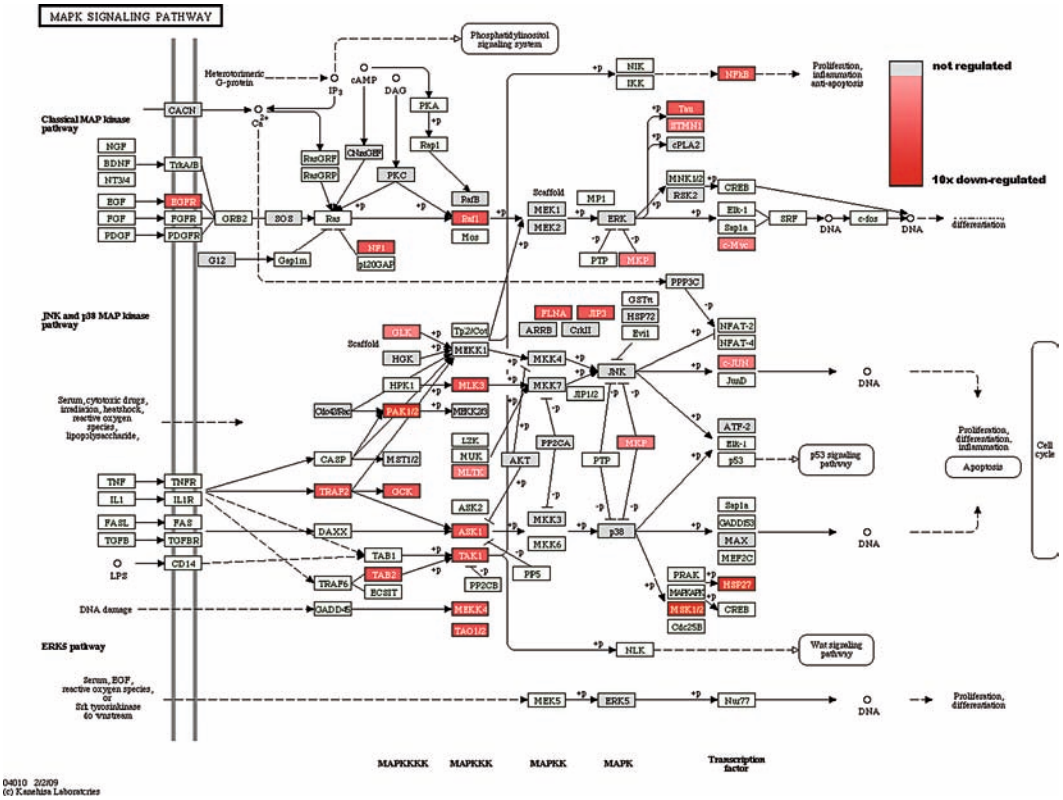


Fig. 5. KEGG MAPK signaling pathway (31). Proteins with detected phosphorylation sites are colored *dark gray*, if they are downregulated after the treatment with sorafenib, and *light gray* if they are not regulated at all. See the online version of this chapter for a colored figure.

ribosomal protein S6 (UniProt id P62753). Whereas five of them, all between amino acid position 235 and 242, are downregulated, two of them, at 244 and 246, are upregulated. Furthermore, KEGG, like other publicly available pathway resources, cover the pathways not in all details. In case of the mTOR pathway, we therefore draw the essential parts of the pathway based on the current literature (e.g., (32) using Inkscape http://www.inkscape.org, see Fig. 6). The identified phosphorylation sites can then be added to the protein nodes as small ellipses labeled by the position and colored by regulation factor. This allows capturing all details of the regulated pathway and supports the understanding of the mode of action of the substance. Mapping of the regulated phosphorylation sites to signal transduction pathways reveals that sorafenib treatment leads to severe downregulation of the MAP kinase pathway in PC3 cells. In addition, several other pathways are deregulated. In particular, the mTOR pathway is significantly affected by sorafenib in PC3 cells. Obviously, these hypotheses have to be validated with independent technologies that confirm the downstream effect on transcription or translation.

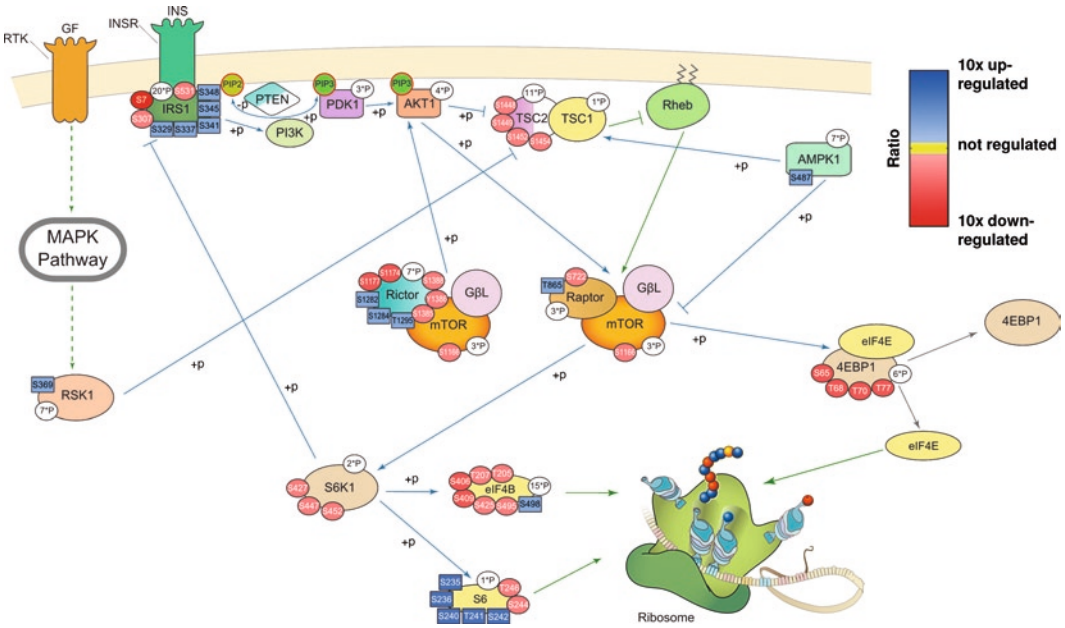


Fig. 6. mTOR pathway with identified phosphorylation sites. Sites, that are downregulated after the treatment with sorafenib, depicted as ellipses, upregulated sites as rectangles. See the online version of this chapter for a colored figure.

4. Conclusion

The analysis of global phosphoproteomes is a relatively new field within bioinformatics. In the last few years, technical advances have led to a steady increase in the number of detectable phosphorylation sites. It has recently become possible to detect and quantify 6,600 sites (8) or even 16,000 sites (5) in a single experiment. The processing of phosphoproteomics raw data requires software that combines standard search engines, such as Mascot and Sequest, with specialized algorithms for the identification of phosphorylated peptides, the localization of the phosphorylation sites, and their quantification. Examples of such software are MSQuant, SuperHirn, and MaxQuant.

We have seen that many of the methods that have been developed for gene expression analysis can also be applied to the downstream analysis of phosphoproteomics data. Additional methods that take the particular nature of the data into account have been developed, e.g., the enrichment of kinase motifs in the set of differential phosphorylated sites.

Unlike genetic mutational analysis or gene expression analysis that measure surrogates only, phosphoproteome analysis directly measures the signaling activity in the cell. Therefore, phosphoproteome analysis will be a valuable tool whenever effects on

cellular signaling activity are studied. For example, such an analysis may reveal the mode of action of drugs that inhibit certain kinases. Or, more visionary, such an analysis may discover biomarker signatures that allow to predict the optimal targeted therapy for a patient (personalized medicine, see (6)).

5. Notes

1. The peptide ion counts depend not only on the peptide concentration but on a number of additional parameters, such as the ionization efficiency, the elution behavior in the nanoLC, and the enrichment efficiency. These parameters differ for different peptides. Thus, two different peptides with identical concentration in the sample may have very different ion counts in the MS. On the other hand, these parameters do not differ for chemically identical peptides of different isotope composition. Thus, labeling methods, such as SILAC or iTRAQ™, allow the relative quantification of a peptide in different samples, whereas absolute quantification is impossible in principle (see however Note 2). The situation is analogous to the situation with microarray-based gene expression data, where due to the differences in the hybridization efficiency only comparisons between samples rather than between features are possible.
2. If defined amounts of synthetically produced, isotopically labeled peptides are spiked into the samples, absolute quantification of the corresponding natural peptides is possible (33).
3. If only a certain set of phosphopeptides is to be analyzed, one can use so-called targeted approaches to improve the coverage of this set. This includes the use of inclusion lists (34) or MRM-based methods (35).
4. Depending on the quality of the MS/MS spectra, it is not always possible to assign the phosphorylation to a specific amino acid. MaxQuant calculates the localization probability that the given amino acid is indeed the one that is phosphorylated. It often makes sense to restrict oneself to phosphorylation sites that are identified and localized with high confidence. Therefore, so-called class I sites are defined as the ones that have a localization probability of at least 75% and a score difference of at least 5 (8).
5. A very different approach has been taken by Zhou et al. (30) who proposed a "global rank test" for microarray data. Here, the sites are ranked by ratios within each replicate. Sites that are consistently ranked top or bottom T are identified as

differentially phosphorylated sites. The parameter T is fixed by an appropriate FDR that is estimated parametrically or based on permutations. A nice feature of this test procedure is that the FDR actually decreases with the number of tested sites. Standard FDR procedures show the opposite behavior.

6. There are a number of databases containing signal transduction pathways, including KEGG (http://www.genome.jp/kegg/pathway.html), BioCarta (http://www.biocarta.com/), and PANTHER (http://www.pantherdb.org/pathway/). By identifying pathways in which differentially phosphorylated proteins are overrepresented, one can expect that the corresponding biological processes differentially respond to the tested conditions.
7. Many motifs are known and the above approach can be used to identify motifs for which differential phosphorylation sites are overrepresented. Another approach is to de novo identify motifs from all differentially phosphorylated sites (36).

References

1. Hunter T (2000) Signaling–2000 and beyond. Cell 100:113–127
2. Pawson T, Nash P (2003) Assembly of cell regulatory systems through protein interaction domains. Science 300:445–452
3. Blume-Jensen P, Hunter T (2001) Oncogenic kinase signalling. Nature 411:355–365
4. Kaminska B (2005) MAPK signalling pathways as molecular targets for anti-inflammatory therapy – from molecular mechanisms to therapeutic benefits. Biochim Biophys Acta 1754:253–262
5. Tebbe A, Klammer M, Kaminski M, Wandinger S, Eckert C, Müller S, Gorray M, Enghofer E, Schaab C, Godl K (2009) Mode of action analysis of sorafenib by integrating chemical proteomics and phosphoproteomics. Eur J Cancer 7:14–15
6. Lim YP (2005) Mining the tumor phosphoproteome for cancer markers. Clin Cancer Res 11:3163–3169
7. Huang PH, Mukasa A, Bonavia R, Flynn RA, Brewer ZE, Cavenee WK et al (2007) Quantitative analysis of EGFRvIII cellular signaling networks reveals a combinatorial therapeutic strategy for glioblastoma. Proc Natl Acad Sci U S A 104:12867–12872
8. Olsen JV, Blagoev B, Gnad F, Macek B, Kumar C, Mortensen P, Mann M (2006) Global, in vivo, and site-specific phosphorylation dynamics in signaling networks. Cell 127:635–648
9. Macek B, Mann M, Olsen JV (2009) Global and site-specific quantitative phosphoproteomics: principles and applications. Annu Rev Pharmacol Toxicol 49:199–221
10. Rush J, Moritz A, Lee KA, Guo A, Goss VL, Spek EJ et al (2005) Immunoaffinity profiling of tyrosine phosphorylation in cancer cells. Nat Biotechnol 23:94–101
11. Ong SE, Blagoev B, Kratchmarova I, Kristensen DB, Steen H, Pandey A, Mann M (2002) Stable isotope labeling by amino acids in cell culture, SILAC, as a simple and accurate approach to expression proteomics. Mol Cell Proteomics 1:376–386
12. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S et al (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5:R80
13. Mueller LN, Rinner O, Schmidt A, Letarte S, Bodenmiller B, Brusniak MY et al (2007) SuperHirn – a novel tool for high resolution LC-MS-based peptide/protein profiling. Proteomics 7:3470–3480
14. Cox J, Mann M (2008) MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. Nat Biotechnol 26:1367–1372
15. Cox J, Matic I, Hilger M, Nagaraj N, Selbach M, Olsen JV, Mann M (2009) A practical guide to the MaxQuant computational plat-

form for SILAC-based quantitative proteomics. Nat Protoc 4:698–705

16. Tusher VG, Tibshirani R, Chu G (2001) Significance analysis of microarrays applied to the ionizing radiation response. Proc Natl Acad Sci U S A 98:5116–5121
17. Bonferroni CE (1936) Teoria statistica delle classi e calcolo delle prababilita. Publicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenzi 9:3–62
18. Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6:65–70
19. Hochberg Y (1988) A sharper Bonferroni procedure for multiple tests of significance. Biometrika 75:800–803
20. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc B 57:289–300
21. Storey JD, Tibshirani R (2003) Statistical significance for genomewide studies. Proc Natl Acad Sci U S A 100:9440–9445
22. Xie Y, Pan W, Khodursky AB (2005) A note on using permutation-based false discovery rate estimates to compare different analysis methods for microarray data. Bioinformatics 21:4280–4288
23. Jiao S, Zhang S (2008) On correcting the overestimation of the permutation-based false discovery rate estimator. Bioinformatics 24: 1655–1661
24. Fisher RA (1935) The logic of inductive inference. J Royal Stat Soc 98:39–54
25. Zeeberg BR, Feng W, Wang G, Wang MD, Fojo AT, Sunshine M et al (2003) GoMiner: a resource for biological interpretation of genomic and proteomic data. Genome Biol 4:R28
26. Maere S, Heymans K, Kuiper M (2005) BiNGO: a Cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. Bioinformatics 21:3448–3449
27. Ackermann M, Strimmer K (2009) A general modular framework for gene set enrichment analysis. BMC Bioinformatics 10:47
28. Goeman JJ, Buhlmann P (2007) Analyzing gene expression data in terms of gene sets: methodological issues. Bioinformatics 23:980–987
29. McDermott U, Sharma SV, Dowell L, Greninger P, Montagut C, Lamb J et al (2007) Identification of genotype-correlated sensitivity to selective kinase inhibitors by using high-throughput tumor cell line profiling. Proc Natl Acad Sci U S A 104:19936–19941
30. Zhou Y, Cras-Meneur C, Ohsugi M, Stormo GD, Permutt MA (2007) A global approach to identify differentially expressed genes in cDNA (two-color) microarray experiments. Bioinformatics 23:2073–2079
31. Kanehisa M, Goto S (2000) KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res 28:27–30
32. Hay N, Sonenberg N (2004) Upstream and downstream of mTOR. Genes Dev 18: 1926–1945
33. Stemmann O, Zou H, Gerber SA, Gygi SP, Kirschner MW (2001) Dual inhibition of sister chromatid separation at metaphase. Cell 107:715–726
34. Mueller LN, Brusniak MY, Mani DR, Aebersold R (2008) An assessment of software solutions for the analysis of mass spectrometry based quantitative proteomics data. J Proteome Res 7:51–61
35. Kitteringham NR, Jenkins RE, Lane CS, Elliott VL, Park BK (2009) Multiple reaction monitoring for quantitative biomarker analysis in proteomics and metabolomics. J Chromatogr B Analyt Technol Biomed Life Sci 877:1229–1239
36. Ritz A, Shakhnarovich G, Salomon AR, Raphael BJ (2009) Discovery of phosphorylation motif mixtures in phosphoproteomics data. Bioinformatics 25:14–21

Part II

Databases

Chapter 4

The Origin and Early Reception of Sequence Databases

Joel B. Hagen

Abstract

Emerging areas of scientific research never arise in a social or intellectual vacuum, but must establish themselves in relation to well-established disciplines. This necessity poses challenges for scientists who must not only create a new disciplinary identity, but must also defend their research from criticism and even condescension from other scientists. The early use of sequence databases provides an excellent case study for examining the challenges facing novel sciences. The need for sequence databases grew out of protein sequencing in biochemistry beginning in the late 1950s. The rapid increase in the number of sequences made databases an attractive resource, but protein biochemists often considered building, managing, and doing research with databases a "second-rate" science. Similarly, computational biologists who used databases and digital computers to study evolutionary phenomena faced criticism from more traditional evolutionary biologists. In retrospect, one can see this early computational biology as laying important foundations for the bioinformatics, molecular evolution, and molecular systematics of today. However, within the context of the 1960s, establishing a scientific identity posed serious challenges for Margaret Dayhoff, Walter Fitch, and Russell Doolittle and other computational biologists who used computers and databases to investigate evolutionary problems.

1. Introduction

"Changes in technology in the past decade have had such an impact on the way that molecular evolution research is done that it is difficult now to imagine a world without genomics or the internet"(1).

"The Internet has become so commonplace that it is hard to imagine that we were living in a world without it only 10 years ago"(2).

Scientists have done an admirable job of documenting the recent histories of new fields, such as bioinformatics, genomics, and proteomics (1–3). Written by participants in emerging areas of research, these histories highlight the rapid technical advances that are dramatically reshaping our understanding of the living world.

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696,
DOI 10.1007/978-1-60761-987-1_4,

They also document the ways that various groups of scientists define novel areas of research to form new disciplinary identities.

Despite the usefulness of participant histories, these accounts leave much unsaid about the origins of the fields they describe. As the above quotations suggest, it is often difficult for scientists to take a longer view of history and see important connections with scientific developments that took place long before the advent of large molecular databases, powerful desktop computers, and the internet. Today, some biologists are able to do all of their research on computers using publicly accessible databases (1). How did this computational approach to biological research originate? What opportunities and challenges did the earliest pioneers face in this new area of research?

It may be useful to consider how sequence databases arose in the first place, and how scientists mined these new collections of data using early mainframe computers. Several decades before the advent of the internet and personal computers, an earlier generation of scientists used the burgeoning collections of protein sequence data to lay the preliminary foundations for the sciences of today. Looking back half a century, we can see that the challenges these scientists faced were not simply technical, but also involved institutional, disciplinary, and other important social factors. A new area of science never emerges in an intellectual or social vacuum, and so it must differentiate itself both from older practices and from existing disciplines. The early history of computational biology provides a particularly rich case study for examining the difficult challenges of discipline formation. In a shifting disciplinary landscape, individual scientists – particularly younger scientists – faced both opportunities and serious challenges because they often had to choose between staying with a well-established field, or moving to an emerging area of research that was viewed with skepticism or condescension by other scientists.

2. The First Molecular Databases

The need for sequence databases became obvious soon after Frederick Sanger reported the complete primary structure of insulin in a series of articles in the 1950s (4–8). During this path-breaking research, Sanger and his colleagues discovered that there were minor differences in the amino acid sequences of insulin drawn from cows, pigs, sheep, horses, and whales. Sanger believed that these differences held the key to explaining how insulin functioned as a hormone (4, 5). Assuming that the "active center" of the hormone must remain invariant, perhaps he could identify this critical part of the protein by disregarding those regions that varied from species to species. Ultimately, this next step in Sanger's research program was not as fruitful as he had hoped, and he

turned his attention to devising methods for sequencing nucleic acids (for which he won a second Nobel Prize). Nonetheless, Sanger's comparative approach marked the beginning of molecular databases as other protein biochemists began informally collecting amino acid sequences as a means for studying molecular structure and function (8). Although sequences were important, Sanger probably would not have considered collecting them to be a serious scientific activity (9). For him, science meant experimentation at the bench. Ironically, although his research was the point of departure for larger, more comprehensive sequence databases, Sanger's philosophy of science minimized the scientific status of collecting sequences and managing databases (8–10).

During the late 1950s and early 1960s, several laboratories worked to determine the amino acid sequences of other proteins, notably glucagon, ribonuclease, cytochrome *c*, and hemoglobin. Looking back, Emil Smith (57) described the two decades following World War II as the "heroic period" of protein chemistry. Sanger's work epitomized this image of a master chemist deciphering the primary structure of a complex molecule. However, the Edman degradation reaction quickly replaced Sanger's less elegant methods, and the entire sequencing process became fully automated by 1970. At the same time, research on DNA was eclipsing protein studies – both in prestige and funding (12). As Smith ruefully acknowledged, sequencing proteins, which a decade earlier was worthy of a Nobel Prize, had become a routine task suitable for a competent laboratory technician. By today's standards, the number of known protein sequences was miniscule in 1970. However, keeping track of slightly more than 1,000 sequences was a serious challenge that posed both technical and vexing social problems for scientists who tried to create and manage the first comprehensive molecular databases.

Together with her colleagues at the National Biomedical Research Foundation (NBRF), Margaret Dayhoff gathered all of the known protein sequences in a book published in 1965: *The Atlas of Protein Sequence and Structure* (53). The collection, which can be viewed as the first attempt to create a comprehensive molecular database, included the amino acid sequences of about 70 proteins from various species; mostly variants of hemoglobin, cytochrome *c*, and fibrinopeptides. Dayhoff's small team laboriously collected these sequences from the published literature, although Dayhoff also encouraged biochemists to submit unpublished sequences. Finding sequences was not easy. Indeed, one of the justifications for creating the *Atlas* was the troublesome and time-consuming literature searches that individual scientists had to undertake to find sequences for their research. Although Dayhoff's team used the newly computerized Medical Literature Analysis and Retrieval System (MEDLARS) established in 1964 at the National Institutes of Health, they complained about the difficulty of locating all of the relevant literature (13). For example, the

term "sequence" was not widely used as a keyword during the mid-1960s. Managing the collection was also a time-consuming task. Dayhoff's team used computers to store the molecular data, but they had to manually type and proofread each sequence. There was no efficient way of sharing data, although later editions of the *Atlas* were available on magnetic tape. Thus, from the very beginning, database managers faced familiar problems: how to organize, catalog, and distribute overwhelmingly large amounts of data (8, 9). When the second edition of the *Atlas* doubled in size in 1966, the editors described the influx of new sequences as an "information explosion" (13). Subsequent editions continued to grow rapidly. Still, by the end of the decade the collection contained only about 1,000 sequences.[1]

The NBRF, which published the *Atlas*, was on the cutting edge of using computers in biology and medicine. The director, Robert Ledley, was a leading advocate for computational biology and was writing a general survey, *Use of Computers in Biology and Medicine* (56). Dayhoff's career spanned the transition from an older tradition of electrical and mechanical calculators to the early mainframe computers. During the late 1940s, before digital computers were available, she had used IBM punched card business machines to calculate the resonance energies of polycyclic organic molecules for her Ph.D. dissertation in quantum chemistry at Columbia University (14, 15). After fellowships at Rockefeller University and the University of Maryland, Dayhoff was hired by Ledley to write FORTRAN programs for mainframe computers to aid in determining the amino acid sequences of protein molecules. This work led directly to collecting sequences and using them for evolutionary research.[2] Thus, although the NBRF was a biomedical center and Dayhoff used the potential medical benefits of her research as a justification for publishing the *Atlas*, most of her research with protein sequences dealt with evolutionary questions that had no immediate medical applications.

Despite its small size, the *Atlas* proved to be more than a collection of data. Dayhoff and her colleagues at the NBRF also used the book as a vehicle for reporting their diverse research using computers to study proteins. For Dayhoff, computer programming went hand-in-hand with building databases. Her early work with the *Atlas* marked two important transitions in computational biology. First, the growing number of sequences allowed a more comprehensive comparative analysis of proteins than had been possible for Sanger and other biochemists with the handful of sequences available just a few years earlier. For example, the small electron transport protein cytochrome *c* turned out to be relatively easy to sequence (16) and the second edition of the *Atlas* listed the sequences from eighteen species, including bacteria, fungi, plants, and a wide variety of animals. This growing collection of data allowed Dayhoff to explore methods of aligning

sequences, developing statistical models for amino acid substitutions, and building phylogenetic trees showing evolutionary relationships among the proteins and species that contained them. The second important transition was from hand calculations to computer-assisted analysis. Even before complete sequences were available, biochemists had used similarities and differences in amino acid sequences to align homologous proteins and infer phylogenetic relationships (17–19). These early phylogenetic trees were intuitively constructed on the basis of hypothetical stepwise substitutions of one amino acid for another. Usually, no attempt was made to weight substitutions based on the number of mutations or frequency with which they occurred. Using the data from the *Atlas* and the calculating power of a mainframe computer, Dayhoff and her colleagues were able to develop probabilistic models of amino acid substitutions. This approach eventually led Dayhoff to develop what she referred to as the Point Accepted Mutation (PAM) matrix, an innovation that formed an early foundation for later approaches for aligning sequences and searching for homologies among proteins. But even in 1966, Dayhoff and her colleague Richard Eck used their earliest substitution models to contruct phylogenetic trees based on cytochrome *c* (13, 20). Their computational technique also allowed inferences about ancestral sequences at the nodes of the tree and estimations of the times of divergence.

Although biochemists recognized that Dayhoff's *Atlas of Protein Sequence and Structure* was an important innovation, its scientific status remained ambiguous. From the perspective of experimental biochemists, collecting sequences that other scientists had discovered was not fundamental research, but was more akin to natural history collecting or editorial work (8–10). The fact that Dayhoff and most of her colleagues were women undoubtedly contributed to the perception that building sequence databases was not science.[3] Using computers to generate phylogenetic trees based on amino acid sequences was a novelty that fell outside mainstream biochemistry and traditional evolutionary biology (21). Although Dayhoff predicted that this new type of research would have practical benefits for biomedical sciences, she could provide few concrete examples of what such an evolutionary medicine might contribute to society. The novelty and ambiguity of this new computational research also led some biochemists to question the importance of Dayhoff's work (8, 9). The problem of establishing a scientific niche for databases also had important consequences for funding the *Atlas*. Collecting the sequences was originally supported as part of a grant to Dayhoff from NIH to develop computational methods for sequence analysis. Other agencies also indirectly supported the database through grants for Dayhoff's studies on molecular evolution. However, as the number of new sequences rapidly increased, the cost of collecting,

proofreading, and entering the sequences to the database escalated. NIH became increasingly unwilling to support this part of Dayhoff's work and repeatedly threatened to terminate the project.

This funding problem was compounded by Dayhoff's expectation that biochemists would submit unpublished sequences to the *Atlas*. At first, she used a reward system where contributors received free copies of the *Atlas* in exchange for submitting sequences. However, because of lack of financial support from funding agencies Dayhoff had to sell later editions of the book. Dayhoff's attitude toward sequences was reminiscent of the natural history tradition of collecting specimens, but it violated well-entrenched attitudes toward privacy, priority, and intellectual property typical of the experimental sciences (9, 10). For biochemists, a sequence was private property until it was published and priority was assigned to the discoverer. Critics charged that Dayhoff was making unfair use of unpublished sequence data for her own research, while at the same time selling the *Atlas* as a commercial venture (8, 9, 22).

3. The Problems of Establishing a Disciplinary Identity

David Lipman, Director of the National Center for Biotechnology Information, has famously described Dayhoff as both "the mother and father of bioinformatics" (23). However, in the 1960s bioinformatics did not yet exist, and using computational methods to investigate evolutionary questions lay at the fringes of other well-established disciplines. Indeed, the use of computers and databases for evolutionary research was not even discussed in general surveys of computational biology, such as Ledley's *Use of Computers in Biology and Medicine* – this despite the fact that Ledley was the director of the NBRF where Dayhoff worked. Combining perspectives from evolutionary biology, protein biochemistry, and the biomedical sciences would eventually grow into successful lines of research that we now identify as bioinformatics and molecular evolution, but both historians and participants have documented the difficulty of forging these disciplinary identities (22, 24). In order to do so, scientists needed not only large databases and powerful computers, but also a compelling intellectual rationale that would set the new computational biology apart from well-established disciplines, such as comparative biochemistry and traditional evolutionary biology.

Shortly after publishing the structure of DNA, Francis Crick provided an early justification for studying the evolutionary implications of protein sequences. Reflecting on Sanger's successful sequencing of insulin, Crick wrote: "Biologists should realize that

before long we shall have a subject which might be called "protein taxonomy," the study of the amino acid sequences of proteins of an organism and the comparison of them between species. It can be argued that these sequences are the most delicate expression possible of the phenotype of an organism and that vast amounts of evolutionary information may be hidden away within them" (25). Historians have documented how Crick and other molecular biologists used the idea of "information" to distance their new science from the traditional practices of biochemistry (4). However, Crick's remarks also highlighted a division between a new molecular approach to studying evolution and the organismal approaches to evolutionary biology that were part of the naturalist tradition. During the 1960s, interactions between organismal and molecular evolutionists were often contentious (21, 26–29). Less obvious is the fact that sequence analysis also had to compete with a number of other experimental techniques for studying molecular evolution (30). Thus, Dayhoff and other computational scientists faced serious challenges when they tried to carve out a disciplinary niche in an already crowded field.

Among those who were most interested in creating a new discipline combining biomedical, evolutionary, and computational perspectives were Linus Pauling and Emile Zuckerkandl. In a series of influential articles, they outlined a new field that they dubbed "chemical paleogenetics" (19, 31, 32). Like Crick's idea of "protein taxonomy," chemical paleogenetics was a conscious attempt to molecularize areas of biology that were part of the natural history tradition (21). Indeed, Dayhoff, Zuckerkandl, and Pauling used the rhetoric of objectivity and ties to experimental laboratory science to argue that their new sequence-based approaches would revolutionize the study of phylogeny – and perhaps replace natural history (20, 31). Although the primary focus of this new science was to be the evolution of proteins such as hemoglobin and the species that contained them, biomedical issues also influenced the thinking of chemical paleogeneticists (28). Pauling's interest in hemoglobin arose within the context of studying the molecular basis of sickle cell anemia. He was committed to the belief that understanding the structure of proteins would provide the key to a new "molecular medicine" (33). During the 1960s, Pauling became particularly interested in evolution because of his concerns about the mutagenic effects of radiation and his strong, public opposition to atomic weapons (28).

Neither protein taxonomy nor chemical paleogenetics became recognizable scientific disciplines, and these nascent fields were quickly absorbed by a more encompassing molecular evolution. Nonetheless, Zuckerkandl and Pauling's articles from the early 1960s provided a compelling rationale for using computational approaches to studying molecular evolution and disease (21, 24, 28). From this perspective, proteins and nucleic acids were "documents of

evolutionary history" that contained all of the information needed to detect homology, reconstruct phylogeny, date important evolutionary events, and reconstruct ancestral macromolecules from the deep past. By focusing attention on the genetic information contained in a linear sequence of amino acids, Zuckerkandl and Pauling not only highlighted a particular subset of evolutionary problems for future study, but they also purposely circumscribed what would become central to molecular evolution (28). For example, despite some early interest in studying evolutionary questions about protein structure and function, this potentially fruitful area of research became marginalized in the molecular evolution that emerged during the late 1960s. As we shall see, this process of defining which problems were central and which were peripheral had a significant impact on the career trajectories of the early molecular evolutionists who used databases for their research.

Today, biologists take the idea of genetic information for granted, but Sanger and the other biochemists who first sequenced proteins were working from quite a different perspective that emphasized protein structure rather than genetic information (4, 8). This difference in perspective was a source of considerable tension between experimental biochemists, and the newer molecular evolutionists who thoroughly implemented the idea of molecular information in their work. From the perspective of molecular evolution, understanding how the history of proteins was caused by evolutionary mechanisms became more important than understanding the chemical mechanisms by which a protein contributed to cellular function. For example, comparing myoglobin and the various amino acid sequences making up the oxygen-carrying protein hemoglobin, molecular evolutionists demonstrated how all of the various globin polypeptides had descended from a common ancestral molecule through a process of gene duplication, mutation, and natural selection (18, 19). However, although these scientists were interested in the biochemistry of the globins, sequence comparisons were of limited value for understanding protein function.

Assuming that mutations occurred at a relatively constant rate, Zuckerkandl and Pauling also claimed that they could use sequence differences as a molecular clock to date when various globins had diverged. The molecular clock quickly became one of the most important concepts in molecular evolution, even though the details of how the clock worked remained controversial (28, 34, 35). For example, Zuckerkandl and Pauling assumed that natural selection was the basic mechanism for molecular evolution, but the molecular clock soon became closely associated with the neutral theory of molecular evolution as developed by Motoo Kimura, Thomas Jukes, and Jack King (27, 34, 36, 37). Sequence data from Dayhoff's *Atlas* played an important role in the development of the neutral theory and served as an important

link between the new molecular evolution and classical population genetics (36).

Thinking in terms of macromolecular information also provided an important conceptual link between studying sequences and using computers. Not surprisingly, Dayhoff and other computational biologists were strongly attracted to the ideas of Zuckerkandl and Pauling (20, 38). If proteins were truly "documents of evolutionary history," computers quickly became necessary tools for deciphering them. The idea that proteins carry a record of evolutionary history also complemented a new emphasis on hypothesis testing in evolutionary biology and systematics (39, 40). When scientists compared sequences of a protein such as cytochrome *c* from several species, an enormous number of possible phylogenetic trees could be generated. If these trees were thought of as alternative phylogenetic hypotheses, then computers were required for evaluating the thousands or millions of alternatives and identifying the most likely possibility. Developing programs to infer phylogenies from sequence data soon became a primary focus – and often a source of controversy – in molecular evolution and systematics (39, 41).

Despite its appeal to Dayhoff and other computational biologists, Zuckerkandl and Pauling's new perspective faced criticism from many evolutionary biologists and experimental biochemists. Traditional evolutionary biologists found the idea of a molecular clock simplistic, and they questioned the validity of phylogenetic conclusions based on comparisons of amino acid sequences (21, 26–29, 34). Some experimental biochemists were equally critical of molecular evolution. Comparing sequences to answer evolutionary questions was a major departure from the way that Frederick Sanger and other experimental biochemists studied protein function in a mechanistic and largely nonevolutionary context. From this perspective, the new molecular evolution seemed to hark back to a nonexperimental natural history tradition. Perhaps not surprisingly, Zuckerkandl and Pauling (31) complained that their research faced skepticism not only from traditional evolutionary biologists who doubted that sequences could be used to answer important evolutionary questions, but also from experimental biochemists who considered evolutionary studies to be "second-rate" science.

4. The Challenges of Using Computers in Molecular Evolution

The tensions among traditional evolutionary biologists, experimental biochemists, and the new computational biologists had important consequences for young scientists who entered the field during the 1960s. For example, Walter Fitch established his

early scientific reputation by writing a computer program to reconstruct the phylogeny of various plants, animals, and fungi using amino acid sequences of cytochrome *c* (42). Although published after Dayhoff's similar attempt to reconstruct phylogeny using cytochrome *c*, the article became a citation classic (43). Publishing the work in the high-profile journal *Science*, ensured that Fitch's computational approach would reach a very broad audience. The study quickly became a textbook example of how to deduce evolutionary history using amino acid sequences, and it propelled Fitch on a career path that relied heavily on computational methods to solve evolutionary problems (43, 44).

Fitch had earned his Ph.D. in comparative biochemistry and was an assistant professor in physiological chemistry at the University of Wisconsin when he began collaborating with Emanuel Margoliash on the cytochrome *c* project. Margoliash was one of the pioneers of protein sequencing and had elucidated the first cytochrome *c* sequence from horses. By the time he met Fitch in 1966, Margoliash had an informal collection of twenty cytochrome *c* variants available for the phylogenetic analysis – including ten unpublished sequences (43). Fitch described this new data set as a "windfall" for his plan to use a computer to generate phylogenetic trees.

Margoliash had very broad biochemical interests, and he was pursuing an ambitious research program on the structure, function, and evolution of cytochrome *c*. Also collaborating with Margoliash was Richard Dickerson, who had just completed a low resolution X-ray diffraction analysis of cytochrome *c*. Margoliash, Fitch, and Dickerson (45) were confident that their combined evolutionary perspectives could unify traditional biochemistry and the new information-based approach championed by Zuckerkandl and Pauling: "Molecular evolution is thus likely both to bridge the gulf between the informational and structural areas of knowledge and to provide a fascinating frontier." Yet, when the three scientists attempted to use sequence data to understand three-dimensional structure and function, they encountered a paradox. Cytochrome *c* from different species varied considerably in primary structure, yet molecules from fungi, plants, and invertebrates reacted identically with mammalian cytochrome oxidase in vitro. All of the various sequences apparently folded into precisely the same three-dimensional conformation. Although comparisons of cytochrome sequences from different species provided important hints about the structure and function of the molecule, Dickerson needed models of both the oxidized and reduced forms of cytochrome *c* to understand how this "molecular machine" worked (46). The collaboration between Dickerson and Fitch was short-lived, and by 1970 Fitch moved decisively toward computational studies of evolutionary mechanisms that largely ignored questions of structure and function.

The abortive partnership of Margoliash, Fitch, and Dickerson highlights the difficulties of bridging the differences between a well-established experimental biochemistry and a new molecular evolution based heavily on the idea of molecular information. By the end of the 1960s, sequencing proteins was relatively easy and sequence databases were growing rapidly, but X-ray diffraction studies remained arduous and time consuming. Thus, although scientists like Dickerson were confident that there was direct causal chain linking primary structure with the three dimensional shape and function of a protein, technical limitations prevented this from becoming a central focus of research using sequence databases. At the same time, molecular evolution was coalescing around a core of problems that were readily amenable to study using the growing body of amino acid sequences and digital computers. For molecular evolutionists, "molecular information" increasingly meant information about how proteins evolved, not how they worked within the cell. Thus, establishing a new disciplinary identity for molecular evolution meant emphasizing the differences between informational and structural approaches.

Given the success of his first attempt to use a computer to reconstruct phylogenies based on amino acid sequences of cytochrome *c*, it is perhaps not surprising that Fitch continued this fruitful line of research. However, moving from biochemistry to molecular evolution and systematics had important consequences for his career. Traditional evolutionary biologists had their own approaches to reconstructing phylogeny. Indeed, the 1960s was a decade of extreme ferment in systematic biology pitting rival schools of evolutionary taxonomists, numerical taxonomists, and cladists (24, 47). Because his research forced him to interact closely with these competing groups, Fitch had to worry about evolutionary concepts and methods that were of little concern to biochemists like Margoliash or Dickerson. For Margoliash (38), the differences between different algorithms for constructing phylogenetic trees were not crucial because they all produced the same general results, but Fitch soon learned that there were important philosophical implications of different computational approaches. For example, although many biochemists used homology as a synonym for similarity of amino acid sequences, evolutionary biologists (particularly cladists) defined homology as descent from a common ancestor. Because he began publishing articles in *Systematic Zoology* and other journals read by evolutionary biologists and systematists, Fitch had to take this distinction seriously. He significantly modified his tree-building algorithms to reconstruct ancestral sequences at the nodes of the tree (41). Detecting molecular homologies became not only an interesting computational problem for Fitch, but also one with important conceptual and philosophical implications that he could not ignore if his work was to be taken seriously by other systematists (43, 44, 48).

Russell Doolittle was another biochemist whose reputation became closely linked with the use of computers and sequence databases. However, his experiences during the 1960s were different from Fitch's in a number of important ways. Doolittle became interested in evolution while he was completing his Ph.D. in biochemistry at Harvard University (22, 49). His dissertation research involved comparisons of the blood-clotting mechanism in various vertebrates (50). Using thrombin from lampreys, Doolittle compared its effect on clotting rates when combined with fibrinogen from cows and lampreys. Thrombin removes a piece of the fibrinogen molecule to produce the active clotting protein, fibrin. Doolittle used paper electrophoresis to separate the small fibrinopeptides that were removed from fibrinogen during this process. Although he did not determine the amino acid sequences of the fibrinopeptides, Doolittle was able to estimate the amino acid composition of the molecules.

While on a postdoctoral fellowship in the laboratory of Birger Blombäck at the Karolinksa Institute in Stockholm, Doolittle learned how to use the Edman degradation reaction to sequence fibrinopeptides. Doolittle and Blombäck quickly built up an informal database of fibrinopeptide sequences from a variety of mammals. Sequences from different species varied in length, but Doolittle and Blombäck used invariant amino acids as "alignment markers" to identify regions of the molecules resulting from insertions or deletions (17). Comparing the aligned peptides, Doolittle and Blombäck proposed a stepwise evolutionary process leading to a branching phylogenetic tree. Constructed intuitively and without the use of a computer, Doolittle and Blombäck followed an informal method of tree building used earlier by Vernon Ingram (18) and Zuckerkandl and Pauling (19) to reconstruct the evolution of the various globin polypeptides. Doolittle and Blombäck used their sequences to hypothesize the evolutionary relationships among several cloven-hoofed mammals (artiodactyls) from which they sampled the fibrinopeptides. The results, some of which contradicted well-established phylogenetic relationships, were controversial. For example, the simplest phylogenetic tree based on fibrinopeptides suggested that goats and sheep were more closely related to reindeer than to cows. George Gaylord Simpson, a leading evolutionary biologist and the foremost expert on mammalian paleontology and taxonomy, was highly critical of this evolutionary claim. In their article, Doolittle and Blombäck (17) acknowledged that their simplest fibrinopeptide tree was contradicted by "a very large body of biological evidence," and they cited personal communication with Simpson. Simpson's letter to Doolittle provides a detailed critique of the biochemists' evolutionary and taxonomic claims and of the use of fibrinopeptides for phylogenetic reconstruction, more generally.

The disagreement between Simpson and Doolittle involved an empirical question open to testing and refutation. However, in the context of the 1960s, the hypothesis-testing was embedded in a broader debate among evolutionary biologists about the validity of new molecular techniques (21, 27). Simpson actively engaged molecular evolutionists at meetings and in publications in a critique that he characterized as a "clarifying confrontation." This confrontation involved philosophical commitments as well as purely scientific issues. Molecular evolutionists who viewed proteins as "documents of evolutionary history" often argued that protein sequences had a privileged status that set them apart from other biological characteristics (25, 31, 32). Because fibrinopeptides accumulated mutations quite rapidly, Doolittle and Blombäck were confident that their method could accurately reconstruct the phylogenetic history of a group of very closely related mammals. Conversely, Simpson and other organismal evolutionists argued that molecular data should carry no more weight than paleontological, morphological, and other forms of evidence. Because he believed that the bulk of the evidence contradicted some of Doolittle and Blombäck's hypotheses, Simpson rejected their claims and called into question the usefulness of fibrinopeptides for phylogenetic studies of artiodactyls. Thus, although the phylogenetic relationships among artiodactyls were an empirical question, resolving discrepancies partly depended upon competing philosophies of science. As a biochemist, Doolittle had little knowledge of the rich fossil record of artiodactyls. Therefore, he was arguing with experts in another discipline who not only disagreed with his specific claims, but who were also highly skeptical about the methodology that he employed. This had important practical implications because when Doolittle submitted two grant proposals to the National Science Foundation, Simpson was one of the reviewers. The critical reviews (held in the Simpson archives) reflect Simpson's deep skepticism toward molecular evolution and the use of fibrinopeptide sequences as a method for understanding mammalian phylogeny.

After learning of the computational methods that Fitch had developed for reconstructing phylogenies using cytochrome *c*, Doolittle began using mainframe computers in his research during the late 1960s (22). Although he continued to use fibrinopeptides for phylogenetic analysis of the artiodactyls, Doolittle did not interact with systematists and organismal biologists to the extent that Fitch did. Doolittle did not publish his later phylogenetic articles in *Systematic Zoology*, but in biomedical journals that systematists or mammalogists would not have routinely read. Important as it was, Doolittle's phylogenetic research was only a small part of his broader program to understand the molecular basis of blood clotting in mammals. To this end, he continued to

see himself as a traditional experimental biochemist – albeit one with a strong interest in using computers (22). This disciplinary identification contrasts with Fitch, who moved further away from his biochemical roots as he became increasingly involved with molecular evolution and systematics. The difference is highlighted by the way that the two scientists' approached a common interest in molecular homology. Both Fitch and Doolittle used computers and sequence databases extensively to study homology, and Doolittle was later lionized for discovering unsuspected evolutionary relationships among seemingly unrelated proteins (11, 22, 51). However, because he was not interacting closely with organismal evolutionary biologists, Doolittle was less concerned than Fitch with precisely defining the concept and exploring the philosophical implications of homology.

5. Conclusions

The experiences of Dayhoff, Fitch, and Doolittle during the 1960s illustrate both the opportunities and the challenges that scientists confronted with the advent of protein sequence databases and mainframe computers. None of these scientists was trained in traditional evolutionary biology, but the availability of protein sequences propelled them on career trajectories that were strongly influenced by evolution. The variation in sequences raised compelling evolutionary questions and provided a means for investigating them. Although aligning sequences, searching for homologies, dating evolutionary events, and constructing phylogenetic trees had been done to a limited extent without computers, Dayhoff, Fitch, and Doolittle were at the forefront of efforts to develop a new computational biology. They did this without the benefit of formal training in computer programming, being largely self-taught. Noting the difficulty of getting computer scientists interested in his evolutionary research, Doolittle later described his early computer programming efforts as a "hobby" (22). Without the interactivity provided by the internet and personal computers, Dayhoff, Fitch, and Doolittle used mainframe computers located in centralized computing centers to lay important groundwork for what would become bioinformatics. Yet, even late in his career, Doolittle denied that he was ever a "bioinformatician" (22).

Disciplinary identity in an emerging area of research posed significant challenges for early computational biologists, as the careers of Dayhoff, Fitch, and Doolittle illustrate. Using novel techniques to study evolutionary questions at the fringes of well-established fields opened them to criticism for doing second-rate

science or for applying inappropriate methods to study evolutionary questions. Eventually, the computational methods that Dayhoff, Fitch, and Doolittle pioneered became mainstream tools in molecular evolution, but in the 1960s molecular evolution was just beginning to take form. Computers only gradually became recognized as credible scientific instruments by biologists. Even in the late 1970s, computational biologists sometimes faced condescension from laboratory scientists who considered them "failed researchers" playing with computers (52). Today, online databases and powerful computers are such an integral part of modern biology that it is difficult to imagine doing research without the internet. However, this cutting-edge research rests on a foundation that extends back to the very beginning of the computer age.

6. Notes

1. The Atlas was published at irregular intervals between 1965 and 1978. It gave rise to the online database, The Protein Information Resource (PIR), established by the National Biomedical Research Foundation in 1984 at Georgetown University (8, 14, 15).
2. Dayhoff's research interests spanned a very wide range of evolutionary questions, including the evolution and classification of proteins, the origins of life, and the thermodynamics and evolution of atmospheres on other planets (14, 15).
3. Dayhoff was acutely aware of the challenges facing women in science. After her death in 1983, the Biophysical Society (of which she was the first female president) established an award in her name to given annually to an outstanding woman at the beginning of her research career (8, 14, 15).
4. Correspondence between Simpson and Doolittle, as well as Simpson's reviews of grant proposals written by Doolittle, are part of the George Gaylord Simpson Papers at the American Philosophical Society library.
5. Both scientists and historians have emphasized the controversies and conflicts between traditional evolutionary biologists and molecular evolutionists. Real as these controversies were, it is equally important to note that many molecular evolutionists deeply respected the expertise of Simpson, Ernst Mayr, and other organismal biologists. An extensive correspondence with numerous molecular evolutionists can be found in the Simpson papers (21).

References

1. Wolfe KH, Li WH (2003) Molecular evolution meets the genomic revolution. Nat Genet Suppl 33:255–265
2. Kanehisa M, Bork P (2003) Bioinformatics in the post-sequence era. Nat Genet Suppl 33:305–310
3. Patterson SD, Aebersold RH (2003) Proteomics: the first decade and beyond. Nat Genet Suppl 33:311–323
4. de Chadarevian S (1996) Sequences, conformation, information: biochemists and molecular biologists in the 1950s. J Hist Biol 29:361–386
5. de Chadarevian S (1999) Protein sequencing and the making of molecular genetics. Trends Biochem Sci 24:203–206
6. Sanger F (1959) The chemistry of insulin. Science 129:1340–1344
7. Sanger F (1988) Sequences, sequences, sequences. Ann Rev Biochem 57:1–28
8. Strasser BJ (in press) Collecting, comparing, and computing sequences: the making of Margaret O. Dayhoff's atlas of protein sequence and structure. J Hist Biol.
9. Strasser BJ (2006) Collecting and experimenting: the moral economies of biological research, 1960s–1980s. Preprints Max-Planck Inst Hist Sci 310:105–123
10. Strasser BJ (2008) GenBank – natural history in the 21st century? Science 322:537–538
11. Smith TF (1990) The history of the genetic sequence databases. Genomics 6:701–707
12. Schachman HK (1979) Summary remarks: a retrospect on proteins. In: Srinivasan PR, Fruton JS, Edsall JT (eds) The origins of modern biochemistry: a retrospect on proteins, vol 325. Annals of the New York Academy of Sciences, New York, pp 363–373
13. Eck RV, Dayhoff MO (1966) The atlas of protein sequence and structure 1966. National Biomedical Research Foundation, Silver Spring, MA
14. Hunt LT (1983) Margaret O. Dayhoff, 1925–1983. DNA 2:97–98
15. Hunt LT (1984) Margaret O. Dayhoff, 1925–1983. Bull Math Biol 46:467–472
16. Margoliash E, Schejter A (1996) How does a small protein become so popular?: a succinct account of the development of our understanding of cytochrome *c*. In: Scott RA, Mauk AG (eds) Cytochrome c: a multidisciplinary approach. University Science Books, Sausalito, CA
17. Doolittle RF, Blömback B (1964) Amino acid sequence investigations of fibrinopeptides from various mammals: evolutionary implications. Nature 202:147–152
18. Ingram VM (1961) Gene evolution and the haemoglobins. Nature 189:704–708
19. Zuckerkandl E, Pauling L (1963) Chemical paleogenetics: molecular "restoration studies" of extinct forms of life. Acta Chem Scand 17:S9–S16
20. Dayhoff MO (1969) Computer analysis of protein evolution. Sci Am 221:87–95
21. Hagen JB (1999) Naturalists, molecular biologists, and the challenges of molecular evolution. J Hist Biol 32:321–341
22. Doolittle RF (2000) On the trail of protein sequences. Bioinformatics 16:24–33
23. Moody G (2004) Digital code of life: how bioinformatics is revolutionizing science, medicine, and business. Wiley, Hoboken, NJ
24. Hagen JB (2000) The origins of bioinformatics. Nat Rev Genet 1:231–236
25. Crick FHC (1958) On protein synthesis. Symp Soc Exp Biol 12:138–163
26. Aronson J (2002) Molecules and monkeys: George Gaylord Simpson and the challenge of molecular evolution. Hist Philos Life Sci 24:441–465
27. Dietrich MR (1998) Paradox and persuasion: negotiating the place of molecular evolution within evolutionary biology. J Hist Biol 31:85–111
28. Morgan GJ (1998) Emile Zuckerkandl, Linus Pauling and the molecular evolutionary clock, 1959–1965. J Hist Biol 31:155–178
29. Sommer M (2008) History in the gene: negotiations between molecular and organismal anthropology. J Hist Biol 41:473–528
30. Hagen JB (in press). Waiting for Sequences: Morris Goodman, Immunodiffusion Experiments, and the Origins of Molecular Anthropology. J Hist Biol.
31. Zuckerkandl E, Pauling L (1965) Evolutionary divergence and convergence in proteins. In: Bryson V, Vogel HJ (eds) Evolving genes and proteins. Academic Press, New York, pp 97–166
32. Zuckerkandl E, Pauling L (1965) Molecules as documents of evolutionary history. J Theor Biol 8:357–366
33. Strasser BJ (1999) Sickle cell anemia, a molecular disease. Science 286:1488–1490
34. Dietrich MR (1994) The origins of the neutral theory of molecular evolution. J Hist Biol 27:21–59
35. Kumar S (2005) Molecular clocks: four decades of evolution. Nat Rev Genet 6: 654–662

36. Kimura M (1983) The neutral theory of molecular evolution. Cambridge University Press, Cambridge
37. Suárez E, Barahona A (1996) The experimental roots of the neutral theory of molecular evolution. Hist Philos Life Sci 18:55–81
38. Margoliash E (1972) The molecular variation of cytochrome *c* as a function of the evolution of species. Harvey Lect 66:177–247
39. Hagen JB (2001) The introduction of computers into systematic research in the united states during the 1960s. Stud His Philos Biol Biomed Sci 32:291–314
40. Hagen JB (2003) The statistical frame of mind in systematic biology from quantitative zoology to biometry. J Hist Biol 36:353–384
41. Felsenstein J (2004) Inferring phylogenies. Sinauer, Sunderland, MA
42. Fitch WM, Margoliash E (1967) Construction of phylogenetic trees. Science 155:279–284
43. Fitch WM (1988) This week's citation classic. Curr Contents 19(27):16
44. Fitch WM (1987) This week's citation classic. Curr Contents 18(27):14
45. Margoliash E, Fitch WM, Dickerson RE (1968) Molecular expression of evolutionary phenomena in the primary and tertiary structures of cytochrome *c*. Structure, function, and evolution in proteins. Brookhaven Symp Biol 21(2):259–305
46. Dickerson RE, Geis I (1969) The structure and action of proteins. Harper & Row, New York
47. Hull DL (1988) Science as a process: an evolutionary account of the social and conceptual development of science. University of Chicago Press, Chicago
48. Fitch WM (2000) Homology: a personal view on some of the problems. Trends Genet 16(5):227–231
49. Doolittle RF (1997) A Delicate Balance. Boston Rev (February–March).
50. Doolittle RF, Oncley JL, Surgenor DM (1962) Species differences in the interaction of thrombin and fibrinogen. J Biol Chem 237:3123–3127
51. Doolittle RF (1997) Some reflections on the early days of sequence searching. J Mol Med 75:239–241
52. Bairoch A (2000) Serendipity in bioinformatics, the tribulations of a Swiss bioinformatician through exciting times! Bioinformatics 16:48–64
53. Dayhoff MO, Eck RV, Chang MA, Souchard MR (1965) Atlas of protein sequence and structure. National Biological Research Foundation, Silver Spring, MD
54. Fitch WM (1970) Distinguishing homologous from analogous proteins. Syst Zool 19:99–113
55. Fitch WM (1971) Toward defining the course of evolution: minimum change for a specific tree topology. Syst Zool 20:406–416
56. Ledley RS (1965) Use of computers in biology and medicine. McGraw-Hill, New York
57. Smith EL (1979) Amino acid sequences of proteins – the beginnings. In: Srinivasan PR, Fruton JS, Edsall JT (eds) The origins of modern biochemistry: a retrospect on proteins, vol 325. Annals of the New York Academy of Sciences, New York, pp 107–118

Chapter 5

Laboratory Data and Sample Management for Proteomics

Jari Häkkinen and Fredrik Levander

Abstract

Proteomic experiments can be difficult to handle because of the large amount of data in different formats that is generated. Samples need to be managed and generated, data needs to be integrated with samples and annotation information. A laboratory information management system (LIMS) can be used to overcome some of the data handling problems. In this chapter, we discuss the role of a LIMS in the proteomics laboratory, and show two step-by-step examples of usage of the Proteios Software Environment (ProSE) to handle two different proteomics workflows.

1. Introduction

The data management problem in proteomics is significant because of several factors; (i) proteomics methods are evolving rapidly with new workflows, (ii) proteomics experiments and analysis involves many steps that generate large amounts of data, and (iii) instruments produce data in different formats (1). Consequently, a major task for the proteomics researcher is to merge heterogeneous data into meaningful information and to collect meta-data for critical evaluation of results.

A Laboratory Information Management System (LIMS) is a software used in the laboratory for the management of samples, users, instruments, protocols, data analysis, and work flow automation. The goal of a lims is to create an environment where all laboratory and analysis information is tracked from biosources to final results (2). A LIMS can be a key element in an enterprise setting with connections to other information systems for streamlining production, yield, and enforce regulatory restriction, but here we restrict the discussion of LIMS to a laboratory setting. A LIMS should support:

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_5,

- Instrument integration – Information from the instrument should be useful for the LIMS, and the LIMS should generate information for instruments, such as inclusion lists for targeted tandem mass spectrometry (MS).
- Analysis tools – Users perform calculations, document, and review results using information from instruments, reference databases, and Web-based services.
- Information sharing and searching – A research group needs to share data, external partners need access to data, some users should be able to monitor progress, review results, and other documentation. Users search for samples, proteins, and other relevant information, and display sample relationships based on analysis results.
- Tracking – The information flow and data generation throughout experiments must be tracked and researchers data tracking work should be supported by the LIMS.
- Standards adoptions – For proteomics data, there are open XML file formats developed for sharing and publication of information (http://www.psidev.info). A LIMS system should support such open standards but also be able to store files in other commonly used data formats, such as comma/tab-separated (csv/tsv/txt), word processor (doc/odt), PDF and Postscript, and spreadsheet (xls/ods) formats.

The foremost advantage of using a LIMS is that the automation of experiments and data analysis can dramatically increase a laboratory's productivity. Accessibility to data is significantly improved, particularly if a Web-based interface allows access from remote locations. In addition, traditional laboratory notebooks are not compatible with a multiuser, multitask environment, so an electronic means of storing and sharing data is an attractive option.

Which LIMS to choose depends on laboratory requirements, system capabilities, integration and data needs, flexibility, standard compliance, and security requirements on the LIMS. How to choose the proper application is out of scope for this chapter, and we choose to describe how to use Proteios Software Environment (ProSE, (3)) as a LIMS. There are many other applications that perform similar services, see (2, 4) for more information about other LIMSs. ProSE is built around a Web-based local data repository for proteomics experiments and features many of the requirements on a LIMS. A feature of the system apart from pure information tracking are analysis possibilities like the combination of search results from different search engines, which are integrated into different proteomic workflows. Using two example scenarios, 2D gel and LC-MS-based experiments, we describe our best practices on solving issues related to information tracking from sample to results to public data repository submission.

2. Materials

To make the most of the remainder of this chapter you need access to a ProSE server. This document is based on ProSE version 2.8.0 but is kept at a general level, so later versions of ProSE should also work. Either follow the installation outlined in the Note 1 or use the demo server available through the ProSE Web site http://www.proteios.org. However, the demo server does not support protein identification searching directly from the application, but you can run searches outside ProSE and upload result files. The data used in the examples is also available at http://www.proteios.org (see Note 2).

We assume that you have access to an account on a ProSE server with a set of plug-ins available for your use (as outlined in Note 1). Throughout the examples below we show one way to perform actions, but there are usually several ways to achieve the same effect.

3. Methods

To get the most of a LIMS, not only the laboratory practice must be adapted to the tool, but also the LIMS need to be adapted to support laboratory practices. These adaptations are mostly related to tracking issues and rise because there is no standard for performing tracking in the laboratory. For example, by using file naming conventions, information usable for tracking can be added to the file name, even if the information is not present in the file itself (see Note 5).

ProSE spans the whole proteomics experiment, from hypothesis to actual protein identifications. ProSE manages sample information, raw data, images, analysis results, as well as connectivity to protein identification, data viewing, and analysis tools. The organisation and interface of ProSE is designed to closely follow the natural workflow of the proteomics researcher, and is compatible with both liquid chromatography (LC)–tandem MS and two-dimensional (2D)-gel experiments (see Note 1).

The ProSE data model is designed to map the steps of a proteomics experiment. The ProSE development team has specifically considered the fact that some parts of data in an experiment are generated automatically, whereas other data is collected manually. Also, we take into account that experiment steps occur at different points in time and different locations, which corresponds to a typical researcher's work situation.

Here, we describe in a step-by-step fashion and the usage of ProSE in two different workflows. Parts which require more

attention, like sample annotations, are discussed further in Subheading 4 below.

3.1. 2D-Gel Electrophoresis Case Study

For detailed description of the proteomics approaches, please refer to Chap. 1 (Schönebeck et al.).

3.1.1. Laboratory Work

Our sample is a complex protein mixture extracted from human tissue. The sample is run on a 2D gel to separate proteins into distinct spots. The gel is scanned and passed through image analysis for the detection of spots, and a gel picking robot is set up to pick spots chosen for identification. The robot digests the proteins and extracts the peptides into wells on microtiter plates. The digests were in this study analyzed using LC-MS/MS in a quadrupole time-of-flight (Q-TOF) instrument. The resulting spectra are subjected to database searching to identify proteins.

Throughout the laboratory work a lot of data is generated of which most information is stored in data files: spot picking and digestion logs, mass spectrum files, and protein identification search results files. ProSE supports upload/import of result files from several search engines, but we recommend running identification directly from ProSE if you have access to local search engines. The generated files and other relevant information should be collected for upload into ProSE.

3.1.2. ProSE Work

The first steps in a new project (experiment) should start by doing some preparation steps in ProSE. ProSE does not enforce specific routes of data upload, but in some instances, some data objects must be in place before new data can be added. We do not care about such constraints here but rather work through data upload in a sequence. ProSE provides a gel project-biased wizard that guides the user though from creating a project for the presentation of identification results. However, we do not cover the wizard here but rather work through using other actions available through the menu and buttons.

1. Log in and create a new project (File → New → Project). Name the project GelProject and save. You are presented with a new page among other things a "Members" tab. This tab allows you to add other users on the server as project members. The information in the project is available for the members with privileges defined by the project owner. We do not share data in this example but more information on sharing information is available in the ProSE user guide found at the ProSE web site.
2. Make sure that the GelProject is active. When the project is created it automatically becomes the active project but if you

later log out and in again, you must select the project as the active project (File → Select Project → GelProject). The active project will be listed on the menu bar. The GelProject menu item has many different actions of which several will be covered throughout this tutorial.

3. Add a new sample (GelProject → LIMS → Samples, and click the "New sample" button). Fill the fields, name it "GelSample", "external id" is an external identifier use "ExtGelSample", and the original field is the amount of sample (in this example use 100, the unit is predefined in ProSE and shown for the fields). Finalize by clicking "save". The biomaterial LIMS optionally keeps track of storage location of material and tracks amounts of material. Material is automatically decreased when new events are created that affect the material. For samples, we create an extract in the next step, this action is an event in ProSE, and all biomaterial events are stored in ProSE. Further information about the sample can be entered as annotations, see Note 3.
4. Select the GelSample and click on the "Make Extract" button to create a new extract. Fill the fields, name it "GelSample.e1", enter an external id, enter the amount of sample used (use 10), and the amount of extract produced (use 35). Click "Save". Return to the GelSample and note that the remaining amount of sample is decreased with the used amount. Clicking on the "Event" tab will show the events associated with the sample, and clicking on the creation event will display details about the event.
5. Now, we add the first dimension separation event for the extract. Click "Next" (or GelProject → LIMS → Extracts), select the "Event" tab, and click on the "New separation event" button. Select the separation technique (here: IPG), click next.
6. The second dimension separation is done similarly to IPG. Select the "Event" tab and click on the "New separation event" button. Select the separation technique (GelElectrophoresis), click next. There is no gel readily available yet, so we need to create one following the wizard. Fill the two forms appropriately (use "pool_test" as the External ID, this is important since the sample data file set expects that for tracking as outlined in Note 5 below). ProSE will report "Gel saved", then finalize by adding the date of the event in the laboratory, while the used quantity should be set to zero, since no more extract was used for this step. Click "Save".
7. Connect the IPG separation to the "pool_test" gel by selecting the gel (GelProject → LIMS → Gels), on the right hand side of the gel information display click on "Add previous separation dimension" and select the IPG event from step 5.

8. Create a staining event by clicking "New Staining Event" on the gel information page.
9. Create a scan event by clicking on the "New Gel Scanning Event" on the gel information page, and select the new scanning event in the list. In GelScanEvent display, click "SelectImageFile" to add a gel scan image to the event. Locate the image file in the directory listing, click on "next" and "next" to get back to the GelScanEvent display page. Now, you can view the image by clicking "view image file". This finalizes most of the manual creation of information. Now, we import all mass spectrometer data.
10. Select GelProject → Hits Import → Gel Based. Enter the gel id "pool_test" in the Gel field. Click on "Next – Select Robot Result File[s]", and in the file listing select the "spot_pick2.xml" file and click on "Import" to import spots. A job listing is presented, click on the "Update" button (bottom left on screen) to update the display. When the "GelSpotPlatePosToHitPlugin" gets the status "Done" the job is finished and the spots are imported.
11. The next step is to register all the peak lists generated by MS. Select GelProject → Hits Import → Gel Based a second time. The files provided in the example data all come from one microtiter plate, with the ID 181150420000TEST, and are in mzData format (see Note 4 about peak list file formats). Select Plate external ID 181150420000TEST in the selection box, and click on "Next – Select PeakList File[s]". Select all files that begins with a string 181150420000TEST_ and click "Import". You will be taken to the job listing display; peak list import is done when all jobs with names like PeakListToHitPlugin File: 181150420000TEST_E2.xml get status "Done".
12. Set up search parameters to run search engines from ProSE. Note that the ProSE installation needs to be configured to access search engines first (see Note 1), in case you do not have access to a ProSE installation with search engine access, you can proceed by uploading the search results supplied and continue at step 15. Search engine parameters are edited by selecting View → Search Setup. Mascot and X!Tandem should be generated, and for this sample data a tolerance of 100 ppm on both precursor and fragment level, and a human database with a decoy section should be used (see Note 6 about combination of search results regarding the choice of database).
13. Run X!Tandem. Select GelProject → Files. Select the files starting with a string "181150420000TEST_" and click on "Extensions". In the pop-up, select "Use spectrum file(s) for X!Tandem search", then the X!Tandem parameter file, and "Next – Create search job[s]" to start an X!Tandem search.

You will again end up in the job listing; wait until the job finishes. When jobs are finished, the search results file will be found in the project top directory.

14. Redo the above with Mascot. Select GelProject → Files. Select the files starting with a string "181150420000TEST_" and click on "Extensions". In the pop-up, select "Use spectrum file(s) for Mascot search", enter your name and email address for the Mascot server, select Mascot parameter file, and "Next – Create search job[s]" to start a Mascot search.
15. Now, the search results need to be imported into the database. So far, the files have been automatically (or manually) uploaded. Select GelProject → Hits Import → Gel Based to import the X!Tandem and Mascot results by clicking on "Next – Select Search Result File[s]", select the files with file type "Tandem result" and "Mascot result", and click import. When the imports finalize, you can select GelProject → Result → Hits to get a listing of your search results, which can be examined more closely by filtering and clicking.
16. Select GelProject → Result → Combined Hits to create combined identification reports (see Note 6 for details). Select an acceptable false discovery rate (FDR), typically 0.05 or 0.01 (Fig. 1). Since proteins have been separated by 2D gel, the search results combination can be done using protein scores

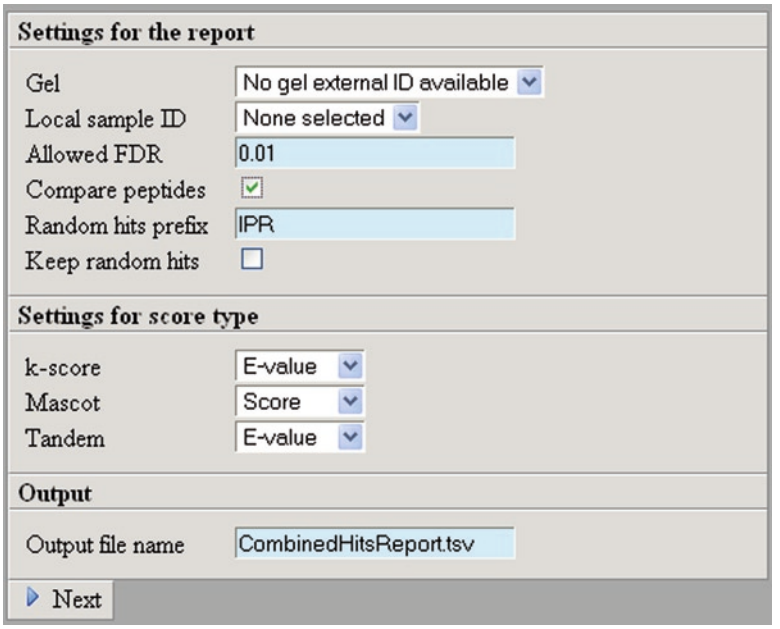


Fig. 1. Form for combining searches in ProSE. The gel or sample is selected in the select boxes. The random hits prefix needs to be adjusted to whatever prefix is used for random hits in the database used for the searches. Search engines to include can also be selected.

at the protein level (peptide level check box not checked). Set the result file name and click "Next" to start the job. Also see Note 6 about combining search results. When the job has finished, a text report will have been generated, and also the Hits table will be updated with combined FDRs for the identifications.

17. The reports will now contain gel spot identifiers and spot coordinates on the 2D gel as well as the associated identifications. To visualize the gel spots on the gel, move to the hits report (GelProject → Result → Hits) and select view gels. All spots that are active using the current filter will be visible on the gel. For example, if "45" is entered in the Spot ID filter, only spot 45 will be shown on the gel. To show all spots where the protein Actin was found, the filter "=*actin*" in the description column can be used. This can also be combined with a combined FDR filter, for example "<0.05".
18. Now, the complete experiment is saved in ProSE. Many files will probably be found in the project main directory, and it is therefore advisable to create subdirectories and move files there. Separate directories can be made for reports, search results, peak lists etc.
19. Hopefully, the results are worth sharing with the rest of the world. Then, uploading to PRIDE (5) is recommended (also see Note 7 about publication of data). To generate files in PRIDE formatted XML, the built in PRIDE XML export can be reached from the Hits report.

3.2. Quantitative LC-MS with Isobaric Labels Case Study

In this second example workflow, the protein levels of four different cell states are compared. For this analysis, the samples were reduced and alkylated using iodoacetamide, digested with trypsin and labeled using the isobaric label TMT (6). The labeled peptides were loaded onto a nano LC system and analyzed online by LTQ-Orbitrap. To get many peptide identifications, CID fragmentation and analysis in the linear trap was used. However, the reporter ions are not visible in most of the spectra, since they are found at lower masses than what can be analyzed in the ion trap using CID fragmentation. To overcome this, each MS/MS scan in the linear trap was followed by an MS/MS scan in the Orbitrap of the same precursor ion using high-energy collision-induced dissociation (HCD) fragmentation.

3.2.1. Laboratory Work

3.2.2. ProSE Work

The major reason for using ProSE, in this project, was to get a large number of confident peptide identifications at a controlled error rate, and to automatically get the reporter ion ratios included in the report. Currently, there is no other software that takes the reporter ion quantities from adjacent scans if they are not present in the spectrum used for identification. Here, we have chosen not

to enter the sample information into ProSE, but rather to generate a report as quickly as possible. The following steps are then used:

1. Generation of the project in ProSE (File → New → Project). Select nongel project as type.
2. The sample data, consisting of raw data from LC-MS/MS of two SCX fractions is first converted to the standard format mzData and uploaded to ProSE (see Note 4 about file formats). We have used Proteome Discoverer (Thermo Scientific) for the conversion, and the centroid mzData spectra (peak lists) are uploaded.
3. Set up the search parameters in ProSE. Activate the project in ProSE, and select View → Search Setup and generate the search parameter files for Mascot and X!Tandem if these are not available. It is important to select the same database for all search engines, and it needs to contain a decoy part, since the decoy database part is used to estimate the false discovery rate (see Note 6 about combining search results). If the X!Tandem installation has the k-score plug-in installed, it could be considered as a separate search engine when this scoring is enabled. For X!Tandem, TMT 6-plex was at the time of writing not found as a selectable modification. Instead, the following needed to be entered as fixed modification: 57.021464@C, 229.162932@K. The cleavage N-terminal mass change was 230.1708.
4. Perform the protein identification searches from ProSE. Go to the project_name → files and check the boxes at the files. Then, choose extensions → Use spectrum files(s) for X!Tandem/Mascot search and select the parameter file from step 3. The search jobs will be added to the job queue, and the search results uploaded when ready.
5. Now, the report generation can begin. First, the peak list files need to be registered. To do this, select project_name → Hits Import → Nongel based. Enter a local sample id for step 1, for example "pool". Then, Press "Next → Select peak list files" and check the check boxes in front of the peak list files in the next step, and "import".
6. When the jobs have finished you can check the results by looking in project_name → Reports → Hits. It should now just contain the two peak list files, but no identifications.
7. The next step is to import the search results. This is done in the second step in the NonGel-Based hits import. Just check the check boxes in front of the search results files and select "import". Note that the search results files were generated in step 4. The import step serves to get the results into the database tables.

8. When the jobs have finished, Reports → Hits will contain a lot of results, which can be navigated and filtered. However, there is still no validation and consensus usage of the search results from different search engines.
9. Combination of the search results from different search engines is performed by selecting Reports → Combined Hits (Fig. 1). Select the local sample Id that you used in step 5, and make sure that combination is performed on the peptide level. If combination is performed on the Protein level, no peptide score cut off will be used, which is dangerous in experiments with complex peptide mixtures. A peptide report will now be built up, where the false discovery rate for peptides will be calculated. Also see the note about combination of search results (see Note 6).
10. Generation of the protein report, including reporter ratios. This is done by selecting Reports → Protein Assembly. Select the local sample ID and select TMT label. The generated report will contain a list of protein groups that include the identified peptides that pass the FDR cut-off. The ratios of the reporter ions will be displayed for all peptides, and an average will be calculated for the protein group. The generated list can now be analyzed to see if it gains any insight into the biological problem under investigation.
11. Export the identifications and peak lists in PRIDE format (http://www.ebi.ac.uk/pride) for submission to the PRIDE database (5). This is done directly from the Hits table in ProSE. An option is to submit the files for biological annotation using PIKE (http://proteo.cnb.uam.es:8080/pike), which can be performed directly from ProSE (see also Note 7).

4. Notes

1. Getting Started with ProSE

   To get a flavor of how to get started with ProSE, we include a short outline of how to install and set up ProSE to work as described in this chapter. The installation procedure is straightforward for an experienced computer user and the procedure is described in detail online at http://www.proteios.org/. The requirements to get ProSE running is a contemporary computer with the following software running: a database management system (MySQL, http://www.mysql.com/), Java version 6 or later (http://java.sun.com/), and Apache Tomcat (http://tomcat.apache.org/).

   - Make sure that the above requirements are fulfilled.
   - Download and uncompress the latest ProSE distribution.

- Run the interactive installation script from a command prompt. You may need administrator privileges to create new directories for ProSE use and deploying the application in Tomcat.
- Start Tomcat. ProSE is now available by directing your Web browser to http://localhost:8080/proteios/app. 8080 is a logical port number and may differ depending on how Tomcat is set up. Replace *localhost* with the computer name or IP-number if you use a Web browser on another machine.

ProSE is now ready for use and some ProSE administration should be done before sharing it to your users, http://www.proteios.org/trac/wiki/ServerAdministration:

- Create user accounts
- Set up search engines; Mascot, X!Tandem, OMSSA...
- Set up links to external information sources; PIKE...
- Start the ftp service

2. Obtaining the Sample Data

   The data files used in the examples, in this chapter, can be downloaded as archives from the http://www.proteios.org/ site. The data can be uploaded to your local Proteios server using ftp. For FTP upload of files, an FTP client is needed, for Mozilla Firefox users, the free plug-in FireFTP (http://fireftp.mozdev.org/) is a convenient ftp client, and another (browser independent) free ftp client is FileZilla (http://filezilla-project.org/).

3. Sample Handling and Annotations

   An important role of the LIMS is to keep track of samples and in which analyses the samples have been used. The LIMS should thus be able to tell which sample has been used for a certain result, and ideally also be able to group and analyze results according to sample categories. The LIMS can also keep track of where to find all the test tubes originating from a certain sample.

   To categorize samples, annotations are needed. Depending on sample types, samples could be annotated with age, sex, disease state, etc. for clinical studies, or substrate, time point etc., for physiological cell experiments. No matter what type of experiment, a controlled vocabulary is needed for automated analysis. Both the annotation types and values need to be consistent for proper interpretation. “Age” and “AGE” could be different things for a computer, as are “11” and “eleven”. To some extent, this can be managed by the LIMS system, in that addition of sample annotation types can be limited to certain users, to avoid the duplication of annotation types.

A policy for sample annotations should be decided at the local laboratory, and then the ProSE permission system allows tuning so that some or all users can administrate annotation types. The laboratory can build their own ontology or use terms from existing ontologies. Once such a system is in place, the advantage of entering all samples into the LIMS soon becomes obvious.

4. Peak List File Formats

   Basic peak list file formats like DTA and PKL do not contain stable spectrum identifiers, and no annotations about where and how the spectra were acquired. The Mascot Generic File format (MGF) allows for some annotation and spectrum titles, but it does not use unique stable spectrum identifiers. It is therefore recommended to use an open peak list file format, which contains information for sample tracking and information about the instrument and data processing. Stable spectrum identifiers are essential to track search results back to spectra. The preferred peak list format in ProSE has been mzData (http://www.psidev.info), and standard formats developed by the Proteomics Standards Initiative (PSI) of the Human Proteome Organisation (HUPO) (7). The mzML file format (http://www.psidev.info/index.php?q=node/257) (8) is replacing mzData as the preferred spectrum format, and we expect that the common practice will be that files should be converted to mzML instead of mzData when this chapter is published. ProteoWizard (9) or instrument manufacturer software perform the file conversion to mzML.

5. File Names

   In some cases, files do not contain information that is needed for sample tracking (see Note 4). Furthermore, files can be difficult to find if they are named randomly. A file naming convention usually helps the laboratory work. ProSE parses some information from file names in certain workflows. In cases where microtiter or target plates have been used as source files for MS, the plate position is parsed from the file name since the plate position is not automatically included in the peak list files. A convention can be to include the acquisition date and initials of the experimenter in the file name, and if the sample was found in a microtiter plate, the position, A1 or similar is kept last in the file name, before file type suffixes. For example, AA_20090601_plate1_A12.mzData, would indicate that the MS file was acquired by AA from a sample found in plate1 at position A12 on the 1st of June 2009. Similarly, fraction number or gel slice number can be included in the file name. ProSE allows users to enter regular expressions to parse out some information from file names.

6. Combining Search Results

   Several publications have shown the complementarity of search engines for peptide identification, and it has become a standard practice to use more than one search engine. However, it is not straightforward to combine the scores from different search engines. In ProSE, a variant of the target-decoy strategy (10) is used to combine search results from more than one search engine (3, 11). The idea is that if the same decoy database is used for more than one search engine, it is possible to estimate the frequency of common random hits. In this way, it is possible to calculate score cut-offs for hits where more than one search engine returns a score. However, to perform such search engine result combination, it is critical that the *same* decoy database was used for all search engines. If a separate random database was used for each search engine, almost no common hits would be found, and the false discovery rate calculations would fail. In the header of the text report generated by ProSE, there will be a warning message if the sizes of the databases used differ between the search results. In some set ups, using different decoy databases may be okay, but only in scenarios where the decoy database is kept and extended with new versions of the target database. Conservation and extension of decoy entries upon database updating can be accomplished by at least one decoy database builder tool (12).

7. Publication of Data

   The LIMS system should optimally help users with formatting of data for publication. Some pieces of information for the methods section of manuscript, like protocols used, are kept track of by the LIMS. Tables containing data about individual peptide and protein identifications, can be generated from the LIMS, if enough data is stored in the system. The journal guidelines (13) and the MIAPE guidelines (14) impose reporting of many parameters, and here a LIMS can take care of much of the work. Furthermore, deposition of the data in public repositories is recommended, and will probably become required for publication in major journals. To allow for such deposition of data, best practice is to convert data into a standard format at an early stage, and make sure that it is properly annotated. After conversion to mzData or mzML (see Note 4 about peak list file formats), the files can be annotated with contact details and sample information, if this was not done by the file converter. ProSE contains extensions for annotating the peak list files in batch. The PRIDE repository (http://www.ebi.ac.uk/pride, (5)) uses sample information embedded in the mzData files, why it is advisable to annotate the mzData files properly. The ProSE PRIDE

XML exporter allows for the addition of experimental protocol information to the PRIDE submission, by selection from protocols present in the ProSE installation.

References

1. Hamacher M, Stephan C, Meyer HE, Eisenacher M (2009) Data handling and processing in proteomics. Expert Rev Proteomics 6:217–219
2. Piggee C (2008) LIMS and the art of MS proteomics. Anal Chem 80:4801–4806
3. Häkkinen J, Vincic G, Månsson O, Wårell K, Levander F (2009) The proteios software environment: an extensible multiuser platform for management and analysis of proteomics data. J Proteome Res 8:3037–3043
4. Stephan, C., Kohl, M., Turewicz, M., Podwojski, K., Meyer, H.E., Eisenacher, M (2010) Using laboratory information management systems as central part of a proteomics data workflow. Proteomics 10:1230–1249
5. Martens L, Hermjakob H, Jones P, Adamski M, Taylor C, States D et al (2005) PRIDE: The proteomics identifications database. Proteomics 5:3537–3545
6. Thompson A, Schafer J, Kuhn K, Kienle S, Schwarz J, Schmidt G et al (2003) Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS. Anal Chem 75:1895–1904
7. Orchard S, Taylor C, Hermjakob H, Zhu W, Julian R, Apweiler R (2004) Current status of proteomic standards development. Expert Rev Proteomics 1:179–183
8. Martens, L., Chambers, M., Sturm, M., Kessner, D., Levander, F., Shofstahl, J., et al. (2010) mzML – A community standard for mass spectrometry Mol Cell Proteomics doi:10.1074/mcp.R110.000133
9. Kessner D, Chambers M, Burke R, Agus D, Mallick P (2008) Proteowizard: open source software for rapid proteomics tools development. Bioinformatics 24:2534–2536
10. Elias JE, Gygi SP (2007) Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. Nat Meth 4:207–214
11. Levander F, Krogh M, Wårell K, Gärdén P, James P, Häkkinen J (2007) Automated reporting from gel-based proteomics experiments using the open source Proteios database application. Proteomics 7:668–674
12. Reidegeld KA, Eisenacher M, Kohl M, Chamrad D, Korting G, Blueggel M et al (2008) An easy-to-use Decoy Database Builder software tool, implementing different decoy strategies for false discovery rate calculation in automated MS/MS protein identifications. Proteomics 8:1129–1137
13. Wilkins MR, Appel RD, Van Eyk JE, Chung MCM, Gorg A, Hecker M et al (2006) Guidelines for the next 10 years of proteomics. Proteomics 6:4–8
14. Taylor CF, Paton NW, Lilley KS, Binz P-A, Julian RK, Jones AR et al (2007) The minimum information about a proteomics experiment (MIAPE). Nat Biotechnol 25:887–893

Chapter 6

PRIDE and "Database on Demand" as Valuable Tools for Computational Proteomics

Juan Antonio Vizcaíno, Florian Reisinger, Richard Côté, and Lennart Martens

Abstract

The Proteomics Identifications Database (PRIDE, http://www.ebi.ac.uk/pride) provides users with the ability to explore and compare mass spectrometry-based proteomics experiments that reveal details of the protein expression found in a broad range of taxonomic groups, tissues, and disease states. A PRIDE experiment typically includes identifications of proteins, peptides, and protein modifications. Additionally, many of the submitted experiments also include the mass spectra that provide the evidence for these identifications. Finally, one of the strongest advantages of PRIDE in comparison with other proteomics repositories is the amount of metadata it contains, a key point to put the above-mentioned data in biological and/or technical context. Several informatics tools have been developed in support of the PRIDE database. The most recent one is called "Database on Demand" (DoD), which allows custom sequence databases to be built in order to optimize the results from search engines. We describe the use of DoD in this chapter. Additionally, in order to show the potential of PRIDE as a source for data mining, we also explore complex queries using federated BioMart queries to integrate PRIDE data with other resources, such as Ensembl, Reactome, or UniProt.

1. Introduction

The Proteomics Identifications Database (http://www.ebi.ac.uk/pride) is a repository for the results of mass-spectrometry-based proteomics experiments, which makes use of public data standards, allowing data from a vast range of approaches, instruments, and analysis platforms to be submitted (1–3).

PRIDE stores three different kinds of information: peptide and protein identifications derived from MS or MS/MS experiments, MS and MS/MS mass spectra as peak lists, and any and all associated metadata. Experiments constitute the basic unit of information and at the time of writing, PRIDE holds around

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_6,

11,200 experiments, containing more than 2.9 million protein identifications supported by 13.2 million peptides, based on more than 78 million mass spectra.

Several proteomics MS data repositories have been established to date, with GPMDB (4), PRIDE, PeptideAtlas (5), and Proteinpedia (6) among the most prominent ones at present. Additionally, the NCBI recently launched their Peptidome (http://www.ncbi.nlm.nih.gov/projects/peptidome) system as a centralized, public proteomics repository not dissimilar from PRIDE. The Tranche (http://tranche.proteomecommons.org) system is used in the field as well, and essentially presents a data transfer layer relying on peer-to-peer internet protocol technology. Apart from these large-scale efforts, there are also smaller, more specialized repositories. For an up to date review covering the capabilities of these different proteomic MS repositories, see (7).

Together with the newly released NCBI Peptidome, the established PRIDE database occupies a special place in the list of proteomics resources, in that it constitutes an actual structured data repository, and does not assume editorial control over submitted data. Because data in PRIDE is not reprocessed or altered in any way after submission, and because PRIDE allows data to remain private while anonymously sharing it with journal editors and reviewers, PRIDE is now the recommended submission point for several journals, such as *Nature Biotechnology* (8), *Nature Methods* (9), and *Proteomics* (http://www3.interscience.wiley.com/cgi-bin/jabout/76510741/2120_instruc.pdf).

Apart from PRIDE itself, several highly influential informatics tools have been developed in support of the PRIDE database: the Ontology Lookup Service (OLS, http://www.ebi.ac.uk/ols) (10, 11), the Protein Identifier Cross-Referencing system (PICR, http://www.ebi.ac.uk/Tools/picr) (12), and more recently, Database on Demand (DoD, http://www.ebi.ac.uk/pride/dod). Additionally, several data submission tools are available for PRIDE, including the powerful and popular PRIDE Converter (http://code.google.com/p/pride-converter). OLS and PICR, and the navigation through the PRIDE Web interface have already been described before (13, 14).

At present, the PRIDE database has developed from its original role as a repository of proteomics identifications arising from mass spectrometry, to a database providing tools for complex queries and data retrieval, data set comparison and access to additional automated annotation of submitted data sets. In this chapter, we first describe the use of DoD, the most recent PRIDE-related tool. We then concentrate on the PRIDE BioMart interface in the BioMart Central Portal (15). The easy-to-use and highly configurable BioMart Web interface provides Wet lab researchers with a convenient tool to efficiently retrieve relevant data, but the feature we specifically highlight here is the ability to perform powerful across-resource queries with other relevant bioinformatics databases.

2. Materials

2.1. Database on Demand

In mass spectrometry, most conventional software algorithms for the identification of recorded mass spectra (also called search engines) are based on searching spectral data against sequence databases. As a result, these sequence databases play an important role in the identification process, and incomplete extraction of information from these databases (e.g, known amino- or carboxy-terminal protein maturation sites) can lead to an inefficient identification process. DoD was built to ensure that the desired information can be easily extracted from the UniProt and IPI sequence databases. DoD, using the existing stand-alone DBToolkit (16) application for its database processing back-end, is implemented as a Web application that creates customized search databases in FASTA format, allowing detailed control over the search space. The user can apply one or more preprocessing steps to the UniProtKB/Swiss-Prot, UniProtKB/TrEMBL and IPI database, and these databases can be combined into a single output database as well. In specific cases, the impact of using such specialized databases can be quite dramatic, increasing the number of peptide identifications by up to 50% (17, 18). DoD can be accessed directly at http://www.ebi.ac.uk/pride/dod.

2.2. PRIDE BioMart Interface

BioMart is a query-oriented data management system that does not require any programming knowledge to interrogate, yet allows for very powerful data retrieval (19). The BioMart interface allows you to build simple or complex queries, with total control over both how the data is filtered (to restrict which records are included) and also which attributes (equivalent to columns in a spreadsheet) are included in the results. A salient point here is that the core BioMart interface does not change between different resources, which means that an understanding of the BioMart interface in any one system automatically allows the user to understand any other BioMart as well. The existence of BioMarts for many different resources, together with the ability to combine two BioMarts in a single query, enables the integration of information across several types of biological data through across-Mart queries. The BioMart Web interface currently allows no more than two resources to be combined in a single query. The PRIDE BioMart interface is accessible at http://www.ebi.ac.uk/pride/prideMart.do. At present, it is possible not only to retrieve data from PRIDE individually, but also to integrate information from PRIDE with Reactome (20), a resource that contains curated information about pathways. Additionally, the PRIDE BioMart is also available via the BioMart Central Portal at http://www.biomart.org/biomart/martview/, where connections with several other resources can be made as well.

3. Methods

3.1. Database on Demand

The DoD Web interface is accessible at http://www.ebi.ac.uk/pride/dod. For more details about how to use DoD, press on the "User Manual" link at the top left block of the page.

As an example query to show the potential of DoD, we are going to create a custom database in FASTA format combining all human sequences from the UNIPROT and IPI databases. The database will be digested in silico by trypsin, allowing two missed cleavage sites, and finally only resulting peptides between 600 and 5,000 Da will be selected.

Step 1. Direct your Web browser to the DoD Web interface: http://www.ebi.ac.uk/pride/dod. The first step of the process is the database selection. Go to the menu, "Choose a database" at the top of the page, and select one or more databases if you need them to be combined. Once you have selected the first database, click on "Add as source". Once the database is selected, different filters can be added depending on each particular database. For instance, for SwissProt and TrEMBL filters can be selected based on "protein accession," "Uniprot keyword," "NCBI tax ID," and "Maturation" (see Note 1). Then, press the "+" button at the right (Fig. 1). Several databases can be added repeating the same process.

*For the sample query (human proteins from UniProt and IPI), you need to select the databases in three steps. First, choose "SwissProt" and add a filter for human (NCBI Tax ID = 9606). Do the same for TrEMBL, first click on "add as Source", and then choose the same filter (*see Note 2*). Finally, add "IPI_Human" as the third database. For each selected database, you can visualize the corresponding selected filters by going to the "Details" column and clicking on "Show". The filters can be edited at that point as well.*

Step 2. The second step of the process is the enzyme selection. This is done in the "PROCESSING" area of the Web interface. You can select an enzyme in the "Choose an enzyme menu". The ones that are available by default are trypsin, Arg-C, Lys-N, Lys-C, and chymotrypsin (see Note 3). When an enzyme is selected, the user can visualize the target cleavage site and if there is any restriction, and if the enzyme cleaves N- or C-terminally of the cleavable residue/s. The number of allowed missed cleavages can also be selected (the default is one).

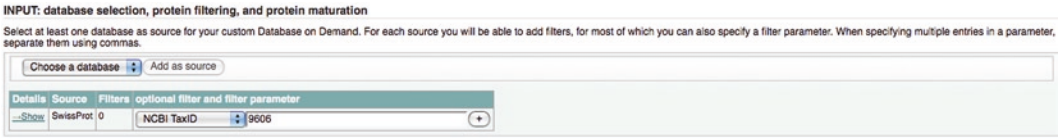


Fig. 1. Screenshot that shows how the filters for the databases can be selected, as specified in Subheading 3.1, step 1.

If the desired enzyme is not included in the list, the user can specify the restriction site using a regular expression. To that end, use the "regex" option (see Note 4). Finally, the user can decide to perform a ragging of the database (see Note 5). Once all the options have been selected, it is necessary to press the "+" button at the right (Fig. 2). More enzymes can be added, following the same protocol (see Note 6).

For the sample query, select trypsin as the enzyme and increase the number of allowed missed cleavages to "two."

Step 3. The third step is the selection of the output. At this point, the user can choose to restrict the results (once you have selected the database/s and the enzyme/s) using a sequence filter (see Note 7) or by specifying mass limits peptides.

For the sample query, we are choosing the default option of selecting peptides between 600 and 5,000 Da (Fig. 3).

Step 4. At the bottom of the Web interface, press on "GENERATE WORKFLOW". This will pop up a menu with some info and warning messages. If the decision is to go ahead, click on "Generate anyway".

Step 5. In a last step, the user is asked to provide an e-mail address. This e-mail address is used to mail a link to the resulting database.

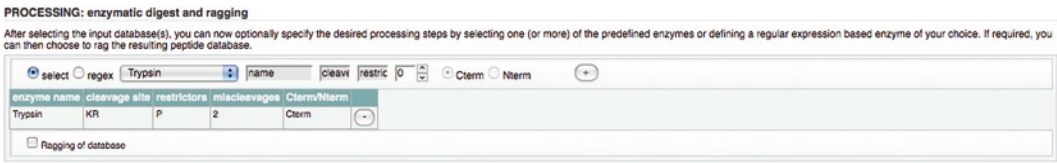


Fig. 2. Screenshot that shows how processing steps (e.g., enzymatic digestion) can be added, as specified in Subheading 3.1, step 2.

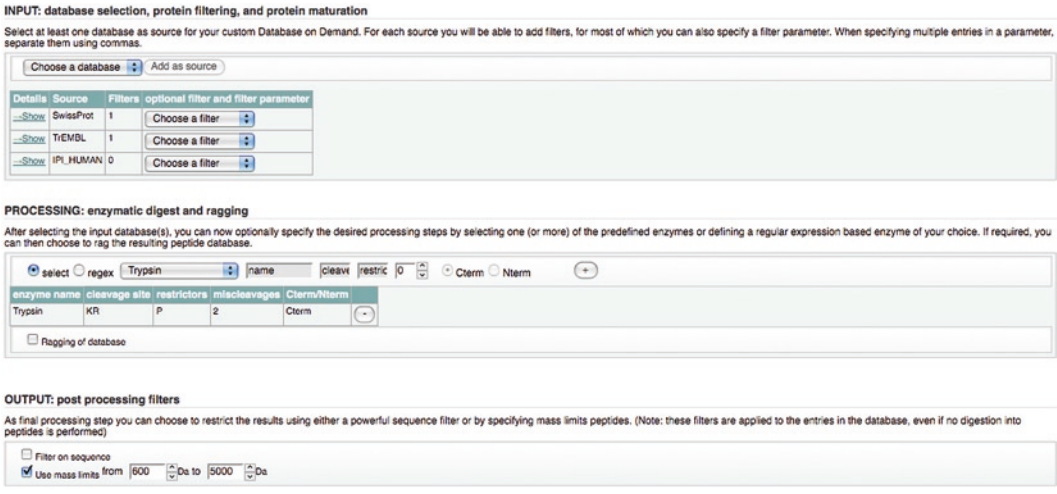


Fig. 3. Screenshot that shows how the appearance of the Web after all the relevant parameters have been specified, according to Subheading 3.1.

Click on "Generate database" and the creation of the custom database will start. The user will first receive an e-mail confirming that the process has started, which includes a unique process ID. When the database is generated, a second e-mail will be sent with the URL where the database can be downloaded from. This e-mail will also contain a description of the user-configured workflow that was used to create the database, for future reference.

For the sample query, at the time of writing, the size of the generated file in FASTA format is 757 MB, and contains 3,483,739 unique entries. It took less than 1 h to generate the file.

3.2. Cross-Resource BioMart Queries in the BioMart Central Portal

The PRIDE BioMart is available via the BioMart Central Portal at http://www.biomart.org/biomart/martview/. As mentioned before, PRIDE can be combined with resources, such as Ensembl, Protein Data Bank in Europe (PDBe), Reactome, or UniProt.

As an example query to show the potential of the BioMart cross-resource queries, we are going to search for all peptide sequences corresponding to proteins identified in PRIDE, found in cerebrospinal fluid (CSF) that are present in UniProt. We further require these proteins to have a size between 300 and 500 amino acids. We also retrieve some extra information about the proteins identified: accession number, definition, gene name, UniProt protein evidence code, and EC number.

The Web interface is used to build your query. There are three main steps involved in query building: the creation of filters to restrict the data included in your results, the selections of attributes (i.e., the selection of columns of data to include), and finally the selection of a format for the results (i.e., HTML table, tab separated values file, comma separated values file, or a Microsoft Excel spreadsheet). You have to do the first two steps once per selected resource, whereas the last step is implicitly shared.

Step 1. Direct your Web browser to the BioMart central portal server: http://www.biomart.org/biomart/martview. Select the "UNIPROT (EBI UK)" BioMart Database from the "Database" or "CHOOSE DATABASE" drop-down on the right hand panel of the interface (see Note 8).

Step 2. Create a "filter" to restrict the results returned to you from the BioMart from UniProt, by clicking on the "Filters" heading on the top left panel of the BioMart MartView window. In the UniProt BioMart, results can be filtered based on four groups of criteria: "Species," "Protein," "Database IDs," and "Others." Each of these headings is adjacent to a + symbol. Click on the + symbol to expand the section and view the available filters.

For our example query, click on "Species" and once the different options are displayed, click on "Complete proteome" and select "Homo sapiens" in the menu. Then, click on "Protein" and click on

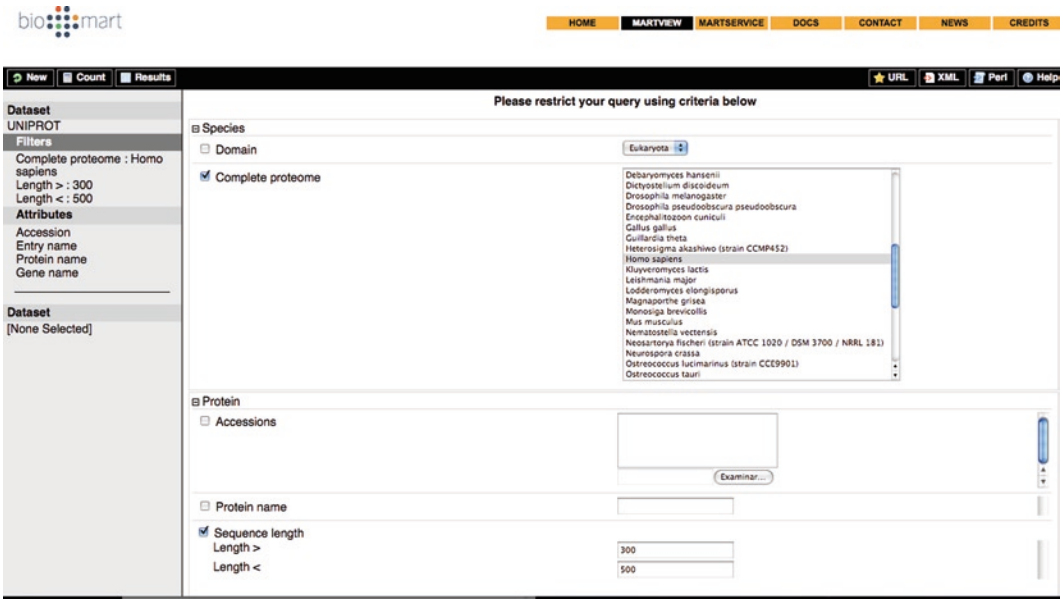


Fig. 4. Screenshot that shows how the selection of attributes to be retrieved from UniProt is performed, as specified in Subheading 3.2.

"Sequence length," and provide values for "Length>" and "Length<" (300 and 500, respectively) (Fig. 4).

Step 3. Select the "Attributes" (data columns) from UniProt that you wish to include in your search results by clicking on the "Attributes" heading on the top left panel of the top BioMart MartView window.

A large number of data columns can be selected from UniProt. There are four groups of data: "Protein attributes," "Gene ontology (GO)," "Database cross references," and "Others." By default accession number, entry name, protein name and gene name are selected. Check (click) all of the attributes you wish to include in your query results.

In this particular example, we are also going to get the "Protein existence" (first click on "Protein attributes" to select it) and "EC number" (select "Others," and click it).

Step 4. Click on the bottom "DataSet" field in the left block and select the "PRIDE (EBI UK)" BioMart Database from the "CHOOSE ADDITIONAL DATASET" drop-down on the right hand panel of the interface.

Step 5. Create a second "filter" to restrict the results returned to you from the PRIDE BioMart by Clicking on the "Filters" heading on the bottom left panel of the BioMart MartView window.

On the right hand panel, six filter sections will be displayed: "Filter by Experiment," "Filter by Sample Details," "Filter by Protein Identification," "Filter by Mapped Protein Database Identifiers," "Filter by Peptide Identification," and "Filter by

Protein Modification." Again, each of these headings is adjacent to a + symbol. Click on the + symbol to expand the section and view the available filters.

For the example query, click on "Filter by Sample Details" and once the different options are displayed, first click on "Filter by Species" and click on "Homo sapiens (Human)" from the right menu. Then, click on "Filter by Tissue" and select "cerebrospinal fluid" from the right menu.

Step 6. Select the "Attributes" (data columns) from PRIDE that you wish to include in your search results by clicking on the "Attributes" heading on the bottom of the left panel on the BioMart MartView window.

On the right hand panel, a large number of individual data items will be listed, each adjacent to a check box. These items are organized into six sections: "Experiment Attributes," "Sample Attributes," "Protein Identification Attributes," "Active Protein Identification Database Cross References," "Peptide Identification Attributes," and "Protein Modification (PTM) Attributes." Again, check (click) all of the attributes that you wish to include in your query results.

For our example query, we will select "PRIDE Experiment Accession," "Experiment Title" (both in "Experiment Attributes"), "Submitted Protein accession" (in "Protein Identification Attributes"), and "Peptide sequence" (at the bottom at "Peptide Identification Attributes") (Fig. 5).

Step 7. Click on the "Count" button at the top left of the BioMart MartView interface. Note that this is not necessarily the same as

Fig. 5. Screenshot that shows how the selection of attributes to be retrieved from PRIDE is performed, as specified in Subheading 3.2.

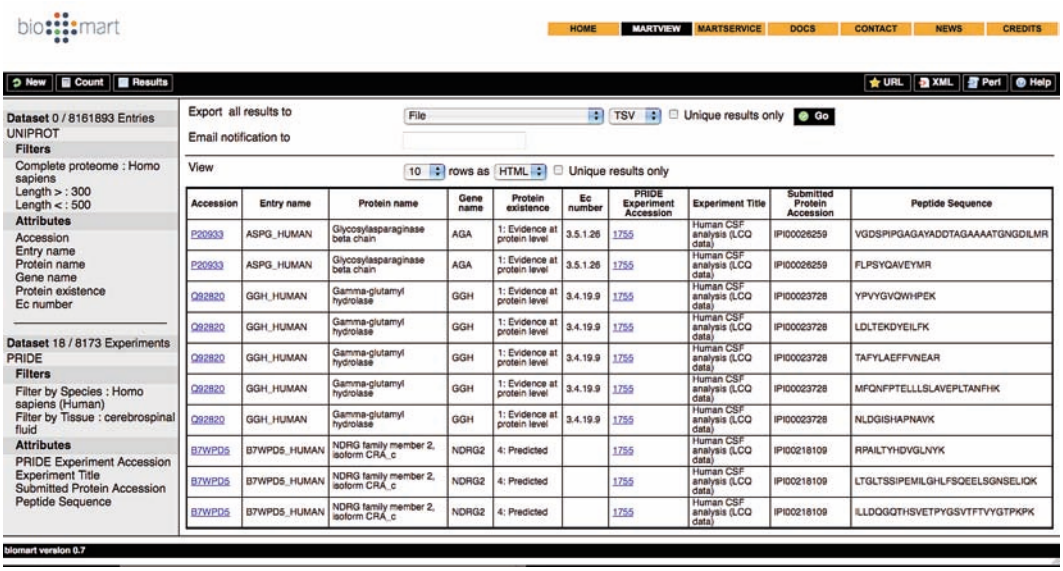


Fig. 6. Screenshot that shows how the final results retrieved using BioMart look like, as specified in Subheading 3.2.

the number of records/rows that are returned by your query, as the count is based on the number of unique Experiments (PRIDE) or proteins (UniProt) returned.

Step 8. Click on the "Results" button at the top left of the BioMart MartView interface (Fig. 6). This will return a preview of the results comprising only the first ten rows of data (by default). At this point, you should examine the data returned and make any required modifications to your selection of filters and attributes if required.

Step 9. Select how you would like the results delivered from the drop-down in the right hand panel, labeled "Export all results to". The options for results delivery are "File", "Compressed File (.gz)", and "Compressed Web File" with the additional functionality of sending an e-mail notification when the results file has been built. This e-mail notification includes a link to the compressed result file and is especially useful for very large data sets (see Note 9).

Step 10. Select the format of the results from the drop-down in the right hand panel. Available options include: HTML (tabulated results), CSV (comma-separated values in a plain text file), TSV (tab-separated values in a plain text file), and XLS (Microsoft Excel spreadsheet).

Step 11. Click the check box labeled "Unique results only" to reduce possible redundancy in the results table.

Step 12. Click on the "Go" button in the right hand panel to retrieve the complete, formatted results.

4. Notes

1. Each database can also be subjected to a maturation step, in which the complete (precursor) protein sequences are in silico matured by removing any pre-, pro-, or signal peptides on either terminus based on the annotations in the database.
2. Since TrEMBL is quite a large database, it is strongly recommended that you use a filter (for example, a species filter for human: TaxID "9606"; for mouse: TaxID "10090"; for *C. elegans*: TaxID "6239"; for *A. thaliana*: TaxID "3702").
3. By default a site will not be cleaved if the following residue after the cleavage site is proline. However, the user can overcome this limitation by selecting from the list the corresponding enzyme plus "/P" (e.g. Trypsin/P, Arg-C/P and Lys-C/P).
4. To create a new enzyme from scratch, simply give it a name, specify the regular expression pattern that is to be used to determine the cleavage site, optionally specify the restricting amino acids (which will prevent cleaving) and choose if you want to cleave on the N-terminal or C-terminal side of the residue/s defining the cleavage site. The enzyme digestion allows a number of missed cleavages, which can be defined for both selected and custom-designed enzymes by changing the number in the miscleavages selection box.
5. The optional ability to process the enzymatic products further by ragging will transform a single sequence into a set of unique sequences, in which each sequence loses a carboxy- (C) or amino- (N) terminal residue compared to the previous sequence. Ragging is extremely useful if you want to detect proteolytic degradation, which results in the formation of a novel N-terminus (and/or C-terminus) and you do not a priori know where this processing will take place. Both for ragging and maturation, customized databases can prove very useful for existing N-terminal proteomics approaches (17). If ragging is selected, the truncation option allows the user to truncate the sequence to the specified number of terminal residues before ragging. For instance, in the case of N-terminal ragging, setting this to 100 will only include the first 100 (N-terminal) residues of the sequence for the ragging process, disregarding the rest of the sequence. This can be useful if you happen to know that the processing you aim to identify, occurs in the first X residues (e.g., mitochondrial target sequences). Note that ragging is applied after enzymatic digest, if a digest is requested.
6. Whenever an enzymatic digestion step is specified, the software will also automatically apply a step to clear peptide-level

sequence redundancy in the database. This means that each peptide sequence will be present only once in the output database, thus maximizing the information ratio of the database (which is defined as the number of unique sequences in a database, divided by the total number of sequences in the database). Whenever a peptide could be derived from more than one protein, the accession numbers and peptide locations for each potential precursor protein are included in the peptide sequence header. These alternative precursor proteins are annotated at the end of the FASTA header description part, and the individual protein accession numbers and locations are separated by "^A" characters (which is the FASTA standard annotation for protein isoforms). For instance, a peptide that matches to both P12345 and P54321 will carry a header like:

>sw|P12345 (17–25)|RNAS1_ONDZI RecName: Full=Ribonuclease pancreatic; EC=3.1.27.5; ^Asw|P54321 (18–26)

7. The first option allows the user to filter sequences by their amino acid composition. You can use a simple yet powerful query language to define your compositional requirements. This language is explained below. Amino acid notation is the single-letter notation, extended with "U" for methionine without initiator methionine. The format supports boolean operators ("AND", "OR", and "^" (NOT)) and is vaguely reminiscent of regular expressions. It does not have the full power of regular expressions however, nor does it have exactly the same syntax, but it is simpler and more powerful in specifying compositional requirements. You can specify residues or sequence stretches and combine these:

   (K and R) or (S or T)→selects all entries having either a K and an R, or that have an S or a T.

   ((K and R) or S) and L→selects all entries carrying an L, as well as an S or a K and an R.

   ^R and ^K→selects all entries lacking both R and K.

   Another feature of the language concerns the counting of residues:

   2K or 2R or (K and R)→selects all entries having either exactly two Rs or two Ks, or that have both R and K.

   Yet another addition of this is logical operations on counts:

   >3K or <5P→selects all entries with strictly more than three Ks or strictly less than five Ps.

   >=2K and <=2L→selects all entries with two or more Ks and two or fewer Ls.

8. In order to combine UniProt and PRIDE, you need to first select UniProt (in the top of the BioMart Web interface) and then select PRIDE as the second database, at the bottom. So these resources can only be combined in one direction, whereas in some other cases the order is not important. It depends on the way the set up for appropriate pipes in and out of each resource was done. At the time of writing, PRIDE data can be integrated with other systems in the following ways:
   (a) With PRIDE as the main dataset (selected at the top of the BioMart page), integration with Reactome, MSD (protein structures), and the Rat Genome Database (RGB) is available.
   (b) PRIDE can be chosen as the second dataset (at the bottom of the BioMart dataset section) not only from the resources mentioned in (a), but also from the highly cross-referenced and information-rich UniProt and Ensembl BioMart interfaces.
9. If there is a long delay after you have clicked GO, this would suggest that your query will result in many rows of data being returned. This may crash your browser on arrival, so it is recommended that you click on the back button and modify your results request to be sent by e-mail.

References

1. Martens L, Hermjakob H, Jones P, Adamski M, Taylor C, States D, Gevaert K, Vandekerckhove J, Apweiler R (2005) PRIDE: the proteomics identifications database. Proteomics 5:3537–3545
2. Jones P, Cote RG, Martens L, Quinn AF, Taylor CF, Derache W, Hermjakob H, Apweiler R (2006) PRIDE: a public repository of protein and peptide identifications for the proteomics community. Nucleic Acids Res 34:D659–D663
3. Jones P, Cote RG, Cho SY, Klie S, Martens L, Quinn AF, Thorneycroft D, Hermjakob H (2008) PRIDE: new developments and new datasets. Nucleic Acids Res 36: D878–D883
4. Craig R, Cortens JP, Beavis RC (2004) Open source system for analyzing, validating, and storing protein identification data. J Proteome Res 3:1234–1242
5. Deutsch EW, Lam H, Aebersold R (2008) PeptideAtlas: a resource for target selection for emerging targeted proteomics workflows. EMBO Rep 9:429–434
6. Kandasamy K, Keerthikumar S, Goel R, Mathivanan S, Patankar N, Shafreen B, Renuse S, Pawar H, Ramachandra YL, Acharya PK, Ranganathan P, Chaerkady R, Keshava Prasad TS, Pandey A (2009) Human proteinpedia: a unified discovery resource for proteomics research. Nucleic Acids Res 37:D773–D781
7. Mead JA, Bianco L, Bessant C (2009) Recent developments in public proteomic MS repositories and pipelines. Proteomics 9:861–881
8. Anonymous (2007) Democratizing proteomics data. Nat Biotechnol 25:262
9. Anonymous (2008) Thou shalt share your data. Nat Methods 5:209
10. Cote RG, Jones P, Apweiler R, Hermjakob H (2006) The Ontology Lookup Service: a lightweight cross-platform tool for controlled vocabulary queries. BMC Bioinform 7:97
11. Cote RG, Jones P, Martens L, Apweiler R, Hermjakob H (2008) The Ontology Lookup Service: more data and better tools for controlled vocabulary queries. Nucleic Acids Res 36:W372–W376

12. Cote RG, Jones P, Martens L, Kerrien S, Reisinger F, Lin Q, Leinonen R, Apweiler R, Hermjakob H (2007) The Protein Identifier Cross-Referencing (PICR) service: reconciling protein identifiers across multiple source databases. BMC Bioinform 8:401
13. Jones P, Cote R (2008) The PRIDE proteomics identifications database: data submission, query, and dataset comparison. Methods Mol Biol 484:287–303
14. Martens L, Jones P, Cote R (2008) Using the proteomics identifications database (PRIDE). Curr Protoc Bioinformatics. Chapter 13, Unit 13.8
15. Haider S, Ballester B, Smedley D, Zhang J, Rice P, Kasprzyk A (in press) BioMart central portal – unified access to biological data. Nucleic Acids Res. 37:W23–7
16. Martens L, Vandekerckhove J, Gevaert K (2005) DBToolkit: processing protein databases for peptide-centric proteomics. Bioinformatics 21:3584–3585
17. Gevaert K, Goethals M, Martens L, Van Damme J, Staes A, Thomas GR, Vandekerckhove J (2003) Exploring proteomes and analyzing protein processing by mass spectrometric identification of sorted N-terminal peptides. Nat Biotechnol 21:566–569
18. Ghesquiere B, Van Damme J, Martens L, Vandekerckhove J, Gevaert K (2006) Proteome-wide characterization of N-glycosylation events by diagonal chromatography. J Proteome Res 5:2438–2447
19. Smedley D, Haider S, Ballester B, Holland R, London D, Thorisson G, Kasprzyk A (2009) BioMart-biological queries made easy. BMC Genomics 10:22
20. Matthews L, Gopinath G, Gillespie M, Caudy M, Croft D, de Bono B, Garapati P, Hemish J, Hermjakob H, Jassal B, Kanapin A, Lewis S, Mahajan S, May B, Schmidt E, Vastrik I, Wu G, Birney E, Stein L, D'Eustachio P (2009) Reactome knowledgebase of human biological pathways and processes. Nucleic Acids Res 37:D619–D622

Chapter 7

Analysing Proteomics Identifications in the Context of Functional and Structural Protein Annotation: Integrating Annotation Using PICR, DAS, and BioMart

Philip Jones

Abstract

For many species, there is a wealth of detailed annotation of individual proteins available to the proteomics researcher. Accessing and making the best use of this annotation can be problematic in the absence of suitable bioinformatics support. This chapter explores some of the technologies and tools that allow protein annotation to be accessed and collated from multiple sources. The intended audience is the proteomics scientist who has limited or no access to bioinformatics/programming support and wishes to make the best use of existing resources to annotate sets of protein identifications derived from mass spectrometry and related techniques.

1. Introduction

This chapter explores the options available to proteomics researchers to enable the analysis of proteomes or lists of identified proteins in the context of existing protein and proteome annotation.

This data can be readily accessed using the tools and techniques available to the bioinformatics researcher. Unfortunately, in the absence of bioinformatics support, the technical hurdles involved in achieving this can be an impediment to the laboratory scientist who wishes to collate third party annotation of the proteins that they have identified.

Fortunately, mechanisms for data integration in bioinformatics are reaching maturity. Technologies, such as BioMart and the Distributed Annotation System (DAS) allow researchers to formulate complex integrative queries across multiple databases, either by writing software to perform the analysis or increasingly by making use of existing software tools to do this for them.

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696,
DOI 10.1007/978-1-60761-987-1_7,

There are a large number of problems that need to be overcome to access and make good use of annotation from disparate sources. Inconsistencies in the naming or identification of proteins and the manner in which protein features are named or categorised make data comparison across data resources very difficult. In the last few years, these two problems have been addressed by the bioinformatics community. Robust solutions now exist for mapping protein identifiers from one database to another and community agreement about feature categorisation is being reached (at least in the context of DAS server annotation.)

Another issue that the proteomics scientist has to overcome is the shear wealth of databases providing annotation. Making informed choices about the best resources to use is difficult, without background knowledge of how the resources are built and maintained. There is a distinction between databases that provide human curated annotation and databases that provide automatic annotation based upon predictive models. Clearly, both of these types of resource are extremely valuable; however, the appropriate use of the annotation from these two sources may differ.

There are of course many different ways to access protein annotation, depending upon the source being queried. Here, we focus upon two technologies, BioMart and DAS, that both provide programmatic access to the data and services provided by a resource, known generally as "web services". Web services allow users of software tools to query a resource via the Internet and retrieve results in a machine-processable, often simple format that the tool they are using can then process. This format typically does not include information about how to display the information (unlike a "normal" Web page for display in an Internet browser) but focuses on transmitting the meaning and structure of the data itself. It is then the responsibility of the software tool (or software "client") to analyse and/or display the information for the user.

The next few sections consider solutions to the problems of mapping protein accessions from one protein sequence database to another, followed by an overview of the use of DAS and BioMart to collate third-party annotation.

2. Solving the Protein Identifier Problem – Just Too Many Databases?

The sheer number of protein sequence databases, with their different protein identifier systems, can be a significant hurdle to collate third-party annotation of proteins. The first step in proteomics data analysis is to select an appropriate protein sequence database to search mass spectra against. This selection takes into account parameters, such as species specificity, database size, and database redundancy. A common choice, for example, is the IPI

database, which provides a "minimally redundant yet maximally complete set of proteins for featured species" (1).

The researcher may then wish to retrieve a broad range of annotation of the identified proteins. This requires that the protein identifiers from the database used (e.g. IPI numbers) are mapped on to the most appropriate species-specific database and/or high quality human-curated database, such as UniProtKB/Swiss-Prot (2). Clearly, the most desirable situation is that the identified protein accessions can be mapped on to all of the relevant protein sequence databases in one step. This can be achieved for the majority of public protein sequence databases using the Protein Identifier Cross Reference Service, PICR (http://www.ebi.ac.uk/Tools/picr/) (3). Following is a step-by-step description of how this can be easily achieved. These instructions assume that you have a list of protein identifiers or accessions from the search database used for protein identification.

1. Build a text file (using, for example, Microsoft NotePad) containing one protein ID on each line. Save this file and note its location and name on your computer. If you wish to create this file using a spreadsheet application, you should paste the protein IDs into the first column and then save the file in "CSV" format.
2. Visit http://www.ebi.ac.uk/Tools/picr/ in an Internet browser (see Fig. 1).
3. In the centre of the screen, you will see a large text area under the heading "Input Data." Under this text area is a "browse" button. Click on this button.
4. Browse to, then open the text file that you saved earlier. (The exact details of the dialogue box used to browse to the file will depend upon the operating system and the Internet Browser you are using, so are not described here.)
5. You should now see the path to the file displayed on the Web page next to the browse button.
6. Select a format for the protein identifier mappings. For the purpose of using the mappings to query further services, such as DAS and BioMart, the CSV format is recommended (plain text, comma-separated values file). This format can be used for importing the data into any spreadsheet software.
7. By default, there is no limitation by taxonomy. Mappings for all species are returned. You may wish, however, to limit the mappings returned to a specific species. There is a pull-down list of species located at the top right hand corner of the PICR Web form. This includes the species described in the Ensembl database. If you are interested in other taxonomic groups, type the name of the taxon into the text box below the pull down list. Suggested species matching your search will start to appear as you type.

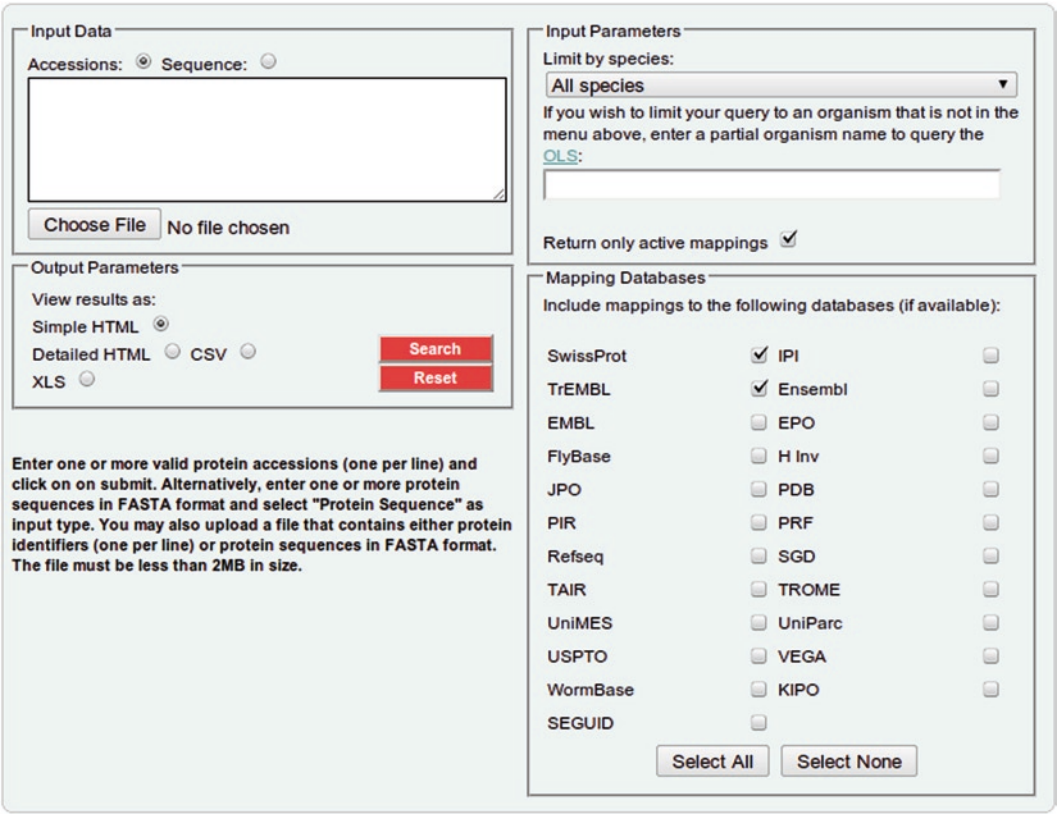


Fig. 1. The PICR service user interface. See the main text for a step-by-step description of how to use this interface. Note that there is also a "web service" interface to PICR for use directly from code.

8. It is recommended that you leave the check box labelled "Return only active mappings" in its default state (checked/ticked). This ensures that only current protein identifiers are returned from PICR.
9. Select the protein sequence database that you wish to map your identifiers to. To use the list of identifiers returned from PICR in a tool, such as DAS or BioMart, it is recommended that you select a single database to map to at a time, to keep the results from PICR simple. Please see Note 1 for a discussion of the default settings, SwissProt and TrEMBL.
10. Click on the red "Search" button which is situated in the "Output Parameters" section in the middle of the screen.
11. The search may take several seconds to perform, or longer if you have supplied a long list of protein identifiers. You will see a progress bar appear on your browser, which is regularly updated. If nothing happens for a long time, click on the "Refresh" link.

To use the data returned from PICR, it is important to understand how the mappings are generated. PICR maps to protein

accessions that are either assigned to exactly the same sequence or have been annotated in UniProtKB/SwissProt as logical cross-references. PICR is not a BLAST service. If you are interested in finding *similar* protein sequences rather than alternative identifiers for the same sequence, PICR is not for you.

In step 6 above, it was recommended that you select the CSV format. This format includes four columns of data:

1. The *input* protein identifier (the one you searched with).
2. The name of the database that the input identifier has been mapped to.
3. The mapped protein accessions.
4. The "status" of the mapping, one of "identical" or "logical." "Identical" indicates that the mapped protein identifier refers to exactly the same protein sequence. "Logical" indicates that the mapped accession is a cross-reference in UniProtKB/Swiss-Prot.

3. Collating Annotation from Multiple Sources – DAS

The Distributed Annotation System (DAS), together with the software tools that have been developed to use this service, allows the user to retrieve annotation on protein sequences or nucleic acid sequences from many physically and geographically separate locations in one request. The real power of this system is that the separate sources of annotation need not be aware of each other in any way, so long as they are using a common naming system and coordinate system for the sequences they describe. The software tool (DAS "client") being used by the researcher is able to locate these separate sources of annotation using a central registry. The tool then requests annotation from all of the registered sources and finally collates this annotation for display or analysis.

DAS has been in common use since it was first used for nucleic acid sequence annotation in 2001 (4), becoming a widely used and stable standard following the release of version 1.53 of the specification in 2002 (5). At this point, the focus of the standard was on serving sequence information and annotations coordinated on to this sequence. Since then, the scope of the standard has been expanded significantly. It is now possible to use DAS to retrieve structural information (at the level of atomic coordinates), to perform sequence alignments, and to retrieve interaction data (6). More recently, groups have been working on DAS writeback to allow researchers to contribute annotation to a remote server. These new facilities have been described in later versions of the DAS specification, including DAS 1.53E (7) (http://www.dasregistry.org/spec_1.53E.jsp) and the DAS 1.6 standard (http://www.biodas.org/wiki/DAS1.6).

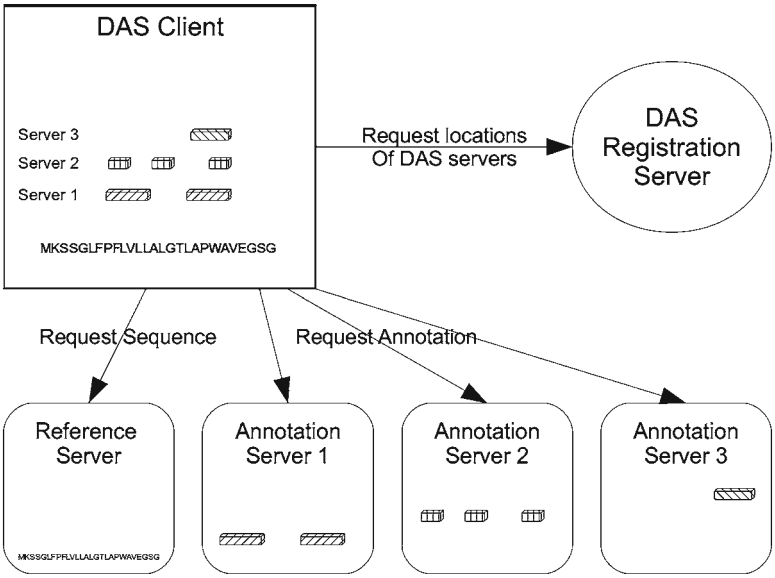


Fig. 2. A DAS client is able to retrieve annotation of a protein (or nucleic acid) sequence from many separate sources and integrate this annotation into a single view. This is a two step process: the client first of all requests the locations of relevant DAS servers from the DAS Registration Server. It then requests sequence information from a single DAS reference server and annotation from any number of DAS annotation servers.

An important concept to understand in DAS is the separation of DAS servers into two types: reference servers and annotation servers. Reference servers provide sequence and version information. More recently, they may also provide structural and alignment information. Reference servers are often run by sequence database maintainers, for example, the UniProt consortium provides a DAS reference server (http://www.ebi.ac.uk/das-srv/uniprot/das) which is kept up-to-date with the UniProt Knowledge Base. Whenever a DAS client requests information for a single protein, it will query one DAS reference server for the sequence. It will also (usually in parallel) query any number of DAS annotation servers for annotation on that sequence. The annotation servers will also reply with version information for the sequence they are annotating, so if there is any version conflict between the reference and annotation server, it can be highlighted by the client (see Fig. 2).

4. The DAS Registry

The DAS Registration Server (http://www.dasregistry.org/) (8), provides a centralised repository describing public DAS servers around the world. It can be browsed using an Internet browser,

or more usually it is used directly by DAS clients to retrieve information about available DAS sources. Generally, it is not necessary for a researcher using a DAS client to visit the DAS Registration Server manually, as the necessary information is automatically retrieved from the registry by the DAS client itself.

However, if you wish to learn more detail about a specific DAS server, or wish to troubleshoot a problem with a particular server, it may be useful to visit the DAS Registration Server directly. The set of steps that follows describes how to manually query the DAS Registration Server for details of a specific DAS service and how to test that the service is operating as expected.

1. Visit http://www.dasregistry.org/ using an Internet Browser.
2. First, find details of all of the DAS servers that are able to serve features using the protein identifiers that you are using (e.g. UniProtKB protein accessions). Hover over the "list" menu item at the top of the screen. A short menu will appear. Click on "list sources."
3. To restrict the list to DAS servers that provide annotation for (for example) UniProtKB protein accessions, click on the pull down list labelled "authority." Select "UniProt" from the list. (See Note 2 for further filtering options.)
4. Click on the "display" button on the right.
5. You will now be presented with a list of DAS servers that you can explore. The DAS Registration Server uses a "traffic light" system to indicate which capabilities the DAS servers posses. To find out the meaning of each "light," hover over it.
6. To examine details and documentation of a specific DAS server, click on the blue "i" icon on the left hand side. This will open a new page, "DAS Source Details," documenting the DAS service.
7. Finally, to validate that the server is operating correctly, click on the green tick at the bottom of the "DAS Source Details" page. This will open a new page "Validate DAS/1 Source." Click on the "Validate" button to test the capabilities of the server, or if you think the server may be failing.

5. Dasty2: A Powerful Web-Based Client

The Dasty2 DAS client (http://www.ebi.ac.uk/dasty/) (9) is a rich and flexible DAS client that runs in an internet browser. It is used to query a single protein at a time and has been developed for use with DAS servers that provide and annotate protein sequence. By default it is configured for query using UniProtKB protein accessions.

The design of the DAS system has largely focused on developing standardised XML formats for data exchange (from servers to clients). For example, individual annotations ("features") include a feature type, feature id, a label, and the start and stop coordinates of the feature and a score, together with other optional fields. Standardising this format is obviously essential to allow clients to be able to query multiple DAS servers; however, it does not solve the problem that separate organisations will use different terminology to describe different feature types. This makes it very difficult to perform a comparative analysis of the feature types served from different institutions. Fortunately, this problem has been recognised and has been addressed through the development of the Protein Feature Ontology (10), by the BioSapiens Network of Excellence (11) and through the use of the Evidence Codes Ontology. See Note 3 for a very brief explanation of ontologies. The DAS servers provided by members of the BioSapiens Network are among the first to take advantage of this standardisation of terminology.

The Dasty2 DAS client (also funded by the BioSapiens Network) incorporates the use of these ontologies in the interface, providing links to term definitions and allowing filtering, sorting, and ordering by both feature type and evidence code ontology terms.

Following is a description of how to query Dasty2 for a specific protein and how to manipulate the user interface to focus upon annotations of interest to the researcher.

Note that by default, Dasty2 retrieves annotation from DAS servers that are registered as being associated with the BioSapiens Network. It is possible to extend your search to include DAS servers from outside this network, so long as they accept UniProtKB protein accessions. If you use non-BioSapiens DAS servers, there is no guarantee that the DAS server will make use of the Protein Feature Ontology or the Evidence Codes Ontology for feature annotation.

1. Visit http://www.ebi.ac.uk/dasty/ using an Internet browser (see Fig. 3).
2. Enter the UniProtKB protein accession that you are interested in into the "Protein ID" text field and click "GO." Note that there are several example accessions given below the text field.
3. The Dasty2 client will immediately start to query all of the available DAS servers with the protein accession that you have entered. This is done in parallel with results from each server being displayed as soon as they arrive. Some of the registered DAS servers may include no annotation of the protein requested or may be inactive for another reason (maintenance down-time for example). This does not impede Dasty2 in any way; however, if you are expecting or looking for annotation

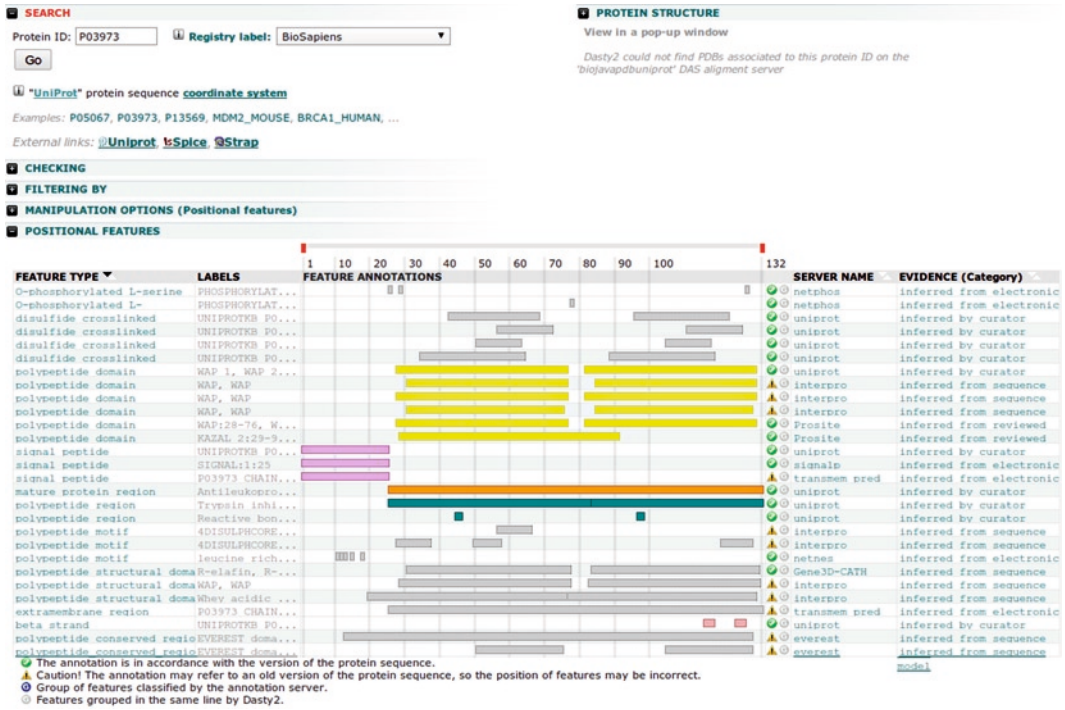


Fig. 3. The Dasty2 DAS client in action. This view shows part of the Dasty2 user interface displaying the DAS tracks from ten different DAS annotation servers for a single protein. In this case, Dasty2 has requested annotation from a total of 33 separate DAS servers.

from a specific DAS server, it is wise to check that the service is responding correctly (see Note 4).

4. Scroll down to the section "Positional Features." This section displays all of the feature annotations that have been loaded in the previous step. Features of the same type may be grouped together on to one row, either using information from the server or according to the configuration of Dasty2. This table includes the columns:
   (a) "Feature Type", which displays (where provided) the Protein Feature ontology term categorising the features displayed on that row.
   (b) "Labels" provides a simple, non-standardised label for the feature type.
   (c) "Feature Annotations" displays the position of the feature relative to the sequence. This is an interactive display with the ability to zoom (grab and slide the red "handles" at the top of the "Positional Features" section and then click the grey "Zoom" button). You can also hover over, or click any of the features displayed for more complete information about the feature. If you click on a

feature, you can view details of the sequence in this region at the bottom of the Dasty2 interface.

(d) "Server Name" indicates which DAS server the annotation has been retrieved from.

(e) "Evidence (Category)" displays the Evidence Codes Ontology term that the features on that row are annotated with, typically differentiating between annotations for which there is direct experimental evidence from annotations that have been inferred by a human curator or annotations that have been derived by automatic means, for example, pattern matching or the use of Hidden Markov models.

Note that it is possible to add additional columns (including "Score" and "Feature ID") or remove columns by expanding the "Manipulation Options (Positional Features)" section.

5. For some highly annotated proteins, there may be many rows of annotation displayed, much of which may be irrelevant to the research problem being addressed. The Dasty2 interface provides several mechanisms to allow you to manage this:

   (a) You may reorder the DAS tracks on the screen by holding down the primary mouse button[1] on a DAS track that you wish to relocate and drag it up or down to a new position. This is useful for viewing a selection of DAS tracks next to each other that you wish to compare directly.

   (b) You can filter the DAS tracks displayed. Scroll up to the "Filtering By" section and expand the section by clicking on the heading. In this section, you can filter by feature types, DAS server name and evidence code. (The feature type and evidence code filters make use of the Protein Feature Ontology and Evidence Code Ontology respectively.)

   (c) You may modify the order of the DAS tracks by clicking on one of the column headings. Repeatedly clicking a heading reverses the sort order.

6. Some annotations may refer to the entire molecule rather than just a section of the sequence, for example, the list of literature citations associated with the molecule. These features can be viewed on Dasty2 by expanding the "Non Positional Features" section. If the DAS server has provided one or more hyperlinks to external sources, these are represented by a purple "i" icon, to the right of the notes section.

[1]Left mouse button on a Microsoft Windows or Linux PC.

Here, some features of the Dasty2 DAS client have been described. Its capabilities extend beyond those described here, including for example the ability to display the protein structure, employing the structure extensions to DAS. As well as Dasty2, other high quality DAS clients exist, including the DAS client built into Ensembl (http://www.ensembl.org/) and the powerful Spice DAS client (http://www.efamily.org.uk/software/dasclients/spice/) (8), which provides a sophisticated protein structure viewer on to which DAS annotation can be projected.

DAS provides a very powerful way of accessing integrated data from many disparate sources. It has the restriction, however, that the available clients all focus on *one protein at a time*. If the researcher wishes to collate annotation for large sets of proteins in a single step for further analysis, BioMart may offer a more suitable alternative as described below.

6. Retrieving Sequence Annotation Using BioMart

BioMart is a powerful query optimised database system that can be used with any kind of data. Individual BioMarts are built and maintained by separate groups around the world, including for example:

- the Ensembl project (http://www.ensembl.org/biomart/martview) (12)
- the UniProt Consortium (http://www.ebi.ac.uk/uniprot/biomart/martview)
- the InterPro database of protein families, domains, regions, repeats, and sites (http://www.ebi.ac.uk/interpro/biomart/martview) (13)
- the PRIDE Proteomics Identifications Database (http://www.ebi.ac.uk/pride/prideMart.do) (14)
- the Reactome curated knowledge-base of biological pathways (http://www.reactome.org/cgi-bin/mart) (15).

A full list of publicly available BioMart implementations can be found on the BioMart home page at http://www.biomart.org. BioMart provides several benefits that are beyond the scope of this chapter, which can be explored on the documentation page on the biomart.org Web site. Two important features of BioMart are its ability to quickly deliver very large datasets in response to queries and the "federation" (linking together) of physically separate BioMarts so that users can build complex queries across two BioMarts at the same time. It should be noted that the federation of BioMarts is a quite different concept to DAS. DAS servers

need not be aware of each other in any way, it is the DAS client that is responsible for collating data from separate DAS servers. Federated BioMarts, however, need to be configured and linked by the mart maintainers.

The real power of BioMart for the biological researcher is its ability to handle queries that return large volumes of data. It is possible, for example, to submit a long list of protein accessions to a BioMart and retrieve information relating to all of the proteins in one query.

BioMart differs from DAS in that it does not define a standard data structure. Each BioMart is structured differently, and so the precise instructions for using any two BioMarts will differ. There are, however, generally applicable techniques to building BioMart queries, which is illustrated in the example below.

This example makes use of the InterPro BioMart to retrieve automatic annotation of protein sequences.

1. Visit http://www.ebi.ac.uk/interpro/biomart/martview in an Internet browser.
2. Click on the "- CHOOSE DATABASE -" pull-down list. You will see three BioMarts listed here. This is a consequence of the fact that the InterPro BioMart has been federated (linked) to both the PRIDE and the Reactome BioMarts. Select the "InterPro BioMart" item.
3. A new pull-down list will appear labelled "- CHOOSE DATASET -". Most BioMarts are organised into several sets of data. The InterPro BioMart is organised into three: a protein-centric dataset "Protein Matches" an InterPro Entry centric dataset and a dataset of annotation on UniParc protein sequences. Select "Protein Matches" from this list.
4. On the left of the screen, two headings will appear, "Filters" and "Attributes."
5. Click on the "Filters" heading. This is where you restrict or "filter" the rows of data returned from the BioMart. In this example, the filter will be a list of UniProtKB protein accessions, such as that generated using the PICR service as described earlier in this chapter.
6. Click on the "Protein Filters" heading on the right hand side.
7. You will now see a large number of different filters that are available to you, including filters by protein accession, sequence length and taxonomy. Find the filter at the top of this page labelled "UniProtKB Protein Accession."
8. At this point, you can either cut and paste a list of UniProt KB protein accessions into the adjacent text area on the right of the screen, or if you have already prepared a text file containing the list, you can browse to this file by clicking on the

"Choose File" button, situated below the text area. Note that the format of the file or the pasted text is flexible, you can either separate the protein accessions by placing one on each line, or you can separate them using commas.

9. Now you have restricted the BioMart to return information about the list of proteins you are interested in, you need to select the data fields that you are interested in. The results will be returned in a table, so you are effectively selecting the columns of this table. Click on the "Attributes" heading on the left of the screen.
10. The right hand side of the screen will now change to display a list of column headings, each with a check box to allow you to select the heading. You can select any data you are interested in from this list; however, if you have filtered by a list of protein accessions, you should include the "UniProtKB Protein Accession" attribute, so you know which protein is being described on each line of the results. For the purposes of this example, click the following attributes:
    (a) UniProtKB Protein Accession
    (b) Signature ID (Name)
    (c) Start Position
    (d) Stop Position
11. Now that you have built a filter to restrict the rows of data, and selected the columns of information that you wish to receive, click on the "Results" button at the top of the page. After a short delay, you will see the *first ten* rows of results displayed. At this point, you may click on the "Filters" or "Attributes" heading on the left to modify the query if it is not quite what you require.
12. Once you are happy with the query, you should now select a format for the full set of results. Find the select pull-down at the top of the page labelled "Export all results to." By default this will be set to "File." A useful feature is to request that the BioMart system sends you an email when the complete set of results is ready to download. This is especially useful if you are expecting a very large set of results, for example, if you have queried the BioMart for the details of several thousand proteins. Next to this select pull-down is a second select pull-down defaulting to "TSV" (tab-separated values file). You may select alternative formats from this pull-down, including "HTML" (for viewing in an Internet browser), "CSV" (comma-separated values file), or "XLS" (Microsoft Excel spreadsheet format).
13. Click on the green "Go" button near the top of the screen to retrieve the full set of results in the selected format.

This is only an introduction to the use of the BioMart interface. Following the basic principles of filtering, selecting attributes and finally selecting an output format you can build a large range of different queries using any BioMart.

7. Notes

1. "SwissProt" and "TrEMBL" can be considered as a single database, being the two components of the UniProt Knowledge Base (UniProtKB), so you can safely leave both of these ticked. It should be taken into account, however, that these two components of UniProtKB are created separately. UniProtKB/SwissProt is human-curated and considered to contain very high quality annotation. UniProtKB/TrEMBL comprises automatically predicted annotation.
2. In addition to the "authority" filter, you can filter the listed DAS sources by organism (taxonomy), type (protein or nucleic acid sequence), capability, and label. Capability allows you to select sources that (for example) provide protein sequence, protein structure, or annotation. Label relates to the projects that the DAS source are a member of.
3. An ontology is a sophisticated controlled vocabulary, comprising unambiguously defined terms with stable identifiers and defining relationships between the terms, for example, parent–child relationships and partitive relationships.
4. To ensure that a specific DAS server has responded successfully to a request made in the Dasty2 DAS client, click on the heading "System Information" near the top of the Dasty2 interface, and located in the "Checking" section. This will expand a list of all of the DAS servers that have been queried. (including 33 separate BioSapiens Network DAS servers at the time of writing). Servers that are listed with a comment in green or red are active and have returned a valid response to Dasty2. Red comments indicate that the DAS server has no annotation for the protein accession you have entered. Comments in orange indicate that the DAS server is not operating correctly.

Acknowledgments

The author would like to acknowledge Richard Côté (the developer and maintainer of PICR) and Rafael Jimenez (the founding developer of Dasty2) for their assistance in proofreading this text.

References

1. Kersey PJ, Duarte J, Williams A, Karavidopoulou Y, Birney E, Apweiler R (2004) The International Protein Index: an integrated database for proteomics experiments. Proteomics 4:1985–1988
2. UniProt Consortium (2009) The Universal Protein Resource (UniProt) 2009. Nucleic Acids Res 37:D169–D174
3. Côté RG, Jones P, Martens L, Kerrien S, Reisinger F, Lin Q, Leinonen R, Apweiler R, Hermjakob H (2007) The Protein Identifier Cross-Referencing (PICR) service: reconciling protein identifiers across multiple source databases. BMC Bioinformatics 8:401
4. Dowell RD, Jokerst RM, Day A, Eddy SR, Stein L (2001) The distributed annotation system. BMC Bioinformatics 2:7
5. Stein LD, Eddy S, Dowell R Distributed Sequence Annotation System (DAS) Version 1.53. http://www.biodas.org/documents/spec.html
6. Blankenburg H, Finn RD, Prlić A, Jenkinson AM, Ramírez F, Emig D, Schelhorn S, Büch J, Lengauer T, Albrecht M (2009) DASMI: exchanging, annotating and assessing molecular interaction data. Bioinformatics 25:1321–1328
7. Jenkinson AM, Albrecht M, Birney E, Blankenburg H, Down T, Finn RD, Hermjakob H, Hubbard TJ, Jimenez RC, Jones P, Kähäri A, Kulesha E, Macías JR, Reeves GA, Prlic A (2008) Integrating biological data – the Distributed Annotation System. BMC Bioinformatics 9(8):S3
8. Prlić A, Down TA, Kulesha E, Finn RD, Kähäri A, Hubbard TJP (2007) Integrating sequence and structural biology with DAS. BMC Bioinformatics 8:333
9. Jimenez RC, Quinn AF, Garcia A, Labarga A, O'Neill K, Martinez F, Salazar GA, Hermjakob H (2008) Dasty2, an Ajax protein DAS client. Bioinformatics 24:2119–2121
10. Reeves GA, Eilbeck K, Magrane M, O'Donovan C, Montecchi-Palazzi L, Harris MA, Orchard S, Jimenez RC, Prlic A, Hubbard TJP, Hermjakob H, Thornton JM (2008) The Protein Feature Ontology: a tool for the unification of protein feature annotations. Bioinformatics 24:2767–2772
11. Thornton J (2009) Annotations for all by all – the BioSapiens network. Genome Biol 10:401
12. Hubbard TJP, Aken BL, Ayling S, Ballester B, Beal K, Bragin E, Brent S, Chen Y, Clapham P, Clarke L, Coates G, Fairley S, Fitzgerald S, Fernandez-Banet J, Gordon L, Graf S, Haider S, Hammond M, Holland R, Howe K, Jenkinson A, Johnson N, Kahari A, Keefe D, Keenan S, Kinsella R, Kokocinski F, Kulesha E, Lawson D, Longden I, Megy K, Meidl P, Overduin B, Parker A, Pritchard B, Rios D, Schuster M, Slater G, Smedley D, Spooner W, Spudich G, Trevanion S, Vilella A, Vogel J, White S, Wilder S, Zadissa A, Birney E, Cunningham F, Curwen V, Durbin R, Fernandez-Suarez XM, Herrero J, Kasprzyk A, Proctor G, Smith J, Searle S, Flicek P (2009) Ensembl 2009. Nucleic Acids Res 37:D690–D697
13. Hunter S, Apweiler R, Attwood TK, Bairoch A, Bateman A, Binns D, Bork P, Das U, Daugherty L, Duquenne L, Finn RD, Gough J, Haft D, Hulo N, Kahn D, Kelly E, Laugraud A, Letunic I, Lonsdale D, Lopez R, Madera M, Maslen J, McAnulla C, McDowall J, Mistry J, Mitchell A, Mulder N, Natale D, Orengo C, Quinn AF, Selengut JD, Sigrist CJA, Thimma M, Thomas PD, Valentin F, Wilson D, Wu CH, Yeats C (2009) InterPro: the integrative protein signature database. Nucleic Acids Res 37:D211–D215
14. Jones P, Côté RG, Cho SY, Klie S, Martens L, Quinn AF, Thorneycroft D, Hermjakob H (2008) PRIDE: new developments and new datasets. Nucleic Acids Res 36:D878–D883
15. Matthews L, Gopinath G, Gillespie M, Caudy M, Croft D, de Bono B, Garapati P, Hemish J, Hermjakob H, Jassal B, Kanapin A, Lewis S, Mahajan S, May B, Schmidt E, Vastrik I, Wu G, Birney E, Stein L, D'Eustachio P (2009) Reactome knowledgebase of human biological pathways and processes. Nucleic Acids Res 37:D619–D622

Chapter 8

Tranche Distributed Repository and ProteomeCommons.org

Bryan E. Smith, James A. Hill, Mark A. Gjukich, and Philip C. Andrews

Abstract

Tranche is a distributed repository designed to redundantly store and disseminate data sets for the proteomics community. It has several important features for researchers, including support for large data files, prepublication access controls, licensing options, and ensuring both data provenance and integrity. Tranche tightly integrates with ProteomeCommons.org, an online community resource that offers a variety of useful tools for proteomics researchers, including project management and data annotation. In this chapter, we discuss the development of Tranche and ProteomeCommons.org, paying particular attention to why it is desirable that data be publicly available and unrestricted as well as the challenges facing data archiving and open access. We then provide a technical overview of Tranche and ProteomeCommons.org as well as step-by-step instructions for using these resources, including the graphical user interface (GUI), command-line tools, and Application Programmer Interface (API). We end with a brief discussion of current and future development efforts and collaborations.

1. Introduction

ProteomeCommons.org was developed and released in 2004 to provide data and software hosting to the proteomics community. Its founding goals were to support data and software reuse and dissemination as well as integration with other proteomics resources (1). Data sets that were hosted on ProteomeCommons.org were available for download via HTTP. While this was a simple and immediately valuable service, a more scalable solution was needed to handle the raw data that is produced by high-throughput mass spectrometry instruments. Furthermore, given the unpredictable nature of hardware failures, redundancy was necessary for the long-term archiving of the hosted data sets. In response to these and other needs, the Tranche-distributed repository project was developed in 2005 and publicly released at the 2006 American Society for Mass Spectrometry (ASMS) conference.

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696,
DOI 10.1007/978-1-60761-987-1_8,

Proteomics data sets tend to be large, expensive to generate, and contain information beyond the immediate needs of the original investigators. There are many reasons to make data publicly available; public access to data sets protects the interests of peer review (e.g., critical evaluation and replication of results) as well as the future reevaluation of data sets as new analytic tools and additional data sets become available. Another important reason to make data publicly available is to satisfy dissemination requirements for funding agencies.

Recent calls for broader data sharing have cited the genome project and other research efforts to share data (2–4). The role of both pre- and postpublication data sharing in the genome project was particularly important in accelerating progress, allowing more rapid development of new tools, and providing broad dissemination of genome data which increased the impact of the genome project. Pre- and postpublication data sharing have different uses and some differences in data infrastructure, with the former usually requiring some degree of security in the form of encryption or limited data access.

The general infrastructure for data sharing has been limited, particularly in reducing the barriers for getting data into repositories and in the dissemination stage (5). The resources available for building the infrastructure have been limited and decision making has been hampered by economics, the volume of data, and the rapidly changing technologies. While there is general consensus that sharing data sets is desirable, the technical and social challenges are significant and pessimism has been expressed over the feasibility (6) or the ultimate usefulness of data sharing in all cases (7). As these authors point out, the cost benefit aspects of data sharing, the time lag involved in building infrastructure, obsolescence times, intellectual property issues, and many other concerns affect the development of an effective data sharing infrastructure. Confounding this situation are the unique data features that must be accommodated in many fields of research. These concerns are being addressed by the proteomics community in several ways beyond Tranche and Proteomecommons.org. For example, centralized resources like Peptideatlas.org (8), TheGPM.org (9), PRIDE (10), HPRD (11), and Peptidome (12) all harvest data and place various levels of metadata in their databases for easy mining and access for investigators. Many of these resources download data sets from Tranche, and some of these resources run the data through their own data pipelines to allow for improved comparisons across data sets.

Currently, some journals recommend public release of proteomics data sets. As of April 2007, *Molecular and Cellular Proteomics Journal* and *The Journal of Proteome Research* officially recommend depositing all mass spectra output data as supplemental material associated with protein identifications (13) following the recommendation of a group of leaders in proteomics

in March 2005. In an editorial in 2007, *Nature Biotechnology* (14) recommended that all proteomics data associated with manuscript submissions be deposited in public repositories, sharing many of the concerns that we outline in this introduction. The submission requirements for the journal *Proteomics* (last updated in 2008) state that peak lists should be deposited in a public repository and submitted as supplemental material, though there is no statement that these peak lists are required or recommended (15). Funding agencies, on the other hand, are beginning to stipulate the submission of data sets to repositories. As of October 2003, any investigators receiving NIH funding of $500,000 or more within a year are required to provide a data sharing plan; if sharing the data is not feasible, the investigator must explain why (16).

There are many barriers to the general adoption of data sharing. Foremost, we should consider the motivations of individual researchers. For example, researchers might not share data due to the understandable perception that ad hoc experimental design and the lack of experimental standards might compromise the reuse of the data sets (7).This is partly addressed by the proper annotation of data sets. While publications provide a depth of information about a data set that is difficult to capture by other means, they are not ideal as primary annotation sources because they are often incomplete, and the annotations are provided in natural language, which is a challenge for data mining applications. This situation is further improved when the experimental parameters are properly stored as metadata. Researchers also may not be sufficiently motivated to share their data when it is neither required nor sufficiently rewarded, which could change if funding and career advancement incorporated not only the generation, but also the public availability of properly annotated data sets (17). Relatively minor shifts in faculty evaluation processes could go a long way in enhancing the public availability of scientific data. Funding agencies, publishers, and professional societies have an opportunity to influence this process through their policies.

Even where there is sufficient motivation to share data, there can be additional obstacles. The investigator may not be prepared to release the data set until all interpretations have been completely exhausted, feeling that unanalyzed or underanalyzed data remains; this is true despite the likelihood that most data sets will not be further analyzed in the original laboratory beyond the primary goals due to priorities and resource constraints. In the case that multiple publications are prepared using the same data set, the author might wish to withhold the data set from public release beyond the first publication. This situation can arise when funding or promotion deadlines require early publication or when logistics interfere with timely follow-ups to initial publications. Some investigators may also have concerns about overlooked

knowledge that can be derived from the data sets by other investigators with different perspectives or computational algorithms that might result in "lost" intellectual property. One response to this concern is that this is one of the anticipated and desired outcomes of data sharing that will result in faster progress in medical research and advancements in treatment. This issue is analogous to the publication of a manuscript which may contain data sets subject to alternative interpretation and is one of the reasons for publication. It might also be addressable by more general access to computational tools. Uncertainty about the interpretation of one's own data sets can also be an inhibitory factor; however, the peer review process exists to help authors confirm that their interpretations are reasonably accurate.

Even when funding sources and publishers do not require that data is publicly released, there is considerable value to the broader community fully understanding and embracing the value of open access to scientific data. This should be particularly clear from the field of genomics, considering how the availability of large genome databases has led to the development of multiple new fields of post-genome research, including proteomics (18, 19).

It is also important to recognize that just because a data set is *available* does not mean it is *usable,* which generally requires that the data set is unrestricted (or minimally restricted) and that it has the appropriate metadata so that it is *findable* and can be placed in an interpretable experimental context. This leads to two important topics: open data and annotations.

When data sets are unrestricted and freely available, they are said to be *open*. This unambiguously allows anyone to freely use the data. Restrictions primarily come in the form of copyright law, though other legal or ad hoc restrictions might apply. This is a complicated topic that will change over time as intellectual property continues to be defined. What is interpreted as a creative work, and hence protected under copyright law, varies by jurisdiction. (For example, database structure might be defined as a creative work, though the same may not be said of the underlying data; however, there is also the database right of the European Union, which is similar to copyright law but protects the underlying data.) Attribution is a related right, and many available licenses include attribution stipulations. In the research community, attribution is a de facto requirement, even when data are not legally protected. Since it is likely that data sets with the least licensing restrictions will see the broadest impact and citation, it is important to retain attribution without restricting the data. Because all data on Tranche is digitally signed upon upload, Tranche inherently supports provenance.

The uncertainties and difficulties associated with ascertaining the appropriate restrictions suggest the value of clearly-defined open data. One tool that protects open data is the Creative Commons CC0 waiver, which is a "no rights reserved" option

that places data in the public domain as completely as is feasible (20). CC0 also provides a machine-readable document so that automated agents can recognize that the associated data can be used without restriction. The association of a machine-readable license or waiver is becoming increasingly important as more tools are developed to aggregate and analyze data.

The most practical justification for open access to data is for the reevaluation of data sets. Without public data sets, there is no foundation for a broader bioinformatics community, which has the potential to contribute discoveries that require subtle statistical analysis of many data sets as well as develop improved algorithms. By comparison, consider how public genetic databases have provided essential evidence for posttranscriptional gene regulation as well as for the discovery of coexpressed genes (21). Furthermore, the cost of producing high quality data sets can be quite high, so encouraging data reuse could prove to be an economical choice for limited proteomics research. This is particularly important when considering biological samples that are rare or difficult to obtain (such as with wild type gastrointestinal stromal tumors (22)) or unique (for example, the tyrannosaurus rex collagen protein (23) and hadrosaur proteins (24)). Mining these data sets to plan future experiments is another application of Tranche that allows investigators to estimate the variance, dynamic range, and other parameters important when optimizing experimental design. Additionally, aggregation of data sets from multiple studies can be used to improve statistical models.

The issue of whether data is findable is a key issue most directly addressed by providing the appropriate metadata in a searchable format, otherwise known as *annotating* (or *curating)* the data set. Much of the data currently generated in proteomics and related fields is not thoroughly annotated, which compromises the value of the data sets; however, as this issue is remedied, future experiments can be designed with more insight, meaning that researchers can be more productive (17). As we discuss later, *semantic searches* are generally more useful than keyword searches, and metadata that is highly structured offers more value, particularly when a controlled vocabulary is used.

Annotations, which provide the context for the data sets and hence aid their findability, involve two separate issues. The first is the definition of the annotations. The Human Proteome Organisation (HUPO) Proteomics Standard Initiative (PSI) has developed the Minimum Information about a Proteomics Experiment (MIAPE) standard to specify the minimal metadata that should accompany proteomics data (25). The second issue is compliance: annotations must be completed and accurate for maximal impact. Considering the potential size and complexity of these annotation standards as well as the long-term collaborative nature of proteomics projects, this is not a trivial task. For example,

MIAPE offers separate modules for various stages and technologies related to a proteomics experiment, such as study design and sample generation, mass spectrometry, gel electrophoresis, and others (http://www.psidev.info/miape/). Each of these modules contains multiple categories of required information, with each category containing multiple related fields. Gathering all the required information typically requires input from several individuals who collaborated in the study. Even in the case that software is used to extract as much information as possible from the data files, much of the annotation will need to be performed manually.

There are many technical and logistical challenges involved with disseminating and archiving large data sets. Foremost, there must be sufficient disk space to hold mass spectrometry data sets, which can be quite large. Current mass spectrometers can generate up to 1 GB (or more) per hour of compressed data (26). For example, the Thermo Fisher Orbitrap (Thermo Fisher Scientific Inc., Waltham, MA, USA) can produce up to 100 MB per hour while Bruker Daltonics' Micro TofQ (Bruker Daltonik GmbH, Bremen, Germany) can produce up to 500 MB per hour (27). Additionally, server hardware failure, in the absence of redundancy, will generally result in data loss. Multiple disk failures over time can wipe out multiple replications; in the absence of a scheme to reintroduce redundancy, every data set will eventually be lost.

In addition to maintaining a valid copy of the data, the challenge of being able to access the data still remains. Changing storage media technologies (e.g., 9.5" floppy, 1.5 MB floppy, zip disk, CD, DVD, flash drives, etc.) can make accessing data difficult that is even only a few years old. Additionally, there is the issue of the large and constantly evolving variety of mass spectrometer output file formats. Several standard formats (mzXML (26), mzData (28) and most recently, mzML (29)) have been developed to deal with this problem, though their long-term success will likely be determined by their adoption by mass spectrometer manufacturers as well as the development of appropriate tools to convert from existing native formats. This is not only an issue for the long-term archiving of data, but will also present a continuing challenge for researchers.

Long-term preservation of data also requires an effective infrastructure. This begs the question: who is responsible for providing that infrastructure? The choices include federal agencies, the universities, individual researchers, and private industry. Each of these choices has its strengths and weaknesses and no single solution is clearly applicable across all fields, applications, or even data types. One advantage of distributed storage systems like Tranche is that it provides the potential for all of these stakeholders to participate in supporting a common data infrastructure through investments in hardware and sharing the ongoing costs of maintenance and administration.

These issues, to varying degrees, have directed the development of both Tranche and ProteomeCommons.org. Dissemination and documentation requirements of data sets are still being defined by funding sources as well as journals, and issues like annotation standards are attracting the attention of standards organizations within the proteomics community, but these issues are far from settled. Despite the ongoing developments, the value of these resources has already been clearly demonstrated: during the 6-month period starting in February through the end of July, over 3.9 TB of data were downloaded from the ProteomeCommons.org Tranche repository.

Below, we discuss how Tranche and ProteomeCommons.org were developed, including what we have learned during the process of hosting and sharing data. We also discuss how to get started using Tranche and ProteomeCommons.org. We will conclude with planned development efforts that will further address the challenges and issues outlined.

2. Methods

Tranche is a distributed repository designed using principles from peer-to-peer networking (redundancy and load balancing) combined with client-server architecture (authentication and reliability), which can best be described as a *distributed server model*. Data sets are uploaded to and downloaded from our network using any of our client tools. Figure 1 shows a screenshot of our graphical user interface (GUI) with a list of data sets available for download.

Our model is strongly decentralized – in the event that any number of servers are offline, the remaining servers can still provide service (though some data might be temporarily unavailable). The network is *federated*, as every server has its own list of trusted users and can be managed separately.

The key to data availability on a Tranche network is redundancy. Like a RAID array, we assume that servers will fail, so every chunk of data that is uploaded must be replicated a minimum number of times. Choosing the proper number of replications is complex and not easily modeled. Hardware failures and server downtime are generally unpredictable, with an innumerable set of factors, including lightning strikes and rodent damage as well as staff turnover and funding. The more servers that are online (to accommodate more data), the more redundancy that is required to accommodate the additional uncertainty. In other words, total number of servers online should be proportional to the number of replications. (With our current model, an increase in the number of servers requires a linear increase in the replications for data

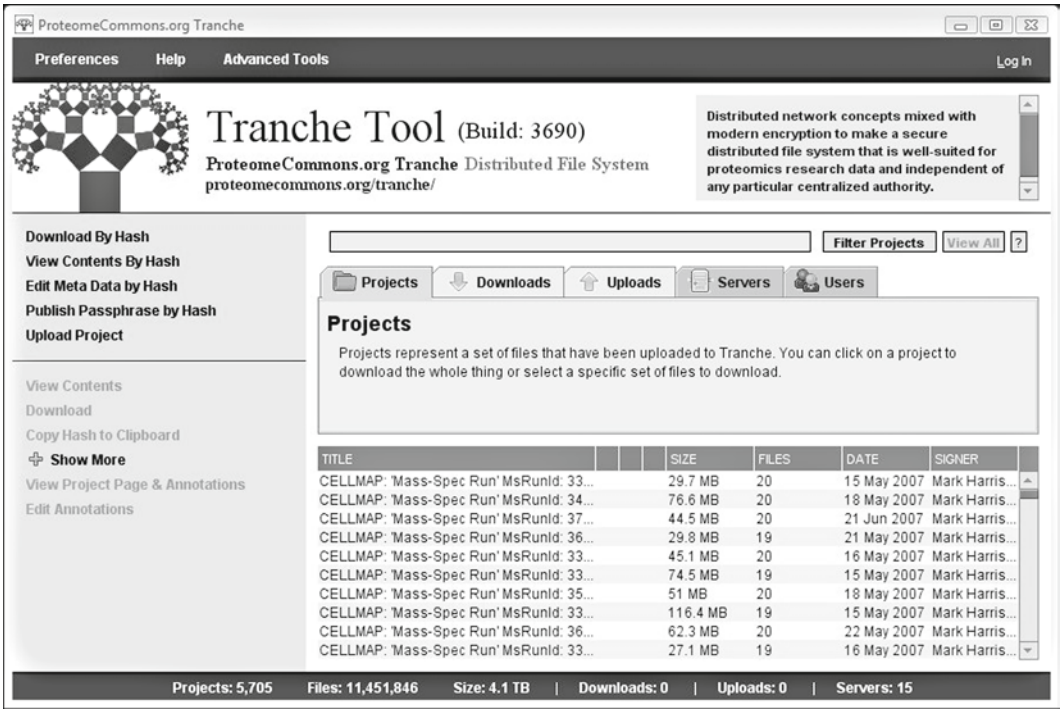


Fig. 1. The graphical interface with the projects tab selected. Data sets are listed here for download.

to remain available; the network model for the next version of Tranche, however, will only require a logarithmic increase in replications to accommodate additional servers.) In the case that it is rare for more than two servers to be offline, then three replications should be enough. The ProteomeCommons.org Tranche network currently requires three replications when data are uploaded.

Load balancing is another strength of using multiple servers. This allows multiple servers to share the burden of user requests. As a heuristic, during an upload or a download, a client will select a server on a moment-by-moment basis based on how much outstanding work the server has combined with its latency. This is a simple implementation of the "nearest neighbor," where the cost of using a server is approximated based on recent performance.

Tranche does not store intact files on servers. Instead, for reasons of performance and scalability, a file is separated into data chunks, which have a maximum size of 1 MB. (For example, a 5.3 MB file would require five 1 MB-data chunks and one 0.3 MB data chunk, for a total of six data chunks.) All the data chunks in a file are described by a metadata chunk. Given a metadata chunk, all data chunks can be downloaded and reassembled into the original file.

A data set is any directory and all of its contents; in Tranche, there are no structural requirements imposed on a submission.

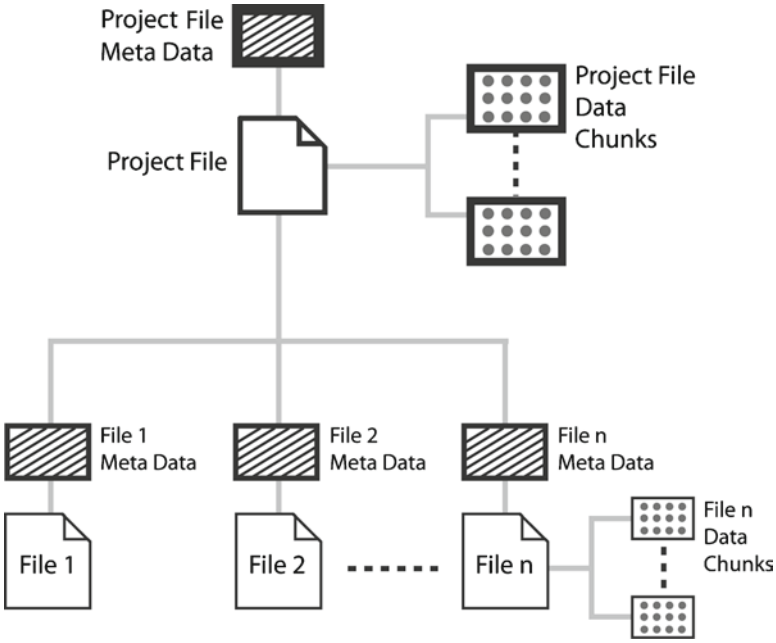


Fig. 2. A data set is simply a collection of files, and is described by a Project File. Each of these (including the ProjectFile) is described by a meta data chunk and is stored as one or more data chunks.

Figure 2 demonstrates how a data set is stored on Tranche. Note that the individual files (each with one metadata chunk and associated data chunks) are described by a *ProjectFile*, which points to the metadata chunks for each file as well as describes their location relative to the root directory of the data set. Just like any other file, the ProjectFile has a metadata chunk and data chunks; you can download and view the ProjectFile just like any other, though it is intended to be used behind-the-scenes by the download tool or for other client applications. (In Fig. 2, the ProjectFile and its constituent metadata chunk and data chunks have a thick outline.)

Tranche servers store chunks in a b-tree structure (30), meaning that insertions, searches and deletes all run in logarithmic time, $O(\log n)$. (This means that the worst-case time to perform these actions is the logarithmic value of the total number of nodes in the collection.) This b-tree structure is made up of hierarchically-arranged nodes, each containing a maximum of 1,000 chunks, with 256 nodes branching from each parent node. The tree is rebalanced as data is added, which is important since most servers will hold millions of chunks. For example, assume the average data chunk is 400 KB (which is not too far from what we have observed), and that its associated metadata is 4 KB. A 4 TB server could then hold, on average, over 21 million chunks. If we were to search for a particular chunk, it would take around an

average of four operations to identify the node containing the chunk. Though a node holds a maximum of 1,000 chunks, a node always branches to 256 new nodes, so the effective seek time for a node when n chunks are in the tree is $\log_{256} n$. At that point, a linear search of the header of the node will quickly identify the chunk.

Each node in the b-tree structure is stored as a separate file. One potential source of data loss is the corruption of one of these "data block" files, such as might happen if the server is killed in the middle of a write operation. Upon starting up, each data block file is checked for corruption and repaired using data from other servers when possible.

There are other sources of potential data loss. On a large network, disk failures must be continually accommodated. Furthermore, chunks can be corrupted during transmission. The redundancy on the network is only maintained in the long-run if lost replications are repaired or reintroduced. To this end, each server spends time downloading desired chunks, deleting unnecessary chunks, and searching for corrupted chunks and replacing them.

The first and last activities help mitigate the above sources of data loss, though with considerable latency. (We will discuss a better solution when we discuss the future of Tranche.) It is worth mentioning an additional source of data loss: a malicious attack. An attack would probably only impact a single replication of any given chunk on a compromised server; but in the worst-case scenario in which an attacker gains authenticated access to the Tranche network, the attacker might delete some or all copies of a chunk. If this occurs, we have a separate Tranche network with different authentication that actively copies over any missing data that it has available.

As mentioned above, servers will attempt to download "desired" chunks as well as delete "unnecessary" ones. This brings us to two very important concepts: the *hash* and the *hash span*. Every chunk (and indeed every file and every data set) has a hash associated with it. These hashes are unique identifiers, and they can be recalculated at any time (i.e., they are deterministic). Each associated hash is generated by combining the four sources: the MD5, SHA-1, and SHA-256 hashes of the chunk along with the number of bytes. (Though it is possible that two separate chunks might have the same hash, resulting in a collision and hence data loss, it is highly unlikely. Since a hash is 76 bytes, the odds of randomly generating the same hash is 1 out of $1.06 * 10^{183}$.)

Since each chunk has an identifier, it is possible to allow a server to accommodate a portion of all the network by assigning it a range of desired hashes, which is the server's hash span. By analogy, this is similar to providing multiple lines to pick up reserved tickets to a concert based on the first letter of your last name. If there are going to be many people, you might want two

lines: A–M and N–Z. If there are many reserved tickets, then there might need to be several lines with shorter ranges. Hash spans not only allow a network to find a chunk more quickly, but they also allow each server to accommodate a portion of the network based on available disk space or any other factor. Additionally, an administrator can remove a hash span from a particularly troublesome server so that it will not likely receive many more chunks, but its current data will continue to be available. (A server can also be flagged as "read-only" to entirely prevent any data from being stored on it.)

When a server is downloading "desired" chunks, it is downloading chunks with hashes within its hash span ranges. If it is deleting an "unnecessary" chunk, it is removing a chunk that does not belong to its hash span – but only if there are already enough copies on the network.

Figure 3 illustrates the storage of data sets and files as data and metadata chunks as well as the upload and download process.

The upload tool processes each file in a data set separately. The tool must first identify the data chunks, and generate a metadata chunk so that the data chunks can be reassembled back into the original file upon download. Each chunk is then uploaded to three servers in the network. Preferably, the chunks will be put on three servers with hash spans that contain the chunk, but if this is not possible, then other servers will be heuristically selected for

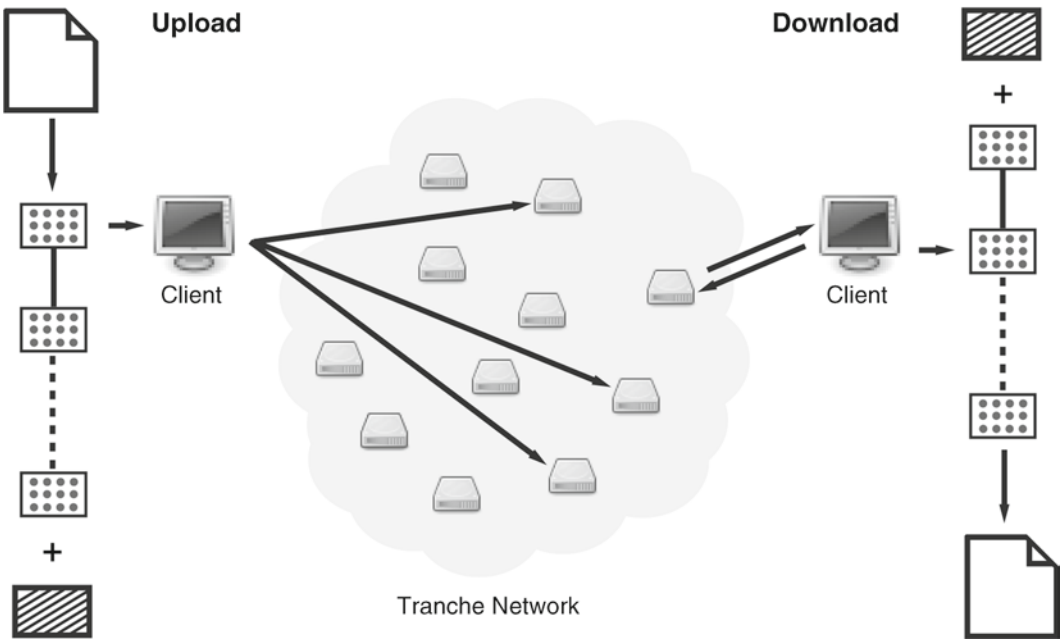


Fig. 3. When uploading (*left*), every chunk is stored on a minimum of three servers. When downloading (*right*), a client will download each chunk from the most appropriate server on the network and use all of the chunks to reassemble the file. The file exists only as chunks on the Tranche network.

time efficiency. To complete the upload process, the tool generates a ProjectFile, which describes the contents of the data set.

Earlier, we mentioned that each server has a list of trusted users. Tranche uses public-key cryptography (using X.509 public key infrastructure) so that every chunk that is uploaded to a server is authenticated. If a user is not recognized, meaning their certificate is not signed by one of the appropriate Tranche certificates nor has insufficient permissions, the request to store the chunk will be denied. The business of certificates is handled by the client tool; when the user logs in, a certificate is downloaded from our Web server. Note that while authentication is required for uploads, it is not required for downloads.

The download tool essentially works in reverse. It starts with the hash for a data set, which is used to retrieve the ProjectFile's metadata chunk. From this, the ProjectFile can be downloaded and reassembled, which will provide a list of all the metadata chunks and the relative path for each file. Each metadata chunk, in turn, provides enough information to download and reassemble the individual files. When downloading any given chunk, the tool will simultaneously query the entire network with requests to determine which servers have the chunk. The first positive response will be used to retrieve the chunk, and all other requests will be canceled. While quite verbose, this has been experimentally determined to be the fastest method given our current average network load. Even considering that the client tools are highly parallelized, meaning at any given moment there are many requests for each client, these requests are quite small – and as described previously, searching for chunks on a server is quite fast. However, we discuss a better solution when we discuss the future developments planned for Tranche.

Tranche offers several features that are useful for researchers:

- *Prepublication encryption*: data sets can be optionally AES encrypted, and require a passphrase to download and decrypt. Upon publication, a user may "release" a data set, which means it will become publicly available for download without a passphrase.
- *Data pedigree*: data sets are signed so that the individual who uploaded will always be known.
- *Data integrity*: since a hash can always be recalculated, data integrity can be verified at any time.
- *Immutability and versioning*: since Tranche uses hashes to determine data integrity, a data set may not be changed (*immutable*). However, new versions of the data set can be uploaded and linked with the previous data set version.

Three main interfaces are available for Tranche: the graphical interface, command-line tools, and the Java API. Investigators

interested in downloading data sets can use these immediately; however, if you wish to upload data, you must register (https://proteomecommons.org/signup.jsp). (Applications might take several days to process.)

The GUI is of most interest to casual users. The GUI is launched using Java Web Start, which means that anyone with Java 5 (or greater) can use the tool simply by clicking on a link (https://proteomecommons.org/tranche/). No installation is required.

Once the user interface is loaded, it will begin loading information about available servers and data sets from the network. The full process may require several minutes for completion; you do not need to wait for the process to complete, but the process will consume a significant amount of your processor's time.

Figure 4 highlights four areas of the user interface. More detailed guides are available online (https://trancheproject.org/users/). However, this figure covers the majority of tasks users perform:

1. *Log in*: Use your ProteomeCommons.org username and passphrase. Once logged in, the tool will handle all authentications for you, including your public/private key management.
2. *Upload project*: A wizard will launch to walk you through the process of uploading a data set. You will be asked if you would like to encrypt the project, which legal license or waiver you would like to use (with the option to provide your own custom license), as well as several more advanced (and less frequently used) options.

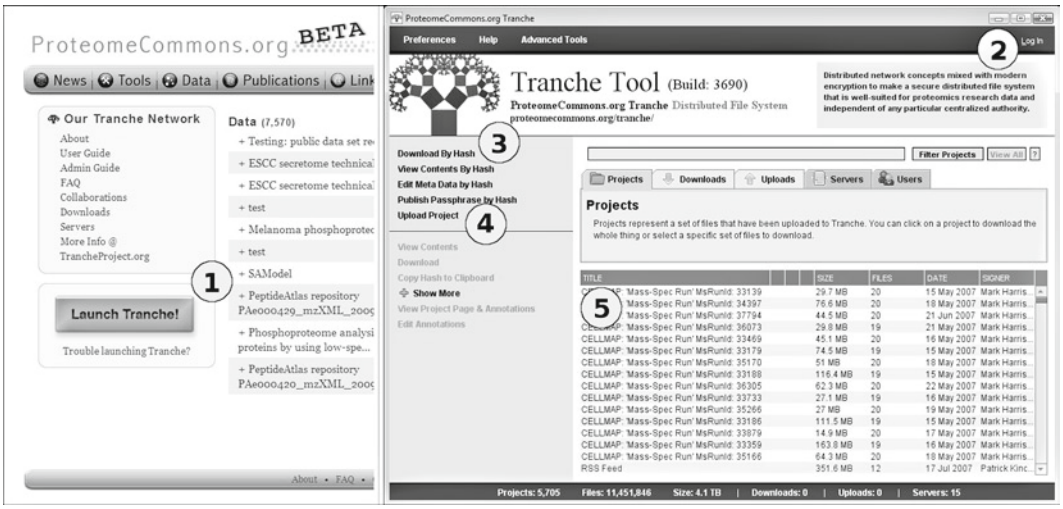


Fig. 4. To launch the ProteomeCommons.org Tranche graphical interface, visit https://tranche.proteomecommons.org and click "Launch Tranche" (1). After around 1 min, when the tool is loaded, click "Log In" (2) to enter your username and passphrase, upon which you can upload (4). You can download anything without an account or logging in. If you have the hash, click "Download by Hash" (3); if you do not have a hash, you can browse and download the available data sets (5).

3. *Download by hash*: Downloads may be accomplished in several ways, but if you have a hash for a data set, then you can quickly download the project. This will launch a wizard to walk you through the download process, similar to the upload tool.
4. *Projects*: If you do not have a hash or do not know which data sets you want, you can browse the list of data sets. By selecting the project you want from the list, you can download the data set, view its contents, as well as access more advanced options.

(Note that ProteomeCommons.org also offers stand-alone versions of both the upload and download tools, as well as more advanced search and browse tools. The upload tool is offered on the member page when a user logs in to ProteomeCommons.org, and the download tool is launched when a user attempts to download a data set from the Web site.)

The command-line tools are useful in several circumstances, e.g., when working remotely over SSH or in a "headless" environment (no windowing environment), or when there is a good deal of work to perform and you wish to automate some of the tasks. The upload (https://www.proteomecommons.org/tranche/files/CommandLineAddFileTool.zip) and download (https://www.proteomecommons.org/tranche/files/CommandLineGetFileTool.zip) tools are easy to use, but are heavily parameterized so that users will likely want to start by simply noting the required parameters. The first thing you will want to do is view the usage. For the upload tool:

```
java -jar -Xmx512m Tranche-Uploader.jar
–help
```

Similarly, for the download tool:

```
java -jar -Xmx512m Tranche-Downloader.jar
–help
```

Note that the command sets 512 MB available for heap space. You may allocate more memory if it is available, though this should be sufficient. Both the upload and download tools provide some advanced parameters which could potentially increase the speed (such as increasing the number of threads available for certain tasks, which increases the amount of work that is done in parallel). If these parameters are changed, it may be necessary to increase available memory. In general, advanced parameters should only be used when instructed by a Tranche developer to troubleshoot an issue.

Lastly, the API allows the integration of Tranche into software or scripts, so long as Java is used or the appropriate language bindings have been established. Though the API is quite extensive, performing an upload or a download is simple. Figure 5 contains code for the simplest use cases.

```
public void downloadDirectory(
        File downloadToDir,
        BigHash dataSetToDownload,
        String passphrase) throws Exception {

    // Load configuration for ProteomeCommons.org Tranche network and wait for servers
    ProteomeCommonsTrancheConfig.load();
    ServerUtil.waitForStartup();

    GetFileTool gft = new GetFileTool();
    gft.setHash(dataSetToDownload);

    // Only provide a passphrase if data set is encrypted
    if (passphrase != null) {
        gft.setPassphrase(passphrase);
    }

    gft.getDirectory(downloadToDir);
}
public BigHash uploadDirectory(
        String userName,
        String userPassphrase,
        File dirToUpload,
        String dataSetTitle,
        String dataSetDescription,
        String dataSetPassphrase) throws Exception {

    // Load configuration for ProteomeCommons.org Tranche network and wait for servers
    ProteomeCommonsTrancheConfig.load();
    ServerUtil.waitForStartup();

    // By logging in, you get a user zip file with certificate/private key
    UserZipFile uzf = UserZipFileUtil.getUserZipFile(userName, userPassphrase);

    AddFileTool aft = new AddFileTool(uzf.getCertificate(), uzf.getPrivateKey());

    aft.setTitle(dataSetTitle);
    aft.setDescription(dataSetDescription);

    // Only provide a passphrase if want data set to be encrypted
    if (dataSetPassphrase != null) {
        aft.setPassphrase(dataSetPassphrase);
    }

    BigHash hash = aft.addFile(dirToUpload);
    return hash;
}
```

Fig. 5. Demonstration of the Tranche Java API. In both these examples, we load the ProteomeCommons.org Tranche network and ensure that the servers are identified and available before proceeding. The download method (*top*) will download a data set to a particular directory. The upload method (*bottom*) will upload a data set and return the hash. Both demonstrate how optional encryption works; if an encrypted data set is downloaded but no passphrase is set, the download will fail. Note that the API is subject to change in future versions of Tranche.

To perform uploads using the API, there must be an account registered with ProteomeCommons.org. All data will be signed by the associated user, so some care should be taken to select the appropriate user name, particularly if it represents a group or organization.

Tranche is used by ProteomeCommons.org to host data sets; in fact, it might be useful to think of ProteomeCommons.org as

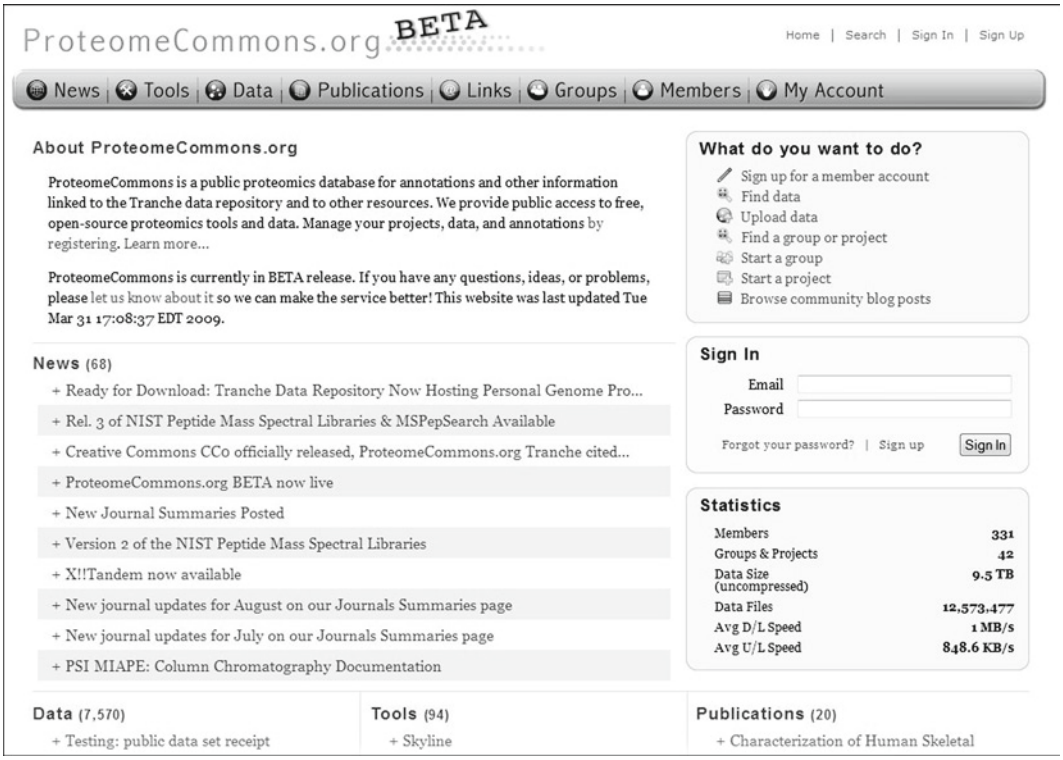


Fig. 6. The ProteomeCommons.org home page.

a layer of functionality that is built on top of Tranche. Many operations, such as deleting a data set and publishing a passphrase (which will allow an encrypted data set to be automatically decrypted for all users), can only be performed from ProteomeCommons.org, since Tranche users will not have sufficient permissions to do this from other client tools. ProteomeCommons.org also provides additional functionality that goes far beyond what Tranche alone can offer, including project management and annotations, as we discuss shortly.

ProteomeCommons.org is an online community, offering public access to user-contributed news, publications, and software. Data sets are automatically added following an upload to the ProteomeCommons.org Tranche network. Figure 6 features a screenshot of the ProteomeCommons.org home page.

Registered users can form groups and projects. Project management through ProteomeCommons.org allows members to share news, publications, tools, messages, and data sets with privacy restrictions. Any project or group can be public or private, and private groups can be hidden from anyone who is not a member. Individual members of groups have finely-defined permissions so that groups can establish rules for the management of all its resources. Subgroups and projects can be added to groups, offering additional control.

A particularly useful feature of groups is the management of annotation duties. Annotations are information that are associated with a data set describing how that data set was produced, processed, and interpreted. The user selects an *annotation standard*, which is a set of categories containing requested fields. Generally, an annotation standard is defined by a standards body, like the previously mentioned MIAPE standard. Every available standard is versioned in the event of new releases. After selecting a standard, a user can edit the annotation set from the Proteomecommons.org annotation editor, as shown in Fig. 7. Progress summaries are shown for individual categories as well as for the entire annotation set.

Administrators of groups and projects can assign duties to members based on individual annotation categories allowing domain experts on the projects to be assigned responsibility for each category. This feature is optional and intended to promote annotation accuracy and completeness. Although domain experts may be assigned responsibility for each annotation category, any member with sufficient privileges can edit an annotation category field regardless of whether it is assigned to anyone or who it is assigned to. When a group member is annotating a particular

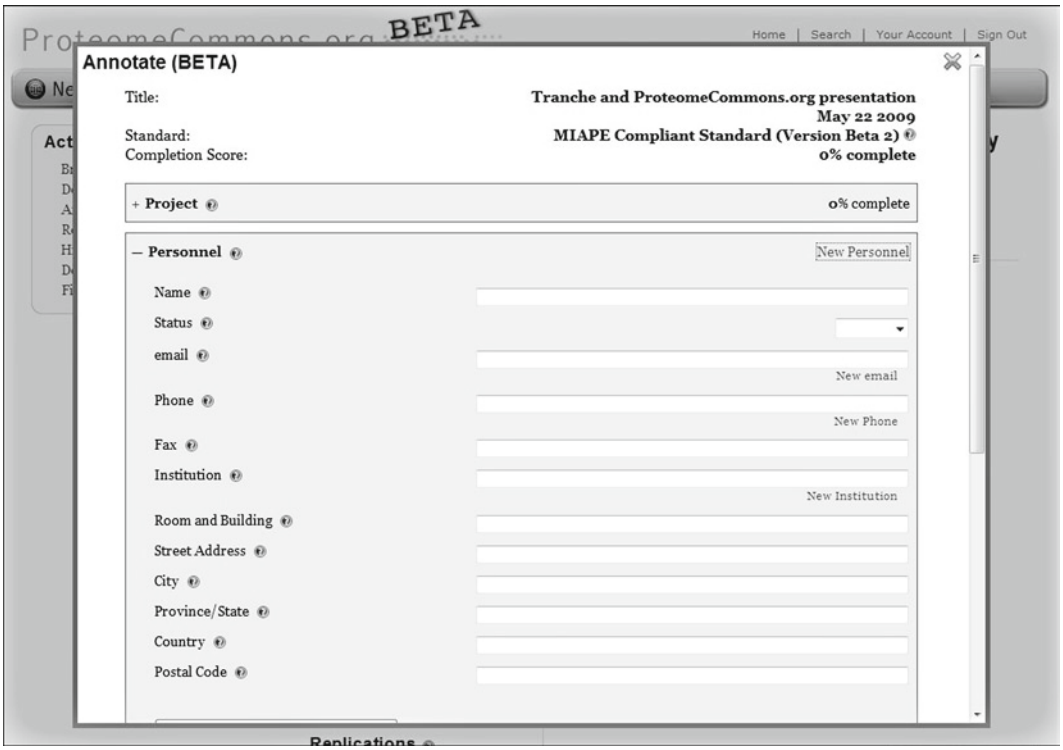


Fig. 7. The annotation editor. When a user uploads a data set, then she can annotate it after logging in to ProteomeCommons.org. If a data set is added to a group or project, a data set can be annotated by any group member with sufficient privileges.

annotation category, that category is locked to prevent concurrent modifications.

A data set is more findable in ProteomeCommons.org if it is accurately and completely annotated. Complete annotation information ranging from the nature of the biological sample to the mass spectrometer instrumentation that appears on the data page is generated by ProteomeCommons.org. This means the data set can be found more easily from an external search engine, such as Yahoo or Google, which index data set pages. Furthermore, anyone can search for data sets matching criteria, and only data sets with the appropriate annotations will be listed. Other nonbrowser interfaces can offer sophisticated semantic searches, which permits more useful bioinformatics applications. For example, the annotations manager was recently granted caBIG silver-level compliance (https://cabig.nci.nih.gov/), and users of caBIG will soon be able to search the annotations for data sets matching their criteria. There is also value in recording as much information about a data set as possible to provide the experimental context necessary for researchers to reinterpret the data sets and to complement the limited annotation often provided in publications describing the data sets. An archived data set missing basic information, such as sample preparation or analysis conditions, will have limited use.

To fully appreciate the value of annotations, it might be useful to contrast a semantic search versus a traditional keyword search. First, it is much easier to explore data sets when they are semantically defined. Though language is ambiguous, the use of ontologies not only offers controlled vocabularies, but also allows the definition of relations between data. For example, if an investigator wished to find all Tandem MS data sets involving yeast, a keyword search would only produce those that matched the descriptive text. The individual who performed the search could not be confident that everything was matched, and might perform additional searches to determine the relevant data sets. With ontologies, not only are the results unambiguous, but they can also be further divided by species of yeast or mass spectrometer manufacturer. Second, semantic searches offer the possibility of new functionality that is not possible with keyword searches. For example, assume that a data set has exactly 51,276 files. Human memory is rarely that specific, though it might remember that the data set had around 50,000 files. Semantically, it would make sense to search for data sets between 50,000 and 60,000 files. Combined with other information, such as the approximate date that the data set was produced, data sets become much more findable. At the present, ProteomeCommons.org search provides limited semantic search capabilities, particularly regarding data set size and dates of uploads; further functionality will depend on community standards and the availability of completed annotations,

but will continue to include keyword searches. caBIG searches of the ProteomeCommons.org annotations, however, will be entirely semantic.

The Tranche and ProteomeCommons.org development team is working with the broader proteomics community to adopt ontologies and controlled vocabularies. This is a long and difficult process, but is important for the development of reliable annotations, particularly considering the findability of data.

There are many more advanced features available with Tranche and ProteomeCommons.org, and these are documented on the Tranche Project Web site (www.trancheproject.org) and on ProteomeCommons.org.

3. Notes

The first version of ProteomeCommons.org was developed and released in 2004 as a Web resource and service. Tranche was developed the following year and released in 2006 to address the specific needs of the proteomics community for storage and dissemination of data sets. Since 2006, the development of both Tranche and ProteomeCommons.org has been based on the feedback from individual users, journals, and funding agencies as well as the anticipated needs of the community.

With over 2 years of operational experience and hundreds of users, several major changes have been made to Tranche to increase system reliability and robustness. We modified the functionality of Tranche to accommodate unforeseen events, including a broad range of hardware failures. Much of our error detection, such as detecting and repairing corrupted data files and chunks helped mitigate these problems. Problems that arose during production took longer to solve than it took to initially develop and release Tranche and involved extensive testing and development. System maintenance is necessarily a major effort for dissemination and archiving of data sets in a production environment.

We began redesigning ProteomeCommons.org in 2008, and the new version was released in February of 2009. Many features remain to be added in response to user input. As indicated in the methods section, the annotation standards, ontologies, and controlled vocabularies are still being defined, though two releases of the MIAPE Mass Spectrometry standard are already available.

We are currently working on the second version of the Tranche Distributed Repository, which will address many challenges related to scalability, performance, and security. One of the most significant developments will be the network model, especially the two specialized roles of Tranche servers: routers and

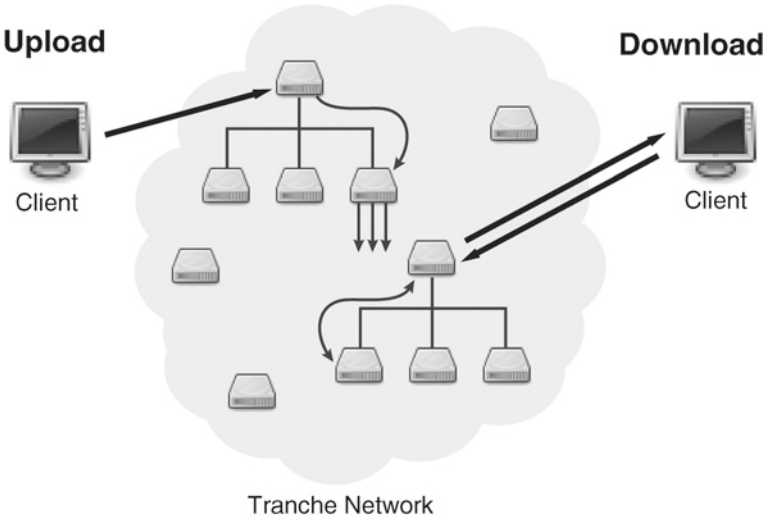


Fig. 8. Diagram of upcoming changes to Tranche network. Data servers can be connected to a routing server or can have an unmediated connection to clients. When a client performs an upload, each chunk will be uploaded to a single server and replicated across the network to the appropriate servers. When downloading, a server will locate the chunk for the user and, if appropriate, add it to its data store for future users.

data servers. As shown in Fig. 8, a router will interface with any data servers to which it is connected, meaning the user will need to be connected to fewer servers. (A client may also make an unmediated connection to any data server, as is also shown in the diagram.) Furthermore, when uploading data, chunks are replicated by the servers, shifting responsibility from the client, further minimizing the number of required connections and improving the performance by simplifying the protocol. This will require less user bandwidth, which will improve the overall performance.

In this new model, servers will have write permissions to other servers. This trust offers a particularly beneficial feature: if a server does not have a chunk that a client is requesting, that server can download the chunk from another trusted server, and then store the chunk and return it to the user. Not only will this result in a higher hit rate for servers having a requested chunk, thus improving the overall performance of the network, but it will also help keep the network more fully replicated. This will be particularly valuable for the long-term availability of data sets.

Servers will store a change log of all activities that impacted their stored data. In the event that a server is temporarily offline, other servers will continue to record changes to the network. When the offline server starts up again and before it makes its data available to the network, it will request all logged activities from other trusted servers on the network. Note that chunks will

always be accepted from trusted servers; however, if other servers log deletions, the credentials of the users who requested the deletes will be provided so that the querying server can ascertain whether it should also delete based on its own managed list of trusted certificates. Not only does this help with the overall replication of data on the network, but this also prevents deleted data from being salvaged by offline servers. Following these and other planned features, the Tranche network will be able to scale into the foreseeable future to accommodate all reasonable growth of users and data sets.

Several future developments are planned for Proteome Commons.org. Since a good deal of this chapter was devoted to the ability of users to find data sets, we plan to add a simple HTTP "RESTful" interface for accessing resources on ProteomeCommons.org, which would permit users to develop more their own data mining applications using our resources. Also, we are aware of the effort that annotations require, and wish to lower the barrier to annotation as much as possible. We plan to add functionality to automatically read in as much information from data sets as possible and add the metadata parsed from the data files to the associated data sets. Additionally, we plan to provide export functionality to formats, such as mzML and mzIdentML, so that stored annotations can be exported in a useful way from ProteomeCommons.org. Using these new tools that read and write tandem mass spectrometry data, we provide more semantically useful information and statistics within ProteomeCommons.org and Tranche, as well as provide some limited file conversions. To increase the usefulness of data sets for the community, we continue to work with publishers to automatically link publications with corresponding data sets in Tranche.

The ProteomeCommons.org Tranche network with PRIDE, PeptideAtlas, and Peptidome are founders of the ProteomExchange consortium. ProteomExchange allows free exchange of metadata between data resources, provides a universal accession number, and links to the raw data sets deposited in Tranche. Thus, investigators have to submit data for a study only once, and it is available in all participating repositories and databases (31). This collaboration is particularly beneficial for users since individual repositories may accommodate different types of data, based on their intended purpose. The ProteomExchange allows other resources to use Tranche to support their work, and in exchange, Tranche gains highly structured interfaces to its data.

Tranche and ProteomeCommon.org provide support for and maintain collaborations with a number of research entities, including Clinical Proteomic Technology Assessment for Cancer (CPTAC), the National Cancer Institute (NCI) Mouse Proteomics Technologies Initiative (MPTI), as well as with the PRIDE and PeptideAtlas repositories. We also have collaborated with Science Commons during the development and adoption of CC0, and

have a current collaboration with the Personal Genome Project (PGP). The Tranche Project is responsive to the needs of the community and new collaborations are welcome.

Acknowledgments

Special thanks to Jayson Falkner, who led the initial development for both Tranche and ProteomeCommons.org. The authors would also like to thank Peter Ulintz, Jared Falkner, Brian Maso, and Panagiotis Papoulias for their contributions. The Proteome Commons.org and Tranche Repository community resources are primarily sponsored by NCRR grant #P41-RR018627 and the NCI CPTC subcontract #27XS115. We also thank all the users of Tranche who have provided invaluable feedback and suggestions for the Tranche Project and ProteomeCommons.org.

References

1. Falkner JA, Ulintz PJ, Andrews PC (2006) A code and data archival and dissemination tool for the proteomics community. Am Biotechnol Lab 38:28–30
2. Toronto International Data Release Workshop Authors (2009) Prepublication data sharing. Nature 461:168–170
3. Schofield PN, Bubela T, Weaver T, Portilla L et al (2009) Post-publication sharing of data and tools. Nature 461:171–173
4. Editorial (2009) Data's shameful neglect. Nature 461:145
5. Salo D (2008) Innkeeper at the roach motel. Libr Trends 57:98–123
6. Heidorn PB (2008) Shedding light on the dark data in the long tail of science. Libr Trends 57:280–299
7. Wiley S (2009) Why don't we share data? The Scientist 23:33
8. Deutsch EW, Lam H, Aebersold R (2008) PeptideAtlas: a resource for target selection for emerging targeted proteomics workflows. EMBO Rep 9:429–434
9. Craig R, Cortens JP, Beavis RC (2004) An open source system for analyzing, validating and storing protein identification data. Proteome Res 3:1234–1242
10. Martens L, Hermjakob H, Jones P, Taylor C et al (2005) The PRoteomics IDEntification database. Proteomics 5:3537–3545
11. Prasad TS, Goel R, Kandasamy K, Keerthikumar S et al (2009) Human Protein Reference Database – 2009 update. Nucleic Acids Res 37:D767–D772
12. Slotta DJ, Barrett T, Edgar R (2009) NCBI Peptidome: a new public repository for mass spectrometry peptide identifications. Nat Biotechnol 27:600–601
13. (2007) Publication guidelines for the analysis and documentation of peptide and protein identifications. Mol Cell Proteomics (http://www.mcponline.org/misc/ParisReport_Final.dtl) accessed on July 13 2009.
14. Editorial (2007) Democratizing proteomics data. Nat Biotechnol 25:262
15. (2008) Instructions to authors. Proteomics (http://www3.interscience.wiley.com/cgi-bin/jabout/76510741/2120_instruc.pdf) accessed on July 13 2009.
16. (2003) Final NIH statement on sharing research data. (http://grants.nih.gov/grants/guide/notice-files/NOT-OD-03-032.html) accessed on July 13 2009
17. Howe D, Costanzo M, Fey P, Gojobori T et al (2008) The future of biocuration. Nature 455:47–50
18. Martin DB, Nelson PS (2001) From genomics to proteomics: techniques and applications in cancer research. Trends Cell Biol 11:61–65

19. Tyshenko MG (2005) Current trends in publicly available genetic databases. Health Inform J 11:295–308
20. (2009) About CC0--"No Rights Reserved". (http://creativecommons.org/about/cc0) accessed on July 13 2009
21. Prince JT, Carlson MW, Wang R, Lu P, Marcotte EM (2004) The need for a public proteomics repository. Nat Biotechnol 22: 471–472
22. Why tumor samples are so important for research. (http://www.pediatricgist.cancer.gov/Source/Research/ResearchArticles/TumorSampleImpArticle.aspx)
23. Schweitzer MH, Suo Z, Avci R, Asara JM et al (2007) Analyses of soft tissue from Tyrannosaurus rex suggest the presence of protein. Science 316:277–280
24. Schweitzer MH, Zheng W, Organ CL, Avci R et al (2009) Biomolecular characterization and protein sequences of the Campanian hadrosaur B. canadensis. Science 324:626–631
25. Taylor CF, Paton NW, Lilley KS, Binz P et al (2007) The minimum information about a proteomics experiment (MIAPE). Nat Biotechnol 25:887–893
26. Pedrioli PGA, Eng JK, Hubley R, Vogelzang M et al (2004) A common open representation of mass spectrometry data and its application to proteomics research. Nat Biotechnol 22:1459–1466
27. Hamacher M, Stephan C, Meyer HE, Eisenacher M (2009) Data handling and processing in proteomics. Expert Rev Proteomics 6, 217–219. (2006) The mzData Standard. (http://www.psidev.info/index.php?q=node/80#mzdata)
28. Orchard S, Taylor C, Hermjakob H, Zhu W et al (2004) Current status of proteomic standards development. Expert Rev Proteomics 1:179–183
29. Deutsch E (2008) mzML: a single, unifying data format for mass spectrometer output. Proteomics 8:2776–2777
30. Bayer R (1971) Binary B-trees for virtual memory. ACM-SIGFIDET Workshop 1971:219–235
31. Martens L, Deutsch E, Hermjakob H, Omenn G (2009) Proteomics data submission strategy for ProteomeExchange.(http://proteomexchange.org/doc/ProteomExchange_data_submission_strategy_final.pdf)

Part III

Standards

Chapter 9

Data Standardization by the HUPO-PSI: How has the Community Benefitted?

Sandra Orchard and Henning Hermjakob

Abstract

The groundwork allowing the systematic capture of proteomics data has now largely been completed, with the design and publication of exchange formats and interchange standards by the Human Proteome Organisation Proteomics Standards Initiative (HUPO-PSI). Our focus can now shift to gathering the ever-increasing amounts of generated data, and finding novel ways to catalog and present it so that a deeper understanding of basic science, health, and disease can be gained by scientists mining these increasingly rich resources.

1. Introduction

The Human Proteome Organisation Proteomics Standards Initiative (HUPO-PSI) has worked since 2002 to develop data standards that enable the collection, storage, and dissemination of proteomics data. Prior to the implementation of such standards, a typical proteomic workflow generated results in a variety of alternative formats, which were dependent on the make and model of instrumentation involved. Comparison at the raw data level, for example of the generated spectra, was impossible if different instrumentation or software had been used, even within the same laboratory examining identical samples. This proved increasingly problematic, as the generation of increasingly large datasets by collaborating laboratories, for example, those involved in the various HUPO tissue initiatives (1, 2), required the ability to collect, collate, and compare data from participating groups with no restriction on the instrumentation used. The main impetus driving these collaborative projects was a desire to leave a legacy of data for subsequent groups, for example, comparative

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_9,

datasets from healthy tissue, against which samples from diseased patients, closely related species, or other healthy humans could be compared. In order to do this, an appropriate data repository, PRIDE (PRoteomics IDEntifications database, (http://www.ebi.ac.uk/pride (3))), was developed, which required the data from many laboratories to be submitted in a common format.

In parallel to the work of the mass spectrometry proteomics groups, the major interaction databases also realized that common data standards were required if they were to meet the needs of their user community. Interactome and network biologists increasingly wished to download and combine the information stored in multiple data resources, all of which operated using their own database layouts and formats. Again, the need for a common format to enable this was identified, requiring the input and cooperation of many groups to realize this need.

From the work undertaken to fulfill these comparatively limited use cases, a set of data standards have been developed which has revolutionized the manner in which the proteomic community can tailor their data collection, compare test data against that held in a number of data resources. Increasingly, the designers and writers of software and analytical tools required to visualize and analyze data that are utilizing the HUPO-PSI data formats, resulting in an ever-growing number of products that can be used to analyze the contents of an increasing number of database resources.

The following sections describe how a community standard is built, and more specifically the design and content of a number of standards which relate to proteomics data. Finally, the relationship between the use of these standards and the data publication process is discussed.

2. What Constitutes a Community Standard?

A standards document, as defined by the HUPO-PSI, actually consists of four separate, but interrelated aspects.

1. A user requirements document – a clearly identified and broadly represented user-community is consulted to decide the needs and wishes of the workers in that field.
2. A Minimum Information About a Proteomics Experiment (MIAPE) document – a checklist of required elements which should be included in every publication to assist the reader in understanding the data presented within it. These have been prepared for each domain within the field of proteomics, firstly by invited experts, followed by an open community review process.

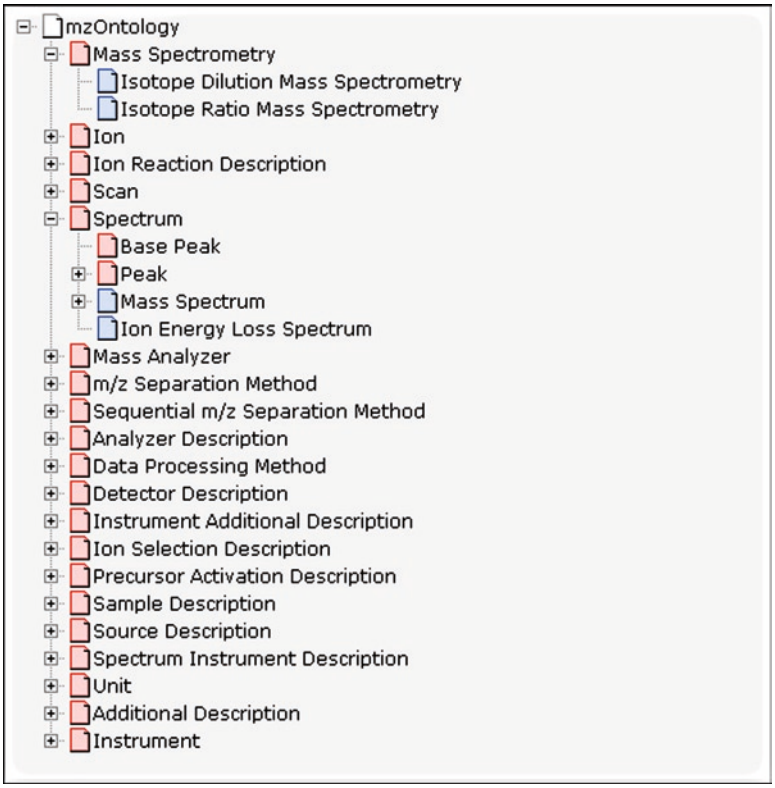


Fig. 1. Controlled vocabularies have been developed to annotate the various aspects of a proteomics experiment.

3. An XML interchange format to allow the transfer of data from one resource to another. This must be capable of holding all the information required by the MIAPE document in appropriate fields.
4. A controlled vocabulary (CV) to enable the standardized annotation of the information exchanged by the XML format (Fig. 1).

All of the above have to be made publicly available for widespread feedback from all potential interest parties – data producers, instrument manufacturers, software and tool developers, and data users – during their development process, to be judged a true community standard. The initial MIAPE parent document (4) was made available for several rounds of input and comment. This included the exposure on both the HUPO-PSI (www.psidev.info) and Nature Biotechnology Web sites, before final journal publication, and all domain specific documents also go through a similar process (5). Additionally, all MIAPE documents are registered with the MIBBI portal (6) and all CVs with the OBO Foundry (7). This ensures that these documents are not only aligned with the

needs of the proteomics community, but also available to a much broader group of biologists who may subsequently find these appropriate for their workflows and data predication pipelines.

3. Mass Spectrometry and Data Standardization

In 2004, the mzData interchange format was published. This allowed the storage of proteomic-related mass spectral data, ranging from basic details about the sample, instrument details, and data processing steps, through to the actual spectral lists of mass-to-charge values and intensities, used base64 encoding to represent the floating point mass-to-charge (m/z) and ion intensity. The format was implemented by a number of manufacturers and open source applications were made available, such as an Eclipse-based mzData editor with validation provided by the Leibniz Institute of Plant Biochemistry, Bioinformatics and Mass Spec Research Group. Additionally, the PRIDE data repository became the first PSI-standards compliant relational database to be implemented (3). However, also in 2004, a second open, generic XML representation of mass spectrometry data was published by the Institute of Systems Biology, mzXML (8). While this was originally designed to be work-flow specific, other workers began to find wider uses for the schema with the result that manufacturers were faced with the prospect of having to implement two separate open-source formats.

In 2006, the two groups decided to merge the two formats into a single, and much improved, XML schema by merging the best aspects of both mzData and mzXML, and addressing the unmet needs of both, which had been identified by their respective user groups. By 2008, this work had matured into the current mzML (9) format, a final stable format (1.1.0) of which was released in 2009. Software incorporating the revised format has been released or is soon to be released by Applied Biosystems, GeneBio, Insilicos, Matrix Science, and Thermo Electron Corporation with other manufacturers in various stages of development. A wide range of open source software has also been adapted or implemented utilizing mzML (for full list see www.psidev.info). A number of converters have also been made available to produce data in this format, such as ReAdW for XCalibur (Thermo) .raw files, wolf for MassLynx (Waters) .raw directories, mzWiff for Analyst (ABI, Agilent) .wiff files, and Trapper for MassHunter (Agilent) .d directories (www.psidev.info). Semantic validators are also available to check completed files (9, 29). Repositories such as PRIDE and the ISB Peptide Atlas, a publicly accessible compendium of peptides identified in a large set of tandem mass spectrometry proteomics experiments, are committed

to upgrading their schema to mzML and converting existing data to this format.

Once standards are in place, the availability of user-friendly software to enable their adoption and use can make the difference between community acceptance and rejection. A European Commission-funded project, ProDaC (www.fp6-ProDaC.eu), lead to the development of a raft of such tools for mzData, initially aimed at easing the flow of data into the standards-compliant repository, PRIDE (3). For example, PRIDE Wizard converts MASCOT files and associated spectra into PRIDE XML (www.mcisb.org/resources/PrideWizard/), allowing direct submission to the repository, while PRIDE Converter creates PRIDE XML files from a range of formats (Fig. 2) (10). These tools will be updated when appropriate, to reflect the adoption of mzML by PRIDE and similar repositories.

Databases utilizing the same interchange format can readily exchange data between themselves, and a collaborative network of mass spectrometry repositories is currently being planned. Meta-data describing each experiment submitted to any one participating database will be accessible for all participants; however, due to the file size, raw data will remain at the original host and accessed from there. The integration of protein identification data

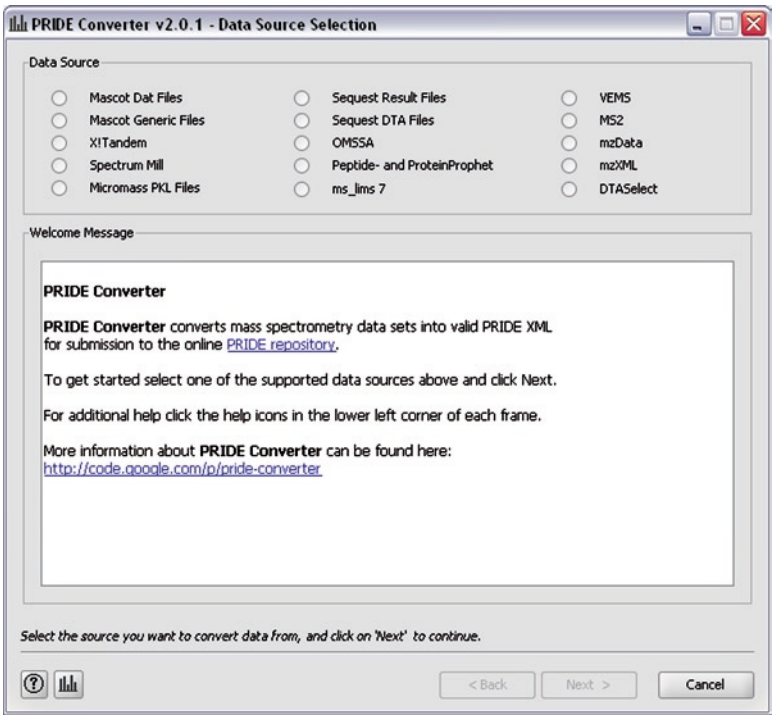


Fig. 2. PRIDE converter.

will enable more comprehensive answers to questions posed by users to be assembled.

The PSI-MS group has now begun developing a draft specification for a standardized format for the exchange and transmission of transition lists (TraML) for selected reaction monitoring (SRM) experiments. A reference implementation is planned for 2009/2010.

The MIAPE-MS document was published in 2007 and is now been achieving community acceptance. To ease the preparation of MIAPE-compliant report generation, the ProteoRed group (www.proteored.org) have implemented a tool to assist in their production. The MIAPE generator tools can also serve as a MIAPE repository and allow immediate comparison of MIAPE documents that have been made publicly available to the community.

4. Proteomics Informatics

The HUPO-PSI Proteomics Informatics group exist to provide a set of minimum reporting requirements that augment the MIAPE reporting guidelines with respect to the analysis of data derived from proteomics experiments and to provide vendor-neutral, standard formats for representing the results of analyzing and processing experimental data. MIAPE-Mass Spectrometry Informatics (MIAPE-MSI) was published in 2008 (11) and the interchange format for protein identification data, mzIdentML, in 2009 (www.psidev.info/index.php?q=node/319). The associated resources now include semantic validation tools, a specification document, tables of conformance to both the MIAPE and MCP guidelines and a number of example instance documents. Most of the major search engines have already implemented support for mzIdentML or are expected to do by mid-2010 and a number of repositories, such as PRIDE, have also committed to supporting this format.

Initially, it was intended that the format would also be capable of transferring both relative and absolute quantitation data at both the protein and peptide level, but this proved a complex task and delayed the release of the format. It has now been decided that this will be separately catered for with its own schema, mzQuantML, with a structure broadly similar to mzIdentML. An alpha version will be produced in 2010 with examples for two quantitation techniques: experts in particular techniques/software packages will then be invited to extend the format.

This group has also worked on the development of PEFF, a common sequence database format designed to overcome problems with the interpretation of current fasta formats by search

engines, with protein identifiers, descriptions, taxonomy, and other annotations, such as PTMs and sequence variants as well as database descriptors and version numbers contained in the file. It has been decided to stay with the relatively straightforward fasta-like format, with a possible move to an XML format later if this becomes a necessity. The format will break current parsers, this is due to constraints which enable it to remain compatible with other PSI formats. Controlled vocabulary terms have been developed and will be added to the PSI-MS CV. Several protein sequence databases, such as UniProtKB (12) and the International Protein Index (13) have committed to supporting this format. Converters for those sequence databases not willing to accept the new format will be made available and maintained on ProteomeCommons.org.

5. Protein Separations

The MIAPE Gel Electrophoresis (MIAPE-GE) guidelines specifying the minimum information that should be provided when reporting the use of *n*-dimensional gel electrophoresis in a proteomics experiment were published in 2008 (14), and MIAPEGelDB has been developed as a public repository and a Web-based data entry tool for documents (15). This guides authors through the publication of the minimal set of information for their proteomics experiments using a simple interface. Similarly, MIAPE-GE documents may be stored in the ProteoRed database with links to corresponding mass spectrometry protein identifications in PRIDE. An interchange format has also been developed, GelML, for describing the results of gel electrophoresis experiments.

Draft MIAPE documentation also exists for separation by column chromatography and capillary electrophoresis, with a consensus interchange format to facilitate the development of effective search and analysis tools.

6. Molecular Interactions

The molecular interaction HUPO-PSI workgroup provides an example of the enormous community benefits that can accrue from the development of common standards and formats. Prior to 2004, there were several protein interaction databases in existence but all used their own database model and data formats, preventing merging or the different datasets held in each. The first PSI-MI XML interchange format (version 1.0) was released

in 2004, enabling the description of protein–protein interaction data with a limited amount of accompanying annotation (16). The format was immediately adopted by all the major database resources with the immediate benefit to the user that separate resources could be searched and the results readily combined. It soon became apparent, however, that the initial format was too restricted and was rapidly expanded to allow interactions between all molecule types to be described with full and detailed annotation. Version XML2.5 was released in 2007 and has remained stable for several years (17). A controlled vocabulary (MI) which describes all aspects of a molecular interaction experiment (MIMIx) has been developed and is in active community usage.

To accommodate those users who wished to download interaction data in a simpler format, a common tab-lineated file scheme was agreed between the participating databases (MITAB2.5). This allows a restricted amount of information to be exchanged and loaded into applications, such as a spreadsheet. Again, usage has proved this format to be both popular and too restrictive so an enhanced version (MITAB2.6) was agreed at a recent workshop.

This global adoption of a standard has proved to be of enormous benefit to the interaction community. With a stable, common format to work with, tool development has been both rapid and largely open-source. Visualization software, such as Cytoscape (18), now uses the PSI-MI format to import network information into the viewer, and it is currently possible to query selected resources, such as the IntAct molecular interaction, from within Cytoscape to further extend an existing network. The recent development of the PSICQUIC – PSI Common QUery InterfaCe, a SOAP or REST-based Web service (http://groups.google.com/group/psicquic), will allow the accessing of all participating databases from a resource, such as Cytoscape with a single query. RpsiXML, a Bioconductor package, allows the conversion of PSI-MI XML2.5 files into R graph objects; the user can then use R methods to determine cohesive subgraphs, compute summary statistics, fit mathematical models to the data or render graphical layouts (19).

The domain-specific MIAPE document detailing the minimum information required to describe an MIMIx was published in 2007 (20) and was prepared by a mixed community of data produces, data providers, and data users. During this process, the discussions lead to the formation of the international molecular exchange (IMEx) consortium, dedicated to the eventual sharing of all curated molecular interaction data such that the users need only to visit a single repository to access all available information (http://www.imexconsortium.org, (21)). As part of this process, common curation standards have been developed by the participating databases (IntAct (22), DIP (23), MINT (24), MatrixDB (25), MPIDB (26), MPact (27)), and an interchange mechanism is now in place, funded by a recent EC grant.

7. MIAPE and the Proteomics Journals

The most effective way of collecting data, is by direct submission by authors as an integral part of the publication process. To this end, a series of meetings (28) and consultations have been held with domain specific editors during both the preparation and implementation phases of standard development. The long-term goal is that appropriate journals will adopt these guidelines as a part of their instructions to authors, although journals may choose to supplement these with their own guidelines, particularly with regard to data quality. These discussions are active, and ongoing, and the journal editors have made valuable contributions to the final content of these documents and formats (28).

8. Summary

The initial work of writing and finalizing a range of standards, exchange formats and CVs for the collation, annotation, and exchange of proteomics data is largely complete, although work will remain ongoing to respond to new technologies and techniques. The initial benefits are already being realized, with the establishment of repositories to hold this information and an increasing range of analytical tools to evaluate the sets of results. Increasing interoperability has lead to the success of collaboration, such as the HUPO tissues initiative, and has encouraged the development of many new tools – now written to query multiple resources rather than specifically designed to serve a single data source. The driving need now is to ensure that these standards are adopted by data producers and used to directly submit datasets to the data repositories, to ensure that valuable, and expensive to produce, information is not lost in supplementary materials and user-maintained Excel sheets. The development of improved submission tools to ease the data flow is seen as an essential part of the process, and is actively being addressed, as is the participation of the journals in this process, but it is ultimately the bench scientist who needs to commit to this process and add to the publicly available pool of data.

9. Notes

1. All controlled vocabularies registered at the Open Biological and Biomedical foundry (www.obofoundry.org/), including all of those produced by the HUPO-PSI, can be viewed using the

Ontology-Lookup Service (www.ebi.ac.uk/ontology-lookup/) which supplies a centralized query interface for ontology and controlled vocabulary lookup. The service also provides a Web service interface to query multiple ontologies from a single location with a unified output format.

2. All of the public domain databases described above actively encourage direct deposition of data, and this is now being encouraged or mandated by the domain journals. Data producers are encouraged to contact the databases early in the submission process as they employ dedicated curators who can assist in the data preparation and deposition process.
3. All work described in this article has been undertaken by members of the community contributing on a voluntary basis. If any reader should wish to contribute to these efforts, or simply stay abreast of developments, should visit http://psidev.sf.net/ to review our activities, join the discussion groups listed, and contribute to the further development of community standards for proteomics data.

References

1. Omenn GS, States DJ, Adamski M, Blackwell TW, Menon R, Hermjakob H, Apweiler R, Haab BB, Simpson RJ, Eddes JS, Kapp EA, Moritz RL, Chan DW, Rai AJ, Admon A, Aebersold R, Eng J, Hancock WS, Hefta SA, Meyer H, Paik Y-K, Yoo J-S, Ping P, Pounds J, Adkins J, Qian X, Wang R, Wasinger V, Wu CY, Zhao X, Zeng R, Archakov A, Tsugita A, Beer I, Pandey A, Pisano M, Andrews P, Tammen H, Speicher DW, Hanash SM (2005) Overview of the HUPO Plasma Proteome Project: results from the pilot phase with 35 collaborating laboratories and multiple analytical groups, generating a core dataset of 3020 proteins and a publicly-available database. Proteomics 5:3226–3245
2. Hamacher M, Apweiler R, Arnold G, Becker A, Bluggel M, Carrette O, Colvis C, Dunn MJ, Frohlich T, Fountoulakis M, van Hall A, Herberg F, Ji J, Kretzschmar H, Lewczuk P, Lubec G, Marcus K, Martens L, Palacios Bustamante N, Park YM, Pennington SR, Robben J, Stühler K, Reidegeld KA, Riederer P, Rossier J, Sanchez J-C, Schrader M, Stephan C, Tagle D, Thiele H, Wang J, Wiltfang J, Yoo JS, Zhang C, Klose J, Meyer HE (2006) HUPO Brain Proteome Project: summary of the pilot phase and introduction of a comprehensive data reprocessing strategy. Proteomics 6:4890–4898
3. Vizcaino JA, Cote R, Reisinger F, Foster JM, Mueller M, Rameseder J, Hermjakob H, Martens L (2009) A guide to the Proteomics Identifications Database proteomics data repository. Proteomics 9:4276–4283
4. Taylor CF, Paton NW, Lilley KS, Binz P-A, Julian RK Jr, Jones AR, Zhu W, Apweiler R, Aebersold R, Deutsch EW, Dunn MJ, Heck AJ, Leitner A, Macht M, Mann M, Martens L, Neubert TA, Patterson SD, Ping P, Seymour SL, Souda P, Tsugita A, Vandekerckhove J, Vondriska TM, Whitelegge JP, Wilkins MR, Xenarios I, Yates JR III, Hermjakob H (2007) The minimum information about a proteomics experiment (MIAPE). Nat Biotechnol 25:887–893
5. Vizcaino JA, Martens L, Hermjakob H, Julian RK, Paton NW (2007) The PSI formal document process and its implementation on the PSI website. Proteomics 7:2355–2357
6. Taylor CF, Field D, Sansone SA, Aerts J, Apweiler R, Ashburner M, Ball CA, Binz P-A, Bogue M, Booth T, Brazma A, Brinkman RR, Michael Clark A, Deutsch EW, Fiehn O, Fostel J, Ghazal P, Gibson F, Gray T, Grimes G, Hancock JM, Hardy NW, Hermjakob H, Julian RK, Kane M, Kettner C, Kinsinger C, Kolker E, Kuiper M, Le Novère N, Leebens-Mack J, Lewis SE, Lord P, Mallon A-M, Marthandan N, Masuya H, McNally R, Mehrle A, Morrison N, Orchard S, Quackenbush J, Reecy JM, Robertson DG, Rocca-Serra P, Rodriguez H, Rosenfelder H, Santoyo-Lopez J, Scheuermann RH, Schober

D, Smith B, Snape J, Stoeckert CJ, Tipton K, Sterk P, Untergasser A, Vandesompele J, Wiemann S (2008) Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. Nat Biotechnol 26:889–896

7. Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W, Goldberg LJ, Eilbeck K, Ireland A, Mungall CJ; OBI Consortium, Leontis N, Rocca-Serra P, Ruttenberg A, Sansone SA, Scheuermann RH, Shah N, Whetzel PL, Lewis S (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol 25:1251–1255
8. Pedrioli PG, Eng JK, Hubley R, Vogelzang M, Deutsch EW, Raught B, Pratt B, Nilsson E, Angeletti RH, Apweiler R, Cheung K, Costello CE, Hermjakob H, Huang S, Julian RK, Kapp E, McComb ME, Oliver SG, Omenn G, Paton NW, Simpson R, Smith R, Taylor CF, Zhu W, Aebersold R (2004) A common open representation of mass spectrometry data and its application to proteomics research. Nat Biotechnol 22:1459–1466
9. Orchard S, Deutsch EW, Binz PA, Jones AR, Creasy D, Montechi-Palazzi L, Corthals G, Hermjakob H (2009) Annual Spring Meeting of the Proteomics Standards Initiative, 27-29 April 2009, Turku, Finland. Proteomics 9:4429–4432
10. Barsnes H, Vizcaino JA, Eidhammer I, Martens L (2009) PRIDE Converter: making proteomics data-sharing easy. Nat Biotechnol 27:598–599
11. Binz P-A, Barkovich R, Beavis RC, Creasy D, Horn DM, Julian RK Jr, Seymour SL, Taylor CF, Vandenbrouck Y (2008) Guidelines for reporting the use of mass spectrometry informatics in proteomics. Nat Biotechnol 26:862
12. The UniProt Consortium (2009) The Universal Protein Resource (UniProt) 2009. Nucleic Acids Res 37:169–174
13. Kersey PJ, Duarte J, Williams A, Karavidopoulou Y, Birney E, Apweiler R (2004) The International Protein Index: an integrated database for proteomics experiments. Proteomics 4:1985–1988
14. Gibson F, Anderson L, Babnigg G, Baker M, Berth M, Binz P-A, Borthwick A, Cash P, Day BW, Friedman DB, Garland D, Gutstein B, Hoogland C, Jones NA, Khan A, Klose J, Lamond AI, Lemkin PF, Lilley KS, Minden J, Morris NJ, Paton NW, Pisano MR, Prime JE, Rabilloud T, Stead DA, Taylor CF, Voshol H, Wipat A, Jones AR (2008) Guidelines for reporting the use of gel electrophoresis in proteomics. Nat Biotechnol 26:863–864
15. Robin X, Hoogland C, Appel RD, Lisacek F (2009) MIAPEGelDB, a web-based submission tool and public repository for MIAPE gel electrophoresis documents. J Proteomics 71:249–251
16. Hermjakob H, Montecchi-Palazzi L, Bader G, Wojcik J, Salwinski L, Ceol A, Moore S, Orchard S, Sarkans U, von Mering C, Roechert B, Poux S, Jung E, Mersch H, Kersey P, Lappe M, Li Y, Zeng R, Rana D, Nikolski M, Husi H, Brun C, Shanker K, Grant SG, Sander C, Bork P, Zhu W, Pandey A, Brazma A, Jacq B, Vidal M, Sherman D, Legrain P, Cesareni G, Xenarios I, Eisenberg D, Steipe B, Hogue C, Apweiler R (2004) The HUPO PSI's molecular interaction format–a community standard for the representation of protein interaction data. Nat Biotechnol 22:177–183
17. Kerrien S, Orchard S, Montecchi-Palazzi L, Aranda B, Quinn AF, Vinod N, Bader G, Xenarios I, Wojcik J, Sherman D, Tyers M, Salama JJ, Moore S, Ceol A, Chatr-aryamontri A, Oesterheld M, Stümpflen V, Salwinski L, Nerothin J, Cerami E, Cusick ME, Vidal M, Gilson M, Armstrong J, Woollard P, Hogue C, Eisenberg D, Cesareni G, Apweiler R, Hermjakob H (2007) Broadening the horizon–level 2.5 of the HUPO-PSI format for molecular interactions. BMC Bioinform 5:44
18. Yeung N, Cline MS, Kuchinsky A, Smoot ME, Bader GD (2008) Exploring biological networks with Cytoscape software. Curr Protoc Bioinformatics 8 unit 8.13
19. Chiang T, Scholtens D (2009) A general pipeline for quality and statistical assessment of protein interaction data using R and Bioconductor. Nat Protoc 4:535–546
20. Orchard S, Salwinski L, Kerrien S, Montecchi-Palazzi L, Oesterheld M, Stümpflen V, Ceol A, Chatr-aryamontri A, Armstrong J, Woollard P, Salama JJ, Moore S, Wojcik J, Bader GD, Vidal M, Gerstein M, Gavin A-C, Superti-Furga G, Greenblatt J, Bader J, Uetz P, Tyers M, Legrain P, Fields S, The GO Consortium, Gilson M, Hogue C, Mewes H-W, Apweiler R, Xenarios I, Eisenberg D, Cesareni G, Hermjakob H (2007) The minimum information required for reporting a molecular interaction experiment (MIMIx). Nat Biotechnol 25:894–898
21. Orchard S, Kerrien S, Jones P, Ceol A, Chatr-aryamontri A, Salwinski L, Nerothin J, Hermjakob H (2007) Submit your interaction data the IMEx way - a step by step guide to trouble-free deposition. Pract Proteomics 7:28–34

22. Aranda B, Achuthan P, Alam-Faruque Y, Armean I, Bridge A, Derow C, Feuermann M, Kerrien S, Khadake J, Kerssemakers J, Leroy C, Menden M, Michaut M, Montecchi-Palazzi L, Neuhauser SN, Orchard S, Perreau V, Roechert B, Tashakkori A, van Eijk K, Hermjakob H (2010) The IntAct Molecular Interaction Database in 2010. Nucleic Acid Res 38:525–531
23. Xenarios I, Salwinski L, Duan XJ, Higney P, Kim S, Eisenberg D (2002) DIP: The database of interacting proteins. A research tool for studying cellular networks of protein interactions. Nucleic Acids Res 30:303–305
24. Chatr-aryamontri A, Ceol A, Palazzi LM, Nardelli G, Schneider MV, Castagnoli L, Cesareni G (2007) MINT: the Molecular INTeraction database. Nucleic Acids Res 35:572–574
25. Chautard E, Ballut L, Thierry-Mieg N, Ricard-Blum S (2009) MatrixDB, a database focused on extracellular protein-protein and protein-carbohydrate interactions. Bioinformatics 25:690–691
26. Goll J, Rajagopala SV, Shiau SC, Wu H, Lamb BT, Uetz P (2008) MPIDB: the microbial protein interaction database. Bioinformatics 24:1743–1744
27. Guldener U, Munsterkotter M, Oesterheld M, Pagel P, Ruepp A, Mewes HW, Stumpflen V (2006) MPact: the MIPS protein interaction resource on yeast. Nucleic Acids Res 34:d436–d441
28. Orchard S (2009) Ending the "publish and vanish" culture: how the data standardization process will assist in data harvesting. J Proteome Res 8:3219
29. Montecchi-Palazzi L, Kerrien S, Reisinger F, Aranda B, Jones AR, Martens L, Hermjakob H (2009) The PSI semantic validator: a framework to check MIAPE compliance of proteomics data. Proteomics 9:5112–5119

Chapter 10

mzIdentML: An Open Community-Built Standard Format for the Results of Proteomics Spectrum Identification Algorithms

Martin Eisenacher

Abstract

To deal with the data flood of current mass spectrometry methods, standard data formats are needed. The Proteomics Standards Initiative (PSI) of the Human Proteome Organisation (HUPO) develops open storage and transfer standards for and with the community. The Proteomics Informatics work group of the PSI has recently released an XML-based format to store the parameters and results of spectrum identification algorithms (the so-called search engines), which identify peptides and/or proteins from mass spectra. Here, this format called "mzIdentML" is described by giving principle design concepts and presenting examples of important use cases.

1. Introduction

Proteomics research naturally creates large data sets of mass spectra. In the last years, the amount of data produced for one experiment increased dramatically. The need to establish an efficient data management is obvious. In parallel, it became apparent that data sets of one experiment alone may not lead to an overall understanding of the complex processes in a cell or during emergence of a disease. Therefore, the Proteomics community came to the conclusion – and step by step to the agreement – that data sets and results should be stored in central public repositories like those of the ProteomeExchange consortium (http://www.proteomexchange.org/index.php). Large consortia such as the ProDaC project (1) supported these efforts (http://www.fp6-prodac.eu). Efficient data management and storage in public databases have been complicated – if not prohibited – by heterogeneous data formats and different storage concepts of mass

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_10,

spectrometer and analysis software vendors. Meanwhile the insight that the possibility to export a standardized format for long-time data storage is indispensible has been accepted. Such a format should not only allow storage of data (here: results), but also storage of information on how these data or results were produced. Thus standard formats allow the examination of data by a reviewer or the one-to-one reproduction of analyses.

In the following, the process of standard creation of the Proteomics Standards Initiative (PSI) (2) (PSI, website: http://www.psidev.info) of the Human Proteome Organisation (HUPO, http://www.hupo.org) is shortly sketched (see previous chapters for a more detailed explanation); and the standard format for storing Proteomics results (mzIdentML) is presented.

2. mzIdentML: A Standard Format for Proteomics Results

The PSI defines standards for data representation in Proteomics to support data comparison, data exchange, and result verification. This definition normally covers both a document defining "minimal information" necessary for the unambiguous description of an experiment from a specific Proteomics domain (e.g., MIAPE principles (3), MIAPE-MS (4) for spectra, and MIAPE-PI (5) for results) and the definition of a (usually XML-based) storage format.

The PSI work group for Proteomics results is called "Proteomics Informatics (PI)" (see Note 1). It assembles standard formats for describing the results of identification and quantitation analyses, including workflows for proteins, peptides, and protein modifications based on mass spectrometry. It uses the "Psidev-pi-dev" mailing list (https://lists.sourceforge.net/lists/listinfo/psidev-pi-dev) and the development-supporting Googlecode site http://code.google.com/p/psi-pi/.

An XML-based format for peptide/protein identification analyses has been released recently (specification: http://www.psidev.info/index.php?q=node/403). It was developed using various appellations, where "AnalysisXML" with file extension ".axml" was the penultimate one (see Note 2). It was displaced by "mzIdentML" with extension ".mzid," as soon as it was decided to split the standard into one format for peptide and protein identification results and one for quantification results, the development of which is in its infancy (current name: "mzQuantML").

2.1. Design Principles and Use Cases

mzIdentML has been designed to support a set of principle tasks, for example, "discovery of relevant results," "sharing of best practice," "evaluation of results," "sharing of data sets," and "creation of a format for input to analysis software."

In the Proteomics field, there exist several rather different approaches to identify peptides/proteins from mass spectra. At some point in the discussion (see Note 3), it became obvious that the subset of use cases had to be defined and the standard format should at least be able to cover and – unfortunately – exclude other use cases.

Use cases considered (see Note 4) are defined in the specification document (http://code.google.com/p/psi-pi/source/browse/#svn/trunk/specification_document) and include, for example, the following:

- Examination of results from an MS, MS/MS, or MSn run with sufficient information for a "viewing" tool to create output conforming to the requirements made by MIAPE guidelines or by journals for publishing manuscripts.
- Documentation of enough information to re-run the analysis (software parameters, sequence database, and spectra).
- Storage of the results of a decoy database search (together with results from the original sequences). Investigation of the effect of changing, for example, the threshold on the false-discovery rate.
- It should be possible to save the results from an analysis of a metabolic labeling experiment (e.g., for a 14N/15N experiment).
- It should be possible to derive the spectrum in which a specific peptide or protein was identified (for example, in order to derive the retention time of a peptide in an LC–MS/MS run). When an mzML file was the input for the analysis, the unique "id" of the spectrum should be referenced in the mzIdentML file.

Other use cases are, for example, spectral library searches, top down searches, storage of fragmentation information, searches against nucleic acid databases, and combination of the results from multiple peptide searches into one set of protein results (see Note 5).

2.2. Ideas and Concepts of the mzIdentML Schema (Release 1.0.0)

The previously defined design concepts directly follows that a storage model for the specific use cases needs to store (i) the identity and configuration (parameters) of the *software* used to perform the analysis; (ii) the protocol used to apply this software such as *input data* (e.g., spectra and search databases) or date of search; and (iii) the *output data* such as result molecules (peptides with modifications, and amino acid sequences) with their scores. In the schema (see Fig. 1), these correspond to the elements (i) <AnalysisSoftwareList> together with <AnalysisProtocolCollection>; (ii) <Inputs>; and (iii) <AnalysisData> together with <SequenceCollection>.

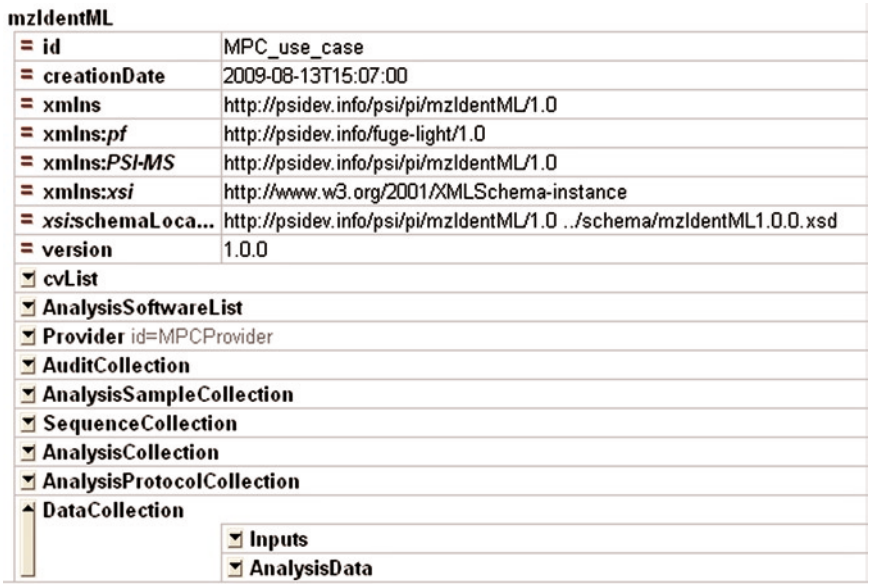


Fig. 1. The high-level elements of an mzIdentML file (*grid view*)

The actual application of the analyses and their connection with parameters, inputs, and outputs are described in <AnalysisCollection>. In <cvList>, the controlled vocabularies (see next section) used in the mzIdentML file are characterized, and <Provider> together with <AuditCollection> describes persons or institutions providing the mzIdentML file and the software used. <AnalysisSampleCollection> allows a simple description of the sample used in the Proteomics experiment.

Nearly all elements below the above-mentioned "parent elements" have an "id" attribute. The schema contains rules to ensure the uniqueness of an "id" value within its respective sub-tree. Additionally, it contains <keyref> elements for referencing attributes like "Peptide_ref" to ensure that the reference points to an element in the correct sub-tree.

One mzIdentML file is meant to store the *final* results of one analysis workflow, and *not intermediate* results, from which further processing with other parameters might be possible. If, for example, a set of proteins is reported, only the peptides necessary to justify these proteins are to be reported. To fulfil one of the design principles, a small exception from this concept is the possibility to define a threshold for a peptide or protein result value and flag the peptides/proteins that pass this threshold. This allows, for example, a protein list with decoy entries, where the false-discovery rate threshold may be changed later.

A mzIdentML file is meant to store at most one set of proteins, more exactly the *results of one protein detection* analysis (possibly a combination of the results of several peptide identification runs).

It is possible to store more than one peptide result sets without having a protein detection analysis, but that should be avoided.

In the future, there is hope that all result-producing tools – especially all search engines – will be able to export mzIdentML files directly. Until this goal is reached, mzIdentML files are converted from existing result files. Therefore, a <SourceFile> element (child of <Inputs>) describes the file from which the mzIdentML file has been produced.

The <SpectraData> element describes the spectra data set used as input (using <cvParam> elements). Especially the file format, e.g., MS:1000774 : "multiple peak list nativeID format" (e.g., for MGF spectra files) or MS:1000775: "single peak list nativeID format" (e.g., for DTA files), and the type of identifier used to reference a certain spectrum within this spectra data set, e.g., "index = xsd:nonNegativeInteger" (e.g., for MGFs) or "file = xsd:IDREF" (e.g., for DTAs), are specified. Defining an identifier is necessary, because spectra are referenced from results of a peptide identification run.

2.3. The PSI–MS Controlled Vocabulary

A "controlled vocabulary" generally contains pre-defined terms to avoid spelling or case ambiguities (see Note 6). The PSI CVs (e.g., the PSI-MOD CV – http://psidev.cvs.sourceforge.net/psidev/psi/mod/data/PSI-MOD.obo, or the PSI–MS CV – http://psidev.cvs.sourceforge.net/*checkout*/psidev/psi/psi-ms/mzML/controlledVocabulary/psi-ms.obo), are hierarchies of controlled terms ("ontologies") (see Note 7) having, for example, "is_a" or "has_a" relationships to one or many "parent terms." Each term has a unique accession number and can have a value (e.g., MS:1001191, "*p*-value," value = 0.05) and a unit for this value (e.g., MS:1001117, "theoretical mass," unit = dalton) (see Note 8). In the mzIdentML file, <cvParam> elements are used to describe further details of a modeled object. Thus, most of the search engine-specific result values are annotated using CV terms, e.g., <cvParam accession="MS:1001171" name="mascot:score" cvRef="PSI-MS" value="13.21"/>, as annotation of a peptide identification. In the CV hierarchy, this term "is_a" "search-engine specific score," which itself "is_a" "spectrum identification result detail." The position within the hierarchy can be used to check the correct use of the CV terms (see next section).

Other ontologies or controlled vocabularies may also be suitable or required for some elements of mzIdentML, for example, Unit Ontology (http://www.obofoundry.org/cgi-bin/detail.cgi?id=unit), ChEBI (http://www.ebi.ac.uk/chebi/), OBI (Ontology of Biological Investigations – http://obi.sourceforge.net/), PSI Protein modifications CV (http://psidev.sourceforge.net/mod/data/PSI-MOD.obo), and Unimod modifications database (http://www.unimod.org/obo/unimod.obo).

2.4. Semantic Validation and Mapping Files

XML schema validation checks the syntax of an mzIdentML file. Whether or not CV terms are used at correct locations cannot be judged by a syntax check. This requires a semantic check: The correct use of CV terms within an mzIdentML file is controlled via a mapping file defining each XML location (XPath notation) where <cvParam> elements can be used, and the terms allowed for that location. The mapping file is interpreted by validation software, which then evaluates that the data annotation in the mzIdentML file is consistent. An example validation tool has been implemented as part of the OpenMS software suite: http://www.psidev.info/validator.

Other semantic checks beyond CV validation are imaginable, e.g., whether "start/end" attributes of peptides are correct within the protein sequence.

2.5. Conformance to MIAPE and Journal Guidelines

An mzIdentML file can be syntactically correct without conformance to MIAPE guidelines. Or it can conform to MIAPE without fulfilling the journal guidelines (e.g., Molecular and Cellular Proteomics guidelines – http://www.mcponline.org/misc/ParisReport_Final.dtl). Elements and attributes that must be filled correctly to let the mzIdentML file conform are listed at http://www.psidev.info/index.php?q=node/386 (mzIdentML Conformance to MIAPE) and at http://www.psidev.info/index.php?q=node/406 (mzIdentML Conformance to MCP Guidelines). mzIdentML files submitted to public repositories or cited in journal articles should at least conform to MIAPE .

2.6. mzIdentML Examples

Together with the release of schema, specification document, and mapping file, several example files have been released as reference for further developments and discussions. In the following, excerpts of these files exemplify some mzIdentML concepts (see Note 9).

3. Examples

3.1. Example 1: Multiple Search Engines, Combination of Peptides, and Decoy Approach

In the following example, two search engines (Mascot and Sequest) have been used for peptide identification on one spectra data set. The identified peptides have been assembled to proteins by ProteinExtractor, a protein assembly algorithm within the ProteinScape LIMS (6) (Bruker Daltonik GmbH, Bremen, Germany). The search database used was a decoy database containing shuffled sequences for each target entry (created with the Decoy Database Builder tool (7), http://www.medizinisches-proteom-center.de/). The reported protein list contains target and decoy entries, where only target entries with a false-discovery rate above 0.05 are marked as final results of the overall analysis.

```
<SpectrumIdentificationProtocol id="SEQUEST_proto" AnalysisSoftware_ref="SEQUEST_SW">
  <SearchType>
    <cvParam accession="MS:1001083" name="ms-ms search" cvRef="PSI-MS"/>
  </SearchType>
  <AdditionalSearchParams>
    <cvParam accession="MS:1001211" name="parent mass type mono" cvRef="PSI-MS"/>
    <cvParam accession="MS:1001256" name="fragment mass type mono" cvRef="PSI-MS"/>
  </AdditionalSearchParams>
  <ModificationParams>
    <SearchModification fixedMod="false">
      <ModParam residues="M" massDelta="15.994914622">
        <cvParam accession="UNIMOD:35" name="Oxidation" cvRef="UNIMOD"/>
      </ModParam>
    </SearchModification>
  </ModificationParams>
  <Enzymes>
    <Enzyme id="Trypsin" missedCleavages="1">
      <EnzymeName>
        <cvParam accession="MS:1001251" name="Trypsin" cvRef="PSI-MS"/>
      </EnzymeName>
    </Enzyme>
  </Enzymes>
  <FragmentTolerance>
    <cvParam accession="MS:1001412" name="search tolerance plus value" cvRef="PSI-MS"
        value="0.9" unitAccession="UO:0000221" unitName="dalton" unitCvRef="UO"/>
    <cvParam accession="MS:1001413" name="search tolerance minus value" cvRef="PSI-MS"
        value="0.9" unitAccession="UO:0000221" unitName="dalton" unitCvRef="UO"/>
  </FragmentTolerance>
  <ParentTolerance>
    <cvParam accession="MS:1001412" name="search tolerance plus value" cvRef="PSI-MS"
        value="75.0" unitAccession="UO:0000169" unitName="parts per million" unitCvRef="UO"/>
    <cvParam accession="MS:1001413" name="search tolerance minus value" cvRef="PSI-MS"
        value="75.0" unitAccession="UO:0000169" unitName="parts per million" unitCvRef="UO"/>
  </ParentTolerance>
  <Threshold>
    <cvParam accession="MS:1001494" name="no threshold" cvRef="PSI-MS"/>
  </Threshold>
</SpectrumIdentificationProtocol>
```

Fig. 2. Sequest parameters defined within the <SpectrumIdentificationProtocol> element.

The parameters of the search engine runs are defined in two <SpectrumIdentificationProtocol> elements (see Fig. 2 for the Sequest parameters), referencing the respective <AnalysisSoftware> elements (see Fig. 3) and defining the type of search (MS/MS), the mass types (monoisotopic parent and fragment masses), the modifications (oxidation as variable modification), the digestion enzyme (Trypsin) and the mass tolerances for parent and fragment masses (0.9 Da and 75 ppm, respectively). Further search engine-specific parameters can be reported as <cvParam> children of <AdditionalSearchParams>.

The same <SpectraData> and <SearchDatabase> are used for both search engine runs. They are specified within the <Inputs> element (see Fig. 4), a child of <AnalysisData>. The source of the

```
<AnalysisSoftware id="SEQUEST_SW" name="ThermoFisher TurboSequest" version="PVM Slave v.27 (rev. 12)"
    URI="http://www.thermo.com/com/cda/product/detail/1,,16483,00.html">
    <ContactRole Contact_ref="THERMO">
        <role>
            <cvParam accession="MS:1001267" name="software vendor" cvRef="PSI-MS"/>
        </role>
    </ContactRole>
    <SoftwareName>
        <cvParam accession="MS:1001208" name="Sequest" cvRef="PSI-MS"/>
    </SoftwareName>
</AnalysisSoftware>
<AnalysisSoftware id="Mascot_SW" name="Mascot" version="2.2.0" URI="http://www.matrixscience.com/">
    <ContactRole Contact_ref="MATRIXSCIENCE">
        <role>
            <cvParam accession="MS:1001267" name="software vendor" cvRef="PSI-MS"/>
        </role>
    </ContactRole>
    <SoftwareName>
        <cvParam accession="MS:1001207" name="Mascot" cvRef="PSI-MS"/>
    </SoftwareName>
</AnalysisSoftware>
```

Fig. 3. The <AnalysisSoftware> element describes the software used.

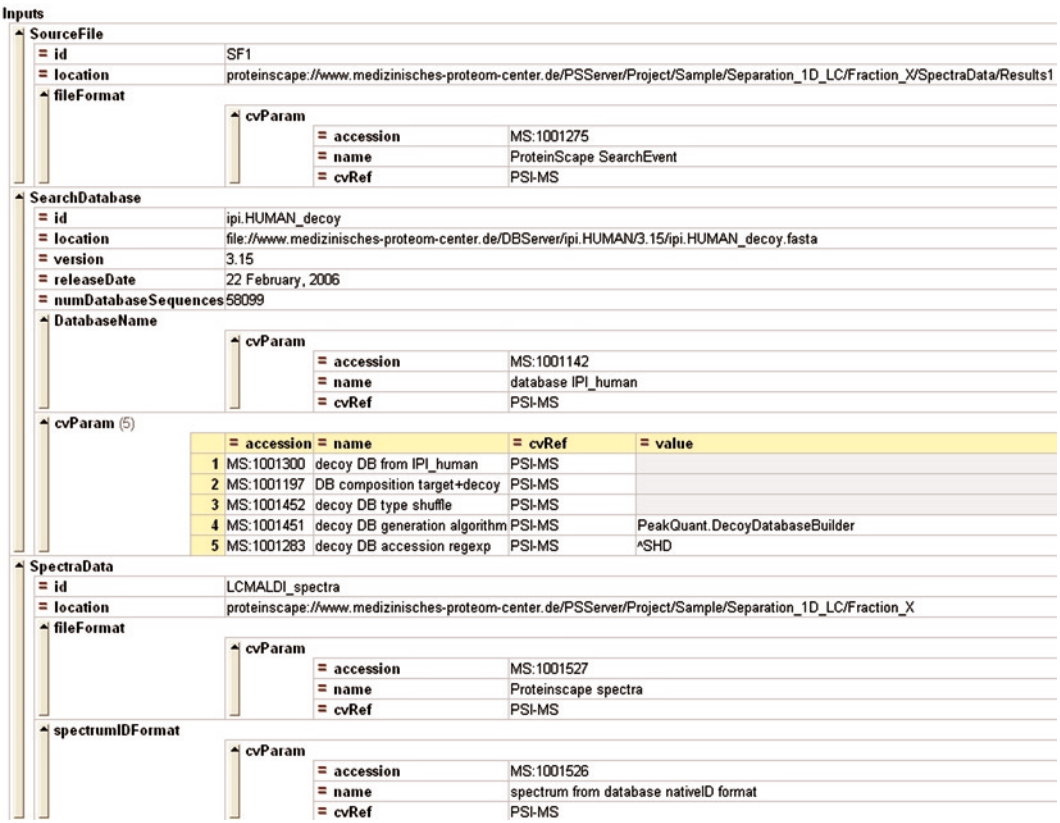


Fig. 4. The <Inputs> element specifying source file, search database, and spectra data (*grid view*).

mzIdentML file itself is specified using the <SourceFile> element. In this example, source file and spectra data are data sets stored in a database, the ProteinScape LIMS. The search database is a human IPI database (version 3.15). The process of generating decoy entries is described in <cvParam> elements of <SearchDatabase> (e.g., the regular expression for identifying decoy entries is "^SHD").

All specified parameters, input data, and a reference to the result data ("SpectrumIdentificationList_ref" attribute) are assembled in a <SpectrumIdentification> element, describing the actual runs of the analysis (see Fig. 5); therefore, an "activityDate" attribute can be given.

The results of a <SpectrumIdentification> analysis are stored within a <SpectrumIdentificationList> element (see Fig. 6); more exactly, this element contains all peptides of all spectra. Its child element <SpectrumIdentificationResult> contains all peptides identified in one spectrum, which itself is specified using the attributes "SpectraData_ref" and "spectrumID." The <SpectrumIdentificationItem> element as child of <SpectrumIdentificationResult> contains the information of one identified peptide. The <...Item> elements have a "rank"

```
<AnalysisCollection>
  <SpectrumIdentification id="SEQUEST_analysis"
    SpectrumIdentificationProtocol_ref="SEQUEST_proto"
    SpectrumIdentificationList_ref="SEQUEST_results"
    activityDate="2007-05-12T13:00:00">
    <InputSpectra SpectraData_ref="LCMALDI_spectra"/>
    <SearchDatabase SearchDatabase_ref="ipi.HUMAN_decoy"/>
  </SpectrumIdentification>
  <SpectrumIdentification id="Mascot_analysis"
    SpectrumIdentificationProtocol_ref="Mascot_proto"
    SpectrumIdentificationList_ref="Mascot_results"
    activityDate="2007-05-12T14:00:00">
    <InputSpectra SpectraData_ref="LCMALDI_spectra"/>
    <SearchDatabase SearchDatabase_ref="ipi.HUMAN_decoy"/>
  </SpectrumIdentification>
  <ProteinDetection id="ProteinExtractor_analysis"
    ProteinDetectionProtocol_ref="ProteinExtractor_proto"
    ProteinDetectionList_ref="ProteinExtractor_results"
    activityDate="2007-05-12T15:30:00">
    <InputSpectrumIdentifications SpectrumIdentificationList_ref="SEQUEST_results"/>
    <InputSpectrumIdentifications SpectrumIdentificationList_ref="Mascot_results"/>
  </ProteinDetection>
</AnalysisCollection>
```

Fig. 5. The three actual analyses with specified protocol parameters, input, and output data.

```
<SpectrumIdentificationList id="SEQUEST_results">
  <SpectrumIdentificationResult id="SEQ_spec1"
    spectrumID="databasekey=1" SpectraData_ref="LCMALDI_spectra">
    <SpectrumIdentificationItem id="SEQ_spec1_pep1" rank="1" Peptide_ref="prot1_pep1"
      chargeState="1" calculatedMassToCharge="1507.6950" experimentalMassToCharge="1507.696"
      passThreshold="true">
      <PeptideEvidence id="PE1_SEQ_spec1_pep1" start="67" end="79" isDecoy="false" DBSequence_Ref="prot1_IPI" missedCleavages="0"/>
      <cvParam accession="MS:1001505" name="ProteinScape:IntensityCoverage" cvRef="PSI-MS" value="0.3919545603809718"/>
      <cvParam accession="MS:1001506" name="ProteinScape:SequestMetaScore" cvRef="PSI-MS" value="7.59488518903425"/>
    </SpectrumIdentificationItem>
  </SpectrumIdentificationResult>
  <SpectrumIdentificationResult id="SEQ_spec2a"
    spectrumID="databasekey=2" SpectraData_ref="LCMALDI_spectra">
    <SpectrumIdentificationItem id="SEQ_spec2a_pep1" rank="1" Peptide_ref="prot1_pep2"
      chargeState="1" calculatedMassToCharge="1920.9224" experimentalMassToCharge="1920.923" passThreshold="true">
      <PeptideEvidence id="PE1_SEQ_spec2a_pep1" start="67" end="83" isDecoy="false" DBSequence_Ref="prot1_IPI" missedCleavages="1"/>
      <cvParam accession="MS:1001505" name="ProteinScape:IntensityCoverage" cvRef="PSI-MS" value="0.5070386909133888"/>
      <cvParam accession="MS:1001506" name="ProteinScape:SequestMetaScore" cvRef="PSI-MS" value="10.8810331335713"/>
    </SpectrumIdentificationItem>
  </SpectrumIdentificationResult>
            :
            :
            :
```

Fig. 6. The beginning of a <SpectrumIdentificationList>, showing the identified peptides of two spectra.

attribute to rank multiple identifications in one spectrum. In this example, only one peptide is reported per spectrum. The "Peptide_ref" attribute links this spectrum/peptide pair with a peptide stored in the <SequenceCollection> of the mzIdentML file (here, "prot1_pep1" is the amino acid sequence "AGTQIENIDEDFR" without modifications). This avoids redundant repetition of sequences. "chargeState," "calculatedMassToCharge," and "experimentalMassToCharge" are the most prominent characteristics of peptide identification from mass spectra and are, therefore, reported as attributes. Other result values like search engine scores are reported as <cvParam> elements. Here, only the ProteinScape-specific result values Intensity Coverage (the intensity of covered peaks) and SequestMetaScore (a combination of the original Sequest scores) are reported.

At first glance, the <PeptideEvidence> element seems to be misplaced in the peptide result part of mzIdentML, as it links the identified peptide with a protein sequence. But it is a very efficient possibility to specify, in which protein (or proteins) a peptide occurs, whether these proteins are decoy sequences, or which reading frame and translation table is applied in case of a nucleotide search database. This information can be very important, although no protein detection has been performed. On the contrary, if one has been performed, the <PeptideEvidence> elements can be very efficiently referenced from the results of a protein detection analysis (see below).

The parameters of the *protein* detection step are described in the <ProteinDetectionProtocol> element (see Fig. 7). In this example, there are several parameters of the ProteinExtractor algorithm reported as <cvParam> elements. Most interesting is the <Threshold> element specifying a threshold of 0.05 (or 5%)

ProteinDetectionProtocol

= id	ProteinExtractor_proto
= AnalysisSoftware_ref	ProteinExtractor_SW

AnalysisParams

cvParam (22)

	= accession	= name	= cvRef	= value
1	MS:1001424	ProteinExtractor:Methodname	PSI-MS	SEQUEST_2.5
2	MS:1001425	ProteinExtractor:GenerateNonRedundant	PSI-MS	true
3	MS:1001426	ProteinExtractor:IncludeIdentified	PSI-MS	false
4	MS:1001427	ProteinExtractor:MaxNumberOfProteins	PSI-MS	3000
5	MS:1001429	ProteinExtractor:MinNumberOfPeptides	PSI-MS	2
6	MS:1001430	ProteinExtractor:UseMascot	PSI-MS	false
7	MS:1001431	ProteinExtractor:MascotPeptideScoreThreshold	PSI-MS	0.0
8	MS:1001432	ProteinExtractor:MascotUniqueScore	PSI-MS	0.0
9	MS:1001433	ProteinExtractor:MascotUseIdentityScore	PSI-MS	false
10	MS:1001434	ProteinExtractor:MascotWeighting	PSI-MS	0.0
11	MS:1001435	ProteinExtractor:UseSequest	PSI-MS	true
12	MS:1001436	ProteinExtractor:SequestPeptideScoreThreshold	PSI-MS	2.5
13	MS:1001437	ProteinExtractor:SequestUniqueScore	PSI-MS	2.5
14	MS:1001438	ProteinExtractor:SequestWeighting	PSI-MS	1.0
15	MS:1001439	ProteinExtractor:UseProteinSolver	PSI-MS	false
16	MS:1001440	ProteinExtractor:ProteinSolverPeptideScoreThreshold	PSI-MS	0.0
17	MS:1001441	ProteinExtractor:ProteinSolverUniqueScore	PSI-MS	0.0
18	MS:1001442	ProteinExtractor:ProteinSolverWeighting	PSI-MS	0.0
19	MS:1001443	ProteinExtractor:UsePhenyx	PSI-MS	false
20	MS:1001444	ProteinExtractor:PhenyxPeptideScoreThreshold	PSI-MS	0.0
21	MS:1001445	ProteinExtractor:PhenyxUniqueScore	PSI-MS	0.0
22	MS:1001446	ProteinExtractor:PhenyxWeighting	PSI-MS	0.0

Threshold

cvParam

= accession	MS:1001447
= name	prot:FDR threshold
= cvRef	PSI-MS
= value	0.05

Fig. 7. ProteinExtractor parameters and Threshold defined within the <ProteinDetectionProtocol> element (*grid view*).

for the false-discovery rate on the sorted protein list. Instead of spectra data sets, the <ProteinDetection> element – describing the actual analysis – references peptide sets as input data using "SpectrumIdentificationList_ref" attributes (see Fig. 5). The result of a protein detection (stored in one <ProteinDetectionList> element, see Fig. 8) is a set of <ProteinAmbiguityGroup> elements. This reflects the fact that a set of identified peptides can be part of more than one protein (e.g., in case of homologs or isoforms) and it may not unambiguously be decided which protein was part of the sample. The <ProteinDetectionHypothesis> describes the proteins within an ambiguity group (in this example, each group contains only one possible protein). The "passThreshold" attribute expresses whether the reported protein passed the defined threshold, which is true for proteins 1 + 2. Only these two proteins pass the threshold in this example, as the third is a decoy protein and thus the false-discovery rate is exceeded for proteins 3–7. The "DBSequence_ref" attribute references the protein sequence within the <SequenceCollection> element. It is optional, as the <PeptideEvidence> element

ProteinDetectionList

= id	ProteinExtractor_results							
ProteinAmbiguityGroup (7)								
= id	ProteinDetectionHypothesis							
1 group1	ProteinDetectionHypothesis							
= id	id_prot1							
= passThreshold	true							
= DBSequence_ref	prot1_IPI							
PeptideHypothesis (4)								
	= PeptideEvidence_Ref							
	1 PE1_SEQ_spec1_pep1							
	2 PE1_SEQ_spec2a_pep1							
	3 PE1_SEQ_spec3a_pep1							
	4 PE1_SEQ_spec10_pep1							
cvParam (5)								
	= accession	= name	= cvRef	= value	= unitAccession	= unitName	= unitCvRef	
	1 MS:1001093	sequence coverage	PSI-MS	0.13				
	2 MS:1001301	protein rank	PSI-MS	1				
	3 MS:1001097	distinct peptide sequences	PSI-MS	4				
	4 MS:1001250	local FDR	PSI-MS	0.0	UO:0000187	percent	UO	
	5 MS:1001507	ProteinExtractor:Score	PSI-MS	200.4				
2 group2	ProteinDetectionHypothesis id=id_prot2 passThreshold=true							
3 group3	ProteinDetectionHypothesis id=id_prot3 passThreshold=false							
4 group4	ProteinDetectionHypothesis id=id_prot4 passThreshold=false							
5 group5	ProteinDetectionHypothesis id=id_prot5 passThreshold=false							
6 group6	ProteinDetectionHypothesis id=id_prot6 passThreshold=false							
7 group7	ProteinDetectionHypothesis id=id_prot7 passThreshold=false							

Fig. 8. Results of a protein detection analysis stored in a <ProteinDetectionList> (*grid view*).

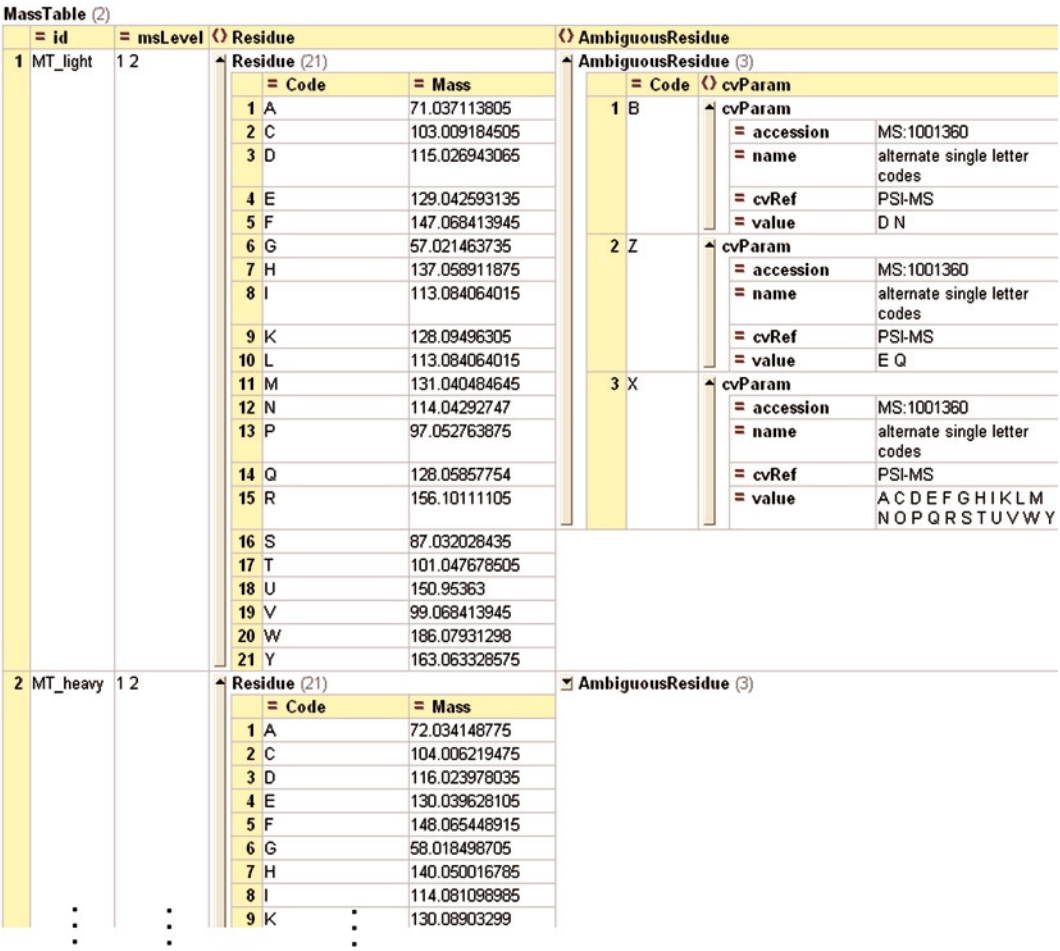
referenced in the following <PeptideHypothesis> section already references a protein sequence. As for the peptide results, <cvParam> elements are used to state the scores and other characteristics of the protein result.

3.2. Example 2: 14N/15N

The most important differences between an LC–MS/MS run, as described in the previous example, and a run of mixed samples labeled with 14N/15N isotopes (8) (http://code.google.com/p/psi-pi/source/browse/trunk/examples/Mascot_N15_example.mzid) are the use of two mass tables, defined in <SpectrumIdentificationProtocol> (see Fig. 9), and the mechanism of referencing them (using "MassTable_ref" attributes of <SpectrumIdentificationItem> elements). With mzIdentML, only the results of the identification part of the 14N/15N experiment can be described, and not that of the quantitation part.

3.3. Example 3: Fragmentation Information

Fragmentation information (example: http://code.google.com/p/psi-pi/source/browse/trunk/examples/Mascot_MSMS_example.mzid) is given in a <Fragmentation> element (see Fig. 10) of <SpectrumIdentificationItem>. The ion type, charge state, and – using a <FragmentationTable> element (see



MassTable (2)

	id	msLevel	Residue	AmbiguousResidue
1	MT_light	1 2	Residue (21)	AmbiguousResidue (3)

Residue (21)

	Code	Mass
1	A	71.037113805
2	C	103.009184505
3	D	115.026943065
4	E	129.042593135
5	F	147.068413945
6	G	57.021463735
7	H	137.058911875
8	I	113.084064015
9	K	128.09496305
10	L	113.084064015
11	M	131.040484645
12	N	114.04292747
13	P	97.052763875
14	Q	128.05857754
15	R	156.10111105
16	S	87.032028435
17	T	101.047678505
18	U	150.95363
19	V	99.068413945
20	W	186.07931298
21	Y	163.063328575

AmbiguousResidue (3)

	Code	cvParam	
1	B	cvParam	
		accession	MS:1001360
		name	alternate single letter codes
		cvRef	PSI-MS
		value	D N
2	Z	cvParam	
		accession	MS:1001360
		name	alternate single letter codes
		cvRef	PSI-MS
		value	E Q
3	X	cvParam	
		accession	MS:1001360
		name	alternate single letter codes
		cvRef	PSI-MS
		value	A C D E F G H I K L M N O P Q R S T U V W Y

2	MT_heavy	1 2	Residue (21)	AmbiguousResidue (3)

Residue (21)

	Code	Mass
1	A	72.034148775
2	C	104.006219475
3	D	116.023978035
4	E	130.039628105
5	F	148.065448915
6	G	58.018498705
7	H	140.050016785
8	I	114.081098985
9	K	130.08903299

Fig. 9. Definition of two mass tables for a 14N/15N analysis (*grid view*).

Fig. 11) – previously defined characteristics, like at least m/z, together with, for example, intensity and error, of the identified ions are specified in the <Fragmentation> element.

4. Outlook

As the next step after the release of the standard, a publication was drafted and is in submission. Tool and search engine developers are going to finish their implementations, which were already begun (see list on http://www.psidev.info/index.php?q=node/408). In parallel, the definition of mzQuantML will go on with the help of the Proteomics community.

```
<Fragmentation>
  <IonType index="1 2 3 4 8" charge="1">
    <cvParam cvRef="PSI-MS" accession="MS:1001229" name="frag: a ion"/>
    <FragmentArray values="214.8 286.1 342.8 444.1 814.1 " Measure_ref="m_mz"/>
    <FragmentArray values="18 83 48 75 277" Measure_ref="m_intensity"/>
    <FragmentArray values="-0.3026 -0.0397 -0.3612 -0.1089 -0.3305" Measure_ref="m_error"/>
  </IonType>
  <IonType index="8" charge="2">
    <cvParam cvRef="PSI-MS" accession="MS:1001229" name="frag: a ion"/>
    <FragmentArray values="408.2 " Measure_ref="m_mz"/>
    <FragmentArray values="187" Measure_ref="m_intensity"/>
    <FragmentArray values="0.4811" Measure_ref="m_error"/>
  </IonType>
  <IonType index="1 2 3 4 5 7 8 9 10 11" charge="1">
    <cvParam cvRef="PSI-MS" accession="MS:1001224" name="frag: b ion"/>
    <FragmentArray values="242.8 314.2 371.127841 472.096295 584.997427 729.2 842.9 956.1 1056 1169.1 " Measure_ref="m_mz"/>
    <FragmentArray values="417 1308 1663 29390 39060 626 4893 2241 492 11230" Measure_ref="m_intensity"/>
    <FragmentArray values="-0.2975 0.0653 -0.0283 -0.1075 -0.2904 -0.1414 0.4746 -0.3684 0.4632 0.4792" Measure_ref="m_error"/>
  </IonType>
  <IonType index="11" charge="2">
    <cvParam cvRef="PSI-MS" accession="MS:1001224" name="frag: b ion"/>
    <FragmentArray values="584.997427 " Measure_ref="m_mz"/>
    <FragmentArray values="39060" Measure_ref="m_intensity"/>
    <FragmentArray values="0.1834" Measure_ref="m_error"/>
  </IonType>
  <IonType index="11 10 9 8 7 6 5 4 3 2" charge="1">
    <cvParam cvRef="PSI-MS" accession="MS:1001220" name="frag: y ion"/>
    <FragmentArray values="1100.3 1029.3 972.292156 871.2 758.2 671.2 614.4 501.4 387.2 288.2 " Measure_ref="m_mz"/>
    <FragmentArray values="1290 1846 1874 25390 73560 65440 1403 10430 656 2137" Measure_ref="m_intensity"/>
    <FragmentArray values="-0.3422 -0.3051 -0.2915 -0.3360 -0.2519 -0.2199 0.0016 0.0856 -0.0714 -0.0030" Measure_ref="m_error"/>
  </IonType>
  <IonType index="11 8 7" charge="2">
    <cvParam cvRef="PSI-MS" accession="MS:1001220" name="frag: y ion"/>
    <FragmentArray values="551.3 436.4 380.1 " Measure_ref="m_mz"/>
    <FragmentArray values="800 11 46" Measure_ref="m_intensity"/>
    <FragmentArray values="0.4752 0.1284 0.3704" Measure_ref="m_error"/>
```

Fig. 10. The <Fragmentation> element of a <SpectrumIdentificationItem> (*excerpt*).

FragmentationTable

Measure (3)

	= id	{} cvParam	
1	m_mz	cvParam	
		= cvRef	PSI-MS
		= accession	MS:1001225
		= name	product ion m/z
2	m_intensity	cvParam	
		= cvRef	PSI-MS
		= accession	MS:1001226
		= name	product ion intensity
3	m_error	cvParam	
		= cvRef	PSI-MS
		= accession	MS:1001227
		= name	product ion m/z error
		= unitAccession	MS:1000040
		= unitName	m/z
		= unitCvRef	PSI-MS

Fig. 11. The <FragmentationTable>, defining characteristics reported in the fragmentation table (*grid view*).

5. Notes

1. Although PSI work groups are open for everyone, often there crystallizes a core group, which changes over time. At the time of writing, the core PI group consisted of Andrew Jones (University of Liverpool, UK), David Creasy (MatrixScience, London, UK), Andreas Bertsch (Eberhard Karls-Universitaet Tuebingen, Germany), Jenny Siepen (University of Manchester, UK), Phil Jones (European Bioinformatics Institute, Hinxton, UK), and Martin Eisenacher (Medizinisches Proteom-Center, Ruhr-Universitaet Bochum, Germany).
2. Predecessors of mzIdentML appellations: mzAnalysis, mzProtID, mzIdent, and AnalysisXML; suggestions: piaML (proteomics informatic analysis ML), paML or even piML, analysisML, suggestions for extensions: .AML, .AnaML, .AnML, .PIML, .aML, .anaML, .anML, .AXML, .AnaXML, .AnXML, .PIXML, .aXML, .anaXML, .anXML.
3. The decision to restrict mzIdentML to a subset of the originally formulated set of use cases was not easy. After more or less fruitful discussions and years of development, finally, at the PSI spring meeting in Toledo 2008, there was consensus for this restriction. Most painful was the decision to drop the quantification use case from milestone 1 release of the standard, as it was reflected to be an essential part of a Proteomics results standard. During the PSI spring meeting in Turku 2009, the first summary of the most common quantification use cases was assembled and a sketch of their essential characteristics was contemplated. It turned out that for all of them, it would be possible to have a separate schema for quantification results, which reference into one or more mzIdentML file(s).
4. The previous mzIdentML format allowed rather complex workflows, such as more than one protein detection and further types of analysis, for example, quality estimation analyses. Therefore, the standard was heavily based on FuGE, which is naturally designed to model complex workflows. Additionally, the design process was based on designing the model using an UML (Unified Modeling Language) tool and to export the schema from that tool. To ease the development process, the UML modeling was replaced by the "traditional" schema modeling; the remaining necessary FuGE elements or types (e.g., <Provider> or ProtocolType) are now incorporated by referencing a schema called "FuGElight," including only the necessary FuGE elements and types.
5. De novo peptide sequencing results are supported but produce extremely large files (later versions should improve that).

Use cases not supported in release 1.0.0 of mzIdentML (although it was planned in the past) are storage of relative and absolute quantitation information, support for biomarker discovery, and support for "sequence tagged searches."

6. There is no way to define an objective rule to judge whether a result characteristic should be an attribute or should be specified by a <cvParam> element. So this decision was discussed for each characteristic and then decided by community consensus.
7. The CV term hierarchies (ontologies) are stored in the OBO file format (9) (http://www.geneontology.org/GO.format.obo-1_2.shtml).
8. The PSI–MS controlled vocabulary (http://psidev.cvs.sourceforge.net/*checkout*/psidev/psi/psi-ms/mzML/controlledVocabulary/psi-ms.obo) contains annotations for mzML and mzIdentML files. The terms that require a value are denoted by having a "value-type" xref entry in the OBO file in the form "xref: value-type:xsd\:string." Units for values are denoted by having a "has_units" relationship ("relationship: has_units: UO:0000221 ! dalton"). As recommended by the PSI CV guidelines, psi-ms.obo should be dynamically maintained via the psidev-ms-vocab@lists.sourceforge.net mailing list on which any user can request new terms.
9. Some of the released examples have been produced by first conversion or export scripts (such as the Mascot and OMSSA examples) and some have been only handcrafted at the moment using fantasy values for some elements or attributes (like the "MPC example" presented here as example 1).

Acknowledgments

Martin Eisenacher is funded by CLIB ("Cluster Industrielle Biotechnologie") project 616 40003 0315413B (Q-ProM). The author wants to thank David Creasy, Andy Jones, Andreas Bertsch, and all other members of the Proteomics Informatics work group for fruitful discussions.

References

1. Eisenacher M, Martens L, Hardt T et al (2009) Getting a grip on proteomics data – Proteomics Data Collection (ProDaC). Proteomics 9(15):3928–3933
2. Orchard S, Deutsch EW, Binz PA et al (2009) Annual spring meeting of the Proteomics Standards Initiative. Proteomics 9(19): 4429–4432
3. Taylor CF, Paton NW, Lilley KS et al (2007) The minimum information about a proteomics experiment (MIAPE). Nat Biotechnol 25(8):887–893
4. Taylor CF, Binz PA, Aebersold R et al (2008) Guidelines for reporting the use of mass spectrometry in proteomics. Nat Biotechnol 26(8):860–861

5. Binz PA, Barkovich R, Beavis RC et al (2008) Guidelines for reporting the use of mass spectrometry informatics in proteomics. Nat Biotechnol 26(8):862
6. Stephan C, Kohl M, Turewicz M, Podwojski K, Meyer HE, Eisenacher M. Using Laboratory Information Management Systems as central part of a proteomics data workflow. Proteomics 2010;10:1230–49
7. Reidegeld KA, Eisenacher M, Kohl M et al (2008) An easy-to-use Decoy Database Builder software tool, implementing different decoy strategies for false discovery rate calculation in automated MS/MS protein identifications. Proteomics 8(6):1129–1137
8. Beynon RJ, Pratt JM (2005) Metabolic labeling of proteins for proteomics. Mol Cell Proteomics 4(7):857–872
9. Smith B, Ashburner M, Rosse C et al (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol 25(11):1251–1255

Chapter 11

Spectra, Chromatograms, Metadata: mzML-The Standard Data Format for Mass Spectrometer Output

Michael Turewicz and Eric W. Deutsch

Abstract

This chapter describes Mass Spectrometry Markup Language (mzML), an XML-based and vendor-neutral standard data format for storage and exchange of mass spectrometer output like raw spectra and peak lists. It is intended to replace its two precursor data formats (mzData and mzXML), which had been developed independently a few years earlier. Hence, with the release of mzML, the problem of having two different formats for the same purposes is solved, and with it the duplicated effort of maintaining and supporting two data formats. The new format has been developed by a broad-based consortium of major instrument vendors, software vendors, and academic researchers under the aegis of the Human Proteome Organisation (HUPO), Proteomics Standards Initiative (PSI), with full participation of the main developers of the precursor formats. This comprehensive approach helped mzML to become a generally accepted standard. Furthermore, the collaborative development insured that mzML has adopted the best features of its precursor formats. In this chapter, we discuss mzML's development history, its design principles and use cases, as well as its main building components. We also present the available documentation, an example file, and validation software for mzML.

1. Introduction

1.1. Motivation for a New File Format

Mass spectrometry produces a huge amount of raw data that is not manageable without computerized automation. Hence, computer-aided data management and data analysis are indispensable for the efficient scientific work. The basic challenge for a computer-aided workflow is to ensure storage and exchange of all important data concerning mass spectrometry experiments in a suitable, efficient, and generally accepted and supported data format. The need for a generally accepted data format was large because each instrument vendor had his own proprietary output format.

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696,
DOI 10.1007/978-1-60761-987-1_11,

Having many different and proprietary formats severely inhibited data exchange. In addition to the recorded mass spectra, which are usually acquired to perform the identification and quantitation of the analyzed (bio-) molecules, further information about the experiment, the so called metadata, is equally essential, since it is usually required for the result interpretation as well as for the identification- and quantitation-related calculations. Typical metadata, which may vary, are information about the respective instrument, experimental conditions and the software (including its settings) which led to the acquired and/or preprocessed data. Therefore, a suitable data format for mass spectrometry should store both the spectra and their corresponding metadata. Transforming mass spectrometer data into pure text-based peak lists is not a desirable method, since peak list transformation means data reduction (by discarding metadata and peak centroiding or isotoping), and important data and metadata are lost during this procedure. Thus, another approach has been proposed: the design of an XML-based data format. XML is designed to describe hierarchically structured data in a textual data format. It was designed to facilitate simplicity, generality, and usability for electronic data exchange. It is easily parsed by software and easy to read. Hence, XML appears to be the natural choice for the purposes discussed above.

To address this, two different institutions independently proposed two different XML data formats during 2003–2005. On the one hand, Human Proteome Organisation Proteomics Standards Initiative (HUPO-PSI, (1, 2)) developed the mzData format (3). The other data format was mzXML, developed at the Institute for Systems Biology (ISB, (4)). Although both approaches were XML-based and were designed to be vendor-neutral mass spectrometer raw output formats, they aimed at different use cases and followed different design philosophies concerning the flexibility regarding new kinds of important information. mzData was designed primarily as a universally applicable data exchange and archive format and was approved by the PSI as an official standard. Its design philosophy was characterized by its relative flexibility due to its controlled vocabulary approach, which kept the XML schema quite stable. It allowed for the introduction of innovations via a controlled vocabulary update. The disadvantage of this approach was the spread of different dialects.

On the other hand, mzXML was designed as an intermediate format for the needs of ISB's software pipeline, the TransProteomic Pipeline (TPP) software suite (5). Its design philosophy relied on a rigid schema, where most metadata was stored in enumerated attributes. This made document validation and software development easy, since there was only one way to represent an annotation. But there was also a tremendous disadvantage: the addition of new metadata options required a schema update with

a new version number. Software developers had to deal with the problem of a spreading number of possibly very similar mzXML versions. However, both were used widely as de facto standards.

Nevertheless, the community was dissatisfied, since there were two different formats for the same kind of data. Vendors and programers had to provide support for two different data formats. Data exchange, tool development, and data storage were still handicapped by this inconclusive state. The coexistence of two data standards was a suboptimal situation. The solution was to create a new data format merging together the best aspects of mzData and mzXML: mzML.

1.2. History of mzML

The history of the file format began with a PSI workshop in San Francisco (CA, USA) on April 21–23, 2006, (6–8) where a comparative analysis of mzXML and mzData was performed and the possibility of merging them into a single data format was discussed. On this occasion, the participants, (representatives of instrument vendors, the PSI, the ISB, software developers, and end users), reached an agreement toward the unification of mzXML and mzData and a roadmap to the new file format was determined (7, 9). As early as the follow-up PSI workshop (PSI Fall workshop, September 25–27, 2006, Washington DC, USA, (10)) concrete progress (contrasting of both schemas and analysis of mzData and mzXML features which were to be kept and/or enhanced in the new format) toward unification could be made. During the next years, a group of designers met regularly at workshops under the aegis of HUPO-PSI and advanced the project (9, 11–13). Finally, mzML 1.0.0 was released at the American Society for Mass Spectrometry (ASMS) conference in Denver (14), CO, USA, on June 1, 2008, and it was rapidly accepted as the new general standard. Although it was hoped that the first schema would remain stable as long as possible, it remained valid merely for a year, when mzML 1.1.0 was released on June 1, 2009. This update was caused by several deficiencies in mzML 1.0, which became evident during the implementation of different software projects for mzML 1.0 in early 2009. The differences between mzML 1.0.0 and mzML 1.1.0 (a complete listing of differences can be found in the specification document of mzML 1.1.0 (15)) make them incompatible. Hence, all software implemented so far had to be updated. However, the PSI Mass Spectrometry Standards working group will continue to support mzML, including documentation, controlled vocabulary maintenance and deployment of up-to-date semantic validators. News on the file format can be tracked on the regularly updated web page about mzML (16, see Note 2).

1.3. The Creation of PSI Standards

The PSI defines standards for data representation in Proteomics to support data comparison, data exchange, and result verification.

This definition normally covers both a document defining "minimal information" necessary for the unambiguous description of an experiment from a specific Proteomics domain (e.g., MIAPE-MS (17)) as well as the definition of a (usually XML-based) storage format.

Work on standards and documents is done at PSI spring meetings (6, 7, 12, 13, 18–20) – where interested scientists from all over the world can join – and in between these meetings, by the use of mailing lists and development-supporting sites, such as SourceForge ((21)) or Google Code ((22)).

PSI standard formats are developed within the PSI work groups called "Protein Separation (PS, (23))," "Mass Spectrometry (MS, (24))," "Molecular Interactions (MI, (25))," "Protein Modifications (MOD, (26))," and "Proteomics Informatics (PI, (27))." One work group may work on more than one standard (e.g., the MS work group is developing both mzML and TraML, (16, 24, 28)). Activities spanning all of the work groups include the editing of controlled vocabularies and "Minimal Information" (i.e., MIAPE) documents.

Once a standard format or document is near maturity, it enters the "PSI document process," a defined workflow containing phases of internal review (PSI editors and steering group) and then external review (specific reviewers and public review). Additionally, a journal publication might be initiated. Comments and suggestions regarding corrections are then built into the standard, and finally it gets released. Finalized standards can be found at http://psidev.info/index.php?q=node/100.

2. Design of mzML

The design of mzML has benefited tremendously from the initial experience obtained during the design and maintenance of mzXML and mzData. The advantages and disadvantages of both implementations were known. Therefore, the yardstick of performance was to merge the best aspects from precursors and to implement the lessons learned from experience with them.

mzML has been designed to store and describe mass spectrometry data output and its experimental context (metadata) as well as to support long-term data storage and data sharing, rather than short-range data management (although it is flexible concerning this matter). A corresponding set of principle tasks was formulated in the specification document (15) as follows:

– "Discovery of relevant results": All relevant data sets in databases or data repositories acquired by diverse acquisition techniques or combinations of diverse acquisition techniques should be identifiable and reviewable.

- "Sharing of best practice": Methods that have been successful at identifying low abundance peptides or proteins should be reviewable for sharing of best practice.
- "Evaluation of results": Sufficient additional information about a particular acquisition method should be provided to allow critical evaluation of the acquired data.
- "Sharing of data sets": Public repositories should be able to import or export the data, multisite projects should be able to share the results to support integrated analysis and meta-analysis of previously published data should be possible.
- "Most comprehensive support of the instruments output": Data should be ascertainable in all relevant forms of mass spectrometry representation, especially in centroid mode and profile mode.

Furthermore, an agreement concerning the compatibility of these principle tasks with the two precursor philosophies had to be reached among the designers. The outcome of this discussion was a set of design principles formulated as follows:

- *Simplicity*: Although the introduction of new features was discussed, the designers decided to abandon most extensions proposed during the design process. Finally, the conviction prevailed, that a simple, but robust, implementation would be a better basis for the new format.
- *Uniqueness*: The same information should always be encoded in a unique way. The designers preferred inflexible unambiguousness to inappropriate flexibility.
- *Stability*: The data format should be as stable as possible and the expected frequency of software updates should be limited. This is ensured by the concept of controlled vocabularies. Nevertheless, it was obvious that some kind of flexibility for encoding new important information must be incorporated. This is provided for by the concept of the <userParam> element (*see* Subheading 4.2).
- *Preservation of functionality*: All features of the precursor formats should be supported. However, coevally the designers decided to refrain from introducing new features in mzML 1.0.
- *Rapid development*: The designers recognized the duality of mzData and mzXML as the main problem for the community and the primary target for their efforts. Therefore, they decided to spend all resources to release a new standard data format and make its precursors obsolete. The rapid development of version 1.0 had higher priority than supporting new features. Support for new features has been halted until the release of version 2.0.

– *Validity*: The designers decided to validate mzML first by implementing software to read and write the new format before its release.

Finally, we want to outline the application field of mzML by listing several of its essential use cases. The example files referred to in the following can be found on the mzML web page (16). These essential use cases include the following:

– The ability of encoding both possible ways for spectrum representation: profile mass spectra and centroid mass spectra.
– Information about all current mass spectrometers (e.g., LTQ-FT mass spectrometers) and their settings as well as their experimental output should be encodeable and their (proprietary) mass spectrometer output should be convertible into mzML in an easy way. Example files: small.pwiz.1.1.mzML, small_miape.pwiz.1.1.mzML, and small_zlib.pwiz.1.1.mzML (generated via conversion with the msconvert tool from ProteoWizard (29) of a Thermo RAW file from an LTQ FT instrument).
– Possibility to convert not only a single source file into a single mzML file, but also sets of files into a single mzML file. Example file: dta_example_1.1.0.mzML (folder of DTA files generated by Proteios Software Environment (30) and converted into a single mzML file).
– Possibility to convert an arbitrary common peak list file into mzML format. Example file: plgs_example_1.1.0.mzML (generated by conversion of a Protein Lynx Global SERVER (31) XML peak list which was generated by Proteios).
– Provision of full support for different data and metadata from different spectrum types, such as the neutral loss spectrum, which is achieved by neutral loss scans. Example file: neutral_loss_example_1.1.0.mzML (hand crafted).
– Another important spectrum type is the precursor spectrum. Spectra acquired by precursor scans should be supported. Example file: precursor_spectrum_example_1.1.0.mzML (hand crafted).
– Storage of quantitation-related data and metadata should be possible. All important modes of scanning and acquiring data, e.g., Selected Reaction Monitoring (SRM), Total Ion Current (TIC), and Selected Ion Monitoring (SIM), should be supported. Example file: MRM_example_1.1.0.mzML (hand crafted).
– Another type of important instrument metadata is the information about the used detector type. It should also be possible to support all the common and different types of detectors like photodiode array (PDA) detectors, position and time-resolved

ion collector (PATRIC) detectors, Faraday cups (or cages), electron multipliers (EMs) or microchannel plate (MCP) detectors. Example file: The "PDA example file" (hand crafted).

– Finally, encoding the same information in mzML as in mzData and mzXML should be possible. Three example files containing the same information in mzML, mzData, and mzXML have been uploaded on the mzML web page to demonstrate this: tiny1.mzML1.1.0.mzML, tiny1.mzData1.05.xml, and tiny1.mzXML3.0.mzXML.

3. The PSI-MS Controlled Vocabulary

A "controlled vocabulary" generally contains predefined terms to avoid spelling or case ambiguities. The PSI controlled vocabularies are hierarchies of controlled terms ("ontologies") having for example "is_a" or "has_a" relationships to one or many "parent terms." Each term has a unique accession number and can have a value (e.g., MS:1000031, "instrument model") and a unit for this value (e.g., MS:1001117, "theoretical mass", unit = dalton). In an mzML file, <cvParam> elements are used to describe further details of a modeled object. Thus, most of the data concerning a mass spectrometry experiment are annotated using controlled vocabulary terms, e.g.,: <cvParam cvRef="MS" accession="MS:1000285" name="total ion current" value="16675500"/>, stating the sum of all the separate ion currents carried by the ions of different m/z contributing to a complete mass spectrum or to a specified m/z range of a mass spectrum. In the controlled vocabulary hierarchy, this term "is_a" "spectrum attribute," which itself "is_a" "object attribute" and has a "part_of" relationship to "spectrum." The position within the hierarchy can be used to check the correct use of controlled vocabulary terms (important for file validation). If a new important term should be added to the PSI-MS controlled vocabulary, the PSI-PI workgroup must be informed (see Note 4).

The following ontologies or controlled vocabularies may also be suitable or required for some elements of mzML:

- Unit Ontology (http://www.obofoundry.org/cgi-bin/detail.cgi?id=unit)
- ChEBI (http://www.ebi.ac.uk/chebi/)
- OBI (Ontology of Biological Investigations – http://obi.sourceforge.net/)
- PSI Protein modifications CV (http://psidev.sourceforge.net/mod/data/PSI-MOD.obo)
- Unimod modifications database (http://www.unimod.org/obo/unimod.obo)

4. The mzML Schema (Release 1.1.0)

4.1. The XML Backbone in mzML

The schema of mzML is a well-defined order of nested XML-elements and subelements with different appropriate attributes that may be optional or mandatory. They are designed to contain all the information for a single MS run, including metadata about the spectra and all the spectra themselves. Extensible Markup Language (XML) is a markup language for the description of hierarchically structured data in a textual data format. It was designed to facilitate simplicity, generality, and usability for electronic data exchange between computer systems, especially via the internet. The so-called XML specification defines a meta-language for the description of application-specific languages by different constraints. These constraints are described by the so-called schema languages, such as DTD or XML schema. There is always a schema language document to describe an application-specific language. In XML schema, these documents are called "XML schema documents" (XSD) and are tagged with the extension ".xsd". An XSD file defines a set of rules a document must conform in order to be considered as "valid" against that schema. It constrains, which set of elements may be used in a document, which attributes containing which data types (e.g., xsd:string or xsd:integer) may be applied to them, the order in which they must appear and the allowable parent–child relationships. Hence, the mzML syntax is mainly defined by its XSD file – mzML1.1.0.xsd (see Note 1).

4.2. Parameter- and List-Elements: mzML's Key Components

In this subsection, we discuss element structures in mzML that recur repeatedly forming its syntactical backbone. Hence, we want to term them mzML's key components.

4.2.1. The Parameter Elements

The three parameter-elements <cvParam>, <userParam>, and <referenceableParamGroup> are key components in mzML, hence they represent the key concepts of stability (ensured by controlled "CV terms"), generic structure (ensured by free "user terms"), and space efficiency (ensured by the possibility to refer to repetitive groups of parameter elements). All of them have been designed to store additional data or comments. However, it is intended to principally use CV terms to describe this kind of data – corresponding to one of mzML's design principles (stability). Hence, the most important parameter-element is <cvParam> (see Fig. 1), which is designed to contain comments or ancillary data by referring to a controlled vocabulary term and specifying its "value" attribute. There are seven attributes for <cvParam>. Three of them ("accession", "cvRef", and "name") are mandatory, and they may be considered as key attributes for its functionality. "accession" holds the accession number of the controlled

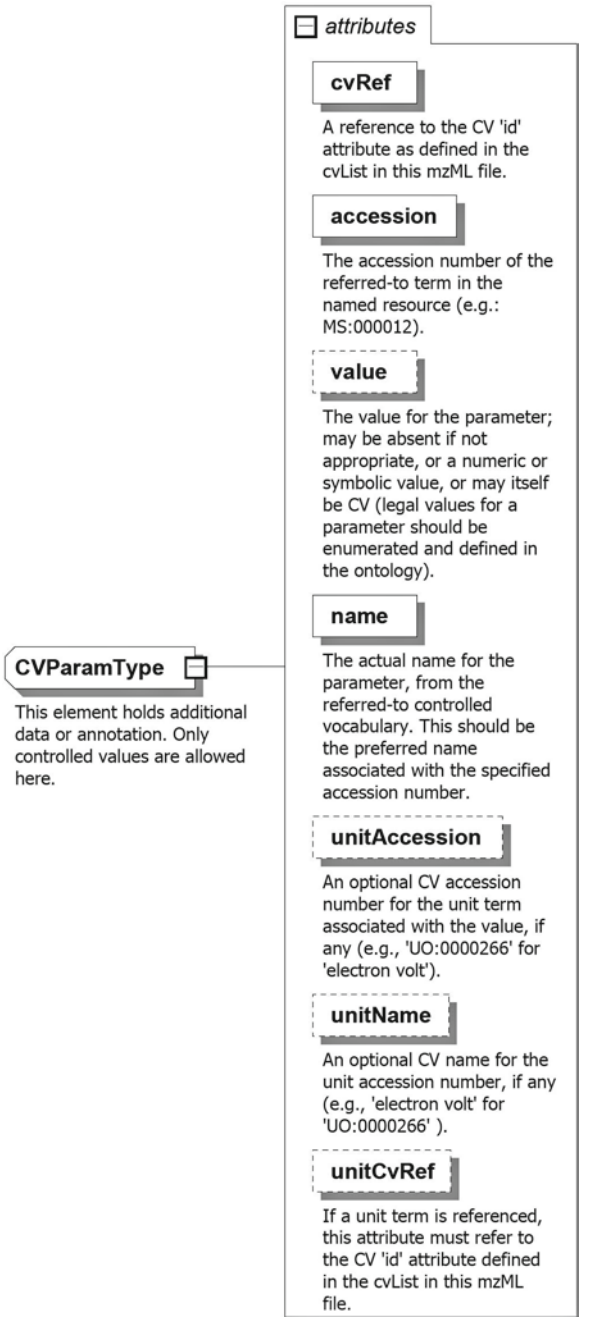


Fig. 1. The <cvParam> element.

term (e.g., “MS:000020” for “scanning method”), “cvRef” points to the “id” attribute of one <cv> element in <cvList> defining the corresponding controlled vocabulary, and “name” states the current name of the CV term (note that the accession number is the important attribute and refers to an abstract

concept, whose name may change (e.g., "multiple reaction monitoring" to "selected reaction monitoring"); so the name attribute is just used for human readability). Four other attributes ("unitAccession", "unitCvRef", "unitName", and "value") are optional. The attribute "value" is intended to hold a potential value of the parameter. All legal values for a parameter are defined in the ontology (e.g., "xref: value-type:xsd\:float" is the ontology entry for "scan start time"). In case of specifying a parameter value, the other optional attributes may hold information about the unit of this parameter. For example, "unitAccession" may hold a controlled vocabulary accession number (e.g., "MS:0000040" for "m/z" or "UO:0000021" for "gram"); "unitName" then contains a – human-readable – name for the unit accession number (e.g., "gram" for "UO:0000021" or "m/z" for "MS:0000040"); and "unitCvRef" then points to a controlled vocabulary's "id" in <cvList>. It has to be stressed that "unitCvRef" loses its "optional" status, if a unit is specified.

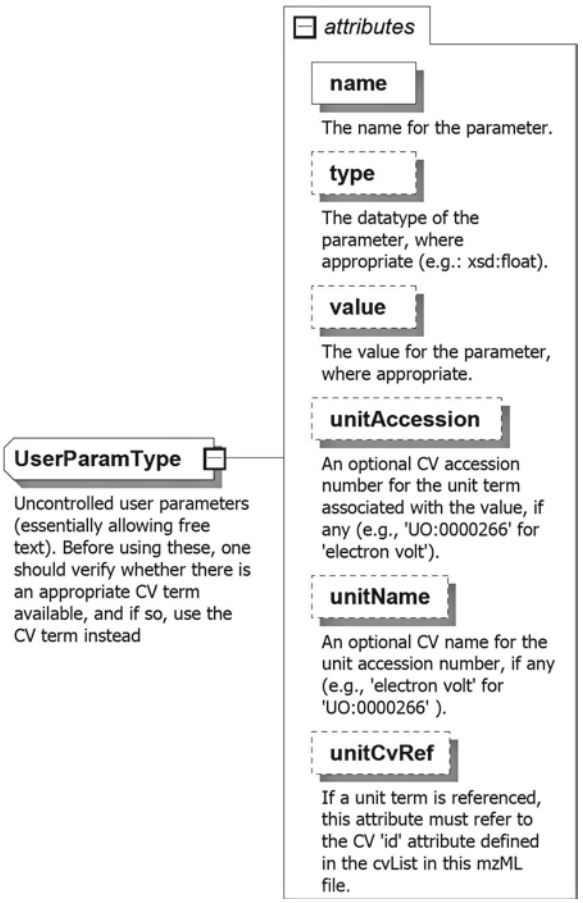


Fig. 2. The <userParam> element.

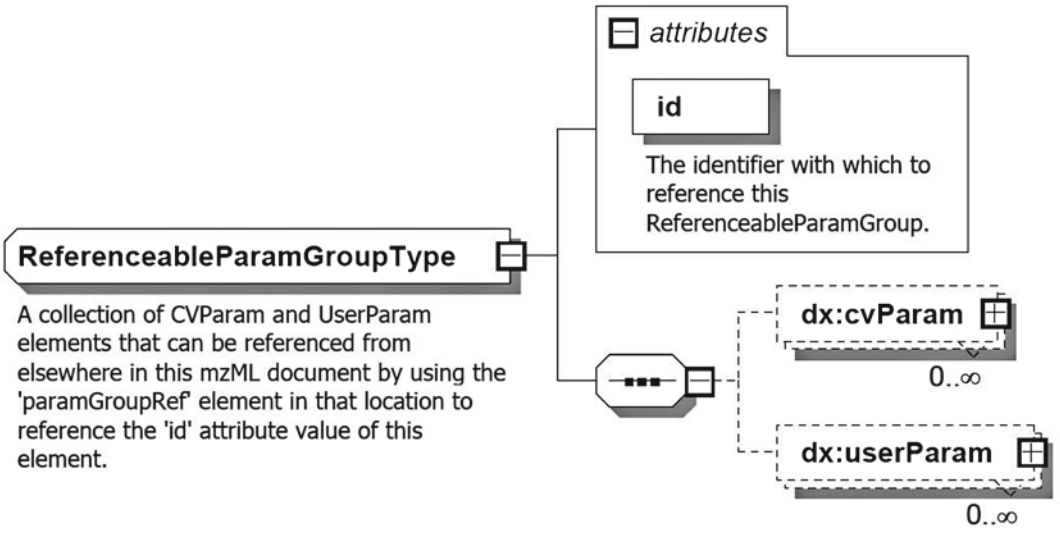


Fig. 3. The <referenceableParamGroupRef> element.

The second parameter-element is <userParam> (see Fig. 2). It provides an opportunity to describe parameters without corresponding controlled vocabulary term. If there is an evident need for an additional controlled vocabulary term, PSI should be informed (see Note 4), so it is recommended to use this element extremely reservedly in exceptional cases only (e.g., when it is not possible to wait for a controlled vocabulary update). A spreading use of uncontrolled terms would lead to uncontrolled dialects and contradict the fundamentals of mzML's design philosophy. Hence, it should be used in cases of emergency only. Besides this general remarks, <userParam> is very comparable to <cvParam>. They share almost the same attributes except for "cvRef" and "accession" of cause. Its only attribute not occurring in <cvParam> is "type", which is optional and is intended to contain information about the data type of the uncontrolled parameter (e.g., xsd:integer or xsd:date).

The last parameter-element is <referenceableParamGroupRef> (see Fig. 3). It has been designed to point to a reusable container for a set of <cvParam> and/or <userParam> elements. This is a comfortable feature to handle frequently repetitive sets of parameters and keep the file simpler. Its sole attribute "ref" is intended to point to the "id" attribute in a <referenceableParamGroup> element, which is that reusable container of <cvParam> and <userParam> elements. As a result of this relationship, referenceable parameter groups can be referenced from elsewhere in an mzML document.

4.2.2. The List Elements

The other class of key building components is the class of list elements. All list structures share the same basic concept. It can be formulated as follows: The whole list structure is delimited by a

Table 1
List elements in mzML

Root element	"Count" attribute	Other attributes	Listed subelement	Number of listed subelements	Other subelements
<binaryDataArrayList>	Yes	No	<binaryDataArray>	2 – ∞	No
<chromatogramList>	Yes	Yes	<chromatogram>	1 – ∞	No
<componentList>	Yes	No	<source>, <analyzer>, <detector>	1 – ∞ each	No
<cvList>	Yes	No	<cv>	1 – ∞	No
<dataProcessingList>	Yes	No	<dataProcessing>	1 – ∞	No
<instrumentConfigurationList>	Yes	No	<instrumentConfiguration>	1 – ∞	No
<precursorList>	Yes	No	<precursor>	1 – ∞	No
<productList>	Yes	No	<product>	1 – ∞	No
<referenceableParamGroupList>	Yes	No	<referenceableParamGroup>	1 – ∞	No
<sampleList>	Yes	No	<sample>	1 – ∞	No
<scanList>	Yes	No	<scan>	1 – ∞	Yes
<scanSettingsList>	Yes	No	<scanSettings>	1 – ∞	No
<scanWindowList>	Yes	No	<scanWindow>	1 – ∞	No
<selectedIonList>	Yes	No	<selectedIon>	1 – ∞	No
<softwareList>	Yes	No	<software>	1 – ∞	No
<sourceFileList>	Yes	No	<sourceFile>	1 – ∞	No
<sourceFileRefList>	Yes	No	<sourceFileRef>	0 – ∞	No
<spectrumList>	Yes	Yes	<spectrum>	0 – ∞	No
<targetList>	Yes	No	<target>	1 – ∞	No

This comparison of all list elements illustrates, that all of them contain a "count" attribute (see column "'count' attribute") and usually one listed subelement (see column "listed subelement"), whose name is derived from the root element and whose cardinality is normally "1 – ∞" (see column "number of listed subelements"). They contain usually no other attributes besides the "count" attribute (see column "other attributes") and no other subelements besides the listed one (see column "other subelements")

root element. Its name is usually tagged with "List", e.g., <softwareList>. The root element contains at least the mandatory "count" attribute, which is indicating the number of listed subelements (thus, it is an integer attribute). Furthermore, the root element includes always a sequence of listed subelements, all of which are named with the same name like the root, but without

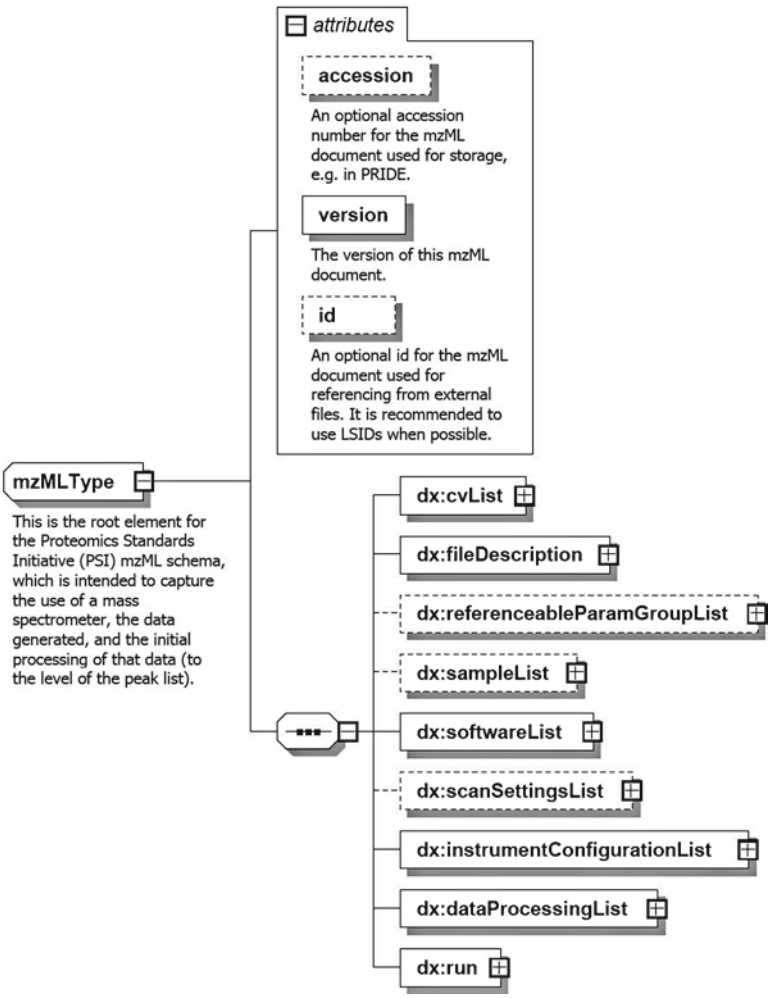


Fig. 4. The <mzML> element.

"List", e.g., <software>. There must be at least one of them. Table 1 gives an overview of all list structures in mzML and demonstrates their common basic concept.

4.3. Top Level Elements

4.3.1. The <mzML>Element

This is the root element of each mzML file (see Fig. 4). Thus, it has to contain all the information mzML is intended to store. It is designed to serve as a container for its top level elements. First of all, there is an element for storing information about all the controlled vocabularies used at different positions through the entire file (<cvList>, see Fig. 5). The second element, the <fileDescription> (see Fig. 6) element, stores information about the kind of spectra that the file contains. As mentioned above, the optional <referenceableParamGroupList> (see Fig. 7) element is intended to hold a list of referenceable parameters used in the file.

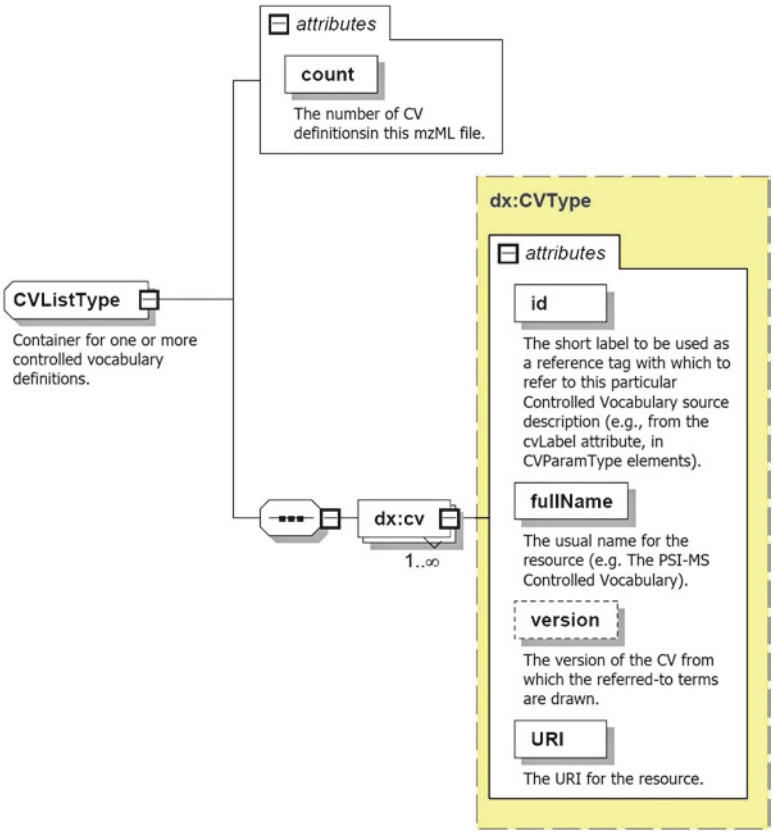


Fig. 5. The <cvList> element. This list element lists the controlled vocabularies used in this document and refers to them via its sole subelement <cv>. For each controlled vocabulary used in this document, a **<cv>** subelement has to be included to the list and its attributes specified. It has four attributes: "URI", "fullName", "id", and "version". All of them besides "version" are mandatory. By specifying the URI of the source (via the "URI" attribute), its ordinary name (via the "fullName" attribute) and an ID for it (via the "id" attribute) a particular controlled vocabulary is defined, and it is possible to refer to it via the reference tag "id" from anywhere in the file.

The next element is <sampleList>. It has been designed for storing information about the samples used to generate the data set. There is also an element for storing information about the software used in the referenced experiment (<softwareList>). Another metadata element is <scanSettingsList> (see Fig. 8). It is intended to store information about the scanner settings prior to the data acquisition process. It is followed by the <instrumentConfigurationList> element (see Fig. 9), which has been designed to store information about the instrument configurations. The penultimate element (<dataProcessingList>, see Fig. 10) is a container for the information about data processing applied to the stored data. Finally, there is an element for storing information about each

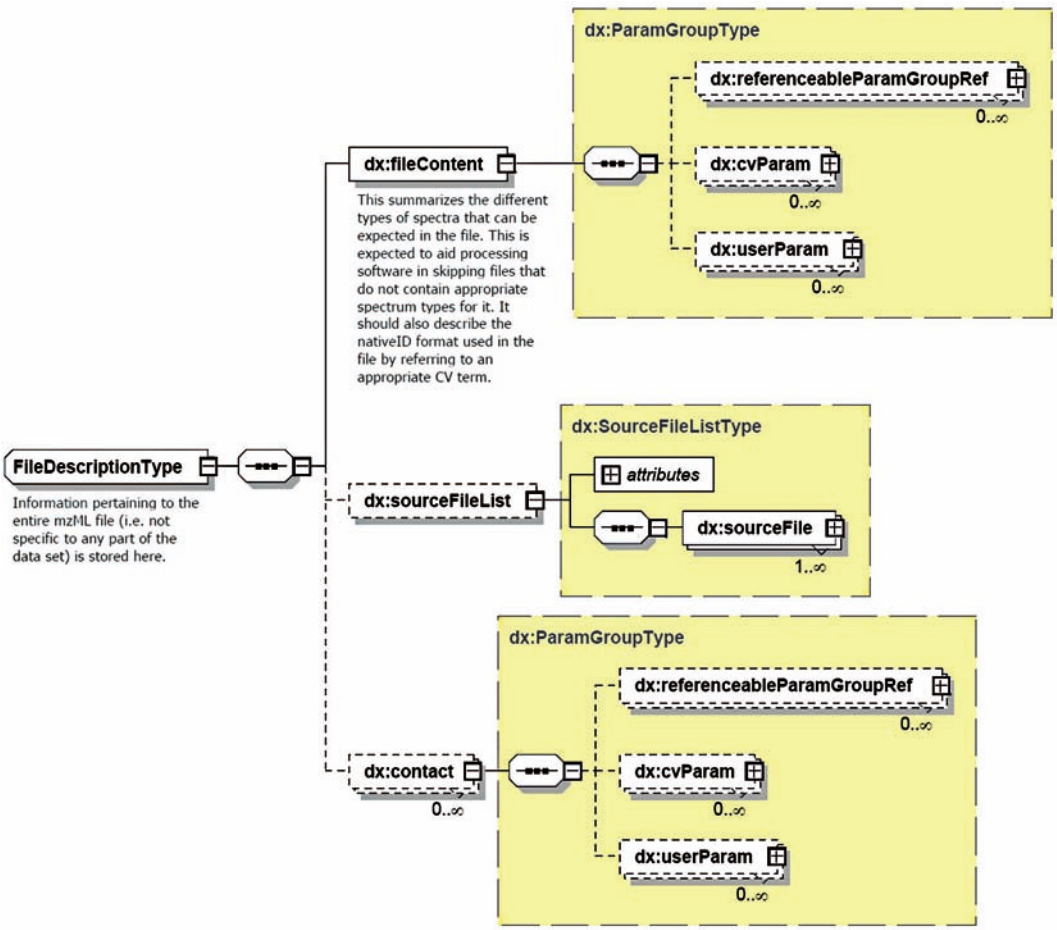


Fig. 6. The <fileDescription> element. Here, different information according to the entire mzML file is stored. <fileDescription> has three subelements. The mandatory subelement <fileContent> for instance has to be used to list and describe the different spectra the mzML file contains via its parameter subelements. The second subelement in <fileDescription>, <sourceFileList>, is an optional list element and should be used to list all the source files the mzML file was generated, converted or derived from. <sourceFileList>'s third and last subelement, <contact>, is also optional and is designed to store contact information in its parameter subelements.

scan performed on an instrument (<run>, see Figs. 11–13). In its highly branched subtree structure among other things the recorded spectra are stored. Besides its subelements, the <mzML> element contains three attributes, namely, "accession", "id", and "version". "accession" is an optional attribute used, e.g., for data sets from public repositories like PRIDE (32). The second attribute "id" is also optional. It is intended for referencing this document from external files. However, "version", is mandatory, so a version number for an mzML document has always to be specified.

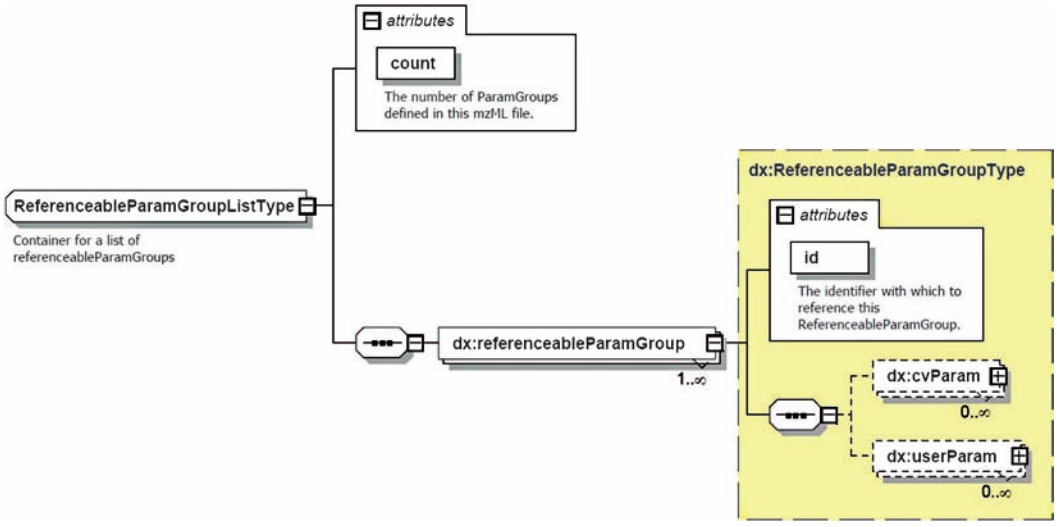


Fig. 7. The <referenceableParamGroupList> element. The optional <mzML> subelement <referenceableParamGroupList> has been designed to list user-defined sets of referenceable controlled vocabulary elements and/or uncontrolled vocabulary elements. For each set, the <referenceableParamGroup> subelement has to be listed. Its parameter subelements, <cvParam> and <userParam>, are both optional (a group may be declared without being implemented).

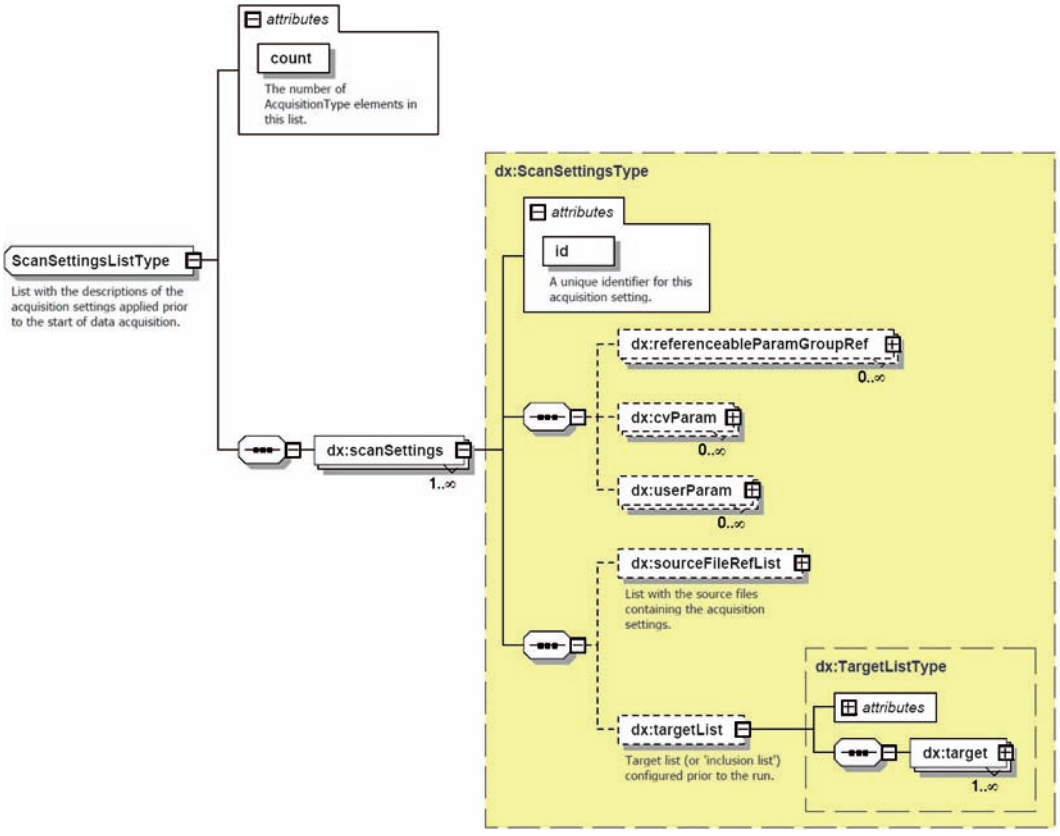


Fig. 8. The <scanSettingsList> element. This is an optional list element for descriptions of the scan settings that have been set for the acquisition of the data archived in this file. Its listed element <scanSettings> is thought to contain the

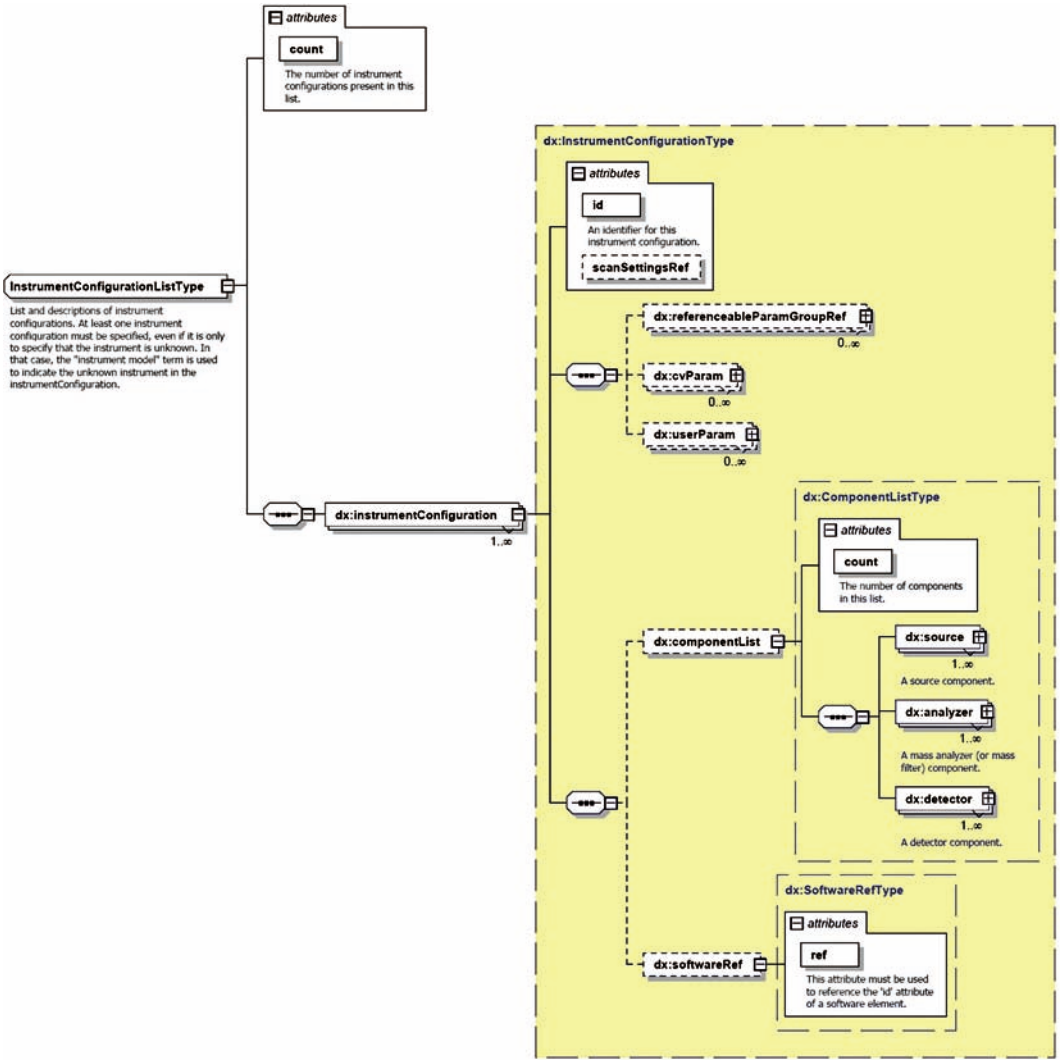


Fig. 9. The <instrumentConfigurationList> element. This list element is mandatory. Its listed element, <instrumentConfiguration>, is designed to describe a particular configuration of a mass spectrometer device (at least one). It may also be specified that the instrument is unknown, if necessary. In this case, the controlled term "instrument model" (its controlled vocabulary id: MS:1000031) should be used via the <cvParam> subelement of <instrumentConfiguration> to indicate that the instrument is unknown (value="unknown"). Besides its parameter subelements, <instrumentConfiguration> contains an optional nested list structure (<componentList> for important component descriptions) and an additional optional subelement (<softwareRef> to reference to a previously defined software element), which may occur only once.

Fig. 8. (continued) description of the instrument settings before the start of the run in its parameter subelements. Alternatively, these scan settings may be described by two list structures which are nested in the <scanSettings> element (additionally to its parameter subelements): on the one hand, <sourceFileRefList> is intended to list source files with the corresponding acquisition settings and on the other hand, <targetList> is designed to list the targets that have been used for this run.

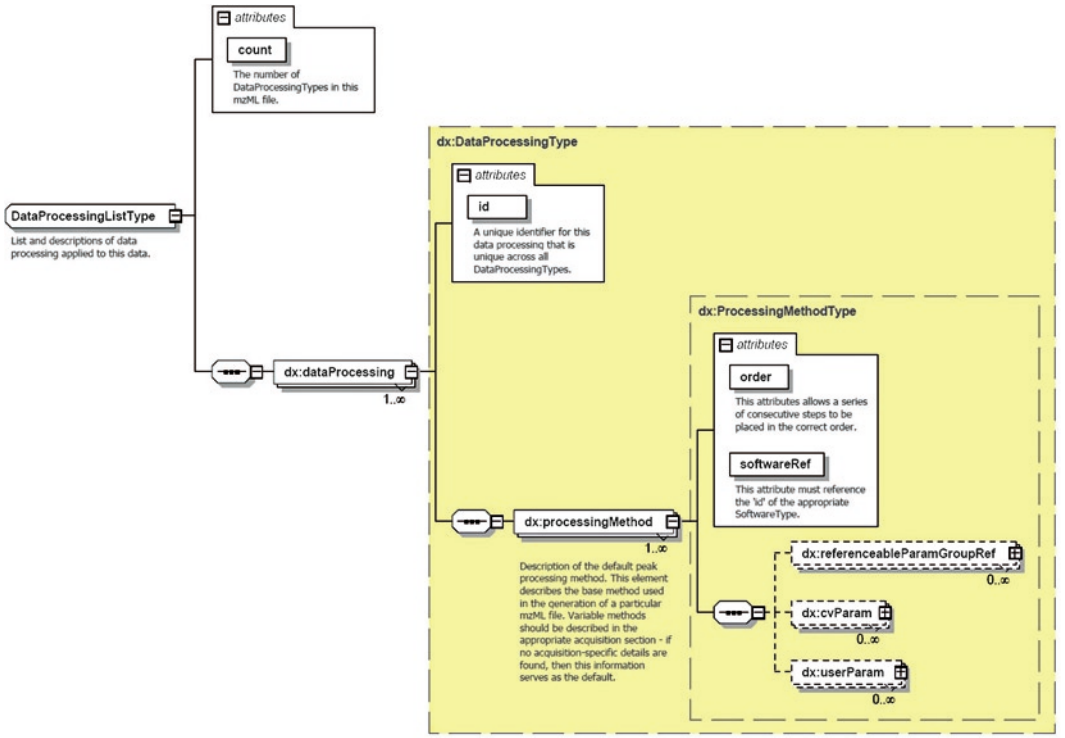


Fig. 10. The <dataProcessingList> element. The mandatory list element <dataProcessingList> lists and describes the data processing procedures, which have been applied to the data stored in this mzML file. Its list element is <dataProcessing>. It describes the way in which particular software (that is listed in <softwareList>) was used, and it holds a sequence of its sole and mandatory subelement, the <processingMethod> element, which is used to describe the default processing method. The settings described in a <processingMethod> element are the default settings, except they are supplemented and potentially overwritten by settings of a <scanSettingsList>-element (where data processing methods are described, which vary between scans).

4.4. Indexing in mzML

Due to the design principle of preserving stability, one of mzXML's features, a mechanism for a random access index was also built into mzML. The advantage of this feature is a speedup of data access, e.g., when a data processing or data management software needs to find an arbitrary spectrum. With the aid of such an index, it is possible to perform more efficient searches for that spectrum in a file rather than by reading it sequentially. However, some designers were convinced that disadvantages regarding a broken index prevail over the advantages of this approach. The compromise is that mzML provides both ways. On the one hand, it has been designed as a stand-alone data format, which does not contain any index. On the other hand, an mzML document can be enclosed in a wrapper schema that provides an index (see Fig. 14). Both ways of usage are separated structurally, since there is a sepa-

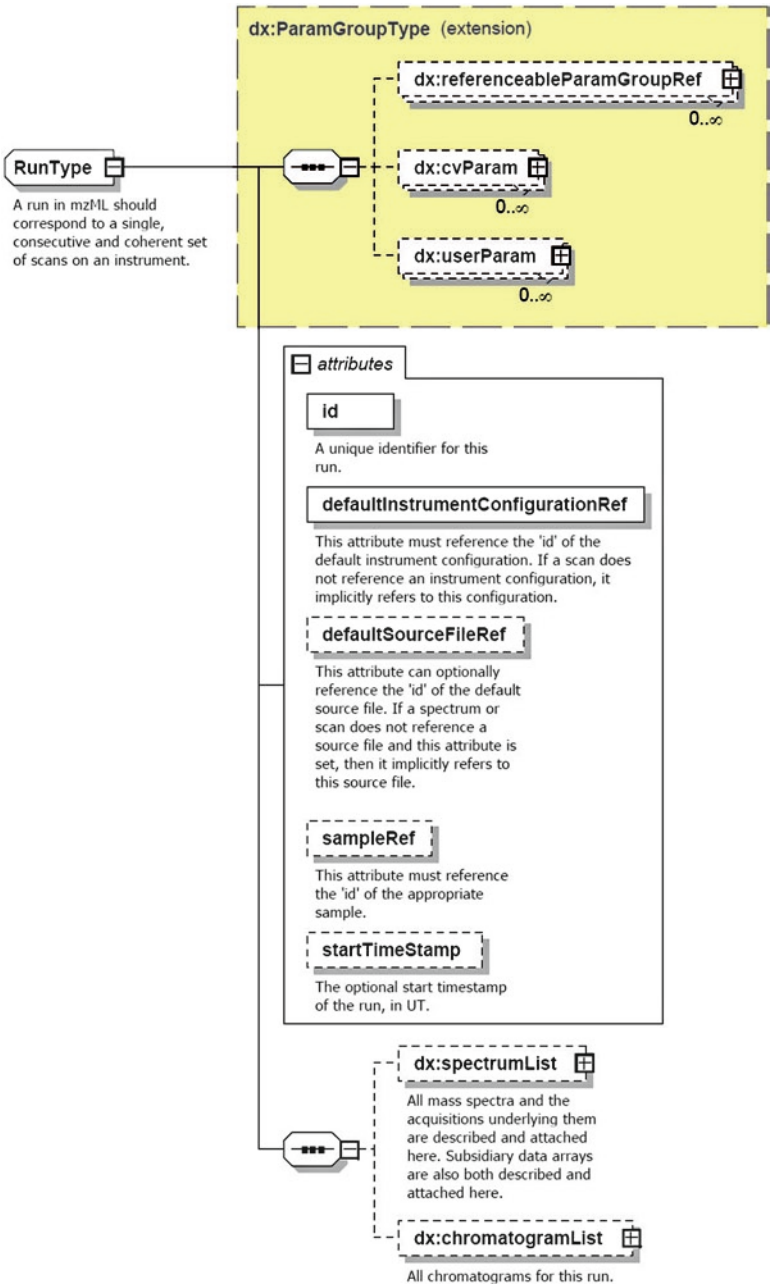


Fig. 11. The <run> element. The mandatory <run> element has been designed to store the information about a particular mass spectrometer run (a single, successive, and coherent set of scans performed by a particular instrument). Since there should be one mzML file per run, there is only one <run> element in each mzML file. Its attributes refer to corresponding elements, which contain the corresponding metadata recorded for this run. An optional sequence of param elements has been designed for further metadata description. Finally, there are two extremely nested list structures, which are designed to describe and store the recorded binary data in the <run> element: <spectrumList> (see Fig. 12) for successively recorded spectra (at least one) and <chromatogramList> (see Fig. 13) for all chromatograms for this particular run.

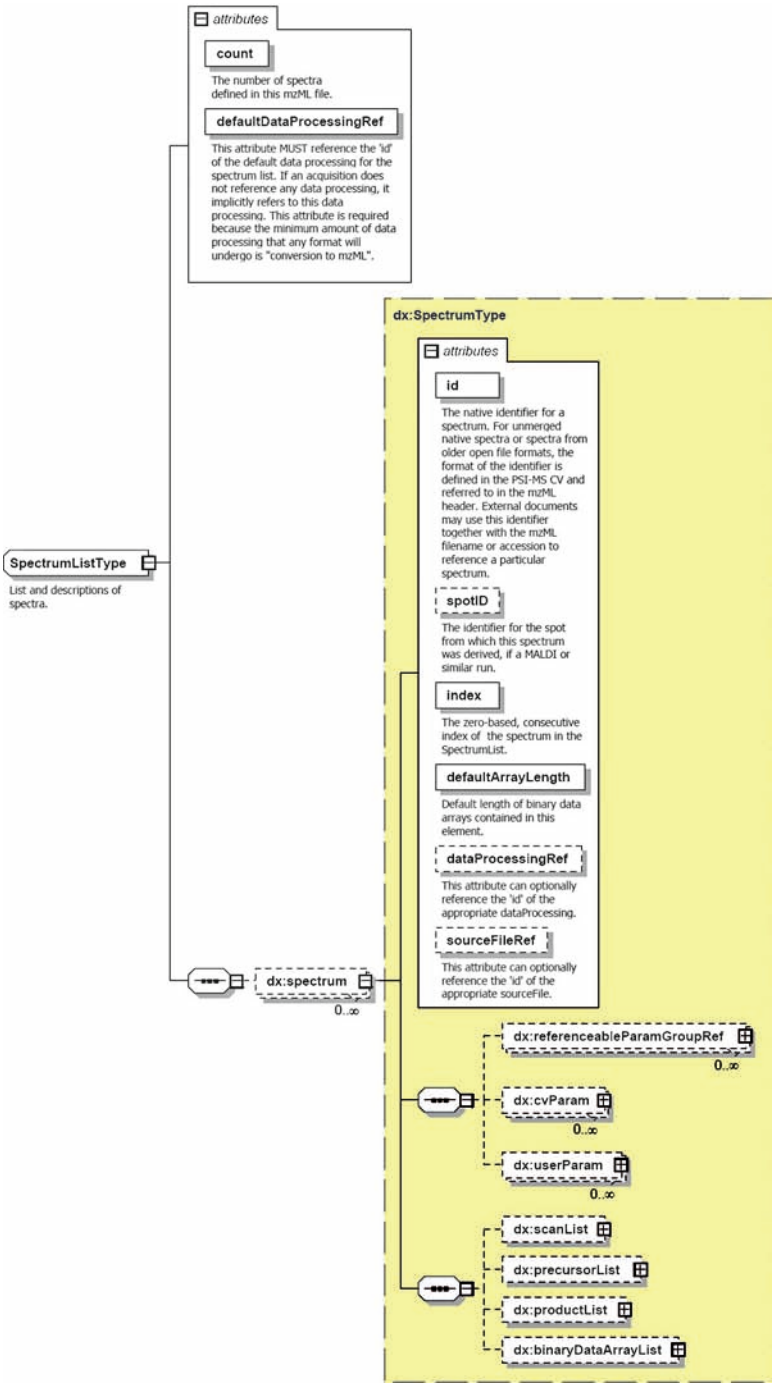


Fig. 12. The <spectrumList> element. This list structure stores all mass spectra in a binary data list and all the underlying data (list of scans, list of precursors and list of products) in convenient list elements. A sequence of param elements makes the <spectrum> element quite flexible.

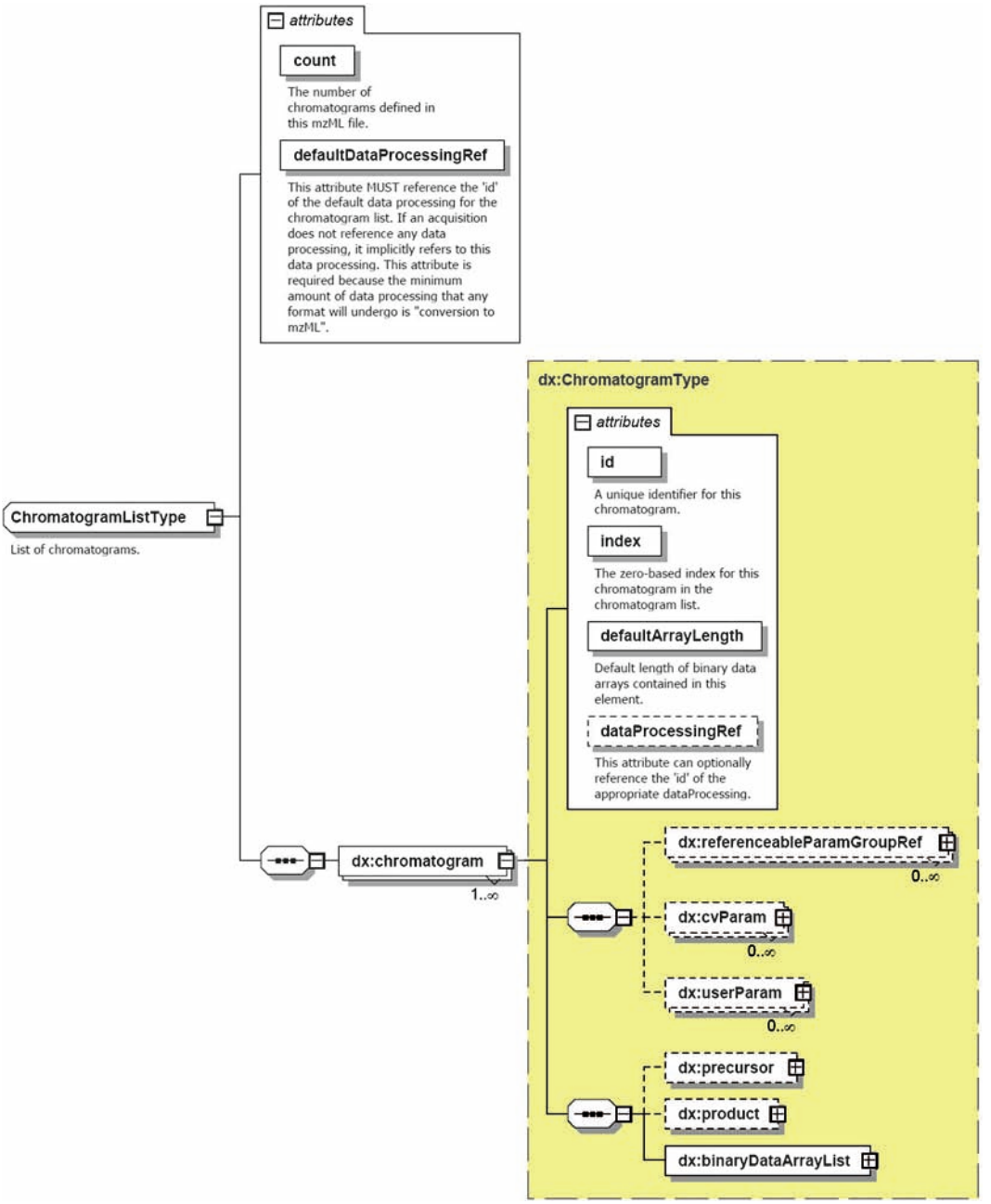


Fig. 13. The <chromatogramList> element. List structure that is designed to store and describe all chromatograms for a particular run. The recorded chromatograms are stored in a binary array list, and there is a list element for underlying data as well as an optional sequence of param elements in the <chromatogram> subelement.

rate XSD file for index support (mzML1.1.0_idx.xsd, (16)). So a particular mzML file may contain an mzML document with or without index support. Software supporting mzML must be designed to support both ways.

Fig. 14. An indexed example file. View on the top level elements of an indexed mzML example (tiny.pwiz.1.1.mzML, which can be found on the mzML web page (16)) with index-relevant parts in the two boxes. This example shows an ordinary mzML document, which is enclosed by the index container element <indexedmzML>. The indices by itself (offsets in bytes for random data access for the entity the index points to) are listed in the list element <indexList>. Besides this, there is an element <indexListOffset> containing a file pointer offset (in bytes) for the <indexList> element and an element (<fileChecksum>) holding an SHA-1 checksum from the beginning of the file to the end of the <fileChecksum> open tag in the container structure.

5. Semantic Validation

As mentioned above, experience with mzData revealed a serious disadvantage of the controlled vocabulary approach, namely, the propagation of several different dialects of the file format. This phenomenon was caused by inconsistently used controlled vocabulary terms and no universal mechanism to ensure that terms were used correctly. The same information was often encoded in slightly different ways. Finally, mzData has taught its designers the lesson that an uninhibited spread of such different dialects can cause serious difficulties in developing and maintaining software. To address this problem for mzML, a semantic validator has been released together with the data format. This

semantic validator is available as a web application, to which an unvalidated file can be uploaded, or as a stand-alone tool, which can be downloaded to validate local files offline. However, the semantic validator has to enforce a set of rules, which has been defined to ensure the semantic validity of an mzML document. It has to be stressed that an ordinary XML schema validation checking the syntax only is not able to decide whether the controlled terms are used correctly. Instead compliance with semantic rules has to be checked. These semantic rules are encoded in one or more (e.g., for different compliance levels) mapping files. However, the semantic validator has to ensure that CV terms are used in the correct location in the document (defined via XPath notation in the mapping file) and the mandatory terms are present the correct number of times. Thus, only files which have been validated by the official semantic validator are considered to be valid. Therefore, this approach implies the demand on the community to be disciplined and to validate every single new or updated mzML document in this way (see Note 3). A spin-off of this approach is the necessity for updating also the semantic validator (by a mapping file update) every time the controlled vocabulary is adjusted to new technologies or new important information. However, its main advantage is that mzML has adopted schema stability from mzData without adopting its difficulties caused by spreading a number of data format dialects. Besides this, there is also an advantage concerning document compliance with publication guidelines. The semantic validator can be configured with different rules mapping files so that different levels of compliance can be defined. Thus, it is possible to configure it to check compliance with MIAPE-MS guidelines ((17)). Generally, an mzML file can be syntactically correct without being MIAPE conform. On the other hand, it can be MIAPE conform without fulfilling journal guidelines, e.g., Molecular and Cellular Proteomics guidelines ((33)). So this kind of flexibility may be helpful to generate custom-made files, which fulfill different levels of compliance. However, there is another advantage. Since it is possible to adjust the metadata regulations in the mapping file for different types of data, the file format can be fitted to encode other types of spectra than mass spectra (e.g., PDA spectra). In summary, it can be said that mzML's semantic validator seems to be a valuable improvement of mzData's controlled vocabulary approach.

6. Notes

1. The best way to study mzML's schema is to load its latest XSD file in a XML editor with XSD graphical browsing functionality (e.g., XMLSpy®, (34)). Then, one can easily browse

through its elements and attributes and read the corresponding annotations. However, this is a more pleasant way than reading the XSD code itself. There is also a very similar way to study the latest controlled vocabulary OBO file. Instead of opening it with a text editor one can use OBO-Edit, a free editor for OBO files, which contains graphical browsing functionality and may be downloaded on the mzML web page (16). Alternatively, one can browse through the latest controlled vocabulary file via the NCBO BioPortal web page for the mass spectrometry ontology (35) and an ordinary web browser.

2. It is always valuable to know which software provides mzML, what is it good for and which vendors are going to add mzML support to their next release. Therefore, one should regularly visit the mzML web page (16). There is a clearly structured and consistently updated table outlining software providing current or future mzML support.
3. There are two comfortable ways to validate a mzML file (13, 20). One can use the ProDaC (Proteomics Data Collection, (36)) online validator (http://eddie.thep.lu.se/prodac_validator/validator.pl) or the online validator that has been developed by Marc Sturm within the OpenMS project ((37), http://www-bs2.informatik.uni-tuebingen.de/services/OpenMS/mzML/).
4. If there is a need to add a new term to the PSI-MS controlled vocabulary, one should send a request to the PSI-PI workgroup by filling out an online form (http://www.psidev.info/index.php?q=node/440). The request will be sent to the PSI-PI workgroup mailing list and discussed within the group. Usually, decisions are made contemporary.

References

1. Web page of the Human Proteome Organisation (HUPO) http://www.hupo.org/
2. Web page of the Human Proteome Organisation Proteomics Standards Initiative (HUPO-PSI) http://www.psidev.info/
3. A HUPO PSI web page with information about mzData http://www.psidev.info/index.php?q=node/80
4. Web page of the Institute for Systems Biology in Seattle, WA, USA http://www.systemsbiology.org/
5. Web page of the Trans-proteomic Pipeline (TPP) software tool chain, which has been designed at the ISB http://tools.proteomecenter.org/wiki/index.php?title=Software:TPP
6. Web page of the HUPO PSI Spring Workshop 2006 in San Francisco, CA, USA http://psidev.sourceforge.net/meetings/2006–04/
7. Orchard S, Apweiler R, Barkovich R, Field D, Garavelli JS, Horn D et al (2006) Proteomics and Beyond: a report on the 3rd Annual Spring Workshop of the HUPO-PSI 21–23 April 2006, San Francisco, CA, USA. Proteomics 6:4439–4443
8. Deutsch E (2008) mzML: a single, unifying data format for mass spectrometer output. Proteomics 8:2776–2777
9. Deutsch EW (2010) Mass spectrometer output file format mzML. In: Hubbard SJJ, Andrew R (eds) Proteome bioinformatics,

1st edn. Springer Science+Business Media LLC, New York

10. Orchard S, Taylor CF, Jones P, Montechi-Palazzo L, Binz PA, Jones, AR et al (2007) Entering the implementation era: a report on the HUPO-PSI Fall workshop 25–27 September 2006, Washington DC, USA. Proteomics 7:337–339
11. Orchard S, Jones AR, Stephan C, Binz PA (2007) The HUPO pre-congress Proteomics Standards Initiative workshop. HUPO 5th annual World Congress. Long Beach, CA, USA 28 October-1 November 2006. Proteomics 7:1006–1008
12. Orchard S, Montechi-Palazzi L, Deutsch EW, Binz PA, Jones AR, Paton N et al (2007) Five years of progress in the Standardization of Proteomics Data 4th Annual Spring Workshop of the HUPO-Proteomics Standards Initiative April 23– 25, 2007 Ecole Nationale Superieure (ENS), Lyon, France. Proteomics 7:3436– 3440
13. Orchard S, Albar JP, Deutsch EW, Binz PA, Jones AR, Creasy D et al (2008) Annual spring meeting of the Proteomics Standards Initiative 23–25 April 2008, Toledo, Spain. Proteomics 8:4168–4172.
14. Deutsch E, Souda P, Montecchi-Palazzi L, Tasman J, Binz, PA, Hermjakob H, Martens L (2008) Design and implementations of the new Proteomics Standards Initiative's mass spectrometer output file standard format: mzML 1.0., In ASMS 2008, Denver, Colorado, USA
15. Eric W, Deutsch LM, Pierre-Alain B, Darren K, Matthew C, Marc S, Frederik L (2009) mzML: Mass Spectrometry Markup Language (mzML 1.1 specification document), PSI Mass Spectrometry Standadrds Working Group
16. mzML web page of the HUPO-PSI Mass Spectrometry Standards working group http://www.psidev.info/index.php?q=node/257.
17. Taylor CF, Paton NW, Lilley KS, Binz PA, Julian RK Jr, Jones AR et al (2007) The minimum information about a proteomics experiment (MIAPE). Nat Biotechnol 25:887–893
18. Orchard S, Hermjakob H, Taylor CF, Potthast F, Jones P, Zhu W et al (2005) Second proteomics standards initiative spring workshop. Expert Rev Proteomics 2:287–289
19. Orchard S, Hermjakob H, Taylor CF, Potthast F, Jones P, Zhu W et al (2005) Further steps in standardisation. Report of the second annual Proteomics Standards Initiative Spring Workshop (Siena, Italy 17-20th April 2005). Proteomics 5:3552–3555
20. Orchard S, Deutsch EW, Binz PA, Jones AR, Creasy D, Montechi-Palazzi L et al (2009) Annual spring meeting of the Proteomics Standards Initiative. Proteomics 9:4429–4432
21. Web page of SourceForge http://sourceforge.net/
22. Web page of Google Code http://code.google.com/intl/en/
23. Web page of the Protein Seperation Workgroup of the HUPO Proteomics Standards Initiative (PSI-PS) http://www.psidev.info/index.php?q=node/83
24. Web page of the Mass Spectrometry Working Group of the HUPO Proteomics Standards Initiative (PSI-MS) http://www.psidev.info/index.php?q=node/80
25. Web page of the Molecular Interaction Workgroup of the HUPO Proteomics Standards Initiative (PSI-MI) http://www.psidev.info/index.php?q=node/31
26. Web page of the Protein Modifications Workgroup of the HUPO Proteomics Standards Initiative (PSI-MOD) http://www.psidev.info/index.php?q=wiki/Protein_Modifications_Workgroup
27. Web page of the Proteomics Informatics Standards Group of HUPO Proteomics Standards Initiative (PSI-PI) http://www.psidev.info/index.php?q=node/40
28. TraML web page of the HUPO-PSI Mass Spectrometry Standards working group http://www.psidev.info/index.php?q=node/405
29. Web page of ProteoWizard http://proteowizard.sourceforge.net/index.html
30. Web page of Proteios Software Environment http://www.proteios.org/
31. Web page of ProteinLynx Global SERVER™ http://www.waters.com/waters/nav.htm?cid=513821&lset=1&locale=en_US
32. Web page of the PRoteomics IDEntifications database (PRIDE) http://www.ebi.ac.uk/pride/
33. Web page of the Molecular and Cellular Proteomics guidelines (MCP guidelines) http://www.mcponline.org/misc/ifora.dtl
34. Web page of XMLSpy® http://www.altova.com/xml-editor/
35. NCBO BioPortal web page for the mass spectrometry ontology http://stage.bioontology.org/visualize/39281/?id=MS%3A1000128
36. Web page of the Proteomics Data Collection (ProDaC) http://www.fp6-prodac.eu/
37. Web page of the open-source framework for mass spectrometry (OpenMS) http://www.fp6-prodac.eu/

Chapter 12

imzML: Imaging Mass Spectrometry Markup Language: A Common Data Format for Mass Spectrometry Imaging

Andreas Römpp, Thorsten Schramm, Alfons Hester, Ivo Klinkert, Jean-Pierre Both, Ron M.A. Heeren, Markus Stöckli, and Bernhard Spengler

Abstract

Imaging mass spectrometry is the method of scanning a sample of interest and generating an "image" of the intensity distribution of a specific analyte. The data sets consist of a large number of mass spectra which are usually acquired with identical settings. Existing data formats are not sufficient to describe an MS imaging experiment completely. The data format imzML was developed to allow the flexible and efficient exchange of MS imaging data between different instruments and data analysis software.

For this purpose, the MS imaging data is divided in two separate files. The mass spectral data is stored in a binary file to ensure efficient storage. All metadata (e.g., instrumental parameters, sample details) are stored in an XML file which is based on the standard data format mzML developed by HUPO-PSI. The original mzML controlled vocabulary was extended to include specific parameters of imaging mass spectrometry (such as x/y position and spatial resolution). The two files (XML and binary) are connected by offset values in the XML file and are unambiguously linked by a universally unique identifier. The resulting datasets are comparable in size to the raw data and the separate metadata file allows flexible handling of large datasets.

Several imaging MS software tools already support imzML. This allows choosing from a (growing) number of processing tools. One is no longer limited to proprietary software, but is able to use the processing software which is best suited for a specific question or application. On the other hand, measurements from different instruments can be compared within one software application using identical settings for data processing. All necessary information for evaluating and implementing imzML can be found at http://www.imzML.org.

1. Introduction

Imaging mass spectrometry is the method of scanning a sample of interest and generating an "image" of the intensity distribution of a specific analyte. The principle of this method is shown

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_12, © Springer Science+Business Media, LLC 2011

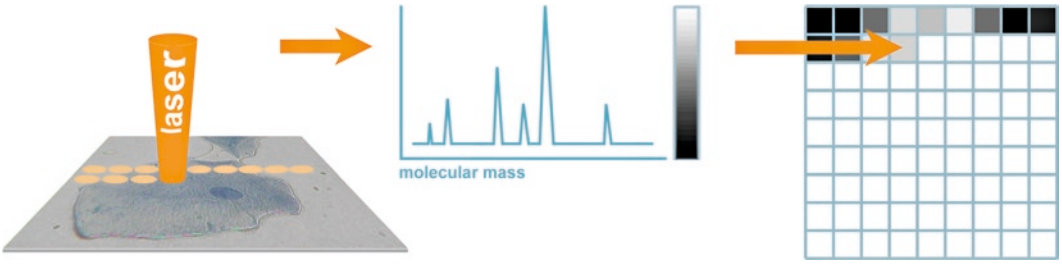


Fig. 1. Schematic process of scanning microprobe MALDI-MS. A desorption laser scans the surface of the target (e.g., a cell or tissue). The intensity of a selected peak in the resulting mass spectrum is transformed into a pixel of a grayscale image.

in Fig. 1. The application of MS imaging is rapidly growing with a constantly increasing number of different instrumental systems and software tools. An overview of methods and applications of mass spectrometry imaging has been recently published (1). This method results in a large number of spectra which are typically acquired with identical measurement parameters. The data format described in this chapter was developed within the EU-funded project COMPUTIS (2). The goal of this project was to develop new and improved technologies for molecular imaging mass spectrometry. An important task was the comparison of images generated by diverse types of mass spectrometers. Therefore a standard format for the exchange of MS imaging data was needed. Both the DICOM standard for in-vivo imaging data (3) and the mzML standard (4) by HUPO-PSI (5, 6) are not able to completely represent an imaging MS experiment. Therefore a standardized data format was developed to simplify the exchange of imaging MS data between different instrument and data analysis software. The following institutions were involved in the development of imzML: Justus Liebig University (JLU), Giessen, Germany; FOM Institute for Atomic and Molecular Physics (FOM), Amsterdam, The Netherlands; Commissariat à l'Énergie Atomique (CEA), Saclay, France; and Novartis Institutes for BioMedical Research (Novartis), Basel, Switzerland.

Several data formats for MS imaging utilize two separate files: a small file (ini or XML) for the metadata and a larger (binary) file for the mass spectral data (e.g., Biomap (7) and internal data formats at FOM and JLU). This structure proved to be very useful for flexible and fast handling of the imaging MS data and it was decided to follow this approach for the new data format. In order to keep as close as possible to existing formats, we decided that the (small) metadata file should be based on the mass spectrometry standard mzML developed by HUPO-PSI (8). A more detailed discussion on why mzML was not fully implemented and about the relation between the two data formats (mzML and imzML) is found in Note 1. A new controlled vocabulary was

compiled for imzML to include parameters that are specific for imaging experiments (see Subheading 2.1.2). All relevant information about imzML including specifications and example files can be found at http://www.imzML.org.

The following section describes the design philosophy of imzML. The data structure is discussed in more detail in Subheading 2. Properties and possibilities of imzML files are discussed in Subheading 4. Available software applications including an example for a file converter are presented in Subheading 5.

2. imzML Data Format

The fundamental goal while developing imzML was to design a data format for the efficient exchange of mass spectrometry imaging data. At the same time, the format should be easily interchangeable with mzML.

The main goals can be summarized as

1. Ensure complete description of imaging MS experiments
2. Minimize file size
3. Ensure fast and flexible data handling
4. Keep the (XML part of) imzML as close as possible to mzML

2.1. Data Structure

imzML consists of two separate files: one for the metadata and one for the mass spectral data. The metadata is saved in an XML file (*.imzML). The mass spectral data is saved in a binary file (*.ibd). A schematic representation of the imzML file structure is shown in Fig. 2. The connection between the two files is made via

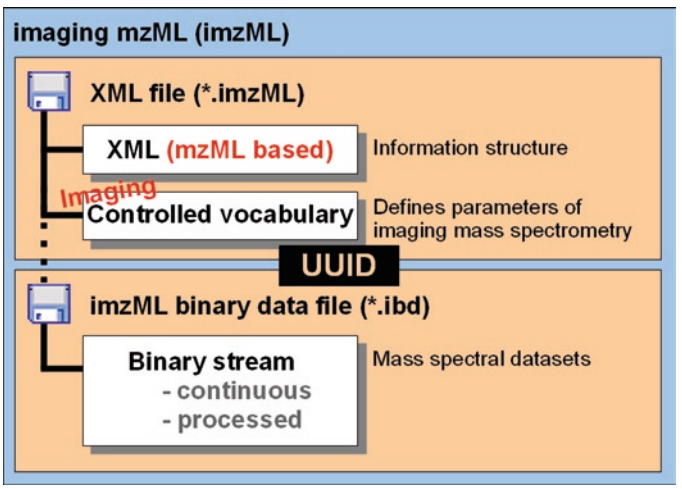


Fig. 2. Data structure of imzML.

links in the XML file which contain the offset positions of the mass spectra in the binary file. It is important to keep in mind that the information is only valid if both files are available. Therefore the user should be very careful when copying or moving those files; inaccurate file handling can result in data loss. It is recommended to keep both files in the same folder and to use the same names for the .imzML part and the .ibd part.

Corresponding XML and binary files contain the same universally unique identifier (UUID) (9) in order to link them unequivocally. The UUID is a controlled vocabulary entry in the <fileContent> tag of the XML file and is also stored at the beginning of the binary file. Comparing both UUIDs allows finding out if the two files belong to the same measurement/data set. More details on the implementation of UUIDs are discussed in Note 2.

2.1.1. XML

The XML file holds the metadata of a MS imaging experiment which is described by the mzML-based XML structure and the controlled vocabulary. The XML model of imzML is the same as for mzML (see the mzML version 1.1.0 documentation for further details) (8). The controlled vocabulary was extended in order to include additional parameters which are needed to describe an MS imaging experiment (see Subheading 2.1.2). Most of the changes in the XML part are related to cvParam mapping rules for the newly introduced parameters of the imaging controlled vocabulary. One of the most important changes compared to mzML is the function of the <binary> element which not contain base64-encoded binary data anymore. It stays empty, which is compatible to mzML 1.1.0. This results in predefined values for "encoded length" and "array length" of zero in the parent tags <spectrum> and <binaryDataArray>. The XML part of imzML passes mzML validators without errors (only warning messages for unknown cv entries are displayed). An example of XML code is given in Fig. 3. Modifications (compared to mzML) are printed bold and will be discussed in more detail in the following section.

2.1.2. Controlled Vocabulary

The controlled vocabulary is used to unequivocally describe the information in the XML file. The additional imzML CV terms are stored in an open biomedical ontology ((10, 11)) – the imagingMS.obo file (12). They complement the mass spectrometry parameters of the MS controlled vocabulary provided by HUPO-PSI (13) in order to allow a complete description of MS imaging experiments. An overview of CV entries concerning imaging-specific parameters and image properties is given in Table 1. These parameters include information about the image itself and acquisition parameters. For example: How many pixels does the image contain in the x and y dimension? Which position in

```
...
...
<scanSettingsList count="1">
  <scanSettings id="as1">
    <cvParam cvRef="IMS" accession="IMS:1000401" name="top down" value=""/>
    <cvParam cvRef="IMS" accession="IMS:1000480" name="horizontal line scan" value=""/>
    <cvParam cvRef="IMS" accession="IMS:1000491" name="linescan left right" value=""/>
    <cvParam cvRef="IMS" accession="IMS:1000413" name="flyback" value=""/>
    <cvParam cvRef="IMS" accession="IMS:1000042" name="max count of pixel x" value="2"/>
    <cvParam cvRef="IMS" accession="IMS:1000043" name="max count of pixel y" value="5"/>
  </scanSettings>
</scanSettingsList>
...
<binaryDataArrayList count="2">
  <binaryDataArray encodedLength="0">
    <cvParam cvRef="MS" accession="MS:1000576" name="no compression" value=""/>
    <cvParam cvRef="MS" accession="MS:1000514" name="m/z array" value="" unitCvRef="MS"
      unitAccession="MS:1000040" unitName="m/z"/>
    <cvParam cvRef="MS" accession="MS:1000523" name="64-bit float" value=""/>
    <cvParam cvRef="IMS" accession="IMS:1000103" name="external array length" value="1537"/>
    <cvParam cvRef="IMS" accession="IMS:1000104" name="external encoded length" value="12296"/>
    <cvParam cvRef="IMS" accession="IMS:1000102" name="external offset" value="16"/>
    <cvParam cvRef="IMS" accession="IMS:1000101" name="external data" value="true"/>
    <binary/>
  </binaryDataArray>
  <binaryDataArray encodedLength="0">
    <cvParam cvRef="MS" accession="MS:1000576" name="no compression" value=""/>
    <cvParam cvRef="MS" accession="MS:1000515" name="intensity array" value="" unitCvRef="MS"
      unitAccession="MS:1000131" unitName="number of counts"/>
    <cvParam cvRef="MS" accession="MS:1000523" name="64-bit float" value=""/>
    <cvParam cvRef="IMS" accession="IMS:1000103" name="external array length" value="1537"/>
    <cvParam cvRef="IMS" accession="IMS:1000104" name="external encoded length" value="12296"/>
    <cvParam cvRef="IMS" accession="IMS:1000102" name="external offset" value="12312"/>
    <cvParam cvRef="IMS" accession="IMS:1000101" name="external data" value="true"/>
    <binary/>
  </binaryDataArray>
</binaryDataArrayList>
...
```

Fig. 3. XML code new CV entries are printed bold.

the image belongs to which spectrum? In which pattern was the image scanned? Which Matrix was used in which concentration? It is also necessary to give information about the instrumentation used. In addition to the mass spectrometer, the ablation laser type and parameters have to be documented such as wavelength, energy, and impact angle. The sample stage also plays an essential role when generating an image: step size and position accuracy. The parameters concerning the scanning process are explained in more detail in Subheading "Image Orientation".

Additional entries were included, which are necessary for handling the external binary file. Four parameters were introduced

Table 1
Additional parameters of the controlled vocabulary

ibd file:	
External binary uri	Location as an URI where to find the ibd file.
ibd checksum:	
ibd MD5	MD5 (Message-Digest algorithm 5) is a cryptographic hash function with a 128-bit hash value used to check the integrity of files.
ibd SHA-1	SHA-1 (Secure Hash Algorithm-1) is a cryptographic hash function designed by the National Security Agency (NSA) and published by the NIST as a U. S. government standard. It is also used to verify file integrity.
ibd binary type:	
Continuous	Way of saving spectra in an imzML binary data file (ibd). The m/z values for all spectra are saved at the beginning of the ibd file. Then the spectral values follow.
Processed	Way of saving spectra in an imzML binary data file (ibd). Every spectrum is saved with its own m/z and intensity values.
ibd identification:	
Universally unique identifier	Universally unique identifier is unique throughout the world and allows to doubtlessly identify the ibd file.
ibd offset handle:	
External array length	Describes how many fields the external data array contains.
External data	Shows that there is no data in the <binary> section of the file but saved in an external file.
External encoded length	Describes the length of the written data stream in byte.
External offset	The position in byte where the data of the data array of a mass spectrum begins.
Image:	
Absolute position offset x	Describes the position at the x-axis of the upper left point of the image on the target.
Absolute position offset y	Describes the position at the y-axis of the upper left point of the image on the target.
Max count of pixels x	Maximum number of pixels of the x-axis of the image.
Max count of pixels y	Maximum number of pixels of the y-axis of the image.
Max dimension x	Maximum length of the image in x-axis.
Max dimension y	Maximum length of the image in y-axis.
Pixel size x	Describes the length in x-direction of the pixels.
Pixel size y	Describes the length in y-direction of the pixels.
Image shape	Describes the shape of the image.
Laser shot mode:	
Pixel mode	The laser keeps the position while firing at the same spot one or several times.
Raster mode	The laser is moved while continuously firing at the sample.
Stigmatic mode	The laser is moved around one point firing until moved to the next position (pixel).

(continued)

Table 1
(continued)

Spectrum position:	
Position x	Attribute to describe the position of a spectrum in the direction of the x-axis in the image.
Position y	Attribute to describe the position of a spectrum in the direction of the y-axis in the image.
Position z	Attribute to describe the position of a spectrum in the direction of the z-axis in the image.
Subimage position x	Describes the position of a subimage in the direction of the x-axis of the complete image.
Subimage position y	Describes the position of a subimage in the direction of the y-axis of the complete image.
Subimage position z	Describes the position of a subimage in the direction of the z-axis of the complete image.
Sample stage:	
Target material	Describes the material the target is made of.
Linescan sequence:	
Bottom up	The starting point is at the bottom of the sample and the sequence of the linescans is in up direction (parallel to the y-axis).
Top down	The starting point is at the top of the sample and the sequence of the linescans is in bottom direction (parallel to the y-axis).
Left right	The starting point is at the left of the sample and the sequence of the linescans is in right direction (parallel to the x-axis).
Right left	The starting point is at the right of the sample and the sequence of the linescans is in left direction. (parallel to the x-axis).
No direction	The linescans are performed randomly on the sample without any sequence.
Scan pattern:	
Meandering	The scanning happens in non-stop way. As soon as the end of the sample is reached, the scanning direction will be switched and the scanning is continued. There is no new positioning necessary.
Flyback	The scanning always happens in the same direction. As soon as the end of the sample is reached, the stage is positioned at the starting edge to begin the next run.
Random access	The scanning points are randomly chosen and do not follow a pattern.
Scan type:	
Horizontal linescan	The scanning line is a horizontal one.
Vertical linescan	The scanning line is a vertical one.
Linescan direction:	
Linescan bottom up	The starting point is at the bottom of the sample and the scanning happens in up direction (parallel to the y-axis).
Linescan left right	The starting point is at the left of the sample and the scanning happens in right direction (parallel to the x-axis).
Linescan right left	The starting point is at the right of the sample and the scanning happens in left direction. (parallel to the x-axis).
Linescan top down	The starting point is at the top of the sample and the scanning happens in bottom direction (parallel to the y-axis).

into the controlled vocabulary to describe the position and length of the data in the binary file. The "external data" parameter indicates that the mass spectral data is stored in a binary file. The parameter "external offset" holds the information at which byte in the binary file the data of the corresponding array starts. If one adds the value of the "external encoded length" to the value of the "external offset" the result has to be equal to the "external offset" of the following binary data array. The parameter "external encoded length" describes the byte length, which has to be read to obtain all the data of the array completely. The parameter "external array length" indicates the number of values of the array.

A separate checksum for the binary file was added in order to find out if the external file has been manipulated or corrupted. It can be either a SHA-1 (Secure Hash Algorithm) or a MD-5 (Message-Digest Algorithm) hash.

Ensuring the integrity and authenticity of digital data is of growing importance in today's information management systems especially for companies who have to operate in accordance with GLP regulations (14). Therefore imzML contains a mechanism to monitor (intentional or accidental) modifications of the data. This feature is based on asymmetric cryptosystems, also known as public key systems and hash functions. The checksum of the binary file is encrypted with a personal key. A public key is needed in order to verify the integrity of the checksum (and therefore the data). This feature (in its basic version) only requires three additional CV entries in the XML file and can thus be easily added to existing imzML files. This modular setup also allows for changing the used encryption procedure, e.g., when one method turns out not to be secure anymore. The encryption is an additional optional feature. This means the data is still readable without the public key, but authenticity cannot be tested in this case. This will protect data against loss (e.g., when one of the keys is lost) and also ensures readability of the data with (older) software that does not include the encryption feature.

2.1.2.1. Image Orientation

The pixel in the upper left corner of the MS image is defined as position 1/1 (Fig. 4). This way every application should generate images of identical orientation. This particular orientation was chosen because it is used for image acquisition in several MS imaging systems. The information of x and y position is part of the CV parameters in the <spectrum> tag of each mass spectrum (see also Fig. 3).

2.1.2.2. Scan Pattern

Information about the pattern and sequence in which an image was acquired is not necessary for generating an image from the imzML file (because x and y position are specified for each individual spectrum). But these parameters can be very important for

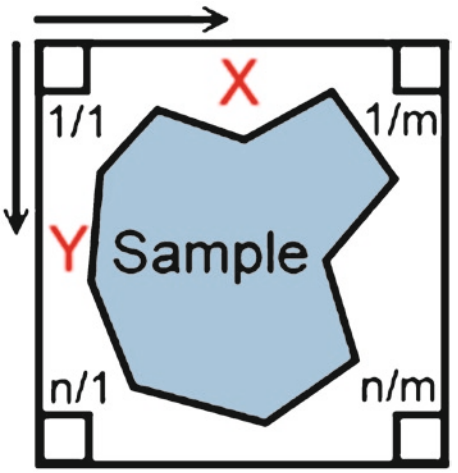


Fig. 4. sample orientation.

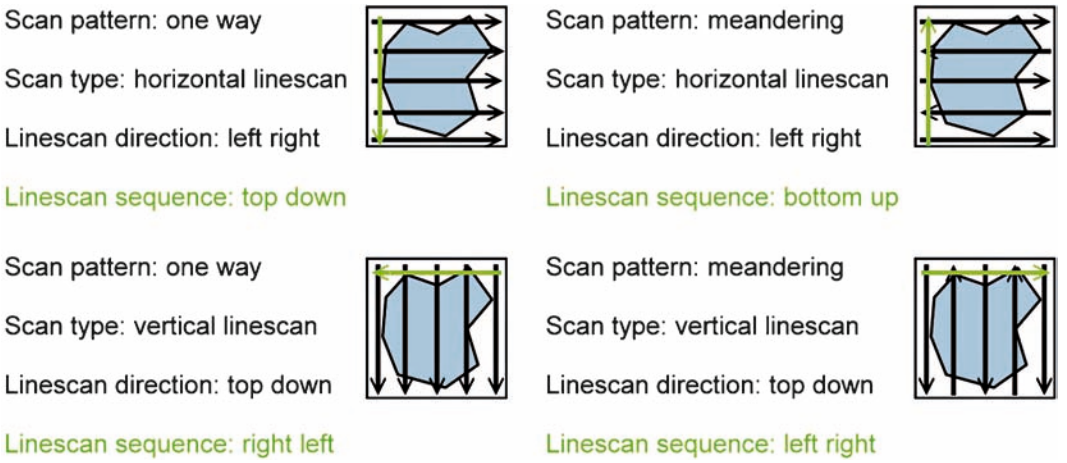


Fig. 5. Examples of scan patterns.

data analysis and interpretation and are therefore part of the imzML controlled vocabulary. The scan process is unambiguously described by four **parameters**. The different scan parameters are illustrated by examples in Fig. 5. The parameter **scan pattern** gives information if the sample was scanned in **fly-back** or **meandering** mode. Fly-back means that the linescans always occur in the same direction. Meandering indicates that the linescans occur in alternating direction. The **scan type** defines **horizontal** or **vertical linescans**. The **linescan direction** defines the direction of the (first) linescan(s). The **linescan sequence** specifies the chronological order of the linescans.

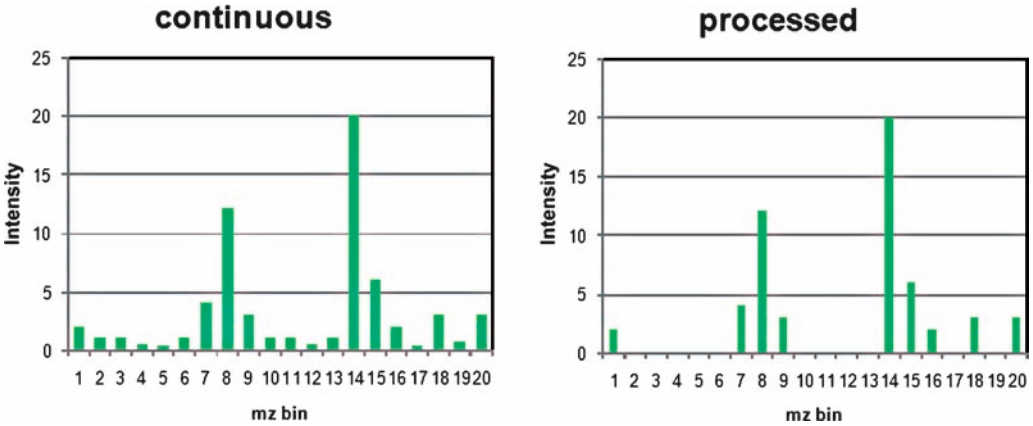


Fig. 6. Data types.

2.1.3. Binary Data File

The imaging binary data file (*.ibd) contains the mass spectral data of the MS imaging measurement. The first 16 bytes of the binary file are reserved for the UUID. This identifier is also saved in the corresponding XML file so that a correct assignment of ibd and XML file is possible even if the names of both files are different. In order to ensure efficient storage, two different **binary modes** are defined: **continuous** and **processed**. Schematic examples of these two data types are shown in Fig. 6. "Continuous" means that an intensity value is stored for each m/z bin even if there is no signal (resulting in an intensity of zero). As a result, the m/z axis is identical for all spectra of one image (if the mass range and bin size is not changed). Therefore it is sufficient to store the m/z array only once in the binary file (directly behind the UUID). For each of the spectra only the corresponding intensity values are stored. This structure can reduce the file size significantly (see Subheading 3).

On the other hand mass spectra are often processed before they are stored, e.g., for noise-reduction, peak-picking, deisotoping. This results in discontinuous and non-constant m/z arrays. In this case, the m/z array has to be stored for each spectrum separately. These data are stored as alternating m/z and intensity arrays in the binary file of imzML. The different storage types are illustrated in Fig. 7. More information on choosing the right binary mode is given in Note 3.

The second parameter influencing the file size is the **binary data type,** which is used to store the values of the spectra. The imzML binary format allows the storage of values in the following signed integer types: 8 bit, 16 bit, and 32 bit. The values can also be saved in floating point data types (IEEE 754): 32 bit (single precision) or 64 bit (double precision). The byte order of mass spectral data in imzML is little endian (see also Note 2). One value saved in the 32 bit integer type needs four bytes of disk space. The disk space needed by all values of all spectra determines the

Fig. 7. Binary data formats.

overall file size. Therefore the choice of the data types for intensity and m/z values directly influences the file size of the binary file. The data type is specified for each binary data array separately by CV parameters. More details can be found in Note 4.

3. imzML File Properties

The efficient data storage of the imzML format is demonstrated by an example file consisting of 7,000 spectra (Fig. 8). This file represents a measurement on a linear ion trap mass spectrometer (LTQ, Thermo Scientific GmbH, Bremen) (linear ion trap) of 50 by 35 pixel with four spectra (profile mode) acquired for each pixel. The original raw data in the proprietary LTQ format has a file size of 215 MB. The imzML files were saved with the following settings: m/z values were stored as 32 bit float and intensity values as 32 bit integer. Conversion of this data to an mzML file results in a file size of 577 MB. The imzML files are 430 MB and 217 MB for the processed and continuous mode, respectively. The smaller size of the *processed* imzML file compared to the mzML file is mainly due to base64 encoding of the mass spectral

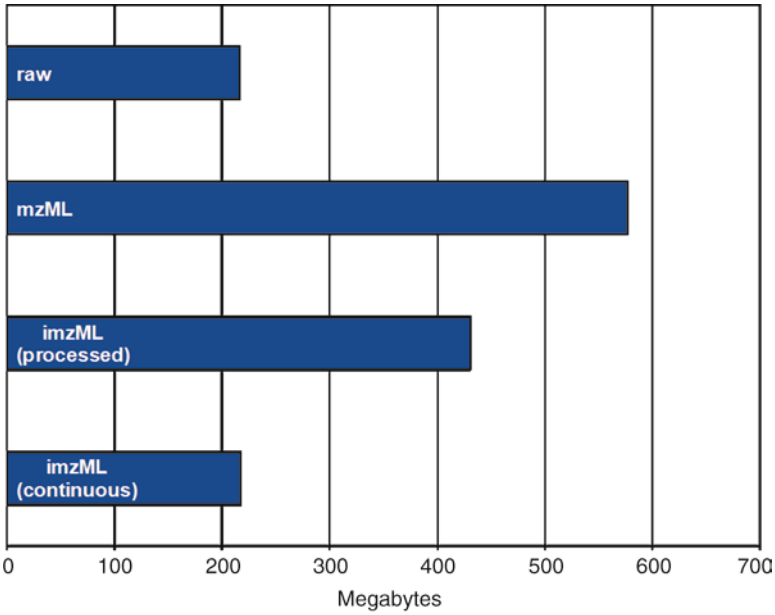


Fig. 8. File size comparison.

data in the mzML file. The even smaller file size of the *continuous* imzML file is due to the fact that the m/z array (which is identical for all mass spectra) is stored only once at the beginning of the binary file (see Subheading 2.1.3 for more details). The continuous imzML file is slightly larger (2 MB) than the original raw file.

In addition to the smaller overall size of the imzML files, the small XML file (several MB) can be used to distribute metadata over the network. This information can be used to select interesting measurements for which the (large) binary files are downloaded selectively (as opposed to downloading the complete dataset for all samples).

4. Implementation

imzML is already implemented in a number of software tools (some of them are available on http://www.imzML.org). Several vendors of mass spectrometry imaging instrumentation support the format (e.g., through export filters). Please check the imzML website (http://www.imzML.org) for updated information on supported vendor platforms and available tools.

4.1. Displaying Tools

There are numerous ways to display and analyze MS imaging data and no single software application can combine all features. Therefore it is a big advantage if one can freely choose the most

appropriate software and is not limited by (proprietary) data formats anymore. Examples of software tools that support imzML to display and analyze MS imaging data are shown in the following. A standard sample (peptide solution on a stainless steel target) was used for a round-robin experiment in order to compare different mass spectrometry imaging systems. The images below show the analysis of this particular measurement on a linear ion trap mass spectrometer with different software tools. The selected ion image of m/z 573 is shown for each tool in order to verify that the data is read and displayed correctly. The specific advantages of each software are illustrated with selected examples.

Biomap by Novartis (Fig. 9) is one of the most widely used software tools for mass spectrometry imaging. It allows browsing through selected ion images as well as coregistration of images and includes a large number of additional analysis tools.

The **Datacube Explorer** tool by FOM allows dynamic scrolling through masses in a dataset for fast and easy screening of a

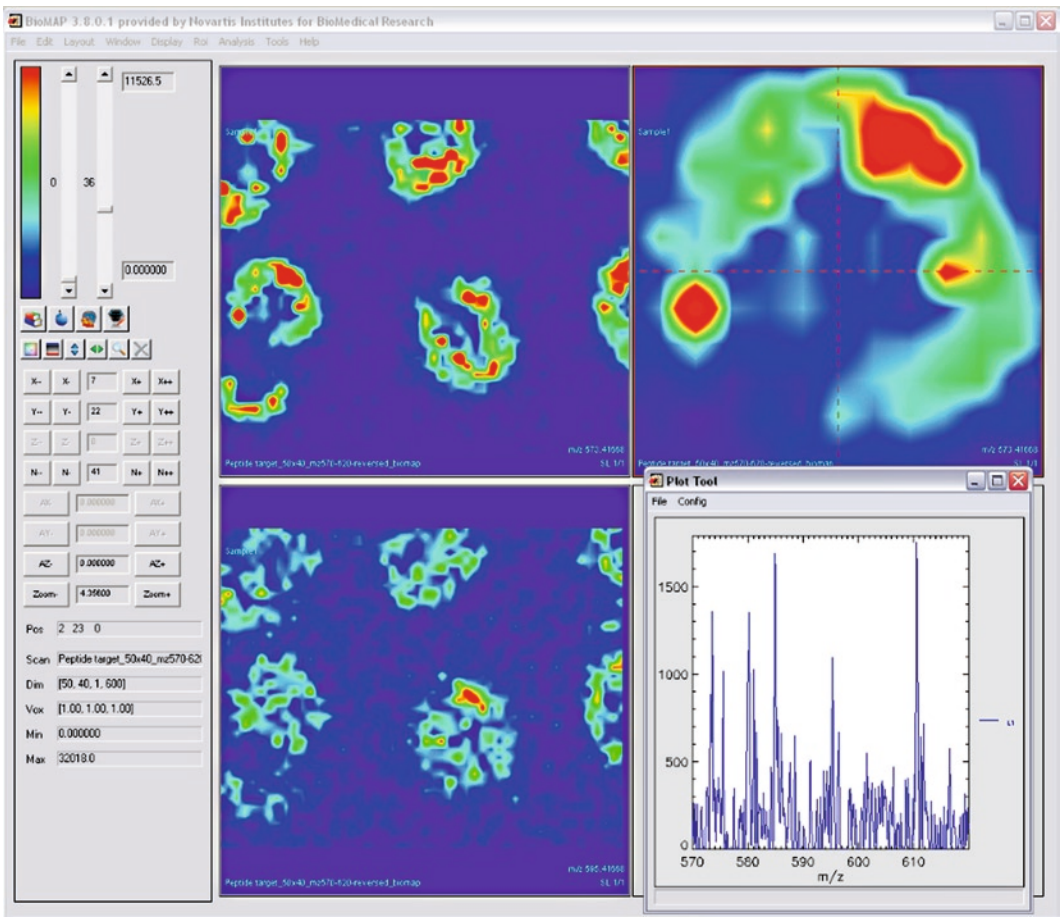


Fig. 9. BioMap (Novartis).

dataset (Fig. 10). It also allows spectral analysis of regions of interest and contains advanced analysis features such as self-organizing maps for image classification. This tool is available for free on http://www.imzML.org.

The **fxSpectViewer** by CEA is especially suited for handling very large data files without the need of binning (Fig. 11). It also

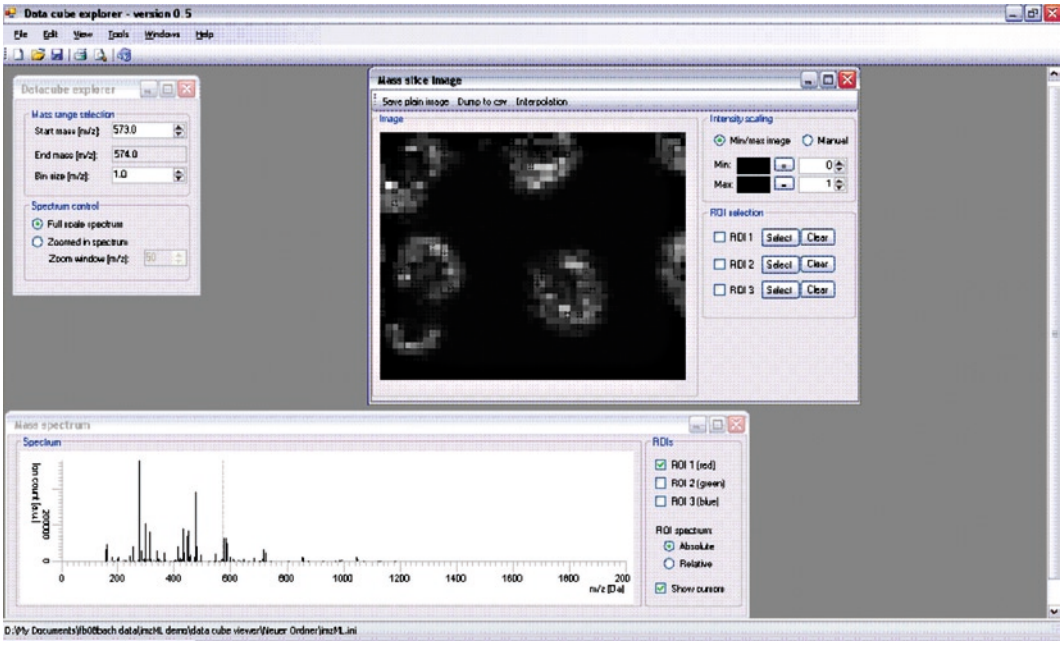


Fig. 10. Datacube Explorer (AMOLF).

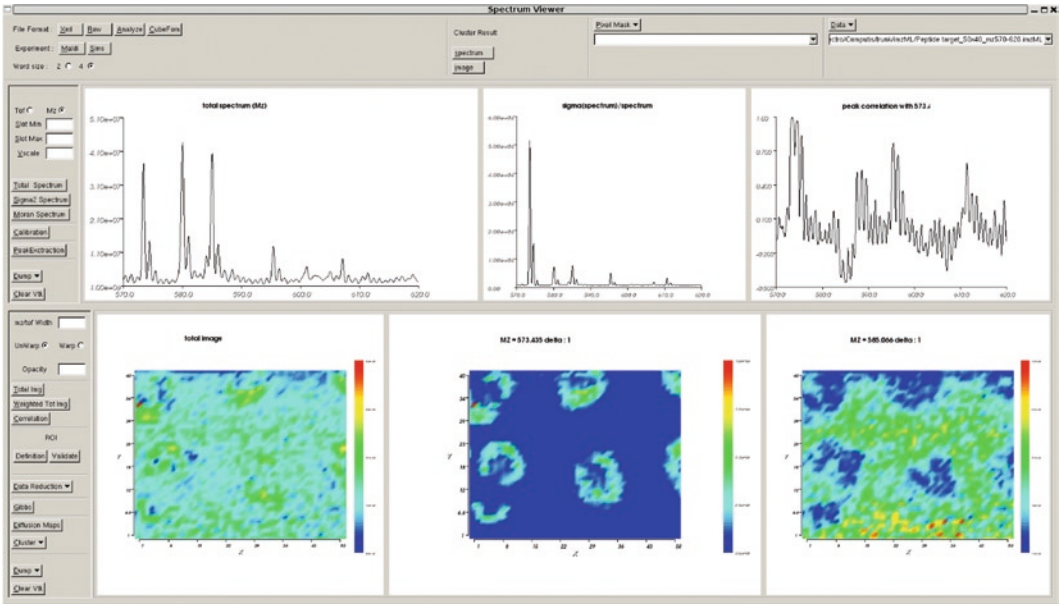


Fig. 11. fxSpectViewer (CEA).

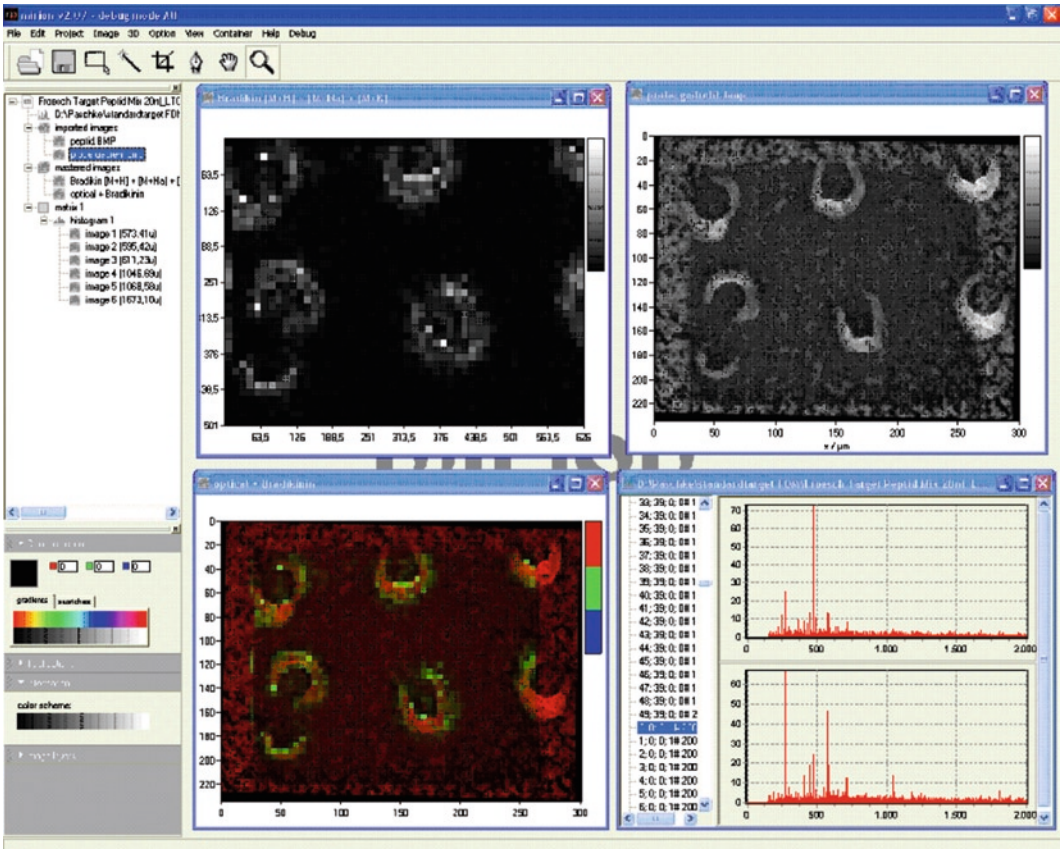


Fig. 12. Mirion (Justus Liebig University).

includes automatic segmentation of images. This software runs under Linux and Windows.

Mirion by JLU was especially developed for analyzing high mass resolution images (Fig. 12). It allows a bin width of 0.001 mass units. This is necessary to take full advantage of the highly accurate mass data from FTMS instruments. It also allows overlaying different MS images as well as optical images. Individual mass spectra are directly accessible from the image.

4.2. Converters

Several converters for imzML are currently developed and some are already available on the imzML website (http://www.imzML.org). An example and general considerations for the conversion of proprietary data to the imzML format is discussed in the following. The example shows a software that converts LTQ-based *.RAW files (proprietary format of Thermo Scientific GmbH, Bremen).

The "Conversion" tab contains details about the input RAW file. The "Imaging" tab (Fig. 13) includes the information that is essential to generate a valid imzML file – and therefore an unambiguous image. "Binary Mode" determines in which way the data

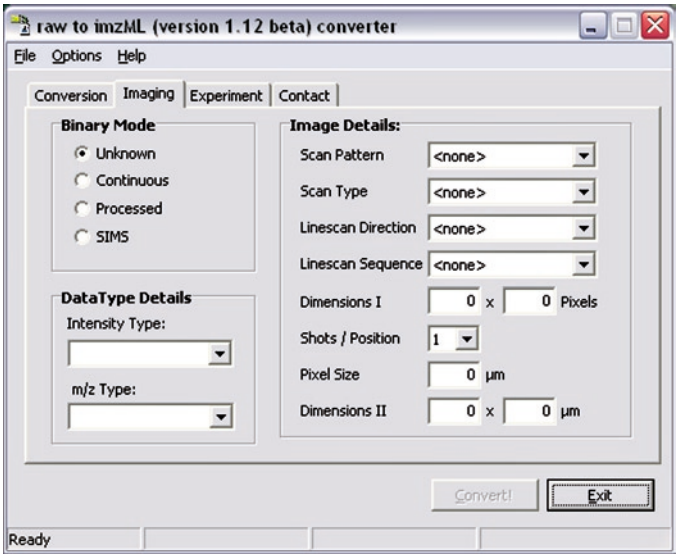


Fig. 13. imzML converter: User interface for imaging details.

is stored in the binary file. This option can have significant influence on the size of the resulting imzML file (see also Fig. 8). A discussion on which binary mode to use (processed or continuous) can be found in Note 3.

The second property that influences the file size is the data type which is used to store the values of the m/z and intensity arrays. This information is included in the "Data Type Details." Considerations for choosing the appropriate parameters are given in Note 4.

The original file might not contain all necessary information to generate a valid imzML file. This implies that the user has to add these details manually, for example, the entries on the right side ("Image Details"). The first four properties specify the characteristics of the scanning procedure which was used when the image was acquired (see Subheading "Scan Pattern").

Further information to be put in manually is included in the "Experiment" tab (Fig. 14). The parameters on the left describe the laser which was used in the MALDI imaging experiment. On the right side, the properties of the sample are listed. These parameters can be included, but they are not required to create a valid imzML file

When programming such a converter it is important to keep in mind that redundant information should be avoided if possible (for the sake of small data files). Some information might be stored redundantly in XML part of imzML. If for example data is acquired within an imaging experiment, all the measurements are

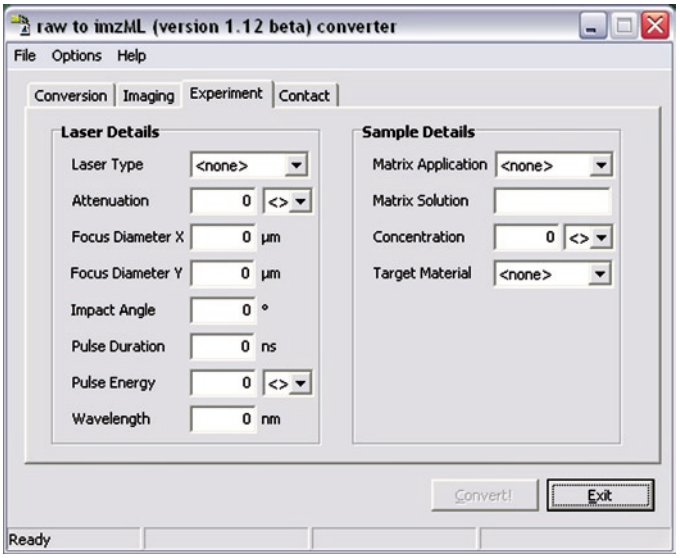


Fig. 14. imzML converter: User interface for experimental details.

usually acquired with the same instrumental settings and measurement conditions. Therefore it is sufficient to store this information only once per experiment and not for each spectrum separately. This redundant kind of information can nicely be merged into referenceable parameter groups (8). The file size can be reduced further by omitting CV param values that can be retrieved from the mass spectral data (e.g., base peak intensity).

5. Notes

1. Why not use mzML?

   First of all mzML was not available at the beginning of the COMPUTIS project (2006). At later stages we evaluated mzML with respect to imaging MS data. The main concern was the file size of converted data sets. Storing the MS data in a separate binary file is crucial for handling the very large imaging MS data sets. After it became clear that our main requirement (a separate binary file) was not possible in mzML, we decided to continue with our own format. However, we decided to store our metadata in the mzML format in order to be able to easily convert between the two formats. During the last years we stayed in contact with HUPO-PSI at various occasions. The result of these discussions was that we call our format imzML (for imaging mzML) and that it will exist in parallel to mzML for specific use cases

(e.g., large data sets). The structure of the XML metadata file in imzML will remain compatible with mzML. The imaging specific CV parameters will be kept in a separate imagingMS obo file. A number of entries from this file (which were of general importance) have already been included in the PSI-MS OBO file.

imzML files can easily be converted to mzML files (with the consequence of increased file size and limited metadata) in order to use mzML-based tools.

2. Byte order of UUID

Binary data can be stored in "little endian" or "big endian" (net standard) byte order. Mass spectral data in imzML binary files is stored in little endian byte order. The universally unique identifier (UUID), however, is stored in big endian according to the RFC 4122 specifications (cf.).

Intel processors and clones use little endian, therefore integers in the computer memory are also little endian numbers. Depending on the programming language which is used, byte order may be automatically "corrected" when reading big endian UUIDs. But some widely used programming languages have no such automatic correction. So the programmer has to take care of this.

In the imzML binary file the first 16 bytes are the binary representation of the UUID. A hexadecimal viewer can be used to examine these bytes, for example: 52 33 F9 E6 09 B9 4A 00 AB 01 AF 5D F4 BE 38 15.

In the corresponding imzML file the textual version of the UUID which consists of 5 blocks delimited by a "-"-char is: {5233F9E6-09B9-4A00-AB01-AF5DF4BE3815}. Be aware that in a correct implementation, both representations should have the same sequence of hexadecimal numerals.

When implementing imzML on a Microsoft operating system, one will usually use the Microsoft implementation of UUID: GUID (general unique identifier). Its memory representation is defined by:

TGUID = structure

Data1 : 4 byte unsigned integer;

Data2 : 2 byte unsigned integer;

Data3 : 2 byte unsigned integer;

Data4 : array of 8 bytes;

end structure;

In this structure, Data1, Data2, and Data3 are numbers and therefore subject to byte order, whereas Data4 is just an array of 8 bytes (and thus independent of byte order). That means when dumping this memory representation on a little

endian computer into a binary file Data1, Data2 and Data3 are inverted whereas Data4 is not inverted.

3. Continuous or processed format

   The binary mode (continuous or processed), which is used in the binary file has significant influence on the overall size of imzML files and is therefore an important parameter. Data with a continuous mass axis is often generated by time-of-flight instruments. Discontinuous data can be the result of data processing. In some cases the appropriate (most effective) data format can vary for one instrument depending on the settings and mode of operation. For example, LTQ-based instruments can generate profile or continuous data in the linear ion trap mode. Measurements performed in the centroid mode always have to be saved as "processed" due to the discontinuous mass axis. When the acquisition is set to profile mode the data format depends on the mass analyzer used for the acquisition. Data from Fourier transform analyzers (ICR or Orbitrap) is always modified before storage and thus has to be stored as "processed" (because of on-the-fly data-processing). If the linear ion trap analyzer is used for detection the data can be saved in continuous mode.

   It is possible to convert continuous into processed data. The m/z values of the first spectrum simply have to be saved for every spectrum. After conversion, all advantages of the processed mode are usable: for example, skipping zero intensity values.

   A special case is the storage of data acquired by secondary ion mass spectrometry (SIMS). This data is typically stored in an "event-based" format due to the much lower number of ions detected in this ionization mode. A data point consists of three values: x, y coordinates, and m/z (or time-of-flight) value. This data has to be converted in order be stored in the binary file. For each pixel the events are sorted by increasing m/z values (after conversion from time-of-flight, if necessary). These data have to be binned in order to be stored as mass spectra: events within a defined mass bin (e.g., 1 u) are summed up. The binned data can be stored in the binary file of the imzML format (usually in the "processed mode").

   It has to be stressed again that the most efficient data format strongly depends on the type of data and has to be evaluated for each set of experiments.

4. How to choose the appropriate binary data type?

   As already mentioned above the size of the binary file is dependent on the applied data. Choosing an unsuitable data type either results in loss of information or unnecessarily large data files. Some considerations on deciding which data type

to use for m/z and intensity values, respectively, are given below. If the data of a mass spectrometry imaging experiment was acquired by a very accurate mass spectrometer (e.g., FT-ICR), the accuracy of the measured **m/z values** (up to eight significant digits) should be taken into account by choosing the "64 bit float" (double precision) data type (up to 15 significant digits). Storing the data in "32 bit float" (single precision) would only allow seven significant digits, resulting in a loss of precision.

Intensity values of acquired spectra can be of integer or floating point data type. This depends on the way the ions are detected and on the digitizer of the used mass spectrometer. The best data type for integer values can be estimated by taking a look at the maximum intensity value of all spectra. If for example, the maximum intensity value is 125 then a data type of 8-bit integer will be sufficient to store these values. The usage of a data type with a range too small for all the values of a mass spectrometry imaging experiment will result in a loss of information. The same applies for the usage of an integer data type for floating point values, because the decimals of the numbers will be cut off by rounding.

References

1. McDonnell LA, Heeren RMA (2007) Imaging mass spectrometry. Mass Spectrom Rev 26:606–643
2. EU Project COMPUTIS (Accessed at http://www.computis.org)
3. Digital imaging and communications in medicine – DICOM. (Accessed at http://medical.nema.org)
4. Deutsch E (2008) mzML: a single, unifying data format for mass spectrometer output. Proteomics 8:2776–2777
5. Hermjakob H (2006) The HUPO proteomics standards initiative – Overcoming the fragmentation of proteomics data. Proteomics 6(Suppl 2):34–38
6. Orchard S, Hermjakob H, Taylor CF et al (2005) Second Proteomics Standards Initiative Spring Workshop. Expert Rev Proteomics 2:287–289
7. Biomap. (Accessed at http://maldi-msi.org/index.php?option=com_content&task=view&id=14&Itemid=39)
8. HUPO PSI mzML documentation version 1.1. (Accessed at http://psidev.info/index.php?q=node/257)
9. RFC 4122 – A Universally Unique IDentifier (UUID) URN Namespace. (Accessed at http://tools.ietf.org/html/rfc4122)
10. Smith B, Ashburner M, Rosse C et al (2007) The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol 25:1251–1255
11. Cote RG, Jones P, Martens L, Apweiler R, Hermjakob H (2008) The ontology lookup service: more data and better tools for controlled vocabulary queries. Nucleic Acids Res 36:372–376
12. Ontology of Mass Spectrometry Imaging. (Accessed at http://www.maldi-msi.org/index.php?option=com_content&view=article&id=187&Itemid=67)
13. PSI MS obo. (Accessed at http://psidev.cvs.sourceforge.net/checkout/psidev/psi/psi-ms/mzML/controlledVocabulary/psi-ms.obo)
14. OECD (1995) The application of the principles of GLP to computerised systems

Chapter 13

Tandem Mass Spectrometry Spectral Libraries and Library Searching

Eric W. Deutsch

Abstract

Spectral library searching in the field of proteomics has been gaining visibility and use in the last few years, primarily due to the expansion of public proteomics data repositories and the large spectral libraries that can be generated from them. Spectral library searching has several advantages over conventional sequence searching: it is generally much faster, and has higher specificity and sensitivity. The speed increase is primarily, due to having a smaller, fully indexable search space of real spectra that are known to be observable. The increase in specificity and sensitivity is primarily due to the ability of a search engine to utilize the known intensities of the fragment ions, rather than just comparing with theoretical spectra as is done with sequence searching. The main disadvantage of spectral library searching is that one can only identify peptide ions that have been seen before and are stored in the spectral library. In this chapter, an overview of spectral library searching and the libraries currently available are presented.

1. Introduction

Proteomics via mass spectrometry has become an important tool in our understanding of how complex biological systems function at all levels. At a more basic level, mass spectrometry-based proteomics provides a tool to better annotate our still basic understanding of genomes and the ability to characterize medically important subproteomes – or indeed our whole proteome.

By far the most prevalent tool for such work is tandem mass spectrometry (MS/MS), also called the "shotgun" method, as the technique involves: breaking intact proteins into smaller peptides; sequencing those peptides; and coalescing the results into protein identifications in a manner reminiscent of genomic shotgun sequencing. One challenge of shotgun proteomics is the formidable informatics analysis that must be applied to the raw data to achieve reliable results (1).

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_13,

The main informatic challenge comes in determining which peptide is represented by each mass spectrum. Most spectra are of insufficient quality to allow a direct reading of the sequence from the spectrum peaks, and therefore the usual approach is to generate theoretical spectra from a subset of possible peptides selected from a reference set of proteins and then to choose the peptide whose theoretical spectrum most closely matches the observed spectrum. In such a way, the search engine returns an answer for every spectrum, and the resulting additional challenge is to separate the correct from the incorrect answers as best as possible and determine the false discovery rate amongst the chosen positive population.

There are two stark shortcomings in the sequence searching technique. First, there is currently no good model for reliably predicting the relative intensities of the peaks in a theoretically generated spectrum, and therefore when comparing real to theoretical spectra, only the m/z values are compared and relative intensity information is ignored. Second, every search is performed from scratch without considering whether the current peptide may have been seen before.

A relatively new technique called spectral library searching addresses these two shortcomings. Spectra previously identified with high confidence are assembled into a library and new data are searched against the library. Since the reference spectra are real spectra, the relative intensity information in each spectrum may be leveraged to its full potential. The technique has recently gained considerable interest and use due to the emerging availability of high quality spectral libraries. The current state of libraries and searching software will be described in this chapter.

2. Spectral Library Searching

The basic technique of spectral library searching (sometimes also called spectrum matching) is briefly described as follows. For each spectrum in a list of new input spectra, a set of possible matches is selected from a reference spectrum library based on similar precursor m/z. Each of these possible matches is compared to the real spectrum using a technique such as a dot product, which includes the relative intensities of the peaks. The best match is (or top N matches are) reported in the same way as sequence searching and most downstream processing is then similar.

The most obvious shortcoming of this approach is that only peptides which have been previously observed can be identified. A high quality spectrum of a novel peptide will not be correctly assigned. This is a serious issue for discovery experiments, but is not serious for a time course experiment where all the peptides are known and the requirement is merely to find and quantify them in each time point.

2.1. Performance

Three clear advantages of spectral library searching are significantly improved speed, specificity, and sensitivity. Speed is increased primarily for two reasons. First, there is no need to exhaustively search an entire protein sequence reference list for peptides of m/z similar to the input spectrum and no theoretical spectra need to be generated for comparison. Rather, a simple indexing scheme can quickly pull only the candidate spectra from the library for immediate comparison with the input spectrum. Second, the overall search space is usually reduced because spectrum libraries only contain spectra that have been previously observed and do not contain the many peptides that have not been previously observed and indeed are not likely to be observed in any experiment.

Specificity is significantly improved because a library spectrum will be a much closer match to a new spectrum than its corresponding theoretical one (Fig. 1). Consider a spectrum that can be readily identified via sequence searching with a certain confidence. Its match to the spectrum library entry will be significantly more confident because peak intensity information is used to better discriminate from the population of incorrect matches.

Finally, sensitivity is markedly improved. Many additional low signal-to-noise spectra can be identified because the intensity information is available to the search engine. Consider a low signal-to-noise spectrum where only a handful of the most intense peaks are discernable. In a sequence search, the confidences of the matches of those peaks do not significantly rise above the noise of other possible matches. However, for a spectral library search if those few peaks are known to be the most intense peaks in the match, then the resulting score can be sufficiently distinguished from the noise of random matches to yield a reasonably confident identification.

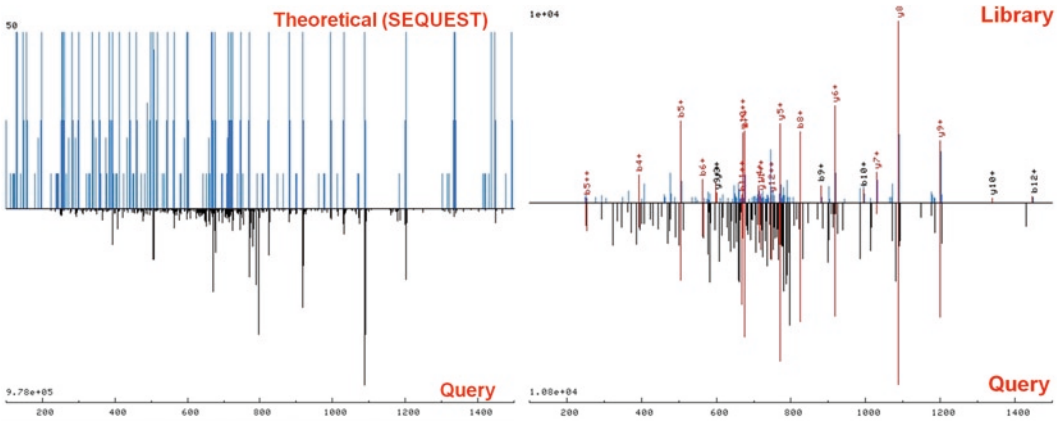


Fig. 1. Comparison of sequence searching and spectral library searching vs. real spectra. *Left panel*: a real query spectrum (*bottom*) compared with a simulated theoretical spectrum typically generated by sequence search engines. Major peaks are present, but it is difficult to see the comparison; yet sequence search engines perform reasonably well. *Right panel*: a spectral library entry compared with another real query spectrum (from the same peptide as the *left panel,* but of significantly lower signal-to-noise ratio).

The significant advantages afforded by spectral library searching make it a desirable addition to any analysis plan. If the missing of identifications of peptides that are not in the spectrum library is a serious concern, then an iterative or parallel approach where the dataset is searched against both the spectrum library and a protein sequence list and the results combined is an attractive alternative. Since spectral library search is so fast, it rarely increases the total search time over sequence searching alone.

2.2. Software

Spectral library searching was initially proposed a decade ago (2), but the lack of freely available software, inability to easily create libraries, and lack of any public libraries caused this technique to see little use. Then in 2006, there was a resurgence of interest in this approach with three articles published in close succession (3–5).

The X!Hunter program (4), part of the GPM suite of tools (6), is very fast and works with the comprehensive libraries distributed by the GPMdb project. The program does not work with other formats.

The Bibliospec suite of tools (5, 7) from the MacCoss lab at the University of Washington provide a library creation program (BlibBuild), a library filtering program (BlibFilter), and a library searching program (BlibSearch) as well as some additional tools all made available at http://proteome.gs.washington.edu/software/bibliospec/documentation/index.html.

The SpectraST program (3) from the Institute for Systems Biology is a comprehensive program that enables both the building of one's own spectral libraries and the searching of new data with those or publicly available libraries. SpectraST can convert most of the known library formats (described below) into its own indexed binary format. SpectraST is included as part of the TransProteomic Pipeline (TPP) software suite (8, 9), which allows it to be easily installed on several platforms, makes it compatible with the other popular downstream processing tools of the TPP such as PeptideProphet (10), and ensures compatibility with the popular pepXML format (8, 11).

The Bonanza program (11) from the Andrews Lab at the University of Michigan takes a somewhat different approach of clustering all input spectra together first and then trying to identify those clusters rather than the individual spectra. Although the Bonanza algorithm is described (11), the original software is not available, but rather integrated into the commercial software Cluster (http://www.singleorganism.com/).

The National Institute of Standards and Technology (NIST) also distributes (http://chemdata.nist.gov/mass-spc/Srch_v1.7/index.html) a spectral library search program with its very high quality libraries (described below). This is a Windows program generally optimized for low-throughput manual exploration of the libraries and potential matches of new spectra. This algorithm

has also been integrated into soon-to-be-released versions of the Open Mass Spectrometry Search Algorithm ((12); OMSSA) sequence search engine, creating a hybrid engine.

3. Spectral Libraries

Spectral library searching approaches can only be as good as the spectral library used for processing a dataset. Lack of good libraries is the primary reason that it took so long for spectral library searching to catch on. It was the advent of public data repositories that spurred the release of raw spectra to public and enabled the creation of large spectral libraries. Several resources provide libraries for some major species, as is described below. If there is no adequate library for the species of interest, then one must create a spectrum library first, and there are now some good tools to do this.

The problem of false positives is important when using the spectral library searching approach. Errors in the library can easily be propagated to new datasets with apparent low error. Consider the case where a multiply observed spectrum with a high quality consensus spectrum is misidentified because the true sequence is not present in the protein list and incorrectly inserted into the library. Subsequent spectral library searching may match a new input spectrum to the library entry with a very high score with an apparent very low chance of being wrong, but the identification will be wrong because the library entry is wrong. One technique to mitigate this is to assign a probability that each identification is correct in the library and then cap the probability of any matches to that of the library entry (13). Thus, if the peptide assigned to the spectrum in the library is only 90% likely to be correct (i.e., $P = 0.9$), then even a very high score match to this spectrum may not be assigned a probability >0.9.

3.1. Availability

At the PeptideAtlas web site there is a "Spectrum Libraries Central" page (http://www.peptideatlas.org/speclib/) that will be maintained in the future (Fig. 2) furnishing all the libraries generated at NIST, as well as libraries created at ISB, and links to additional spectrum library searching resources. As new resources become available subsequent to the writing of this chapter, they will be listed at this PeptideAtlas page.

NIST has systematically collected raw data from many different repositories and via private communication and used all these data to build very high quality spectrum libraries for several different species and samples. The libraries are distributed in MSP format both at NIST and ISB. The format is compatible with SpectraST and the NIST MS Search software.

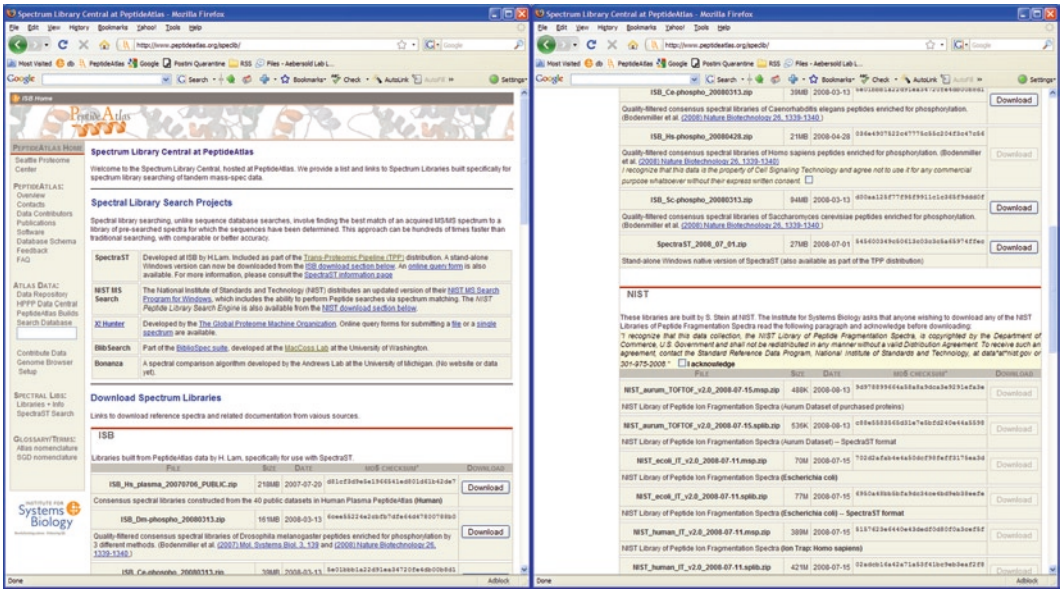


Fig. 2. PeptideAtlas "Spectrum Libraries Central" page providing the most up-to-date list of spectral library searching projects and software as well as lists of downloadable spectrum libraries generated at ISB and also by NIST.

The MacCoss lab at the University of Washington has compiled libraries for seven different species based on data collected in that lab, and makes the libraries available at their web site (http://proteome.gs.washington.edu/software/bibliospec/documentation/libs.html). They are distributed in the binary format for their BiblioSpec tool.

The GPM provides a set of libraries in their HLF format for use with X!Hunter at ftp://ftp.thegpm.org/projects/xhunter/libs/. The SpectraST software can convert this format to its own native format for use by SpectraST. One notable variation with the GPM libraries is that each spectrum only retains the top 20 peaks. This makes the libraries very small and the searching very fast, although it appears that retaining only the top 20 peaks leads to inferior specificity and sensitivity (13).

3.2. Formats

There are no official standard formats for encoding spectrum libraries. Each independent group has created their own format. The Proteomics Standards Initiative (PSI) has not yet, as of this writing, developed a standard file format for spectrum libraries to complement its formats for mass spectrometer output files (mzML (14)) and search engine output (mzIdentML (15)), although the latter might be used for spectrum libraries at some point in the future. NIST distributes their libraries in their MSP format. GPM distributes libraries in their HLF format. ISB distributes libraries in their SpectraST format. The SpectraST software is capable of

reading most of these formats, but can only convert these other formats to its splib indexed binary format and its sptxt plain-text format similar to the MSP format.

3.3. Creating Private or Specialized Spectrum Libraries

If none of the currently available spectrum libraries suits the needs of a certain analysis problem (e.g., if the data are from a species for which there are no libraries) then a custom library must be created. Any data that is processed with the TPP's PeptideProphet can be converted into a spectrum library using the SpectraST software. The BlibBuild software can create a spectrum library from input data that are in the MS2 format.

4. Conclusion

Spectral library searching is a powerful technique that enjoys speed, sensitivity, and specificity advantages over sequence searching. However, it is in many ways less flexible and therefore cannot be an alternative for some kinds of analysis. However, for many analyses it does provide a significant advantage and can also be complementary to conventional search strategies in a way that significantly increases the total number of identified spectra. As the libraries become more comprehensive and combining multiple searches into a single result becomes more prevalent, spectral library searching will become recognized as a crucial element to many analyses.

References

1. Deutsch EW, Lam H, Aebersold R (2008) Data analysis and bioinformatics tools for tandem mass spectrometry in proteomics. Physiol Genomics 33(1):18–25
2. Yates JR 3rd et al (1998) Method to compare collision-induced dissociation spectra of peptides: potential for library searching and subtractive analysis. Anal Chem 70(17):3557–3565
3. Lam H et al (2007) Development and validation of a spectral library searching method for peptide identification from MS/MS. Proteomics 7(5):655–667
4. Craig R et al (2006) Using annotated peptide mass spectrum libraries for protein identification. J Proteome Res 5(8):1843–1849
5. Frewen BE et al (2006) Analysis of peptide MS/MS spectra from large-scale proteomics experiments using spectrum libraries. Anal Chem 78(16):5678–5684
6. Craig R, Cortens JP, Beavis RC (2004) Open source system for analyzing, validating, and storing protein identification data. J Proteome Res 3(6):1234–1242
7. Frewen B, MacCoss MJ (2007) Using BiblioSpec for creating and searching tandem MS peptide libraries. Curr Protoc Bioinformatics. **Chapter 13**: p. Unit 13 7.
8. Keller A et al (2005) A uniform proteomics MS/MS analysis platform utilizing open XML file formats. Mol Syst Biol, 1: p. 2005.0017.
9. Deutsch EW et al (2010) A guided tour of the Trans-Proteomic Pipeline. Proteomics 10(6): 1150–1159
10. Keller A et al (2002) Empirical statistical model to estimate the accuracy of peptide identifications made by MS/MS and database search. Anal Chem 74:5383–5392
11. Falkner JA et al (2008) A spectral clustering approach to MS/MS identification of post-translational modifications. J Proteome Res 7(11):4614–4622

12. Geer LY et al (2004) Open mass spectrometry search algorithm. J Proteome Res 3(5): 958–964
13. Lam H et al (2008) Building consensus spectral libraries for peptide identification in proteomics. Nat Methods 5(10):873–875
14. Deutsch E (2008) mzML: a single, unifying data format for mass spectrometer output. Proteomics 8(14):2776–2777
15. JonesAR et al The mzIdentML data standard for mass spectrometry-based proteomics results. Nat Biotechnol, submitted.

Part IV

Processing and Interpretation of Data

Chapter 14

Inter-Lab Proteomics: Data Mining in Collaborative Projects on the Basis of the HUPO Brain Proteome Project's Pilot Studies

Michael Hamacher, Bernd Gröttrup, Martin Eisenacher, Katrin Marcus, Young Mok Park, Helmut E. Meyer, Kyung-Hoon Kwon, and Christian Stephan

Abstract

Several projects were initiated by the Human Proteome Organisation (HUPO) focusing on the proteome analysis of distinct human organs. The initiative dedicated to the brain, its development and correlated diseases is the HUPO Brain Proteome Project (HUPO BPP). An objective data submission, storage, and reprocessing strategy have been established with the help of the results gained in a pilot study phase and within subsequent studies. The bioinformatic relevance of the data is drawn from the inter-laboratory comparisons as well as from the recalculation of all data sets submitted by the different groups. In the following, results of the single groups as well as the centralised reprocessing effort are summarised, demonstrating the added-value of this concerted work.

1. Introduction

Due to the need for an international proteomic forum to improve understanding of human diseases, the Human Proteome Organisation (HUPO) was launched in February 2001 (1). Several initiatives are focused on the study of human organs and their specific diseases, especially by proteomic means. The initiative dealing with the brain is the Brain Proteome Project (HUPO BPP) (2–4), founded in 2003 and chaired by Young Mok Park, KBSI, and Helmut E. Meyer, MPC.

The brain is of highest interest in medical research and in pharmaceutical industry because of the increasing social impact of the neurological diseases, such as Alzheimer's disease, Parkinson's disease, Multiple Sclerosis, etc. The prevalence of some of these diseases is increasing within the last decades, e.g. every fifth person

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_14,

over 80 years in industrial countries is suffering from Alzheimer's disease (5). The most promising current approach to help understanding the developmental and neurodegenerative changes in the brain is proteomics in combination with other well-established methods of molecular biology and human genetics. In order to reach understanding neurodegenerative diseases and ageing, it is necessary to coordinate neuroproteomic activities worldwide and to enable every participant and active member of the HUPO BPP to access all data and new technologies obtained through these studies (for an activity overview see Table 1). Unfortunately, the exchange of raw data between several groups is often hampered by incompatible data formats and the lack of common software. The HUPO BPP recognised the need for guidelines from the very beginning and declared the elaboration of standards and the bioinformatics infrastructure as one of the primary goals. This is consistent with the general effort to shape a new, reliable proteomics codex (6, 7). In order to evaluate existing approaches in brain proteomics as well as to establish a standardised data reprocessing pipeline, pilot studies had been initiated in 2004, including both mouse and human brain samples (8). Participating groups were free to analyse these samples according to their own approaches. Data generated was collected and centrally analysed. The comparison between the distinct group and the centrally gained results gave new insights into inter-lab projects and led to a new code of conduct in proteomics bioinformatics (9, 10), e.g. also visible in the work of the EU project Proteomics Data Collection ProDaC (11–13). The data flow and the quintessence of the pilot studies is presented in the following (for a deeper insight please refer to the PROTEOMICS special edition *The Human Brain Proteome Project – Concerted Proteome Analysis of the Brain, 2006)*, including an update of HUPO BPP´s recent work.

2. Materials and Methods

In the mouse pilot study, brain tissue from normal mice of three developmental stages had to be analysed by quantitative proteomics, while in the human pilot study, two human brain tissue samples from an autopsy and a biopsy, respectively, had to be analysed by quantitative/qualitative proteomics techniques.

Samples were sent out to several groups that could use their own standard analysis protocols. Data had to be submitted to a Data Collection Center (DCC, data file storage solution at the Medizinisches Proteom-Center, Bochum, www.medizinisches-proteom-center.de) for central reprocessing and were submitted to the PRIDE database (http://www.ebi.ac.uk/pride, experiment

Table 1
Workshops, meetings and achievements 2002–2010

Timeline	Meeting
November 21–24, 2002	1st HUPO World Congress, Versailles, France Meyer and Klose offered to chair the brain initiative
January 15, 2003	PepTalk, San Diego, presenting of the HUPO BPP
April 28, 2003	Kick-off Meeting in Frankfurt/Main, Germany with 25 participants, announcement of project
May 2003	HUPO BPP homepage www.hbpp.org minutes and updates can be found here
July 10, 2003	Planning Committee Meeting in Frankfurt/Main, Germany updates and decision of first steps
September 5/6, 2003	1st HUPO BPP Workshop at Castle Mickeln, Germany with 50 participants establishing of committees (Steering, Specimen, Technology and Standardisation, Bioinformatics, Training) idea of pilot studies and master plan
October 8–10, 2003	Neuroproteomics Session at the 2nd HUPO World Congress in Montreal
January 20, 2004	1st Steering Committee Meeting at the ESPCI in Paris, France organised by Jean Rossier; preparation of 2nd HUPO BPP Workshop
April 23/24, 2004	2nd HUPO BPP Workshop at the ESPCI in Paris, France, organised by Jean Rossier; official begin of pilot study and fixing of master plan (75 participants)
April 26/27, 2004	1st HUPO BPP ProteinScape Training Course for Pilot Study Participants at the MPC, Bochum, Germany
July 29, 2004	1st HUPO BPP Bioinformatics Meeting at the EBI in Hinxton, UK, organised by Rolf Apweiler; elaboration of a Data Collection Concepta time line for reprocessing and publishing, implementation of ProteinScape as general analysis software
July 2004	Internet forum forum.hbpp.org with several discussion forums announcements and downloads
September 30, 2004	2nd Steering Committee Meeting in Frankfurt/Main, Germany updates and agreement of Data Collection Concept
October 15, 2004	2nd HUPO BPP ProteinScape Training Course for Pilot Study Participants at Protagen, Dortmund, Germany
October 23, 2004	Neuroproteomics Session and HUPO BPP Workshop at the 3rd HUPO World Congress in Beijing presentation of project (e.g. pilot studies) and discussions, intensification of exchange/contact with other HUPO projects and international societies
End of 2004	Implementation of Data Collection Center at the MPC, Bochum, Germany

(continued)

Table 1
(continued)

Timeline	Meeting
November 5, 2004	2nd HUPO BPP Bioinformatics Meeting at the EBI in Hinxton, UK organised by Rolf Apweiler
December 15/16, 2004	3rd HUPO BPP Workshop at Castle Rauischholzhausen, Germany
December 17, 2004	3rd HUPO BPP ProteinScape Training Course for Pilot Study Participants at the MPC, Bochum, Germany
January 28, 2005	3rd HUPO BPP Bioinformatics Meeting at Protagen, Dortmund, Germany
April 8, 2005	4th HUPO BPP Bioinformatics Meeting at the EBI in Hinxton, UK
June 1, 2005	"HUPO BPP International Workshop on Mouse Models for Neurodegeneration" in Doorwerth, The Netherlands
July 7, 2005	5th HUPO BPP Bioinformatics Meeting at the EBI in Hinxton, UK
August 27, 2005	4th HUPO BPP Workshop in Munich, Germany during the 4th HUPO World Congress
January 9–11, 2006	Jamboree of the HUPO BPP Bioinformatics Committee at the EBI in Hinxton, U.K.
February 15–16, 2006	5th HUPO BPP Workshop ""Bridging Proteomics and Medical Science" at the UCD, Dublin
September 4–7, 2006	Preliminary: Neuroproteomics Session at the 7th Siena Meeting
October 27–November 1, 2006	6th HUPO BPP Workshop at the 5th HUPO World Congress in Long Beach, USA
March 7–9, 2007	7th HUPO BPP Workshop "High Performance Proteomics in the HUPO BPP" at the EBI in Hinxton, UK
October 7 2007	8th HUPO BPP Workshop "Applications in Brain Proteomics" at the 6th HUPO World Congress in Seoul, Korea
January 9–10, 2008	9th HUPO BPP Workshop "The HUPO BPP Roadmap" Barbados
August 16, 2008	10th HUPO BPP Workshop "New Concepts for Neurodegenerative Diseases" at the 7th HUPO World Congress in Amsterdam, Netherlands
March 3, 2009	11th HUPO BPP Workshop in Kolymbari, Greece
September 26, 2009	12th HUPO BPP Workshop at the 8th HUPO World Congress in Toronto, Canada
To come in 2010:	
March 30–31, 2010	13th HUPO BPP Workshop in Ochang, Korea
September 18, 2010	14th HUPO BPP Workshop at the 9th HUPO World Congress in Sydney, Australia

accessions 1669–1750) serving as reference data for future analysis (14, 15). In the course of these studies and the subsequent central reprocessing, a data collection, submission and storage pipeline has been established, a bioinformatics identification strategy has been elaborated and very interesting insights into today's proteomics approaches could be gained.

2.1. Methods

Participating laboratories used their own protocols that had to be annotated (see Notes 1). Pooling of the samples was not allowed. Data had to be submitted to the DCC for a central reprocessing, preferably using ProteinScape™ (Bruker Daltonics, Bremen) for data collection and submission. In addition, data had to be made publicly accessible at the PRIDE database as mentioned above.

2.2. Data Collection Center and Bioinformatics

Not surprisingly, heterogeneity of the data had been very high because of the diverse analytical strategies. Therefore, it was of greatest interest to show if a central data reprocessing would lead to additional as well as more reliable protein identifications. To reach this goal, a Data Collection Center and a central data reprocessing workflow were elaborated. A common, powerful and automated data analysis strategy was elaborated to collect, to analyse and to reprocess these heterogeneous data sets (also see (16)).

The Data Collection Center was localised at the MPC, Bochum, Germany. The groups were asked to use ProteinScape™ software (Bruker Daltonics, Bremen, Germany) to allow handling, storage, and standardised re-analysis of the submitted data. In order to give some standardised basics, a specified analysis guideline was provided elaborated within the Bioinformatics Committee [International Protein Index (IPI) database release April 2005 to search against, minimal two identified peptides per protein, etc.; full guideline available at www.hbpp.org)]. Moreover, the false discovery rate of proteins should be lower than 5%. This criterion has been used in several other studies, most notably by the HUPO Plasma Proteome Project (HUPO PPP) (17).

The database integration of the submitted several gigabytes of data, including peak lists, gel images, and sample descriptions, was finished at the end of April 2005. Re-analysis of the data sets according to the re-analysis criteria was done in several iterative steps resulting in data sets containing non-evaluated peptide lists. Different search engines were employed at the PAULA cluster, MPC, Bochum (Proteomics Cluster under Linux Architecture) to broaden the number of identified proteins, including Sequest (Thermo Finnigan, Waltham, MA, USA – cluster licence already existing) for MS/MS data, ProteinSolver (Bruker Daltonics, Bremen, Germany) for MS/MS data, Mascot (temporary free cluster licences by Matrix Science, London, UK) for MS/MS and

MS data, as well as ProFound (Rockefeller University, New York, USA) for MS data.

All MS data sets were searched against a specially prepared decoy protein database of the International Protein Index (IPI 3.05) databases for each analysed species. A decoy protein for each protein in the original database was added shuffling all amino acids of this protein.

MS spectra were sent to Mascot and ProFound as described in the data reprocessing guideline (see www.hbpp.org). Protein results with a Metascore higher than 90 were labelled as identified.

The MS/MS data from the distinct separation types (spot/band/fraction) were sent independently to Mascot, ProteinSolver, and SEQUEST. The search parameters were evaluated for each search engine separately prior to their use for the automated approach to estimate the best peptide threshold score, generating a pool of peptides used for assembly of the proteins. This enhances the maximum of identified proteins by a defined false positive rate of 5%. The evaluation of parameters was calculated by analysing a subset of 12,000 spectra. ProteinExtractor had been used for assembling the different protein lists from MS/MS data and for removing redundancies. ProteinExtractor generates a protein list containing a minimal set of proteins with those isoforms that can be distinguished by MS/MS data. This strategy comprises an iterative approach: First, select the most likely protein candidate (highest summed peptide score), write this protein into the result list. Second, mark all spectra explainable by this protein as "used", then select the most probable next protein candidate from the still unused spectra, and repeat this, until all spectra are marked as "used". The algorithm is following rules elaborated by MS/MS experts for those proteins that should appear in a minimal protein list. In this approach, the number of matched peptides for the identified proteins in any of the search engines has been used rather than the sum of the peptide scores. The resulting merged protein list has been sorted by the sum of the individual sum scores of the algorithms, and proteins were marked identified until the list contained 5% decoy proteins. Up to 3% of peptides corresponded to a decoy entry on the peptide level in a list containing less than 5% false positive proteins. The overlap of the search engines was 80–90%, therefore 10–20% more proteins can be found using this approach.

The HUPO BPP was one of the first coordinated initiatives supporting mzData standard format of the HUPO Proteomics Standards Initiative (PSI). This standard format is now succeeded by mzML (http://www.psidev.info/index.php?q=node/80). As mentioned above, collected data were submitted to the PRIDE database for public access.

3. Results

Nine independent laboratories analysed the samples according to their own approach in these pilot studies. The amount of the total MS spectra was 1.350 (0.2%) and of MS/MS spectra 740.000 (99.8%). Approximately 80% of the spectra belonged to the human samples and approximately 20% to the mouse samples. Half of the spectra originated from gel-based or LC-based approaches each.

The spectra can be classified in different ways to observe the diversity of experimental setups. Prior to the MS analysis, different separation techniques were applied: 32% of the spectra were acquired after 1D PAGE separation, 22% after 2D PAGE (PAGE approaches are not discussed here) and 46% after liquid chromatography.

3.1. Centralised Analysis Strategy

Central data reprocessing of such huge amount of data is very time- and resource-consuming. To maximise the statistical reliability of the results, an automated workflow was elaborated (16) in an iterative way. It proved to be very useful that data sets were collected using a common software (ProteinScape and other algorithms therein) supporting the new standard format mzDATA. Though not all groups adhered to the default settings, the conversion into the common standards (e.g. to the right IPI version 3.05) turned out to be relatively easy.

The adaptation and implementation of additional algorithms, such as ProteinExtractor, as well as the determination of the parameters for the different search engines took several rounds. Now completed, this pipeline allows an easy, objective (in the meaning of the strict use of the parameters), comprehensive, and fast way for reprocessing of MS/MS spectra. It assures a highly reliable list of identified proteins in combination with the use of a decoy database to determine a false discovery rate below 5%. The use of different search engines results in more proteins than the single searches alone (see also Notes 2). Each search engine finds a subset of the overall protein lists due to the different algorithms; the combination of all leads to a higher number of identified proteins. In addition, proteins, corresponding gels, and differential expression level alterations can easily be correlated even after submission to the DCC, as this information is represented in the ProteinScape database structure.

3.2. Single Compared to Centralised Approaches

A total of nine participating labs analysed human and mouse brain samples, respectively, using a variety of different techniques. 37 different analytical approaches were accomplished, 17 of these analyses were done differentially, i.e., the protein expression patterns of the different samples (human or mouse) were compared.

The participating groups identified over 1,200 redundant proteins in sum. The comparison of these data had been severely limited due to the different used parameter sets including different protein databases, identification criteria, etc. By applying the described bioinformatics workflow, it was possible to reprocess all available data sets with a fixed parameter set, in a reasonable time frame, with a suitable amount of manpower and with a known quality threshold (false discovery rate). By considering all spectra, the amount of proteins increased fourfold to over 5,400 (redundant), mainly because of the previously non-analysed data. In some data sets, more proteins could be identified (up to over 90%), while others tend to lose identified proteins (up to 35%), probably due to the use of other, more stringent parameters. After removing redundancies by using the software ProteinExtractor, 1,832 were identified in the human brain sample and 792 proteins in the mouse brain samples. Thus, the combination of all (normalised) data allowed the HUPO BPP bioinformatics team to generate a comprehensive, but still extended lists of proteins with an objective quality standard, thus making the best of reliability and effectiveness (see also Notes 3).

3.3. Technology Platforms

The overlap of identified proteins between the different participants is very low, e.g. no protein had been named by all groups in the mouse study performing quantitative 2D PAGE analyses. Nearly, 80% of all proteins were listed by just one group and are uniquely identified, while only a subset of proteins were detected by several groups. Of course, the very different amount of proteins identified by the different groups reduced the possibility of a big overlap. However, this distribution can be found in several big studies (see e.g. (18)) and reflects the different features of the approaches used and the subsequent sources of variation under these conditions in different laboratories. In general, the data reprocessing might be an additional reason for the small overlap, e.g. because of the application of distinct protein assembly algorithms.

In addition, most proteins are very low abundant, while the detection by recent methods is not sufficient for reproducible detection throughout different laboratories. This also means that a weighting or valuation of the techniques is not meaningful as long as the approaches are not accurately standardised. The internal laboratory reproducibility has to be proved by all means.

3.4. Data Mining

Data mining has been one of the main goals of the pilot studies. Mueller and his colleagues from the European Bioinformatics Institute (EBI) in Hinxton, UK, elaborated a workflow for annotating identified proteins, including sequence features, genomic context, GO annotation, protein–protein interactions and disease association (19). In general, the results obtained reflect roughly

the protein composition one would expect when analysing the brain. Mueller et al. reported an enrichment of genes corresponding to the identified proteins that are encoded on Chromosome 1. Moreover, they reported many detected proteins being involved in energy metabolism (mitochondrial electron transport, hydrolases) or in transport mechanisms (cytoskeleton associated proteins), as the biological context "brain" would predict. However, transmembrane proteins are underrepresented, probably due to the used sample preparation and/or analysis procedures. This extended data mining workflow allows a fast and automated analysis of protein lists

Interestingly, a high number of proteins identified in this pilot study are not named in other extensive proteome studies. This phenomenon has been described by Martens et al. (20) reporting, e.g. that the overlap between the HUPO BPP and the HUPO PPP (21) human is slightly above 15%, while nearly 50% between the HUPO BPP and the platelet proteome data set (22) are identical (after normalisation to a common IPI version). Most overlapping proteins belong to housekeeping genes. The higher percentage overlap between brain and platelets could be due to the many shared functions and proteins that are found in both tissues. Omenn and colleagues could show, e.g. that serotonin can be removed by reuptake into serotonergic neurons as well as into platelets (23). Martens also compared the strategies of the HUPO BPP and the HUPO PPP pilot studies. While the HUPO BPP used peak lists for the central reprocessing, the HUPO PPP initially relied solely on peptide sequences as units and gathered identifications in a centralised database, leaving the classification into low and high confident to the submitting laboratory. The number of peptides per protein reported is similar for both HUPO initiatives, reaching more than eight peptides (human), while the overlap of peptides identified does not exceed 5% between the two data sets. In summary, the authors come to the conclusion that the organisation of data management and the synergistic effects of a consortium of collaborators are of outstanding importance.

4. Discussion

Although the Pilot Studies of the HUPO BPP were performed on a voluntary basis, the acceptance and the engagement of the participating laboratories were very impressive leading to the submission of enormous data sets to the Data Collection Center.

One of the severe problems on the bioinformatics site was the heterogeneous data handling within the different groups (data format, data collection, and data interpretation). To avoid possible

obstacles, it was agreed upon using a common data handling concept and to support the mzDATA standard from the HUPO Proteome Standards Initiative. By implementing the ProteinScape software at most participating laboratories, it was possible to assure correct data exchange and storage between the groups and the DCC. Incompatible data formats and long unmanageable Excel list could be bypassed with this approach. The bioinformatics committee of the HUPO BPP made great efforts to elaborate a bioinformatics workflow for the "objective" reprocessing of the heterogeneous data sets. By combining existing software solutions, by adapting the search parameters to the existing data, and by applying the decoy database the committee succeeded in elaborating a fast, reliable (FDR <5%), and automated data reprocessing pipeline. Once defined, this workflow can be applied fast (processing the data set generated in these studies now takes approximately 2 weeks) and is easy to use.

It became clear that the analysis of the same sample could result in different sets of proteins when using different protein separation systems (e.g. by using Klose or IPG gels). This could be due to the different nature of the used systems and the different resulting separation properties thereof. The same is true for the different approaches used by the groups, namely, gel-based vs. non-gel-based analyses. Even within similar approaches, slightly variable handling or parameters lead to changed protein sets detected, especially when analysing low abundant proteins. This could explain the small overlap between the proteins named by the different groups, leaving the single sets unique, but not (necessarily) wrong.

5. Outlook

Concerning the main phase of HUPO BPP, the results of the pilot studies influenced the ongoing strategies. Since then, understanding the pathogenesis of neurodegenerative diseases has improved, but both AD and PD can neither be cured nor be detected in a pre-stadium by prognostic biomarkers. Therefore, an urgent requirement for advanced comprehension of the mechanisms causing neurodegenerative diseases still exists. Several HUPO BPP workshops have taken place since the pilot phase to face these challenges (for an overview, please see Table 1). Workshops being held around the annual HUPO world congress in autumn focus more on techniques applied in proteomics like membrane proteomics, general challenges in proteome analysis of brain samples. The spring workshops, nowadays, deal more with clinical issues, such as discussing the neuropathological aspects of neurodegenerative diseases like Alzheimer's and Parkinson's disease.

From the clinical point of view, general accessibility and the quality of human brain samples collected from autopsies and clinicopatholigical series as well as perspectives for diagnostic and prognostic biomarkers of dementias have been discussed recently (24).

Most recently, a creation of a Human Brain Proteome Map (or Atlas) using Brodmann's Area is discussed to foster detecting and curing neurodegenerative diseases in a preclinical stadium. Whether or not this map will be developed and used in future strategies are discussed in the upcoming workshops in Ochang and Sydney (see Table 1).

6. Notes

1. Annotations concerning sample handling, preparation, separation, and identification are mandatory so that discrepancies and differences can be traced back. This will become mandatory as the journals already recognised the need for reliable data sets (6, 7).
2. Different approaches and search engines have to be seen as complementary. The combination of the generated output results in added-value, as on the one hand identified proteins can be approved, while on the other hand new proteins can be detected by the combination of the peptides identified by different groups. The separation features of the different techniques do overlap and can be applied successively.
3. Every study has to show the reproducibility of its data. As the overlap of the identified proteins in regard to the different laboratories is not optimal, it is extremely important that the own data is handled very critically and that internal SOPs per group (NOT necessarily between the laboratories) are essential. This includes, e.g. independent analyses of the biological samples with an appropriate statistical number of repeats (more than five, no pooling) and taking into account the limitations of the used technique. Only confident protein lists resulting from stringent criteria also benchmarking of MS/MS search algorithms by Kapp and colleagues (25) are of advantage to the scientific community and will lead to biomarkers.

References

1. Hanash S (2002) HUPO initiatives relevant to clinical proteomics. Mol Cell Proteomics 3:298–301
2. Meyer HE, Klose J, Hamacher M (2003) HBPP and the pursuit of standardisation. Lancet Neurol 2:657–658
3. Hamacher M, Klose J, Rossier J, Marcus K, Meyer HE (2004) "Does understanding the brain need proteomics and does understanding proteomics need brains?"–Second HUPO HBPP Workshop hosted in Paris. Proteomics 4:1932–1934

4. Hamacher M, Meyer HE (2005) Great mood in proteomics: Beijing and the HUPO human brain proteome project. Proteomics 5:334–336
5. Beyreuther K, Einhäupl KM, Förstl H, Kurz A. (2002) Prävalenz von Demenzen. In: Demenzen – Grundlagen und Klinik, pp. 2–7, Thieme Verlag, Stuttgart
6. Wilkins MR, Appel RD, Van Eyk JE, Chung MC, Gorg A, Hecker M et al (2006) Guidelines for the next 10 years of proteomics. Proteomics 6:4–8
7. Carr S, Aebersold R, Baldwin M, Burlingame A, Clauser K, Nesvizhskii A (2004) The need for guidelines in publication of peptide and protein identification data: Working Group on Publication Guidelines for Peptide and Protein Identification Data. Mol Cell Proteomics 3:531–533
8. Hamacher M, Apweiler R, Arnold G, Becker A, Blüggel M, Carrette O et al (2006) HUPO Brain Proteome Project: summary of the pilot phase and introduction of a comprehensive data reprocessing strategy. Proteomics 6:4890–4898
9. Hamacher M, Stephan C, Meyer HE, Eisenacher M (2009) Data handling and processing in proteomics. Expert Rev Proteomics 6:217–219
10. Hamacher M, Eisenacher M, Meyer HE, Stephan C (2008) Proteomics today: bioinformatics at its best. Proteomics and Bioinformatics – an inseparable couple. Proteomics 8:4616–4617
11. Stephan C, Eisenacher M, Kohl M, Meyer HE (2010) Proteomics data collection (ProDaC): publishing and collecting proteomics data sets in public repositories using standard formats. Methods Mol Biol 604:345–368
12. Eisenacher M, Martens L, Barsnes H, Hardt T, Kohl M, Häkkinen J et al (2009) Proteomics data collection – 5th ProDaC Workshop: 4 March 2009, Kolympari, Crete, Greece. Proteomics 9:3626–3629
13. Eisenacher M, Martens L, Hardt T, Kohl M, Barsnes H, Helsens K et al (2009) Getting a grip on proteomics data – Proteomics Data Collection (ProDaC). Proteomics 9:3928–3933
14. Martens L, Hermjakob H, Jones P, Adamski M, Taylor C, States D et al (2005) PRIDE: the proteomics identifications database. Proteomics 5:3537–3545
15. Jones P, Cote RG, Martens L, Quinn AF, Taylor CF, Derache W et al (2006) PRIDE: a public repository of protein and peptide identifications for the proteomics community. Nucleic Acids Res 34:D659–D663 (Database issue)
16. Stephan C, Reidegeld KA, Hamacher M, van Hall A, Marcus K, Taylor C et al (2006) Automated reprocessing pipeline for searching heterogeneous mass spectrometric data of the HUPO Brain Proteome Project pilot phase. Proteomics 6:5015–5029
17. States DJ, Omenn GS, Blackwell TW, Fermin D, Eng J, Speicher DW, Hanash SM (2006) Challenges in deriving high-confidence protein identifications from data gathered by a HUPO plasma proteome collaborative study. Nat Biotechnol 24:333–338
18. Chamrad D, Meyer HE (2005) Valid data from large-scale proteomics studies. Nat Methods 2:647–648
19. Mueller M, Martens L, Reidegeld KA, Hamacher M, Stephan C, Blüggel M et al (2006) Functional annotation of proteins identified in human brain during the HUPO Brain Proteome Project pilot study. Proteomics 6:5059–5075
20. Martens L, Müller M, Stephan C, Hamacher M, Reidegeld KA, Meyer HE et al (2006) A comparison of the HUPO Brain Proteome Project pilot with other proteomics studies. Proteomics 6:5076–5086
21. Omenn GS, States DJ, Adamski M, Blackwell TW, Menon R, Hermjakob H et al (2005) Overview of the HUPO Plasma Proteome Project: results from the pilot phase with 35 collaborating laboratories and multiple analytical groups, generating a core dataset of 3020 proteins and a publicly-available database. Proteomics 5:3226–3245
22. Martens L, Van Damme P, Van Damme J, Staes A, Timmerman E, Ghesquiere B et al (2005) The human platelet proteome mapped by peptide-centric proteomics: a functional protein profile. Proteomics 5:3193–3204
23. Omenn GS, Smith LT (1978) A common uptake system for serotonin and dopamine in human platelets. J Clin Invest 62:235–240
24. Kim YH, Marcus K, Gringberg LT, Goehler H, Wilftang J, Stephan C et al (2009) Toward a Successful Clinical Neuroproteomics The 11th HUPO Brain Proteome Project Workshop. Proteomics Clin Appl 3:1012–1016
25. Kapp EA, Schutz F, Connolly LM, Chakel JA, Meza JE, Miller CA et al (2005) An evaluation, comparison, and accurate benchmarking of several publicly available MS/MS search algorithms: sensitivity and specificity analysis. Proteomics 5:3475–3490

Chapter 15

Data Management and Data Integration in the HUPO Plasma Proteome Project

Gilbert S. Omenn

Abstract

The Human Plasma Proteome Project (HPPP) is an international collaboration coordinated by the Human Proteome Organisation (HUPO). Its Pilot Phase generated the 2005 *Proteomics* special issue "Exploring the Human Plasma Proteome" (Omenn et al. Proteomics 5:3226–3245, 2005) and a book with the same title (Omenn GS (ed) (2006) Exploring the human plasma proteome. Wiley-Liss, Weinheim, pp 372). Data management for that Pilot Phase included collection, integration, analysis, and dissemination of findings from participating laboratories and data repositories. Many investigators face the same challenges of integration of data from complex, dynamic serum, and plasma specimens. The PPP workflow assembled a representative Core Dataset of 3,020 protein identifications, overcoming ambiguity and redundancy in the heterogeneous contributed identifications and redundancy and updates in the protein sequence databases. The results were made available with alternative thresholds from the University of Michigan, yielding a range of numbers of protein identifications. Data were submitted to EBI/PRIDE and to ISB/PeptideAtlas. The current phase of the PPP employs Proteome Xchange to link submission of well-annotated primary datasets to EBI/PRIDE, distributed file sharing by Tranche/Proteome Commons.org, and reanalysis from the primary raw spectra at ISB/PeptideAtlas. Such human plasma proteome datasets are available for data mining comparisons with the proteomes of other organs and biofluids in health and disease.

1. Introduction

The database of 3,020 protein identifications from the large collaborative Human Plasma Proteome Project (HPPP)(1, 2), organized as the first initiative of the Human Proteome Organisation (HUPO) in 2002, has been widely utilized and has been cited 252 times as of 21 January 2010. Thus, it is desirable for users to understand its organization and especially the data management and data integration features that are critical to cross-comparison of findings from different studies. The challenges of data management and

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_15,

data integration across dozens of participating laboratories remain highly relevant in the field, especially the objective of obtaining full annotation of samples. The HUPO Protein Standards Initiative (PSI) has addressed many aspects of standardization of data formats and data submission (psidev.sf.net).

Compromises typically must be accepted on the level of detail of experimental methodology, starting with the protocol and variation in collection and processing of blood specimens; the choices of reference specimens; the capture of information that is embedded in free text; the uncertainty of identifications when laboratories are mandated to push the limitation of detectability; the parameters used by various mass spectrometry instruments; the design of data storage systems; and the choice of sequence database (and version) used for analysis (see Note 1).

This chapter describes the guidelines for data submission, the creation of the data repository, the array of specially prepared reference specimens, the handling of MS/MS data, the data integration workflow algorithm, and the consolidation and annotation of datasets from 18 laboratories that submitted MS/MS findings on the HUPO PPP reference specimens. Results from these and other platforms were published in (1).

2. Methods

2.1. Creating a Data Repository

The HPPP adopted a data model (3) focused on identifications of whole proteins with a high-level, concise description of experimental results and a minimum of data input, transmission, and reformatting for the collaborating submitters. Guidance specified protein accession numbers and names, binary description of the confidence of the protein identification (with common parameters), lists of identified peptides, and free text descriptions of experimental protocols, estimates of relative abundance, and any information about posttranslational modifications (see Note 2). Identification datasets were stored as peptide lists, reflecting the fact that many laboratories applied intact protein fractionation before tryptic digestion and mass spectrometry. During the PPP Pilot Phase, we later requested peak lists and raw spectra in the instrument native format. Participating laboratories used different search databases and different algorithms to assemble protein identifications from the search output. The guidance anticipated the guidelines subsequently mandated by *Molecular and Cellular Proteomics* (4) and other journals, and the publication ((3), Table 1) explicitly compared the PPP data model with the Carr et al. guidelines (4). Laboratories received two distinct identifiers: a numeric public identifier used for interactions with the submission centers and other laboratories, and a three-character private

code known only to the laboratory and the central data analysis group, used to create data surveys without disclosing the identity of submitters ((1), Tables 1 and 2).

The data repository was built with a Structured Query Language (SQL) relational database server, an intermediate structure presenting an exact copy of the data submitted, and the main data structure designed to hold the integrated project data. The database captured three sets of protein identifiers from the same experiment: (1) protein IDs made by data producers, in the entity identification; (2) results of peptide list searches performed by the data integration center, in the entity ProteinByPeptides; and (3) analyses by others, through the MsRun branch of the database. The entire repository structure is available in Fig. 1 of Adamski et al. (3). Data were transmitted primarily as Excel or Word documents, even though assistance was available and promoted to prepare XML schema-based file formats.

2.2. PPP Reference Specimens

The investigators collectively decided to have a range of reference specimens to be able to address alternatives in anti-coagulation, compare plasma versus serum, and obtain preliminary results on ethnic group differences. BD Diagnostics prepared the requested specimens from pairs of donors of Caucasian–American (BD1), African–American (BD2), and Asian–American (BD3) backgrounds, after informed consent (1). Sets of four specimens were

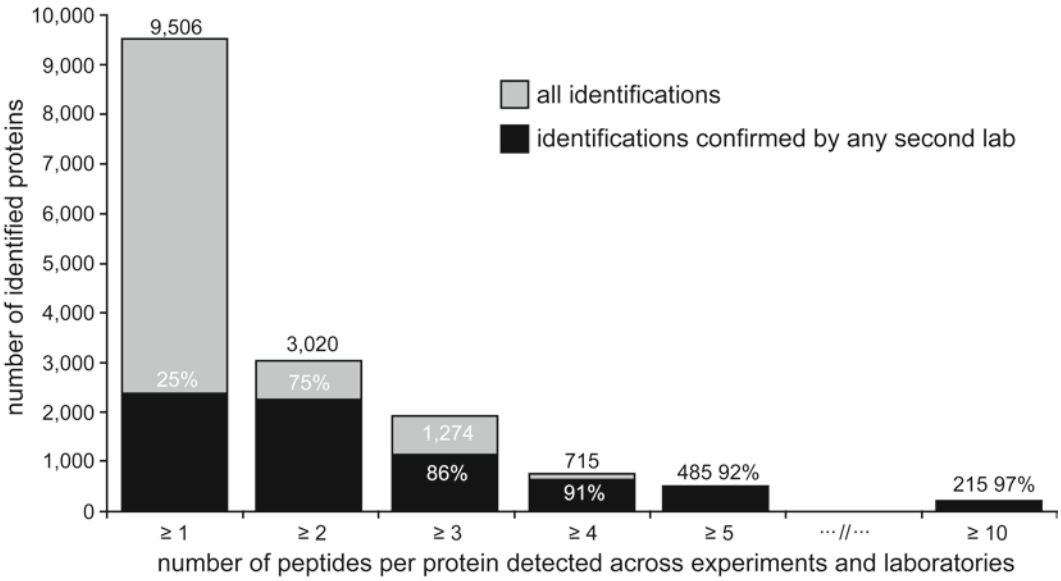


Fig. 1. Distribution of MSMS and FTICR/MS protein identifications as a function of the number of peptides detected per protein (from Fig. 4 from ref. (3)). The dark portion of each bar represents the percentage confirmed in at least one additional laboratory.

prepared: serum, EDTA-plasma, heparin-plasma, and citrate-plasma (making 12, see Note 3). A similar set of four specimens was prepared by the Chinese Academy of Medical Sciences (CAMS). Finally, the UK National Institute of Biological Standards and Control made available a lyophilized citrated plasma prepared from a pool of 25 human donors (NIBSC). Of 55 laboratories that originally committed to participate, 41 requested and received the BD1 specimens, 27 the BD2 and BD3 sets, 15 the CAMS set, and 45 the NIBSC sample. Laboratories varied markedly on how many of the specimens they actually analyzed, and how extensively they fractionated and analyzed each specimen.

2.3. Inference from Peptides to Proteins

MS/MS spectra yield sequence information for peptides, primarily but not only tryptic peptides, to be matched against protein databases. Often the search returns a cluster of proteins, all of which contain the same set of matching peptides. For a uniformly collected dataset, probabilities are readily applied with PeptideProphet/ProteinProphet (5). However, the extremely heterogeneous, collaborative nature of this dataset, with various instruments and various search engines (6), required an alternative, which is shown in Subheading 2.4. The concept of this workflow algorithm is that proteins most likely truly present are more likely to be detected across independent experiments and to have been annotated more extensively. The outcome is the choice of one protein as the representative entry from several overlapping clusters of equivalent protein identifications. As discussed later, we retained the full list to permit comparisons with proteins identified by others using different integration strategies (or none at all).

2.4. HPPP Data Integration Workflow Algorithm from Adamski et al. (3)

1. Assemble peptide sequence lists retaining all source information
2. Search the peptide lists against the IPI v2.21 database (periodically updated). Require 100% identity between the sequences; disregard flanking residues.
3. Select one representative protein from each cluster of equivalent protein matches, or intersection of several clusters.
4. Each protein entry in the reference database receives three integer scores:
   (a) Number of labs reporting a peptide sequence list containing a sequence which maps to a cluster, including this protein
   (b) Number of distinct experiments (labs x specimens x protocols) reporting a peptide list with this protein
   (c) Number of identifications (labs x specimens x protocols x clusters) for clusters, including this protein. Choose cluster member with largest value of (a). In case of tie scores, proceed to (b), (c), and (d) to (h).

(d) Prefer proteins that are products of a well-described gene (not "hypothetical," "similar to," etc.) from EnsEMBL.

(e) Well-described protein-product of any gene

(f) Well-described protein not assigned to any gene

(g) Protein not assigned to any gene, described only as a fragment or similar to, etc.

(h) Select the protein having the lower IPI number (in IPI v2.21).

Score (a) counts each laboratory only once, no matter from how many specimens or with how many different peptide sequence lists the laboratory identified this protein. Next in importance, score (b) counts the number of independent experiments in which the protein was identified. Score (c) counts all reported peptide sequence lists, even if several results are from the same experiment. Criteria (d-g) indicate the level of annotation for each database entry, facilitating the selection of the best-described proteins.

2.5. Summary of Collaborative Data

The 18 laboratories that contributed MS/MS data (MALDI, LC-ESI, and FT-ICR-MS) submitted a total of 12,667 distinct protein accession numbers, using the IPI, SwissProt, and NCBInr databases, with IPI version 2.21 (5) the standard we chose for this project. Over time, new versions of IPI appeared, a problem for any longitudinal study or even a snapshot study with several to many months from data collection to publication. We locked in and referred back to v2.21. After integration, we had 9,504 unique proteins of ≥6 amino acids in length based on spectra for one or more peptides, and 3,020 proteins based on two or more peptides (see Note 4). The article (3) described in great detail the thresholds individual scientists might apply to the publicly available primary datasets. In the course of the project, we held a Jamboree Workshop (June 2004) at which participating scientists and teams from the various laboratories and informatics specialists worked together on the primary data. Several labs agreed to standardize their LCQ-MSMS SEQUEST searches to use Xcorr ≥1.9, 2.2, 3.75 for 1+, 2+, and 3+ ions, respectively, plus deltaCn ≥0.1 and Rsp ≤4 as the threshold for "high confidence" sequences of tryptic peptides. Note that delta Cn and Rsp are not always employed; they increase stringency and confidence of protein identifications. The number of lab-reported high-confidence identifications ranged from 21 to 789.

We gave emphasis to cross-laboratory confirmation of identifications (see refs. (1–3) for many details). Figure 1 shows the numbers of protein identifications according to the number of peptides per protein detected across experiments and laboratories; the dark subset in each bar represents the proportion confirmed in a second lab.

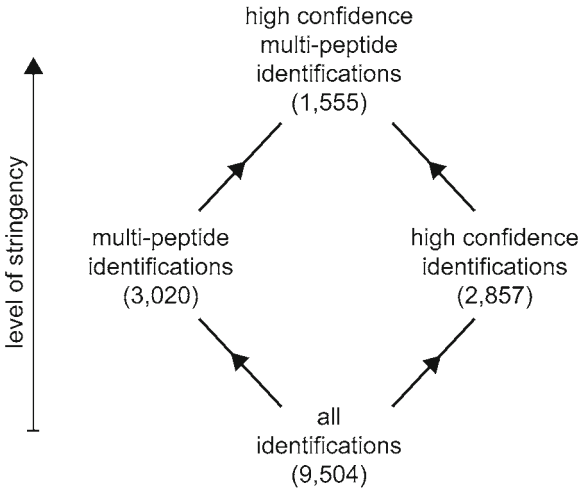


Fig. 2. Alternative protein identification lists with different inclusion criteria from the HUPO Plasma Proteome Project (from Fig. 5 of ref. (3)).

We presented a schema in Fig. 2 in the form of a diamond-shaped parallelogram with sets of proteins. The entire post-integration list of 9,504 ("all identifications") was divided into two more stringent categories, identifications called "high confidence" by the participating investigators (2,857 proteins) and 3,020 proteins for which two or more distinct peptides were reported across all 18 laboratories reporting MSMS results, following integration. The final point in the diamond represents "high-confidence multi-peptide identifications" with 1,555 proteins. This latter set was used for comparison with the number of identifications in the HUPO Human Brain Proteome report (7).

We also published an even more restricted set of 889 proteins (8) in which we applied the Bonferroni adjustment for multiple statistical comparisons of the protein match with $p > 0.95$ among 43,730 IPI entries, as well as an adjustment for protein length to account for more opportunities for matching peptides the longer the protein sequence. The Bonferroni is a very common adjustment in large-scale transcriptomics analyses, but it is seldom utilized in proteomics. Given the many families of proteins, it is likely that this analysis, treating each protein as an independent observation, is overly stringent.

2.6. False-Positive Identifications

False-positive (FP) identifications are a widely acknowledged problem. A standard solution is to match the peptide sequences against a reversed sequence version of the protein database, such that each "reverse hit" would be a representation for a FP. In our HPPP analysis, we were dealing, as noted, with highly heterogeneous datasets and a variety of search engines (9) and database matches, so we applied a different concept. We posited that FP

and true-positive identifications would show opposite behavior as one accumulates large numbers of peptide IDs. FPs would be expected to accumulate roughly proportional to the total peptide IDs, without two or more FP peptide IDs coinciding on the same database entry at any rate greater than random. In contrast, for a protein which is truly present at a detectable concentration in the specimen, increased sampling should identify the same peptides mapping to the same correct database entry, as described in ref. (1). The actual criteria varied across the 18 laboratories.

2.7. Correlating Immunoassay Quantitation of Proteins with Estimates of Abundance Based on Number of Peptides

The HPPP had a specific subproject on quantitative estimates of protein concentrations, utilizing immunoassays from several laboratories (see also Note 5). This topic received a lot of attention at the Jamboree Workshop and in the subsequent publication by Haab et al. (10). The peptide counting method, the average number of different peptides found for that IPI number across the labs reporting that IPI identification, may be regarded as a precursor to the now-popular label-free spectral counting approach. A major challenge is figuring out what epitopes account for the immunoassay results and which of multiple proteins in a family or cluster may cross-react with the antibody, or not do so. The conclusion, with appropriate caveats, was that we obtained a log-linear relationship between immunoassay-based concentrations and number of peptides detected for a wide concentration range of proteins (see Fig. 6b in (1)). The correlation coefficient (of the log-linear relationship) for the total number of peptides matching that protein, based on quantitative immunoassays of 49 proteins among the 3,020 protein dataset, was $r = 0.86$ (1). These proteins cover quite a range of eight orders of magnitude in concentration.

2.8. Comparisons of Protein Identifications Across Different Studies

In the overview paper for the Plasma Proteome Project, we compared the protein identifications of the HPPP with those of several other authors ((1), Table 4). The amount of overlap between and among these reports was not high, reflecting especially incomplete detection of low abundance proteins, as well as uncertain numbers of false positives. An important methodological point in these comparisons is the fact that different investigators use different methods for integration of multiple matches or clusters, if they do integration at all. We found it necessary to go back to our larger datasets, both the unintegrated list of 5,102 proteins for the 3,020 core dataset and the 9,504 integrated IDs, including single peptide hits, to pick up additional matches with these datasets from different sources. Many biologically significant annotations can be generated with data mining of the HPPP (see refs (1, 2)) for numerous examples).

Comparisons of other organ proteomes with the plasma (or serum) proteome remain to be pursued. Such comparisons have

been frequent statements of intent across the HUPO Initiatives, including liver, brain, kidney/urine, and cardiovascular. As noted, there has been a comparison of plasma and brain (7) and a comparison of plasma and the salivary fluid proteome (11), using the HUPO PPP for the plasma comparisons.

The PPP is a major component of the Human Plasma PeptideAtlas (12, 13). It is now desirable to utilize the entire complement of studies in the PeptideAtlas, which has the very special advantage that all of these datasets have been re-analyzed from the raw spectra at the Institute for Systems Biology with the TransProteomicPipeline, eliminating numerous sources of variation due to instrument settings, search engine parameters, and database matching and integration. Deutsch et al. published their first Human Plasma PeptideAtlas as part of the HPPP Pilot Phase publication; they identified 960 proteins from datasets that partially overlapped the datasets contributed to the HPPP (14).

In an update of the Human Plasma PeptideAtlas, Farrah et al. have reanalyzed the data from 14 of the reporting laboratories in the Pilot Phase of the HPPP using the latest TPP pipeline and the SpectraST spectral library searching tool (15) searched against the latest NIST human library (version 3.0; available at http://www.peptideatlas.org/speclib/). Applying extremely stringent PeptideProphet FDR thresholds, they identified 10,893 unique peptides and inferred 1,186 proteins at a 5% decoy-estimated protein false-discovery rate, and 9,807 peptides and 930 proteins at a 1% protein FDR; the entire Human Plasma PeptideAtlas, including additional HPPP current phase submissions, has 2,249 proteins at 1% FDR as of January, 2010 [data provided by Drs. Terry Farrah and Eric Deutsch].

2.9. The Next Phase, now Current Phase, of the HUPO HPPP

Under current cochairs Ruedi Aebersold, Mark Baker (succeeding Young-Ki Paik), and Gil Omenn, the HUPO HPPP continues with the intent to collect large, well-annotated datasets on human plasma in normal individuals and as part of disease-oriented studies with both organ and plasma specimen analyses (16, 17) (also see Note 6). The aims of the PPP-2 are (1) to stimulate submission of high-quality, large datasets of human plasma proteome findings with advanced technology platforms; (2) to establish a robust, value-added informatics scheme involving EBI/PRIDE, University of Michigan/ProteomeCommons/Tranche, and Institute for Systems Biology/PeptideAtlas; and (3) to collaborate with other HUPO organ-based and disease-related initiatives to make plasma the common pathway for biomarker development and application.

The initial datasets and the PRIDE Web site were demonstrated at the HPPP Workshop at the Amsterdam HUPO Congress (18). The data processing and data mining scheme calls for use of the Proteome Xchange (Fig. 3): submission of the fully annotated

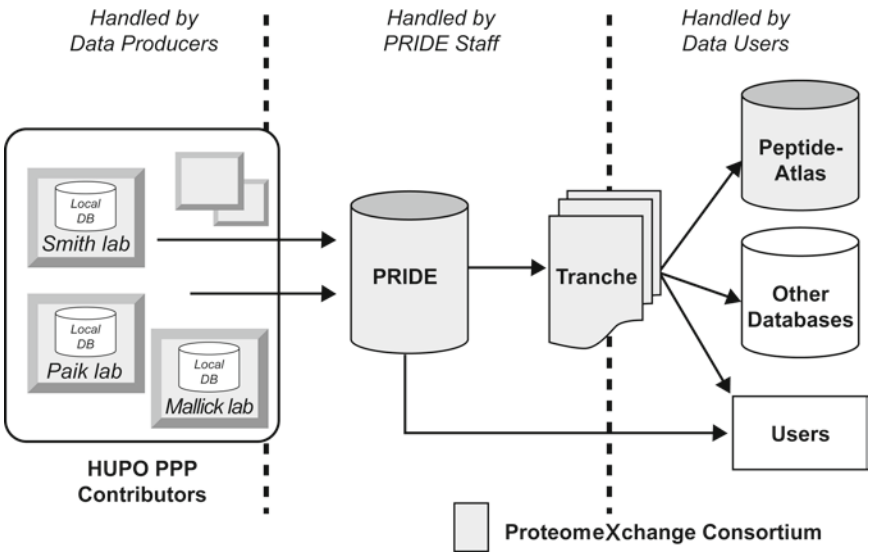


Fig. 3. Scheme for Proteome Xchange, involving EBI/PRIDE, UM Tranche/ProteomeCommons.org, and ISB/PeptideAtlas, with further distribution of dataset files to the interested proteomics and bioinformatics community.

experimental datasets with the investigator's interpretations to EBI/PRIDE; automatic transfer to Tranche/ProteomeCommons.org for distributed file sharing globally; and automatic transfer to PeptideAtlas for full reanalysis from the raw spectra across all submissions to the new HPPP. A major element is the development of heavy-labeled proteotypic peptides based on N-glycosite peptide isolation, which will be a major resource for many kinds of proteomics studies, including high-throughput-targeted proteomics. We plan on consolidated analyses with other plasma proteome datasets already in the PeptideAtlas or received subsequently. Tranche will also contribute these datasets to the Peptidome at NIH/NCBI and to the GPMdb in Canada, and to any scientists requesting such resources. It is expected that all HUPO initiatives will contribute to the large-scale Gene-Centric Human Proteome Project now under discussion (19).

3. Notes

1. Highly collaborative studies utilizing a range of technology platforms and a variety of specimens are hard to fit into a tight uniform protocol. Dealing with heterogeneous datasets requires special procedures and cross-checking, which can be enhanced by targeted data mining and data integration. Some of these features are well demonstrated in the HUPO HPPP.

2. Gaining sufficient annotation of preanalytical variables (20), fractionation of specimens, MSMS analytical and search engine variables, and database matching procedures is another major challenge, with the content often less than desired.
3. Based on the results of the HPPP Pilot Phase, we prefer and recommend the use of plasma over serum for proteomics analyses and the use of EDTA-plasma among the plasma options (1, 21).
4. Protein lists from the HPPP are available at: www.ebi.ac.uk/PRIDE for HUPO HPPP and individual lab submissions; http://www.bioinformatics.med.umich.edu/hupo/ppp includes protein lists for the 3,020 protein core dataset and its peptides, plus its corresponding 5,102 protein matches before integration. The 9,504 and 889 protein lists are also posted in this site; and embedded in www.peptideatlas.org, human plasma proteome datasets from multiple sources (not all of the HPPP datasets were included, see (14)).
5. One of the many interesting side analyses was the matching of our peak lists for six small datasets against microbial genomes in the NCBI Microbial (nonhuman) GenBank (June 2004 release), using X!Tandem for RefSeq protein sequence identification. We found notable bacterial and mycobacterial matches (1), a clue to the usefulness of this approach for the now very popular work on metagenomics of the huge microbial populations who share our bodies and influence many physiological functions.
6. Ensuring intended comparisons of results across pairs or multiples of large complex experimental projects has been frustrating and remains an important goal. Analysis for differences requires replicates to demonstrate the extent of congruence of findings upon repeat analysis of the same specimen.

Acknowledgments

I thank all the investigators and core staff for the Pilot Phase of the HUPO HPPP (see (1)) and especially the bioinformatics team headquartered at the University of Michigan who were coauthors on the original description of the Data Management and Data Integration plan for this project: Marcin Adamski, Thomas Blackwell, Rajasree Menon, and David States of the University of Michigan and Lennart Martens, Chris Taylor, and Henning Hermjakob of the European Bioinformatics Institute (see (3)). I thank Eric Deutsch and Terry Farrah of the Institute for Systems Biology for the current data from the PeptideAtlas Human Plasma Proteome build and for review of the manuscript.

References

1. Omenn GS, States DJ, Adamski MR, Blackwell TW, Menon R, Hermjakob H et al (2005) Overview of the HUPO plasma proteome project: results from the pilot phase with 35 collaborating laboratories and multiple analytical groups, generating a core dataset of 3020 proteins and a publicly-available database. Proteomics 5:3226–3245
2. Omenn GS (ed.) (2006) Exploring the human plasma proteome. Wiley-Liss, Weinheim, p 372
3. Adamski M, Blackwell T, Menon R, Martens L, Hermjakob H, Taylor C et al (2005) Data management and preliminary data analysis in the pilot phase of the HUPO Plasma Proteome Project. Proteomics 5:3246–3261
4. Carr S, Aebersold R, Baldwin M, Burlingame A, Clauser K, Nesvizhskii A (2004) The need for guidelines in publication of peptide and protein identification data: Working Group on Publication Guidelines for Peptide and Protein Identification Data. Mol Cell Proteomics 3:531–533
5. Nesvizhskii AI, Keller A, Kolker E, Aebersold R (2003) A statistical model for identifying proteins by tandem mass spectrometry. Anal Chem 75:4646–4658
6. Kersey PJ, Duarte J, Williams A, Karavidopoulou Y, Birney E, Apweiler R (2004) The International Protein Index: an integrated database for proteomics experiments. Proteomics 4:1985–1988
7. Hamacher M, Apweiler R, Arnold G, Becker A, Blüggel M, Carrette O et al (2006) HUPO brain proteome project: summary of the pilot phase and introduction of a comprehensive data reprocessing strategy. Proteomics 6:4890–4898
8. States DJ, Omenn GS, Blackwell TW, Fermin D, Eng J, Speicher DW, Hanash SM (2006) Challenges in deriving high-confidence protein identifications from data gathered by HUPO plasma proteome collaborative study. Nat Biotech 24:333–338
9. Kapp EA, Schütz F, Connolly LM, Chakel JA, Meza JE, Miller CA et al (2005) An evaluation, comparison and accurate benchmarking of several publicly-available MS/MS search algorithms: sensitivity and specificity analysis. Proteomics 5:3475–3490
10. Haab BB, Geierstanger BH, Michailidis G, Vitzthum F, Forrester S, Okon R et al (2005) Immunoassay and antibody microarray analysis of the HUPO PPP reference specimens: systematic variation between sample types and calibration of mass spectrometry data. Proteomics 5:3278–3291
11. Yan W, Apweiler R, Balgley BM, Boontheung P, Bundy JL, Cargile BJ et al (2009) Systematic comparison of the human saliva and plasma proteomes. Proteomics Clin Appl 3:116–134
12. Deutsch EW (2010) The PeptideAtlas Project. Methods Mol Biol 604:285–296
13. Deutsch EW, Lam H, Aebersold R (2008) PeptideAtlas: a resource for target selection for emerging targeted proteomics workflows. EMBO Rep 9:429–434
14. Deutsch EW, Eng JK, Zhang H, King NL, Nesvizhskii AI, Lin B et al (2005) Human Plasma PeptideAtlas. Proteomics 5:3497–3500
15. Lam H, Deutsch EW, Eddes JS, Eng JK, King N, Stein SE, Aebersold R (2007) Development and validation of a spectral library searching method for peptide identification from MS/MS. Proteomics 7:655–667
16. Omenn GS, Aebersold R, Paik YK (2007) HUPO plasma proteome project 2007 workshop report. Mol Cell Proteomics 6:2252–2253
17. Omenn GS, Menon R, Adamski M, Blackwell T, Haab BB, Gao W, States DJ (2007) The human plasma and serum proteome. In: Thongboonkerd V (ed) Proteomics of human body fluids: principles, methods, and applications. Humana Press, Totowa, NJ, pp 195–224
18. Omenn GS, Aebersold R, Paik YK (2009) 7th HUPO world congress of proteomics: launching the second phase of the HUPO plasma proteome project (PPP-2) 16-20 August 2008, Amsterdam, The Netherlands. Proteomics 9:4–6
19. HUPO – the Human Proteome Organisation. (2010) A Gene-centric Human Proteome Project. Mol Cell Proteomics 9:427–429
20. Gelfand C, Omenn GS (2010) Pre-analytical variables for plasma and serum proteome analyses. In Ivanov A, Lazarev A (eds), Sample preparation in biological mass spectrometry. Springer, NY. (in press)
21. Rai AJ, Gelfand CA, Haywood BC, Warunek DJ, Yi J, Schuchard MD et al (2005) HUPO plasma proteome project specimen collection and handling: towards the standardization of parameters for plasma proteome samples. Proteomics 5:3262–3277

Chapter 16

Statistics in Experimental Design, Preprocessing, and Analysis of Proteomics Data

Klaus Jung

Abstract

High-throughput experiments in proteomics, such as 2-dimensional gel electrophoresis (2-DE) and mass spectrometry (MS), yield usually high-dimensional data sets of expression values for hundreds or thousands of proteins which are, however, observed on only a relatively small number of biological samples. Statistical methods for the planning and analysis of experiments are important to avoid false conclusions and to receive tenable results. In this chapter, the most frequent experimental designs for proteomics experiments are illustrated. In particular, focus is put on studies for the detection of differentially regulated proteins. Furthermore, issues of sample size planning, statistical analysis of expression levels as well as methods for data preprocessing are covered.

1. Introduction

Sometimes, today's bioanalytical research is accompanied by the phantasm that the more data is recorded within an experiment the bigger will the cognition drawn from this experiment be. This phantasm is stimulated by the new technological possibilities of measuring simultaneously the expression levels of thousands of molecules as well as by the opulent information stored in databases. The good intentions behind high-throughput experiments are, however, opposed by the fact that the probability of wrong conclusions increases with the number of hypothesis stated in the context of an experiment. Patterson (1) consequently named data analysis the "Achilles heel of proteomics." Preconditions for tenable inferences are well-defined study problems, adequate experimental designs and the correct statistical methods for data analysis.

One of the challenges for the analysis of data from high-throughput experiments is their high-dimensionality, i.e., many

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_16,

features are observed on only a small number of biological samples or individuals. Historically, statistical methods for the analysis of high-dimensional data were refined or even newly developed for gene expression data from DNA microarrays. Because studied problems in proteomics are often very similar to those in genomics, many of these statistical methods can easily be employed for protein expression data, too. An essential difference between gene and protein expression data, however, results from the different bioanalytical technologies which are used for measuring expression levels. Therefore, different ways of data preprocessing are necessary.

A particular question of proteomics is the comparison of expression levels between different types of biological samples, for example, between samples of mucosa and tumor tissue. In Subheading 2, experimental designs for such problems as well as issues of sample size planning are detailed. The presented designs are applicable when comparing two or more independent or dependent categories of biological samples. An example for dependent categories are repeated measurements of the same samples at different points in time. Furthermore, models which include more than one experimental factor are illustrated. Subheading 3 presents necessary steps for the preprocessing of expression levels recorded by mass spectrometry (MS) of 2-dimensional gel electrophoresis (2-DE). Preprocessing is necessary for making the recording of different biological samples comparable. In Subheading 4, the statistical concepts of hypothesis testing and of *p*-value adjustment for multiple testing are detailed, as well as the quantification of expression ratios.

2. Designs and Planning of Experiments

A classical laboratory experiment consists of measuring a dependent (or endogenous) variable under the influence of other independent (or exogenous) experimental factors. In proteomics experiments, the dependent variable is usually the expression level of a protein (i.e., a metric variable), whereas the independent variables may be either categorical (e.g., group membership or disease state) or also metric (e.g., age). In the following, we regard different experimental designs, starting with the most simple one, which is given by studying one experimental factor with two categories, and turn then toward several further aspects of experiment planning, such as sample size calculation and randomization. We regard especially experimental designs for 2-D DIGE gels. Further designs for experiments with this particular type of gels also are presented in (2, 3).

2.1. One Experimental Factor with Two Categories

One of the most frequent problems in proteomics is the comparison of expression levels from two distinct types of biological samples, for example, cell lines under two different experimental conditions or tissue samples from diseased and healthy individuals. These experiments have thus only one categorical experimental factor with two categories. In the just mentioned examples, all samples are independent from each other. One can, however, also consider the case of dependent biological samples, for example, tumor tissue and mucosa from the same individual or a cell line sample measured at two different points in time. The goal of such experiments is to find those proteins which are significantly up- or downregulated in the one category of samples compared to the other one. The preprocessing and analysis of the resulting data is described in Subheadings 3 and 4. In the following, the concrete handling of these designs within 2-DE and MS experiments is given.

In a classical 2-DE approach, simply one gel is prepared per sample. When using the so called DIGE approach (4) instead (where two or more samples, labeled by different fluorescent dyes, can be studied on the same gel), the experimentator has to distinguish between experiments with independent and those with dependent samples. In the latter case, i.e., when two samples per experimental unit (individual) are studied, both samples can be prepared onto the same gel, and the ratios of expression levels are used for statistical analysis. When regarding independent samples instead, an internal standard (comprised of a mixture of all samples included in the experiment) is additionally incorporated, and the ratios of expression levels from the true samples to those from the standard are analyzed. In the case of independent samples, two types of experimental settings can be considered when using DIGE gels. In the first setting, each sample is prepared together with the internal standard on one gel. This design is especially recommended when samples sizes are very different for the two categories of the experimental factor. Particularly, when samples sizes are equal, it is also possible to put two samples – each representing one of the two categories of the experimental factor – together with the internal standard onto one gel (three different fluorescent dyes are used, here). This second setting needs less gels than the first one; it is, however, necessary that the two different samples for each gel are assigned together randomly. Procedures for randomization are described below in this section.

MS experiments are performed very similar. In classic approaches, each sample is recorded in one MS run. Newer approaches which incorporate an isotope-labeling can analyze two samples – labeled by masses of different weight – in one run (5, 6). When studying dependent samples using such isotope-labeling approaches, again ratios of observed intensities are taken for analysis. When having samples from two independent groups,

it is again necessary to match two samples – one of each group – randomly for one MS run.

2.2. One Experimental Factor with More than Two Categories

In some experiments which study the influence of one experimental factor, more than two categories are studied. Stühler et al. (7), for example, compared expression levels in brains of mice at different developmental stages, embryonic, juvenile, and adult. When using DIGE gels, it is recommended to put always only one sample together with an internal standard onto one gel. Only if combinations of categories are assigned randomly to a gel, it is also possible to put more than one sample onto the same gel.

2.3. Two or More Experimental Factors

Let us next regard experiments, where two categorical experimental factors are to be studied. It is then necessary to distinguish between designs with a cross-classification and those with hierarchical classification.

In a cross-classified experiment, each category of the one factor is combined with each category of the other factor. Assume, for example, that it is desired to observe the effect of two different treatments A and B on the expression levels in samples from a certain cell line. We have thus two experimental factors, A and B, each with two categories, treated and not treated. One can then prepare the samples under four different conditions: (1) not treated, (2) only treated with A, (3) only treated with B, and (4) treated with A and B.

In a hierarchical design, not each combination of categories from the two factors is studied. Let us consider a study with two cohorts of patients, where each cohort is treated with a different therapy (thus, factor A has two categories: therapy one and two). As second factor B, consider "diabetes mellitus status," with the two categories "present" and "not present." It is obvious, that a patient can neither be studied under each category of factor A nor under each category of factor B, here.

One can consider course experimental designs with even more than two experimental factors, also in cross-classified and hierarchical settings, however, these designs are seldom studied in proteomics.

2.4. Repeated Measures Designs

A special type of experiments is when expression levels are studied multiple times on the same sample, but under different conditions. These designs are called repeated measures designs. Basically, the above detailed design with one experimental factor of two categories is a repeated measures design if the samples from the two categories are dependent, for example, if expression levels are studied in tumor and mucosa of the same patients. A frequently used repeated measures design is usually given if one experimental factor is the time, where the categories of this factor are different points in time. Sitek et al. (8) studied, for example,

cell lines at several hours after the treatment. The dependence structure of such measurements has to be taken into account in the analysis of such experiments.

2.5. Randomization

In all of the above-described experiments, the experimenter is usually only interested in the effects of the intentionally incorporated factors. It can, however, happen that a studied factor is overlapping with another uninteresting one. Assume, for example, that protein expression in the liver of mice from a treatment group is compared with that of an untreated control group. And suddenly, the experimentator (after he has spent a lot of time with collecting and preparing samples) gets aware that all treated mice were male and all untreated individuals were female. Is the experimentator then studying the effect of treatment or of that of gender? (Yes, such disasters happen!)

How can such mistakes be avoided in the planning of an experiment? Particularly, when studying treatment effects, the probability of incorporating undesired overlapping effects can be diminished by assigning the samples to the different categories of the treatment factor randomly (by the way: "randomly" is not the same as "arbitrarily"!). An example is given in the notes section.

2.6. Sample Size Calculations

Because sample and gel preparation is expensive and time consuming, an important question when planning a proteomics experiment is how many samples are needed for a particular experiment. This question can be stated more precisely by the question "How many samples are needed to detect an effect of a certain size?" Consider, for example, a design for comparing two categories of samples and a very small expression change of a particular protein between the two categories is supposed to cause overall strong changes within the studied biological system. A considerable larger number of samples is then necessary to detect this small effect than for detecting a very obvious and big effect. Besides the size of the effect that is to be detected, the variance of the expression levels influences the number of samples, too. The higher the variance, the harder it becomes to detect an effect. Both, the influence of the effect size and that of the variance onto the necessary sample size are exemplified in Fig. 1.

When calculating the appropriate sample size for an experiment, it is therefore necessary a) to specify the size of effect that is desired to be detected and b) to know something about the variance of expression levels. Knowledge about the variance can only be earned from earlier studies or, for example, from a small pilot experiment. With this information, one can calculate the so called power, which is the probability that a truly existing effect is detected by a statistical test (see Subheading 4). Let us regard the example in Fig. 2, where power curves are plotted under the assumption that the variance of the log-transformed expression

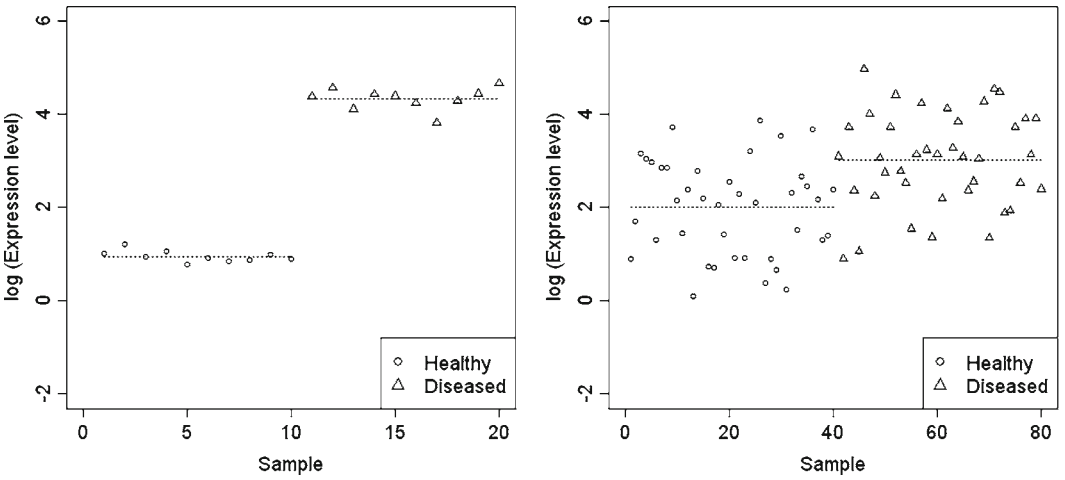


Fig. 1. A small sample size per group is sufficient to detect a big group effect when expression levels scatter very little (*left*), while larger sample sizes are necessary to detect a very small effect or when expression levels scatter very much (*right*).

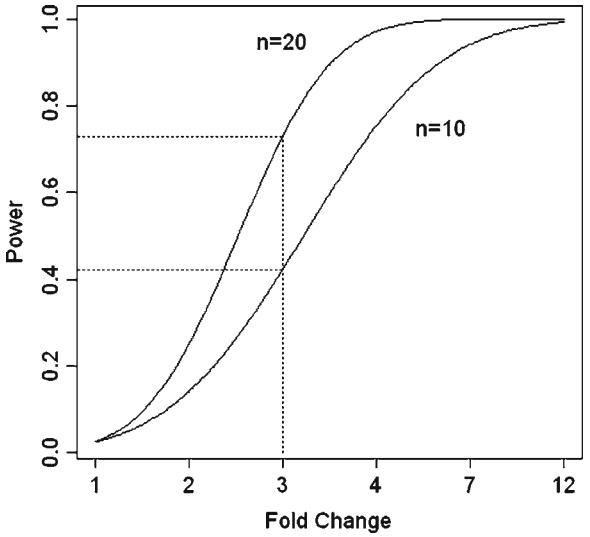


Fig. 2. Theoretical power curves for two different samples sizes n per group. The power is the probability that a particular true fold change is detected by a statistical test.

levels is 1.2 and for two different samples sizes, $n = 10$ and $n = 20$ per group. In this setting, a true 3-fold expression change can be detected with a probability of 0.42 when using 10 samples per group and with a probability of 0.71 when using 20 samples per group. For practical aspects of power calculation, see the notes section.

Another way of determining an appropriate sample size is to control a prespecified false discovery rate (9).

3. Data Preprocessing

Before starting with the concrete statistical analysis, electrophoretic and mass spectrometric recordings must be preprocessed. The raw results of 2-DE experiments are digital images which contain protein spots and the raw result of a mass spectrometric experiment is a mass spectrum with the m/z ratio on the abscissa and the intensity on the ordinate. For the former type of experiment, preprocessing starts with a specified automatic spot detection algorithm and by summarizing the pixel values within a spot boundary as a measure of abundance. For the latter one, a peak detection algorithm is carried out first and then the intensity values within the start and end point of a peak are summarized as a measure of abundance (10, 11). The thus obtained expression levels have to be further transformed by several steps as described in the following.

3.1. Variance Stabilization

In nearly all proteomic experiments, it can be observed that the variance of expression values that have been recorded for a protein depends on the average of these values. In , highly expressed proteins have a higher variance than low expressed proteins. It is therefore a common usage to apply a variance stabilizing transformation to the recorded expression levels. Most common transformation functions are the logarithm or the arsinh function. While the logarithm can, however, produce extreme and negative values for very low expressed proteins, the arsinh is positive and more flat in the lower region.

3.2. Normalization

Normalization is a further necessary transformation of expression levels to make the measurements of several gels or MS runs comparable. Particularly, in experiments where the samples of different categories are prepared by different labels (e.g., different fluorescent dyes or mass tags), normalization can also be used to remove labeling-biases and make the different channels comparable.

The two most frequent used normalization methods for proteomics data are quantile normalization (12) and the vsn normalization (13, 14). Quantile normalization shifts the expression levels of all gels or MS runs to have the same quantiles (see notes section). The vsn method uses affine linear mappings for transforming the expression levels. The latter method directly applies the arsinh function for variance stabilization as described above.

3.3. Standardization

Another method for making several gels comparable is the incorporation of an internal standard. This method can only be applied for techniques in which different labels are used and two or more samples are prepared on the same gel or run within the same MS run. One channel is then usually used for the internal

standard – mostly a mixture of all samples studied within an experiment. Expression values from the true samples are then divided by those of the internal standard. An internal standard is redundant if depended samples are directly compared to each other.

3.4. Missing Values Imputation

Particularly, in 2-DE experiments, resulting data matrixes contain a considerable number of missing values because the number of detected spots or of identified proteins is different from gel to gel (15, 16). Most of the classical statistical methods that were invented in the first half of the twentieth century are, however, designed for complete data matrixes, especially the methods for multivariate data. There are a number of ways missing data can be handled. Perhaps the simplest one is to omit all data rows or columns with missing values. That means, however, a loss of statistical power or a loss of information about certain proteins. Another possibility is to impute missing values and to obtain thus a complete data matrix. Several methods for missing values imputation are possible, e.g., the k nearest neighbor method (see notes section) or principal component regression. More sophisticated methods make imputations repeatedly several times and take the mean of all imputations (17).

4. Statistical Analysis

4.1. Statistical Hypothesis Testing

Let us throw again a glance upon the above-described study problems for comparing samples from two different categories of tissues. The goal of such experiments is the detection of differentially regulated proteins. An easy strategy to find those proteins would be to simply compare the average expression level in both categories of samples, separately for each protein. However, because expression levels are measured on a continuous metric scale, a nonzero difference between the average level in both categories can be expected for almost every protein, even for those which are not differentially regulated. How can the analyst now decide, which of the differences are big enough to call the protein differentially regulated, or how can he distinguish those proteins for which the difference is nonzero just by chance from those for which the difference deviates significantly from zero? This decision can be made by performing a statistical test. For performing a statistical test, first a *null hypothesis* is stated (e.g., "Protein x *is not* deregulated") as well as the complementary *alternative hypothesis* (e.g., "Protein x *is* deregulated"). Based on the measured values and eventually some assumptions about their underlying probability distribution either the null hypothesis is maintained or it is rejected in favor of the alternative hypothesis.

Table 1
Comparison of test decision based on experiment and unknown reality

		Unknown reality	
		Protein *is not* deregulated	**Protein *is* deregulated**
Test decision	Protein *is not* deregulated	True negative decision	False negative decision
	Protein *is* deregulated	False positive decision	True positive decision

Because a test decision is generally based on samples that are taken from a bigger population and because the measured quantity has usually a nonzero variance, the decision may fail to hit the unknown reality. In particular, a false positive or a false negative decision is possible (Table 1). Unfortunately, the probability α for a false negative decision and the probability β for a false negative decision are interdependent, and can thus not be decreased simultaneously. The solution is therefore to predefine a tolerable α (also called level of significance) and to control β by calculating the necessary sample size. A quantity that is usually derived by a test is the so called *p*-value. If this value is smaller than α, the null hypothesis is rejected.

4.2. Comparing Two Groups

In the most frequent problem in proteomics, that is comparing expression levels between two different categories of biological samples, one test is performed for each protein. If one can assume that expression levels are normally distributed, the so called t-test can be used. If expression levels are assumed to be non-normally distributed, e.g., if they show a very skewed distribution, one should rather use the nonparametric Mann-Whitney-U test (MWU). Both, t-test and MWU test offer versions for dependent and independent categories.

4.3. Multiple Hypothesis Testing

When performing thousands of statistical tests simultaneously (i.e., for thousands of proteins), there will usually be a high number of positive test decisions which are made just by chance, though the true situation is not positive. Naturally, these test decisions are false positive ones. How can the number of false positives be diminished? One solution to this problem is to be more conservative when testing. For that purpose, *p*-values can be adjusted in the sense of certain error rates (18), for example, the family-wise error rate (FWER) or the false discovery rate (FDR). The FWER is defined as the probability of having at least one false positive test decision among all test decisions. The FDR,

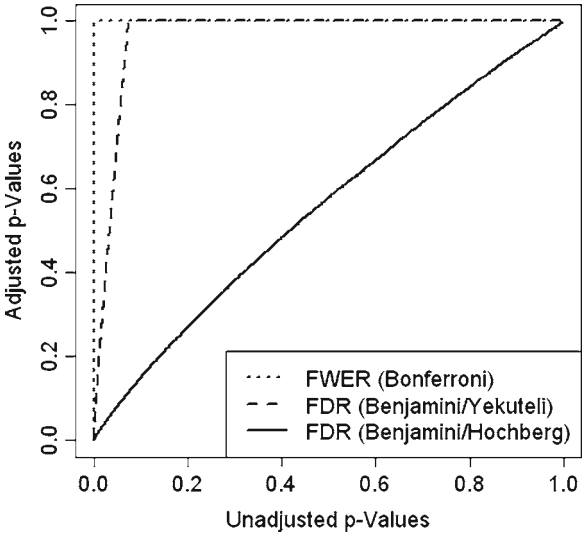


Fig. 3. Relation between unadjusted and adjusted *p*-values.

on the other hand, is the portion of false positives among all positives. An algorithm for adjusting *p*-values with regard to the FWER was given by Bonferroni. For controlling the less strict FDR, there are two different algorithms. One of them assumes that all hypothesis are independent (19) and a more liberal one puts no assumption onto the correlation structure of the hypothesis (20). Adjusted *p*-values from a cell line study of adenocarcinoma (21) are plotted versus the raw *p*-values in Fig. 3. While there seemed to be many significantly regulated proteins when using the unadjusted *p*-values, the FWER- and FDR-adjustments dramatically reduced the set of significant features and thus the number of false positive findings. Formulas for adjusting *p*-values are given in the notes section.

4.4. Analysis of Variance or Covariance

In Subheading 2, we have mentioned experimental designs in which a factor can have more than two levels or in which more than one factor is included. Data from such experiments can statistically be evaluated by analysis of variance (ANOVA) methods or by analysis of covariance (ANCOVA). In both methods, a dependent metric variable (here, these are the expression levels) is related to one or several independent experimental factors. If all independent variables are categorical (e.g., group membership, gender), ANOVA is used. If there is also one or more independent metric variables, ANCOVA is used instead. For each of the independent variables, one statistical test is performed, i.e., one *p*-value is produced. If that *p*-value is smaller than the significance level, the associated variable is supposed to have a significant influence onto the dependent variable. Besides the main effect

given by the independent variables, it is also possible to study interactions between these effects. Assume, for example, that there are two independent categorical factors included in the experiment, each on two levels: group (healthy, diseased) and gender (male, female). A significant interaction between group and gender indicates that the strength of a significant group effect is different within the two levels of gender. In extreme situations, an interaction between two independent variables can mean that the effect of factor A is inverse between the two levels of factor B.

Particular analysis of variance methods for repeated measures designs are, for example, detailed in (22) in the case of a normal distributed dependent variable. Because protein expression levels are often not assumed to be normally distributed, one should also consider nonparametric methods as detailed in (23). The most difficult problem in the analysis of repeated measures designs is to set the correct assumption of the correlation matrix. Different forms for this matrix can be considered. Let us take again the example that protein expression is repeatedly measured at several subsequent points in time. A simple assumption for the correlation structure between the studied points in time is that there is an equal correlation between each of two points in time (this structure is called compound symmetry). More realistic, is however, that points in time that are more distant from each other have a smaller correlation than those that are less distant (autoregressive structure). In some situations, an unstructured correlation matrix is assumed.

4.5. Fold Change and Confidence Intervals

Using statistical tests, it is possible to conclude that a protein is significantly up- or downregulated. One goal of proteomics is furthermore to quantify the strength of regulation, which is usually done by a ratio estimate, for example, the fold change. The fold change is defined as the ratio of the average expression between two categories of an experimental factor. Because expression levels are usually log-transformed, the ratio becomes than a difference. In the context of the fold change, it is important to report this quantity always in combination with a confidence interval (21). A confidence interval covers the true expression change with a probability of $(1-\alpha)$. Using a confidence interval, one can compare the importance of proteins with the same fold change. Assume that protein X and protein Y both have a fold change of 2. For protein X, however, the confidence interval is given by [0.6, 2.8] while the confidence interval for protein Y is given by [1.7, 2.2], i.e., the latter confidence interval is much smaller but with the same level of confidence than the former one. For protein X, it is then not really possible to conclude that it is truly up regulated because the lower bound of the interval is smaller than 1. For protein Y instead, it seems very likely that is up regulated with a high confidence.

5. Notes

5.1. Randomization

In order to avoid the overlapping of effects from the studied experimental factors with other uninteresting effects, a random assignment of experimental units to the study groups is important. Assume, for example, that the effect of one treatment is to be studied on the expression levels in a cell line. Five samples are to be assigned to each group, the treatment and the control group. For random assignment, follow the next steps:

1. Generate a list of ten random numbers (e.g., from a standard normal distribution) and assign ranks 1–10 to these numbers.
2. All samples with rank 1–5 are treated and all with ranks 6–10 are not treated (Table 2).

5.2. Sample Size Calculations

When testing for differential expression between two groups, a *t*-test is usually carried out for each protein. The power of the t-test is the probability that the test detects a certain log(fold change) under a fix sample size and with a given variance of the expression levels. For determining an appropriate sample size, proceed as follows:

1. For each protein in the data from a pilot sample, calculate the variance of its preprocessed expression levels. Power can, for example, be calculated for the minimum, median, or maximum of all variances.
2. Calculate the power for different sample sizes and for different log(fold changes) using the variance estimates from the pilot sample. It is recommended to use a statistical software tool for calculating the power (e.g., the free software R from www.r-project.org).
3. Choose that sample size for your experiment which yield the desired power.

Table 2
Random assignment of treatment or nontreatment to ten samples of a cell line for avoiding undesired overlapping effects

Sample	1	2	3	4	5	6	7	8	9	10
Random Number	1.92	0.15	−0.64	−1.00	−0.83	1.02	0.16	0.54	−0.19	0.34
Rank	10	5	3	1	2	9	6	8	4	7
Treatment	No	Yes	Yes	Yes	Yes	No	No	No	Yes	No

5.3. Quantile Normalization

Assume that your data matrix A consists of m columns (representing gels) and n rows (representing proteins). For making gels comparable, the following steps of quantile normalization can be applied (directly cited from (12)):

1. Sort each column of A, yielding a new matrix A_{sort}.
2. Calculate the means across rows and assign this mean vector to each column of A_{sort}, yielding the matrix M.
3. Rearrange M to have the same ordering as A, yielding the normalized matrix A_{norm}.

This algorithm is also implemented in the "limma"package for the software R (available from www.bioconductor.org).

5.4. Missing Values Imputation

Especially, 2-DE produces data matrixes with many empty entries. To impute these missing values, one can use the k-nearest neighbor method:

1. Calculate the correlation or distance between the expression levels of each pair of proteins by using the available values.
2. Assume that the expression level of protein i in gel j is missing. Determine the k nearest proteins (neighbors) to protein i (according the distance or the correlation). Calculate the mean of the expression levels of these k neighbors in gel j and use it to fill the gap. A k between 10 and 20 has been recommended by Jung et al. (16).

5.5. Adjusting of p-Values

When searching for differentially regulated proteins, p-values should be adjusted to avoid a too high number of false positives. Assume that the raw p-values for n proteins are $p_1, \ldots, p_n$. A very strict adjustment is given by the Bonferroni method to control the FWER:

$$p_i(\text{adjusted}) = \min\{1, n \cdots p_i\} (i = 1, \ldots, n).$$

A less strict method is the FDR-procedure of Benjamini and Hochberg (19).

1. Take the n ordered p-values: $p_{(1)} \le p_{(2)} \le \ldots \le p_{(n)}$.
2. The FDR-adjusted p-values are given by $p_{(i)}(\text{adjusted}) =$

$$\min_{k=i,\ldots,n} \left\{ \min\left(\frac{n}{k} p_{(i)}, 1 \right) \right\}$$

$$(i = 1, \ldots, n).$$

Other adjustment procedures are implemented in the *p.adjust* method of the R-package "stats."

References

1. Patterson SD (2003) Data analysis – the Achilles heel of proteomics. Nat Biotechnol 21:221–222
2. Karp NA, McCormick PS, Russell MR, Lilley KS (2007) Experimental and statistical considerations to avoid false conclusions in proteomic studies using differential in-gel electrophoresis. Mol Cell Proteomics 6:1354–1364
3. Fodor IK, Nelson DO, Alegria-Hartman M, Robbins K, Langlois RG, Turteltaub KW et al (2005) Statistical challenges in analysis of two-dimensional difference gel electrophoresis experiments using DeCyder. Bioinformatics 21:3733–3740
4. Ünlü M, Morgan ME, Minden JS (1997) Difference gel electrophoresis: a single gel method for detecting changes in protein extracts. Electrophoresis 18:2071–2077
5. Gygi SP, Rist B, Gerber SA, Turecek F, Gelb MH, Aebersold R (1999) Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. Nat Biotechnol 17:994–999
6. Ross PL, Huang YN, Marchese JN et al (2004) Multiplexed protein quantitation in Saccharomyces cerevisiae using amine-reactive isobaric tagging reagents. Mol Cell Proteomics 3:1154–1169
7. Stühler K, Pfeiffer K, Joppich C, Stephan C, Jung K, Müller M et al (2006) Pilot study of the Human Proteome Organisation Brain Proteome Project: Applying different 2-DE techniques to monitor proteomic changes during murine brain development. Proteomics 6:4899–4913
8. Sitek B, Apostolov O, Stühler K, Pfeiffer K, Meyer HE, Eggert A, Schramm A (2005) Identification of dynamic proteome changes upon ligand activation of trk-receptors using two-dimensional fluorescence difference gel electrophoresis and mass spectrometry. Mol Cell Proteomics 4:291–9
9. Cairns DA, Barrett JH, Billingham LJ, Stanley AJ, Xinarianos G, Field JK et al (2009) Sample size determination in clinical proteomic profiling experiments using mass spectrometry for class comparison. Proteomics 9:74–86
10. Boehm AM, Pütz S, Altenhöfer D, Sickmann A, Falk M (2007) Precise protein quantification based on peptide quantification using iTRAQ™. BMC Bioinform 8:214
11. Jeffries N (2005) Algorithms for alignment of mass spectrometry proteomic data. Bioinformatics 21:3066–3073
12. Bolstad BM, Irizarry RA, Astrand M, Speed TP (2003) A comparison of normalization methods for high density Oligonucleotide array data based on bias and variance. Bioinformatics 19:185–193
13. Huber W, Heydebreck A, von Sültmann H, Poustka A, Vingron M (2002) Variance stabilization applied to microarray data calibration and the quantification of differential expression. Bioinformatics 18:S96–S104
14. Kreil DP, Karp NA, Lilley KS (2004) DNA microarray normalization methods can remove bias from differential protein expression analysis of 2D difference gel electrophoresis results. Bioinformatics 20:2026–2040
15. Jung K, Gannoun A, Sitek B, Meyer HE, Stühler K, Urfer W (2005) Analysis of dynamic protein expression data. RevStat-Stat J 3:99–111
16. Jung K, Gannoun A, Sitek B, Apostolov O, Schramm A, Meyer HE et al (2006) Statistical evaluation of methods for the analysis of dynamic protein expression data from a tumor study. RevStat-Stat J 4:67–80
17. Schafer JL, Graham JW (2002) Missing data: our view of the state of the art. Psychol Meth 7:147–177
18. Dudoit S, Shaffer JP, Boldrick JC (2003) Multiple hypothesis testing in microarray experiments. Stat Sci 18:71–103
19. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J Roy Stat Soc Ser B 57:289–300
20. Benjamini Y, Yekutelli D (2001) The control of the false discovery rate in multiple testing under dependency. Ann Stat 29:1165–1188
21. Jung K, Poschmann G, Podwojski K, Eisenacher M, Kohl M, Pfeiffer K et al (2009) Adjusted confidence intervals for the expression change of proteins observed in 2-dimensional difference gel electrophoresis. J Proteomics Bioinform 2:78–87
22. Diggle PJ, Liang K-Y, Zeger SL (1994) Analysis of longitudinal data. Clarendon Press, Oxford
23. Brunner E, Domhof S, Langer F (2002) Nonparametric analysis of longitudinal data in factorial experiments. Wiley, New York

Chapter 17

The Evolution of Protein Interaction Networks

Andreas Schüler and Erich Bornberg-Bauer

Abstract

The availability of high-throughput methods to detect protein interactions made construction of comprehensive protein interaction networks for several important model organisms possible. Many studies have since focused on uncovering the structural principles of these networks and relating these structures to biological processes. On a global scale, there are striking similarities in the structure of different protein interaction networks, even when distantly related species, such as the yeast *Saccharomyces cerevisiae* and the fruit fly *Drosophila melanogaster*, are compared. However, there is also considerable variance in network structures caused by the gain and loss of genes and mutations which alter the interaction behavior of the encoded proteins. Here, we focus on the current state of knowledge on the structure of protein interaction networks and the evolutionary processes that shaped these structures.

1. Introduction

In his Nobel lecture in 1968, Jaques Monod pointed out the importance of interactions in biological systems: "… any phenomenon, any event, or for that matter, any 'knowledge,' any transfer of information implies an interaction" (1). Today, more than 40 years later, it has indeed become obvious that biological phenomena can only rarely be attributed to a single-molecule species, like the transport of oxygen by four units of hemoglobin. Instead, cellular processes are most often carried out by "modules" formed by the interactions between genes, proteins, and small molecules (2). With the availability of completely sequenced genomes and high-throughput methods to detect protein interactions, we now have the opportunity to study the cellular complexity in a more holistic way and uncover the organizing principles underlying the network formed by proteins, and their interactions with each other. This approach to understand cellular complexity,

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696,
DOI 10.1007/978-1-60761-987-1_17,

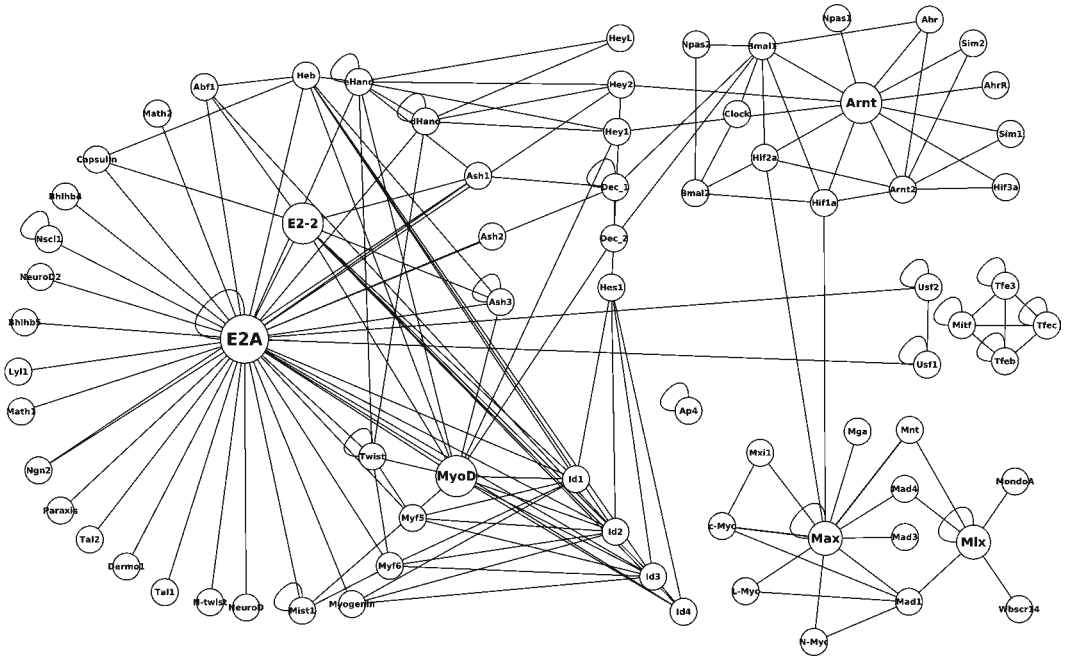


Fig. 1. The protein interaction network formed by human basic helix-loop-helix transcription factors. This well-studied subnetwork of the human protein interaction network (64) shows several structural features that are also apparent on a global level. The network is dominated by a few highly connected proteins (especially the transcription factor E2-alpha) and shows a high degree of modularity. UniProt Ids are used as node labels, and edges that connect a protein with itself refer to homodimerizations. Cytoscape (78) has been used for visualization.

which is sometimes referred to as "Network Biology" (3), has already yielded exciting insights. It is now known, for example, that protein interaction networks are characterized by the presence of highly interconnected subnetworks and that a few number of proteins with a very high number of interactions dominate the large-scale structure of these networks (4), (Fig. 1). Biomedical research has also been fueled by these advances as protein interaction data have been proven to be useful for the prediction of potential drug targets (5). However, our knowledge about the evolutionary processes that shaped the structure of protein interaction networks is still limited, especially because comprehensive interaction data are only available for a few distantly related model organisms (6), (Table 1).

In this chapter, we discuss the basic principles of protein interactions, the structure of protein interaction networks, and the currently known evolutionary processes that influence this structure.

Table 1
Databases providing information on protein–protein or domain–domain interactions

Web server and link	Organism	Number of interactions (as of November 2009)
Databases providing protein–protein interactions based on experimental data		
BIND (82) http://bind.ca	No species restriction	67,739
BioGRID (83) http://www.thebiogrid.org	No species restriction	169,723
DIP (84) http://dip.doe-mbi.ucla.edu	No species restriction	57,683
HPRD (85) http://www.hprd.org	Human	38,806
IntAct (86) http://www.ebi.ac.uk/intact/main.xhtml	No species restriction	202,419
Mint (87) http://mint.bio.uniroma2.it/mint	No species restriction	83,321
Mips (88) http://mips.helmholtz-muenchen.de/proj/ppi	Mammals	N/A
Databases providing domain–domain interactions		
Ipfam (17) http://ipfam.sanger.ac.uk	No species restriction	2,733
3did (18) http://gatealoy.pcb.ub.es/3did	No species restriction	5,313
Domine (89) http://domine.utdallas.edu/cgi-bin/Domine	No species restriction	20,513
Databases providing protein–protein interactions based on computational predictions		
I2d (90) http://ophid.utoronto.ca/ophidv2.201	No species restriction	424,066
PRISM (91) http://prism.ccbb.ku.edu.tr/prism	No species restrictions	N/A
PIPS (92) http://www.compbio.dundee.ac.uk/www-pips	Human	37,606
HPID (93) http://wilab.inha.ac.kr/hpid	Human	N/A

2. Basic Tenets of Protein Interactions and the Structure of Protein Interaction Networks

2.1. Principles of Protein–Protein Interactions

Soon after the first protein complexes had been successfully crystallized and structurally resolved using X-ray crystallography, numerous studies have been conducted to uncover the biochemical and biophysical principles underlying protein–protein interactions. In a pioneering study, Chothia and Janin analyzed the binding interfaces of three protein complexes and concluded that the formation of unspecific hydrophobic interactions is the most important factor in stabilizing protein association (7). Hydrogen bonds and van der Waals contacts contribute much less to the stability of a protein complex, but they are important for the

specificity of protein interactions because they require complementarity of the involved binding interfaces. These basic tenets of protein interactions have been repeatedly confirmed in analyses of more comprehensive data sets (8). However, there are also many properties in which protein interactions can differ and which can be used to classify them.

2.2. Permanent and Transient Protein Interactions

One important concept for the classification of protein interactions is the distinction between *permanent* and *transient* protein associations. Some proteins form very stable interactions and can be found only in their complex form *in vivo*. The interactions that stabilize multisubunit enzyme complexes, such as the ATP synthase, are an example for such permanent protein interactions.

But many protein interactions are transient, and the involved proteins are in an equilibrium where the interactions are broken and re-formed continuously (9). These transient interactions are especially important in cellular signaling cascades (10). Obviously, there is no clear cut boundary between transient and permanent interactions, but trends can be observed. Binding interfaces that mediate transient interactions seem to be smaller on average, with a surface area of 1,500 Å^2 or less, while interfaces that mediate strong interactions are usually much larger, reaching a surface area of up to 10,000 Å^2 (11). Moreover, transient interactions contain more polar residues than permanent ones.

Permanent and transient protein interactions seem to impose different constraints on protein evolution. For example, proteins from *Saccharomyces cerevisiae*, which participate in permanent protein interactions, are on average more similar to their orthologs in *Saccharomyces pombe* than proteins that participate in transient protein interactions, with the average sequence identity being 46 and 41%, respectively (12). The distinction between transient and permanent interactions is important for understanding the evolution of protein-binding behavior but is sadly often overlooked (13).

2.3. Protein Domains as Binding Interfaces

Another concept that is useful for studying the evolution of protein interactions is based on protein *domains*. Protein domains have originally been defined as parts of proteins that are able to fold independently (14). A more recent perspective views domains as units of protein function and of protein evolution (15). These definitions overlap, protein domains are able to carry out a specific function because of their specific fold, and because of their specific function, natural selection acts to preserve the underlying sequence. Different domains are frequently joined in proteins by mechanisms such as gene fusion (16), leading to multi-domain proteins. The domain composition of a protein is often referred to as *domain arrangement* (15). Protein–protein interactions can often be explained by the domain arrangements of the respective

proteins because the binding interfaces involved in protein interactions often correspond to protein domains. Databases have been established that provide information about known domain–domain interactions inferred from structural data (17, 18), (Table 1) and the binding behavior of proteins can thus be predicted by annotating protein domains.

Approximately 19% of the known human protein–protein interactions can be explained by known domain–domain interactions (19). The remaining 81% likely correspond to domain–domain interactions that are currently unknown and to interactions that are based on the association of a protein domain and a short peptide motif, such as the interaction of the SH3 domain with the PxxP motif.

It has been observed that the number of protein domains is much smaller than the number of proteins, and most domains occur in many proteins in many different species (20). The analysis of the human genome, for example, showed that only ~7% of known protein domains appear to be specific for vertebrates. Also, a trend has been observed toward a higher proportion of multidomain proteins in eukaryotic proteins compared to that in prokaryotic ones (21). Some protein domains seem to be especially prone to participate in the formation of new domain arrangements. This is referred to as domain promiscuity or versatility (22). Furthermore, it could be shown that domains that mediate protein interactions are among the most promiscuous classes of protein domains (23). Taken together, the available data suggest that the formation of novel domain arrangements is a key factor in the evolution of protein interaction networks.

3. The Structure of Protein Interaction Networks

3.1. Protein Interaction Data

Research on the global properties of protein interaction networks has for a long time been hindered by a lack of data. Several approaches have been started to remove this handicap. One of these approaches is based on extracting knowledge about protein interactions from the scientific literature. Many publications discuss protein interactions detected in small-scale experiments. This information can be gathered by text mining and several databases have been established that provide these data for the scientific community (24).

The development of high-throughput methods to detect protein interactions, such as the yeast two-hybrid method (25), was also helpful in complementing the small-scale data gathered from the scientific literature. Using these techniques, extensive protein interaction data have been experimentally determined for

several model organisms, including *S. cerevisiae* (26, 27), *Escherichia coli* (28), *Helicobacter pylori* (29), *Drosophila melanogaster* (30), *Caenorhabditis elegans* (31), *Plasmodium falciparum* (32), *Campylobacter jejuni* (33), and *Homo sapiens* (34).

Despite these efforts, the available protein interaction data are still very incomplete and assumed to include many false positives (35). Several methods have been established to assess the accuracy of protein interaction assays and thus distinguish experimental artifacts from biologically meaningful interactions. Popular methods for such an evaluation include the comparison of expression profiles of proteins that are assumed to interact and the comparison of interaction profiles between orthologous proteins (36). When these methods are applied to the results of high-throughput interaction screens, they yield false-positive rates that can be as high as 80% (37).

The yeast *S. cerevisiae* is the best studied model organism in terms of protein interactions; all possible protein pairs have been screened for possible interactions, many of them multiple times (38). The yeast network is, therefore, currently the most suitable one for studying the global structure of protein interaction networks.

3.2. Structural Properties of Protein Interaction Networks

To uncover global properties of protein interaction networks, known protein interactions of one species are usually modeled as an undirected graph, with nodes representing proteins and edges representing protein interactions. This representation is obviously very simplified as it ignores all biochemical properties of the interaction (e.g., transient and permanent interactions are treated identically in such a model), and also does not consider the dynamics of protein interaction networks caused by differential gene expression and cellular localization of proteins.

In a seminal study by Jeong et al. (4), it was shown that the distribution of interaction partners per protein in the yeast protein interaction network can be well approximated by a power law (4). This means that the network is highly inhomogeneous with a small number of highly connected proteins in contrast to a majority of proteins with few interactions. Intuitively, proteins with many interaction partners (often referred to as "hubs") should be more essential for an organism than other proteins, and it could indeed be shown that the loss of a highly connected protein (a protein with more than 15 interaction partners) is about three times as likely to be lethal than the loss of a protein with fewer interactions (4). Subsequent studies based on more comprehensive protein interaction data confirmed the presence of proteins with many more interactions than the average protein in the respective interaction network (39). However, the correlation between connectivity and essentiality could not be verified in recent analyses of high-quality protein interaction data sets

(see Notes 2 and 3). Instead, a strong correlation between connectivity and pleiotropy, the number of phenotypes observed as a consequence of gene knockout, could be shown (38).

Protein interaction networks also seem to be characterized by a high degree of modularity, which corresponds to the presence of highly interconnected subnetworks. The degree of modularity for each protein in an interaction network can be quantified with the *clustering coefficient C*, defined as $C_i = 2n_i/k_i(k_i-1)$ for each protein i,with k_i being the connectivity of this protein and n_i the number of interactions between all interaction partners of protein i. The average clustering coefficient $<C>$ of protein interaction networks is several orders of magnitude larger compared to a random network of the same size (40). Curiously, most protein hubs either have very large or very small values for the clustering coefficient. By integrating protein interaction data with gene expression and cellular localization data, Han et al. (41) showed that these two kinds of protein hubs correspond to proteins that interact with most of their partners simultaneously and hubs that bind their different partners at different times or locations (41). The hubs that bind most of their partners simultaneously, referred to as "party" hubs, are often part of large protein complexes, explaining their high clustering coefficient. Hubs that bind to their partners at different times or locations often correspond to highly connected proteins in signaling pathways; these hubs are referred to as "date" hubs (41, 42).

Another structural property of protein interaction networks is the high abundance of interactions between structurally similar proteins. It could be shown that self-interactions and interactions between paralogous proteins occur significantly more often in protein interaction networks than would be expected by chance (43, 44).

4. Evolution of Protein Interaction Networks

4.1. Gene Duplications and the Evolutionary Conservation of Protein Interactions

Protein interaction networks are constantly rewired in the course of protein evolution in order to integrate new proteins into the network, to compensate for the loss of a protein, or to evolve novel functionalities (Fig. 2). New nodes can be added to a protein interaction network by means of gene duplication and the product of a duplicated gene inherits the binding behavior of its ancestor. This principle has been proposed as an explanation for the presence of hubs in protein interaction networks. If all genes would be equally prone to gene duplication and if their products would retain their binding behavior after duplication, proteins with a higher than average number of interactions would more likely gain a new interaction partner after random duplication events.

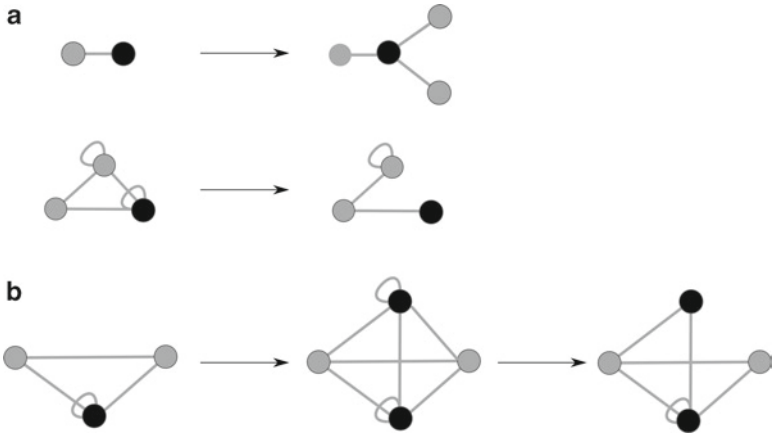


Fig. 2. Elementary processes of protein interaction network evolution. (**a**) Gain and loss of protein interactions, which are usually caused by point mutations in binding interfaces. (**b**) Gene duplication, which first yields a protein with identical binding behavior. After duplication, the proteins are likely to diverge in their binding behavior. Empirical data suggest that gene duplications occur at much lower rates (up to two orders of magnitude) than interaction gain and loss events (53).

These growth dynamics are often referred to as "the rich get richer" or "preferential attachment." Mathematical models based on this principle can generate networks with the same inhomogeneous distribution of interactions per protein. These networks contain a small number of proteins with a very high number of interactions, even if some degree of divergence in binding behavior after duplication is allowed for in the respective model (45, 46). Under certain conditions, such duplication–divergence models can also account for the high degree of modularity observed in protein interaction networks (47).

However, whether duplication–divergence models are really an adequate explanation for the structure of protein interaction networks is controversial (48). If the duplication–divergence model is correct, paralogous proteins should share interaction partners more often as would be expected by chance. As we have pointed out above, this is indeed the case (49). However, despite the overrepresentation of interactions between paralogous proteins, the interaction turnover, i.e., rate at which proteins gain and lose interactions after duplication, is still considerably high. Based on a comparison of protein interaction data from four eukaryotic species, Beltrao and Serrano estimated a rate of 10^{-6}–10^{-5} interaction gains or losses per protein pair per million years (50). As the number of protein pairs for eukaryotic species is in the order of 10^7 (*S. cerevisiae*) to 10^8, one could expect between 100 and 1,000 gains and losses of interactions per million years

per proteome. In comparison, the number of genes that undergo duplication has been estimated to be 10^{-2} per gene per million years for *S. cerevisiae* (51). Since ~90% of single gene duplicates become silenced after duplication (52), 10^{-3} proteins per million years can be assumed to be the growth rate for protein interaction networks by gene duplications. Because of these differences in estimated rates of gene duplication and interaction turnover, the gain and loss of interactions might be the dominating factors in shaping structural properties of protein interaction networks (53).

Duplication–divergence is currently the most popular concept for modeling the evolution of protein interaction networks, but it is not the only concept that has been proposed to explain the structural features of protein interaction networks. Some studies propose that protein hubs have certain features that enable them to evolve higher than average number of interactions, for example, the presence of "sticky" domains (i.e., domains that mediate a very diverse set of protein interactions) like the Zinc-finger domain (54). Another property that could increase the interaction propensity of a protein is the presence of intrinsically disordered regions. The inherent flexibility of disordered regions offers malleable interfaces that can allow binding to several partners or to adopt different conformations, manifested in increased binding capability (55). Also, disordered regions are known to often contain peptide motifs that act as binding regions, such as the canonical SH3-ligand PxxP (56).

The evolutionary rates of protein hubs, measured as the dN/dS ratio (the ratio of nonsynonymous to synonymous substitutions, also referred to as Ka/Ks ratio), have been studied intensively, with conflicting results. Some studies presented evidence that hubs evolve at slower rates than proteins with few interactions (57, 58), which means that once a protein has gained a high number of interaction partners, it is likely to be affected by purifying selection and maintaining its sequence. These results could not be reproduced in other studies and it has been proposed that the seemingly slow rates of evolution in protein hubs might be caused by biased data sets (59, 60). If protein hubs would indeed evolve at slower rates, evolutionary models for the growth of protein interaction networks would have to account for this by decreasing the probability for the gain and loss of interactions for proteins that already have many interaction partners. The controversy revolving around the evolutionary rates of protein hubs was finally resolved by mapping structural data to protein interaction networks. Based on structural information, it is possible to divide proteins into several classes based on whether they have only one or multiple binding interfaces. It was shown that hubs that use multiple interfaces to bind to their interaction partners evolve at slower rates than average, which is not the case for single-interface hubs (61).

It is, of course, also possible to invoke natural selection as an explanation for the structure of protein interaction networks. This would mean that the structure observed in protein interaction networks is not simply the consequence of the duplication and divergence processes driving network evolution and the different binding propensities of proteins, but rather the consequence of natural selection acting on the network level (62, 63). However, no evidence that supports this hypothesis could be presented so far. On the contrary, it was shown that the known duplication–divergence processes are able to explain at least parts of the observed structure of protein interaction networks. For example, Amoutzias et al. showed that the interaction network formed by the family of basic helix-loop-helix (bHLH) proteins, an ancient family of transcription factors, can be elegantly explained by single-gene duplication and domain rearrangement events (64). Interestingly, the bHLH family can be divided into six phylogenetic groups (Fig. 1) with different domain arrangements. Three of the subnetworks formed by the distinct phylogenetic groups show a similar hub-based structure, with one highly connected and many peripheral proteins. These similar structures are probably the result of convergence since phylogenies do not support large-scale duplication. A hub-based structure can thus, in principle, arise as the consequence of single-gene duplication and divergence processes (including domain rearrangements) without natural selection acting on the network level. However, the picture might be different for other gene families.

4.2. Origin of Novel Binding Interfaces in Proteins

So far, we have only discussed gradual changes in the binding specificities of existing binding interfaces. This begs the question how these interfaces originated in proteins in the first place.

In order to evolve a new interface, mutations must change the surface area of a protein in such a way that the association to another protein surface leads to the release of sufficient free energy to compensate the entropy loss upon association. Single substitutions are unlikely to cause a sufficient increase in the free energy upon association to lead to a functional association, and without any functional difference which could be selected for, the respective substitutions are unlikely to be fixed in a population.

One possible explanation for the evolution of novel binding interfaces is based on the mass action law, which states that the binding affinity between two reactants increases with their local concentrations. If the local concentration of proteins is sufficiently high, a single substitution can thus indeed lead to significant changes in the binding behavior (65). One prominent example for this is sickle-cell anemia, a disease caused by a mutation that replaces glutamic acid at position six in the β-chain of hemoglobin with valin. In the cytosol of erythrocytes, the concentration of hemoglobin is very high (~5 mM (66)) and in individuals with

sickle-cell anemia, this leads to the aggregation of hemoglobin and thus to the formation of fibrils. Based on this principle, Kuriyan and Eisenberg argued that colocalization might be important in the evolution of binding interfaces (67).

Colocalization can occur, for example, when two proteins bind to adjacent DNA regions, when two proteins bind to the plasma membrane, or when two proteins are fused together by gene fusion. Under such conditions, single substitutions could yield a selectable difference in binding affinity, which makes a gradual pathway to a specific interface possible (67). Comparative genomics studies have revealed a high abundance of gene fusion and fission events (68), and there are thus many possible scenarios in which two noninteracting proteins are first joined by gene fusion, optimized for a high binding affinity to each other, and finally split up to yield two proteins that specifically bind to each other.

Another explanation for the evolution of specific binding interfaces has been proposed by Bennet and coworkers (69). In their studies on diphteria toxin dimers, they observed that these proteins dimerize by exchanging domains, a mechanism which has been termed *domain swapping*. Domain swapping works by first breaking the intramolecular interactions of a protein domain with the rest of the protein in two monomers, the domains are then relocated and replace each other in the original monomers, leading to dimerization. Wether the resulting dimer is more stable than the single monomers depends on several factors. Translational and rotational energy is lost upon dimerization, leading to a decrease in stability, but this loss could be compensated by interactions between the nonswapped parts of the protein. If the dimer has a new useful function or is more efficient in carrying out the function of the monomers, then mutations that stabilize the dimer would be selected for. Not only dimers but also higher oligomers can evolve in such a way. If a homodimerizing protein is formed in such a way, gene duplication and diversification can lead to families of homo- and heterodimerizing proteins. There are many instances where domain swapping has been observed (70) and this mechanism might be an explanation for the fact that interactions between identical and structurally highly similar proteins are overrepresented in protein interaction networks.

4.3. Conserved Structures in Protein Interaction Networks

The availability of extensive protein interaction data for several model organisms makes it possible to carry out comparative studies on protein interaction networks. However, we already mentioned that comprehensive protein interaction data are only available for a small number of model organisms. Among eukaryotes, these include the yeast *S. cerevisiae*, the fly *D. melanogaster*, the roundworm *C. elegans*, and *H. sapiens*. It has been estimated that these organisms shared a common ancestor ~1.4 billion years ago (71).

Because of this large evolutionary distance and the previously mentioned high rate at which a protein can gain or lose interactions (50), substantial differences between the protein interaction networks of these four organisms can be expected. However, it can also be expected that protein interactions which participate in biological processes, common to all eukaryotes, are evolutionarily conserved, even between distantly related species.

In one of the first attempts to quantify the amount of evolutionarily conserved protein interactions between distantly related species, Cesareni et al. showed that 2% of the known protein interactions in yeast can be mapped to orthologous proteins in the fly, and 8% of the known *D. melanogaster* protein interactions can be mapped to orthologs in *S. cerevisiae* (72). It is very likely that this seemingly low amount of evolutionary conserved protein interactions is affected by the high amount of false-positive interactions in protein interaction data sets (35). This is supported by the observation that the fraction of interactions that can be mapped between both species increases to 5 and 24%, respectively, when only high-confidence interactions are considered (72). Also, it has to be considered that the available protein interaction data are incomplete and that some interactions cannot be mapped simply because they have not been detected yet. The reported fractions of evolutionarily conserved protein interactions between distantly related species can thus be regarded as lower bounds.

Mapping conserved protein interactions between species is not the only approach that has been made in comparing protein interaction networks. Other approaches take the structure of the networks into account and search for similarities on the network level by employing graph comparison algorithms. Kelley et al. introduced such an algorithm called *PathBlast*, which searches for evolutionarily conserved pathways between species (73). The algorithm tries to map a pathway from one species to the orthologous proteins from another species and defines a score for putative matches. This score is decreased by pathway proteins for which no ortholog could be identified and by missing protein interactions between the orthologous proteins. Such a scoring scheme accounts for evolutionary variations and also for the incomplete nature of protein interaction data sets. Evolutionarily conserved pathways between *S. cerevisiae* and the bacterial pathogen *H. pylori* could be identified with this approach (73).

In a later study, this procedure has been generalized to compare more than two species at once and to search not only for conserved pathways but also for conserved clusters of highly interconnected proteins (74). Comparing the *S. cerevisiae*, *C. elegans*, and *D. melanogaster* protein interaction networks with this method yielded 71 network regions that are conserved between all three species, most of which could be identified as

functional modules responsible for central biological processes such as protein folding, intracellular transport, and RNA/DNA metabolism (74).

Another recent procedure to find subnetworks conserved between several species was developed by Gerke et al. (75). Their approach called Protein Interaction Network Analysis (PINA) is based on identifying topologically interesting subnetworks first, like highly clustered regions, and then performing a pair-wise sequence comparison of the respective proteins. Using this approach, they were able to identify conserved subnetworks, like, for example, a cluster of Y-family DNA polymerases that is conserved between *Mus musculus* and *H. sapiens* (75).

Searching for similarities on the network level seems to be a much more fruitful approach for comparing protein interaction networks than simply counting the number of evolutionarily conserved protein interactions between two species, especially because such methods do not rely on perfect matches, which are unlikely to be found due to the incomplete nature of protein interaction data sets. Also, similar network structures can often be mapped to specific biological processes, which makes such results useful for the prediction of protein functions.

5. Conclusions

Much progress has been made in understanding how protein interaction networks evolve. However, research in this area is still limited by the available data. Especially the fact that comprehensive protein interaction data are available only for distantly related model organisms makes it hard to assess the degree of evolutionary conservation in interaction networks. And, as has been mentioned above, it is not only the amount but also the quality of the available interaction data which is a limiting factor. Especially transient protein interactions are still very difficult to detect. For transient interactions that have been detected, it is hard to assess whether those interactions are not only biophysically possible but also biologically relevant (i.e., they do occur *in vivo*). Fortunately, yeast two-hybrid and mass spectrometry-based methods to detect protein interactions are constantly refined and improved. Mass spectrometry-based methods, for example, have already matured to a level where it is possible to detect the precise stoichiometry and even dissociation constants of protein complexes (76).

The human protein interaction network is currently a focus of research, and community-wide efforts such as the proposed "Human Interactome Project" (77) are currently in planning.

A "complete" protein interaction map for humans will certainly be of great value, especially for biomedical research. Further

advances in understanding the evolution of protein interaction networks, however, will depend on the availability of interaction data for closely related species, and it is unlikely that such data will be available soon.

6. Notes

1. Several open-source programs are available for common tasks in research on protein interaction networks, including network visualization and topological analyses. Cytoscape (78) is a very popular program in this context; it provides a basic framework for the analysis and visualization of networks that can be easily extended by plug-ins.
2. It is advisable to not rely on a single data set for protein interactions. Several databases are available as a source for experimentally verified and computationally predicted protein interactions (see Table 1). Integrating this data into a single data set is not straightforward (especially because different databases use different protein/gene IDs). However, several tools are available that automatize the task of extracting all available protein interactions for a given species. Popular tools for this task include APID2NET (79) (a plug-in for Cytoscape) and PINA (80).
3. It is possible to filter out low-confidence interactions in a protein interaction data set to minimize the number of false-positive interactions. Several approaches have been proposed for this task (81). These methods include (a) removal of all interactions that have not been detected at least twice in independent experiments, (b) filtering out interactions that are not co-expressed, (c) filtering out interactions that do not share similar Gene Ontology annotations. Some of these methods are available as plug-ins for the Cytoscape program.

References

1. Monod J (1968) On symmmetry and function in biological systems. Nobel symposium 11, Symmetry and Function of Biological Systems at the Macromolecular Level. 1527
2. Hartwell LH, Hopfield JJ, Leibler S, Murray AW (1999) From molecular to modular cell biology. Nature 402:C47–C52
3. Barabási AL, Oltvai ZN (2004) Network biology: understanding the cell's functional organization. Nat Rev Genet 5:101–113
4. Jeong H, Mason SP, Barabási AL, Oltvai ZN (2001) Lethality and centrality in protein networks. Nature 411:41–42
5. Fry DC (2006) Protein-protein interactions as targets for small molecule drug discovery. Biopolymers 84:535–552
6. Kiemer L, Cesareni G (2007) Comparative interactomics: comparing apples and pears? Trends Biotechnol 25:448–454
7. Chothia C, Janin J (1975) Principles of protein-protein recognition. Nature 256:705–708

8. Larsen TA, Olson AJ, Goodsell DS (1998) Morphology of protein-protein interfaces. Structure 6:421–427
9. Nooren IMA, Thornton JM (2003) Diversity of protein-protein interactions. EMBO J 22:3486–3492
10. Pawson T (2003) Organization of cell-regulatory systems through modular-protein-interaction domains. Philos Transact A Math Phys Eng Sci 361:1251–1262
11. Nooren IMA, Thornton JM (2003) Structural characterisation and functional significance of transient protein-protein interactions. J Mol Biol 325:991–1018
12. Teichmann SA (2002) The constraints protein-protein interactions place on sequence divergence. J Mol Biol 324:399–407
13. Mintseris J, Weng Z (2005) Structure, function, and evolution of transient and obligate protein-protein interactions. Proc Natl Acad Sci USA 102:10930–10935
14. Doolittle RF (1995) The multiplicity of domains in proteins. Annu Rev Biochem 64:287–314
15. Moore AD, Björklund AK, Ekman D et al (2008) Arrangements in the modular evolution of proteins. Trends Biochem Sci 33:444–451
16. Kummerfeld SK, Teichmann SA (2005) Relative rates of gene fusion and fission in multi-domain proteins. Trends Genet 21:25–30
17. Finn RD, Marshall M, Bateman A (2005) iPfam: visualization of protein-protein interactions in PDB at domain and amino acid resolutions. Bioinformatics 21:410–412
18. Stein A, Russell RB, Aloy P (2005) 3did: interacting protein domains of known three-dimensional structure. Nucleic Acids Res 33:D413–D417
19. Schuster-Böckler B, Bateman A (2007) Reuse of structural domain-domain interactions in protein networks. BMC Bioinform 8:259
20. Koonin EV, Wolf YI, Karev GP (2002) The structure of the protein universe and genome evolution. Nature 420:218–223
21. Lander ES, Linton LM, Birren B et al (2001) Initial sequencing and analysis of the human genome. Nature 409:860–921
22. Weiner J, Moore AD, Bornberg-Bauer E (2008) Just how versatile are domains? BMC Evol Biol 8:285
23. Basu MK, Carmel L, Rogozin IB et al (2008) Evolution of protein domain promiscuity in eukaryotes. Genome Res 18:449–461
24. Tuncbag N, Kar G, Keskin O et al (2009) A survey of available tools and web servers for analysis of protein-protein interactions and interfaces. Brief Bioinform 10:217–232
25. Young KH (1998) Yeast two-hybrid: so many interactions, (in) so little time. Biol Reprod 58:302–311
26. Uetz P, Giot L, Cagney G et al (2000) A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. Nature 403:623–627
27. Ito T, Chiba T, Ozawa R et al (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc Natl Acad Sci 98:4569–4574
28. Arifuzzaman M, Maeda M, Itoh A et al (2006) Large-scale identification of protein-protein interaction of Escherichia coli K-12. Genome Res 16:686–691
29. Rain JC, Selig L, Reuse HD et al (2001) The protein-protein interaction map of Helicobacter pylori. Nature 409:211–215
30. Giot L, Bader JS, Brouwer C et al (2003) A protein interaction map of Drosophila melanogaster. Science 302:1727–1736
31. Li S, Armstrong CM, Bertin N et al (2004) A map of the interactome network of the metazoan C. elegans. Science 303:540–543
32. LaCount DJ, Vignali M, Chettier R et al (2005) A protein interaction network of the malaria parasite Plasmodium falciparum. Nature 438:103–107
33. Parrish JR, Yu J, Liu G et al (2007) A proteome-wide protein interaction map for Campylobacter jejuni. Genome Biol 8:R130
34. Stelzl U, Worm U, Lalowski M et al (2005) A human protein-protein interaction network: a resource for annotating the proteome. Cell 122:957–968
35. Bork P, Jensen LJ, von Mering C et al (2004) Protein interaction networks from yeast to human. Curr Opin Struct Biol 14:292–299
36. Deane CM, Salwiński Ł, Xenarios I et al (2002) Protein interactions: two methods for assessment of the reliability of high throughput observations. Mol Cell Proteomics 1:349–356
37. Hart GT, Ramani AK, Marcotte EM (2006) How complete are current yeast and human protein-interaction networks? Genome Biol 7:120
38. Yu H, Braun P, Yildirim MA et al (2008) High-quality binary protein interaction map of the yeast interactome network. Science 322:104–110
39. Yook S, Oltvai ZN, Barabási A (2004) Functional and topological characterization of protein interaction networks. Proteomics 4:928–942

40. Han JJ, Dupuy D, Bertin N et al (2005) Effect of sampling on topology predictions of protein-protein interaction networks. Nat Biotechnol 23:839–844
41. Han JJ, Bertin N, Hao T et al (2004) Evidence for dynamically organized modularity in the yeast protein-protein interaction network. Nature 430:88–93
42. Jin G, Zhang S, Zhang X et al (2007) Hubs with network motifs organize modularity dynamically in the protein-protein interaction network of yeast. PLoS ONE 2:e1207
43. Ispolatov I, Yuryev A, Mazo I et al (2005) Binding properties and evolution of homodimers in protein-protein interaction networks. Nucleic Acids Res 33:3629–3635
44. Lukatsky DB, Shakhnovich BE, Mintseris J et al (2007) Structural similarity enhances interaction propensity of proteins. J Mol Biol 365:1596–1606
45. Pastor-Satorras R, Smith E, Solé RV (2003) Evolving protein interaction networks through gene duplication. J Theor Biol 222:199–210
46. Evlampiev K, Isambert H (2008) Conservation and topology of protein interaction networks under duplication-divergence evolution. Proc Natl Acad Sci USA 105:9863–9868
47. Kashtan N, Alon U (2005) Spontaneous evolution of modularity and network motifs. Proc Natl Acad Sci USA 102:13773–13778
48. Pagel M, Meade A, Scott D (2007) Assembly rules for protein networks derived from phylogenetic-statistical analysis of whole genomes. BMC Evol Biol 7(Suppl 1):S16
49. Pereira-Leal JB, Teichmann SA (2005) Novel specificities emerge by stepwise duplication of functional modules. Genome Res 15:552–559
50. Beltrao P, Serrano L (2007) Specificity and evolvability in eukaryotic protein interaction networks. PLoS Comput Biol 3:e25
51. Lynch M, Conery JS (2000) The evolutionary fate and consequences of duplicate genes. Science 290:1151–1155
52. Lynch M, O'Hely M, Walsh B et al (2001) The probability of preservation of a newly arisen gene duplicate. Genetics 159:1789–1804
53. Berg J, Lässig M, Wagner A (2004) Structure and evolution of protein interaction networks: a statistical model for link dynamics and gene duplications. BMC Evol Biol 4:51
54. Gamsjaeger R, Liew CK, Loughlin FE et al (2007) Sticky fingers: zinc-fingers as protein-recognition motifs. Trends Biochem Sci 32:63–70
55. Mészáros B, Simon I, Dosztányi Z (2009) Prediction of protein binding regions in disordered proteins. PLoS Comput Biol 5:e1000376
56. Neduva V, Russell RB (2005) Linear motifs: evolutionary interaction switches. FEBS Lett 579:3342–3345
57. Fraser HB, Wall DP, Hirsh AE (2003) A simple dependence between protein evolution rate and the number of protein-protein interactions. BMC Evol Biol 3:11
58. Fraser HB, Hirsh AE (2004) Evolutionary rate depends on number of protein-protein interactions independently of gene expression level. BMC Evol Biol 4:13
59. Jordan IK, Wolf YI, Koonin EV (2003) No simple dependence between protein evolution rate and the number of protein-protein interactions: only the most prolific interactors tend to evolve slowly. BMC Evol Biol 3:1
60. Bloom JD, Adami C (2003) Apparent dependence of protein evolutionary rate on number of interactions is linked to biases in protein-protein interactions data sets. BMC Evol Biol 3:21
61. Kim PM, Lu LJ, Xia Y et al (2006) Relating three-dimensional structures to protein networks provides evolutionary insights. Science 314:1938–1941
62. Guelzim N, Bottani S, Bourgine P et al (2002) Topological and causal structure of the yeast transcriptional regulatory network. Nat Genet 31:60–63
63. Wuchty S, Oltvai ZN, Barabási AL (2003) Evolutionary conservation of motif constituents in the yeast protein interaction network. Nat Genet 35:176–179
64. Amoutzias GD, Robertson DL, Oliver SG et al (2004) Convergent evolution of gene networks by single-gene duplications in higher eukaryotes. EMBO Rep 5:274–279
65. Deeds EJ, Ashenberg O, Gerardin J et al (2007) Robust protein protein interactions in crowded cellular environments. Proc Natl Acad Sci USA 104:14952–14957
66. Noguchi CT, Schechter AN et al (1985) Sickle hemoglobin polymerization in solution and in cells. Annu Rev Biophys Chem 14:239–263
67. Kuriyan J, Eisenberg D (2007) The origin of protein interactions and allostery in colocalization. Nature 450:983–990
68. Wang M, Caetano-Anollés G (2009) The evolutionary mechanics of domain organization in proteomes and the rise of modularity in the protein world. Structure 17:66–78
69. Bennett MJ, Choe S, Eisenberg D (1994) Domain swapping: entangling alliances between proteins. Proc Natl Acad Sci USA 91:3127–3131

70. Bennett MJ, Schlunegger MP, Eisenberg D (1995) 3D domain swapping: a mechanism for oligomer assembly. Protein Sci 4:2455–2468
71. Hedges SB (2002) The origin and evolution of model organisms. Nat Rev Genet 3:838–849
72. Cesareni G, Ceol A, Gavrila C et al (2005) Comparative interactomics. FEBS Lett 579:1828–1833
73. Kelley BP, Sharan R, Karp R et al (2005) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. Proc Natl Acad Sci USA 100:11394–11399
74. Sharan R, Suthram S, Kelley RM et al (2005) Conserved patterns of protein interaction in multiple species. Proc Natl Acad Sci USA 102:1974–1979
75. Gerke M, Bornberg-Bauer E, Jiang X et al (2006) Finding common protein interaction patterns across organisms. Evol Bioinform Online 2:45–52
76. Aloy P (2007) Shaping the future of interactome networks. Genome Biol 8:316
77. Ideker T, Valencia A (2006) Bioinformatics in the human interactome project. Bioinformatics 22:2973–2974
78. Shannon P, Markiel A, Ozier O et al (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13:2498–2504
79. Hernandez-Toro J, Prieto C, las Rivas JD (2007) APID2NET: unified interactome graphic analyzer. Bioinformatics 23:2495–2497
80. Wu J, Vallenius T, Ovaska K et al (2009) Integrated network analysis platform for protein-protein interactions. Nat Methods 6:75–77
81. Suthram S, Shlomi T, Ruppin E et al (2006) A direct comparison of protein interaction confidence assignment schemes. BMC Bioinform 7:360
82. Bader GD, Betel D, Hogue CWV (2003) BIND: the Biomolecular Interaction Network Database. Nucleic Acids Res 31:248–250
83. Stark C, Breitkreutz B, Reguly T et al (2006) BioGRID: a general repository for interaction datasets. Nucleic Acids Res 34:D535–D539
84. Xenarios I, Rice DW, Salwinski L et al (2000) DIP: the database of interacting proteins. Nucleic Acids Res 28:289–291
85. Peri S, Navarro JD, Kristiansen TZ et al (2004) Human protein reference database as a discovery resource for proteomics. Nucleic Acids Res 32:D497–D501
86. Hermjakob H, Montecchi-Palazzi L, Lewington C et al (2004) IntAct: an open source molecular interaction database. Nucleic Acids Res 32:D452–D455
87. Chatr-aryamontri A, Ceol A, Palazzi L et al (2007) MINT: the Molecular INTeraction database. Nucleic Acids Resm 35:D572–D574
88. Pagel P, Kovac S, Oesterheld M et al (2005) The MIPS mammalian protein-protein interaction database. Bioinformatics 21:832–834
89. Raghavachari B, Tasneem A, Przytycka TM et al (2008) DOMINE: a database of protein domain interactions. Nucleic Acids Res 36:D656–D661
90. Brown KR, Jurisica I (2007) Unequal evolutionary conservation of human protein interactions in interologous networks. Genome Biol 8:R95
91. Keskin O, Nussinov R, Gursoy A (2008) PRISM: protein-protein interaction prediction by structural matching. Methods Mol Biol 484:505–521
92. McDowall MD, Scott MS, Barton GJ (2009) PIPs: human protein-protein interaction prediction database. Nucleic Acids Res 37:D651–D656
93. Han K, Park B, Kim H et al (2004) HPID: the human protein interaction database. Bioinformatics 20:2466–2470

Chapter 18

Cytoscape: Software for Visualization and Analysis of Biological Networks

Michael Kohl, Sebastian Wiese, and Bettina Warscheid

Abstract

Substantial progress has been made in the field of "omics" research (e.g., Genomics, Transcriptomics, Proteomics, and Metabolomics), leading to a vast amount of biological data. In order to represent large biological data sets in an easily interpretable manner, this information is frequently visualized as graphs, i.e., a set of nodes and edges. Nodes are representations of biological molecules and edges connect the nodes depicting some kind of relationship.

Obviously, there is a high demand for computer-based assistance for both visualization and analysis of biological data, which are often heterogeneous and retrieved from different sources. This chapter focuses on software tools that assist in visual exploration and analysis of biological networks. Global requirements for such programs are discussed. Utilization of visualization software is exemplified using the widely used Cytoscape tool. Additional information about the use of Cytoscape is provided in the Notes section. Furthermore, special features of alternative software tools are highlighted in order to assist researchers in the choice of an adequate program for their specific requirements.

1. Introduction

The advent of several "omics" research fields such as Transcriptomics, Proteomics, and Metabolomics has led to substantial progress in acquiring knowledge about biological functions on different scales ranging from molecular to physiological levels. The ongoing accomplishments toward a comprehensive understanding of biological systems mainly rely on the effectiveness of high-throughput techniques. Application of these technologies results in the generation of huge data sets, and interpretation of this large amount of information is a current challenge.

Biological data are often stored in tabular form. However, simple lists or matrices that contain the results of experiments are often not adequate for tracking interdependencies of biological entities. Obviously, there is a need for computer-based assistance

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696,
DOI 10.1007/978-1-60761-987-1_18,

for both visualization and analysis of biological data. To this end, data visualization software provides important means in order to represent data in an easily interpretable manner. An important task of data visualization software is, therefore, the presentation of biological relationships typically leading to a network representation of related biological processes (biological pathways). In general, pathways are visualized as graphs, i.e., a set of nodes and edges. Nodes are representations of biological molecules (e.g., different classes of nucleic acids or proteins) or larger entities such as molecular machines. Edges connect the nodes, depicting some kind of relationship or interaction (e.g., any type of chemical modification, inhibition, or activation).

This chapter focuses on softwares that assist in visual exploration and the analysis of biological networks. During the last decade, several valuable software tools have been developed varying in both complexity and the suggested fields of application.

The chapter presented here is structured as follows. First, basic requirements for the visualization of biological networks are discussed. Second, the application of Cytoscape (1), a widely used visualization tool, is described followed by the discussion of special features of some alternative software tools. Finally, additional information about the use of Cytoscape for processing of proteomic data is provided in the "Notes" section.

2. Basic Requirements for Computer-Aided Visualization of Biological Pathways

In this paragraph, several tasks are discussed that frequently occur in the context of software-aided visualization, analysis, and assembly of biological pathway data.

1. Biological systems are complex. Each constituent of the system may be characterized by a set of attributes of interest. Adequate software should permit a flexible encoding of such attributes with visual features (such as color, size, or the font used). For example, pathway proteins with abundance in a user-defined range may share the same color.
2. An increasing amount of biological information is available from public repositories. In order to facilitate data integration from various sources, a direct database connection including both querying and download possibilities is desirable.
3. State-of-the-art visualization software not only represents biological relationships from the interpretation of textual notations, but also enables the construction and editing of such networks by the user.
4. In order to relate hypotheses of biological pathways and results of experiments, adequate software should support

mapping of experimental data into the displayed biological network. Moreover, further information about, for example, the subcellular localization of proteins or time course data can be related to both nodes and edges. Integration of such information may help to overcome the limits of static snapshot-like representations (cellular state) and may allow for the establishment of a rather dynamic representation of biological processes.

5. Large molecular interaction networks ranging over several spatial levels (e.g., networks that include a detailed description of biological processes in different compartments of an eukaryotic cell) may comprise thousands of nodes and edges. Therefore, a very important feature of an adequate visualization tool is an advanced scalability function. This implies a comfortable zooming function in order to reduce the degree of complexity, for example. Furthermore, scalability opportunities facilitate access to biological information in different parts of the network.
6. Networks with a large number of nodes and edges are often difficult to comprehend and the arrangement of pathways is generally time consuming. Therefore, an important feature of visualization tools is the availability of appropriate layout algorithms enabling to organize and align the nodes of a network. Layout generation may also support automatic adjustment of the network size.
7. Rapid progress in biomolecular research most likely requires refinement of the basic features (e.g., analysis, graphical representation, and input/output formats) of visualization software packages. An opportunity to extend the software with new features is thus advisable. Open-source software enables the user to adopt the existing functionality quickly. Some software tools provide a possibility for the integration of plugins and allow for extensions to access the core features of the system.
8. Sharing of pathway information and further processing using other software tools involve encoding and export/import in standard data formats. Because visualization tools are used to prepare graphical representations of biological information for publication, support of different image formats is an important feature.
9. For an adequate representation of large biological networks, the application of adequate filtering techniques and thus reduction of network complexity are often essential. A further requirement is the possibility to select subsets of nodes or edges with respect to different criteria, which may likely facilitate the discovery of biological mechanisms.

10. Specific visualization tools also support statistical analysis in order to allow the comparison of different sets of experimental data mapped onto the biological network.

3. Data Visualization Software

In this paragraph, computer-aided visualization of biological networks is illustrated. Cytoscape (1) is chosen as an adequate example, because this software package generally meets the majority of the above given basic requirements in subheading 2.

Furthermore, "Cytoscape" has a vivid community (see Note 1) and is widely used in both proteomic research and a wide range of life sciences applications in general. Currently (December 2009), over 487 publications are referencing Shannon et al. (1). The software is utilized in several software environments that aim at the analysis of biological data. For example, Cytoscape is part of the data mining solution GENPAC (http://www.nalapro.com/index_e.html) and is used as visualization tool in the context of type 1 diabetes research projects (http://www.t1dbase.org/page/Welcome/display).

In order to provide knowledge about alternative software solutions, comparable tools are additionally listed and information about their distinct capabilities is provided (see Subheading 3.2).

3.1. Cytoscape

3.1.1. Description of the Basic Cytoscape Features (Network Establishment, Annotation, Analysis, and Visualization)

The following description refers to Cytoscape version 2.6.3, which was installed on a standard PC (Intel Pentium Dual Core, 2.5 GHz, 4 GB Ram, Windows XP Professional, Service Pack 3). Installation was straightforward on this system. Cytoscape is an open-source software and released under terms of the GNU LESSER GENERAL PUBLIC LICENSE (LGPL) v. 2.1.

Cytoscape is designed for the visualization of biomolecular interaction networks and pathways. The software provides the ability to depict very large networks (100,000+ nodes and edges, see Note 2) and to visualize interactions of the constituents of different molecular networks (e.g., protein–protein or protein–gene interactions). The software supports the generation of biological networks by applying an editor module. Established networks can easily be imported and several file formats (.gpml, BioPAX (e.g., .owl files), .xml, .rdf, .gml, .xgmml, .sif, .sbml, .txt) are supported. Latest versions of Cytoscape (version 2.6 and above) act as a web service client for public biological databases. It is, therefore, possible to import existing networks and/or annotation data from public repositories (e.g., PathwayCommons, IntAct, BioMart, NCBI Entrez Gene and PICR). Cytoscape equally supports the import of experimentally derived data sets (see Note 3 for a quick start).

The software uses *attributes* and *annotations* to include additional biological information. The attributes comprise specific properties of a network constituent (a node or an edge). For example, an edge may have an attribute that quantifies the extent of the interaction between the associated nodes.

In contrast, annotations refer to an ontology, i.e., a set of fixed/controlled terms that are hierarchically structured to reflect their semantic relations (e.g., the Gene Ontology database (2, 3)). Annotations thus apply to groups of nodes or edges that share the same characteristics. Additional information such as experimental data obtained from high-throughput experiments can be integrated into a network, thereby permitting advanced data analysis.

In addition, Cytoscape enables editing of an existing network (e.g., adding or removing nodes or edges from a data set as well as modifying both node names and attributes) and combining smaller networks to a single, larger network (see Note 4).

A large set of layouts is implemented in the standard installation of Cytoscape (Fig. 1, see Note 5), facilitating the proper organization of networks (see Note 6). In order to establish and

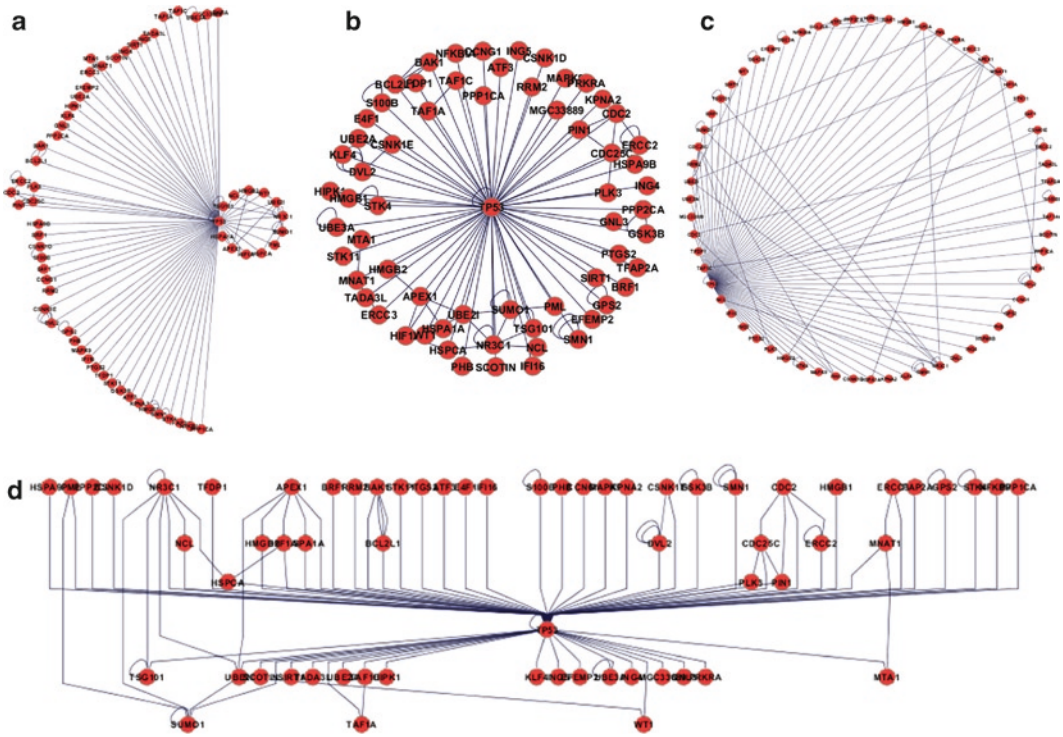


Fig. 1. Screenshot of several layouts provided by *Cytoscape*. Layout algorithms are applied to a data-set reported previously in (31). The screenshots show a selection of this larger network, i.e., the immediate neighbors of TP53 (tumor protein 53). Results of applications of different layout algorithms on this small network are shown: (**a**) yfiles – *circular layout*; (**b**) Cytoscape layout – *Spring embedded*; (**c**) Jgraph – *Circle layout*; and (**d**), yfiles – *hierarchic*.

comprehend large networks, Cytoscape provides several possibilities for data representation. For example, related nodes (i.e., interacting nodes that perform a common function) can be combined into a single parent node.

An important feature of any visualization software is the graphical representation of information, which is mapped to both nodes and edges. To this end, Cytoscape supports the encoding of any attribute with a large set of visual properties (e.g., size, color, or geometrical shape). A given set of encoded attributes can be stored as so-called *Visual Styles*. This software scheme is very powerful and allows the user to change the visual appearance of a visualized network easily.

Cytoscape uses three different kinds of mappers to alter note and edge properties (Fig. 2). The pass-through mapper may be used to label nodes directly with the respective protein name. Discrete mappings allow the use of visual properties to reflect biological characteristics. For example, the node color is indicative of the cellular location of a given protein. The continuous

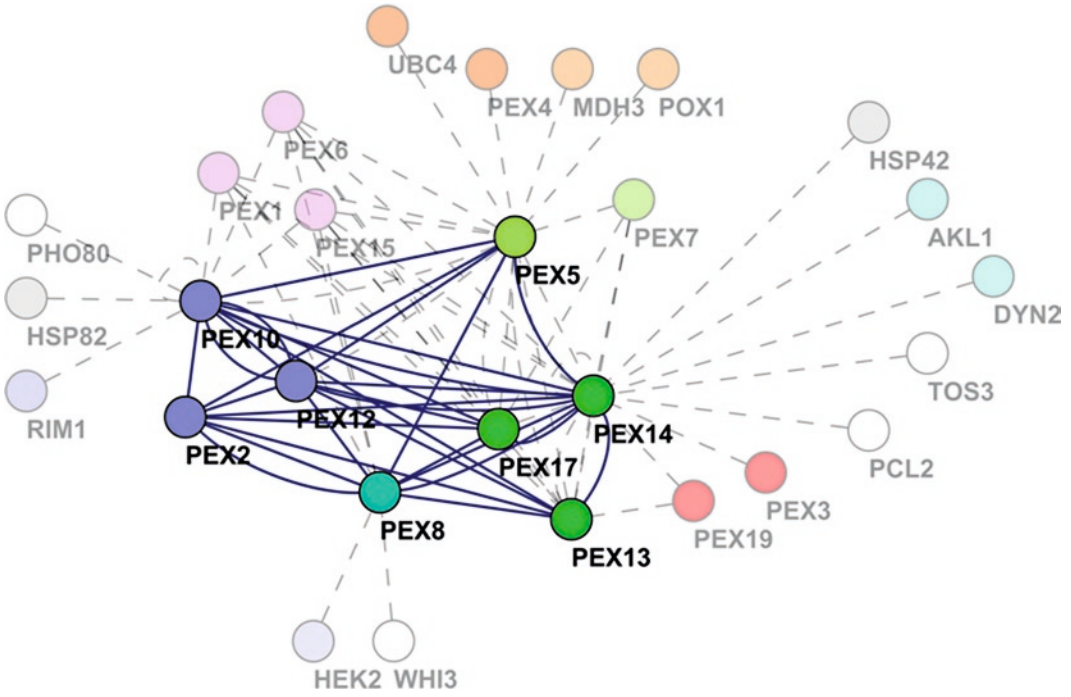


Fig. 2. Visualization of the peroxisomal matrix protein import machinery in the yeast *Saccharomyces cerevisiae* using Cytoscape. The interactions of Pex2p, Pex8p, Pex10p, Pex12p, and Pex14p were analyzed by Agne et al. (32). Interactions (*edges*) and proteins (*nodes*) investigated in this publication are highlighted compared to further interactions retrieved from the Biogrid database. *Nodes*, representing individual proteins, were arranged manually. Several *node* and *edge* attributes were used for visualizing different attributes of the interaction network. The node labels were linked using a pass-through mapper with the protein names. Node colors were used to visualize the respective protein function. Interactions and proteins investigated by Agne et al. (32) were highlighted using edge color, edge line style, edge line width, edge opacity, node border opacity, node label opacity, and node opacity, respectively.

mapper can further be used for the implementation of color gradients reflecting different interaction strengths (see Fig. 2 and Note 7). Such flexible visualization opportunities generally facilitate data analysis since functionally related nodes, as well as the response of biological processes to experimental perturbations is visualized.

Cytoscape further supports versatile filtering methods (see Note 8), which is an important property of pathway analysis. The selection of nodes can be performed with respect to the state of any arbitrary attribute. For example, nodes that show large differences in different experiments can be selected. Selected nodes can then be combined to subnetworks for further data analysis. A filtering toolbox provides several predefined filters, which further facilitate the analysis of the network.

Pathway representations are often used in scientific papers. The Cytoscape export function simplifies the creation of such figures, as high-quality images of the networks ready for publication can be stored in common formats (e.g., .pdf, .jpeg, and .png).

3.1.2. Advanced Features of the Cytoscape Software

This section deals with several techniques used for network generation and data analysis that require installation of Cytoscape extensions. Cytoscape features an advanced plug-in system. Available plug-ins can be downloaded from http://chianti.ucsd.edu/cyto_web/plugins/index.php. On October 14, 2009, 87 plug-ins were hosted at this website (see Notes 9 and 10). In the following, a selection of these plug-ins is described. Furthermore, an example for a possible integration of Cytoscape with other existing software tools is provided.

3.1.2.1. Text Mining

Cytoscape provides flexible and advanced text mining capabilities (4) by including the Agilent Literature Search plug-in. This feature allows for searches in public literature repositories (PubMed – http://www.ncbi.nlm.nih.gov/sites/entrez?db=pubmed, OMIM – http://www.ncbi.nlm.nih.gov/sites/entrez?db=omim, and USPTO – http://patft.uspto.gov) and automatically generates a network with respect to the findings in the literature (see Fig. 3). There are also possibilities for validating the results. Related sentences from the literature can be listed by selecting any node or edge of the network. Each sentence is linked to the data source used, which allows an easy retrieval and evaluation of the research article of interest. Misleading information can be deleted from the list, yielding an immediate update of the network diagram.

3.1.2.2. Identifying Network Modules and Complexes

In this paragraph, *network modules* are defined as a set of cooperating nodes performing a consolidated function. Molecular *complexes* are a sub-category of modules: proteins of a complex constitute in their entirety a macromolecular machine (e.g., the ribosome). The identification of such larger biochemical entities

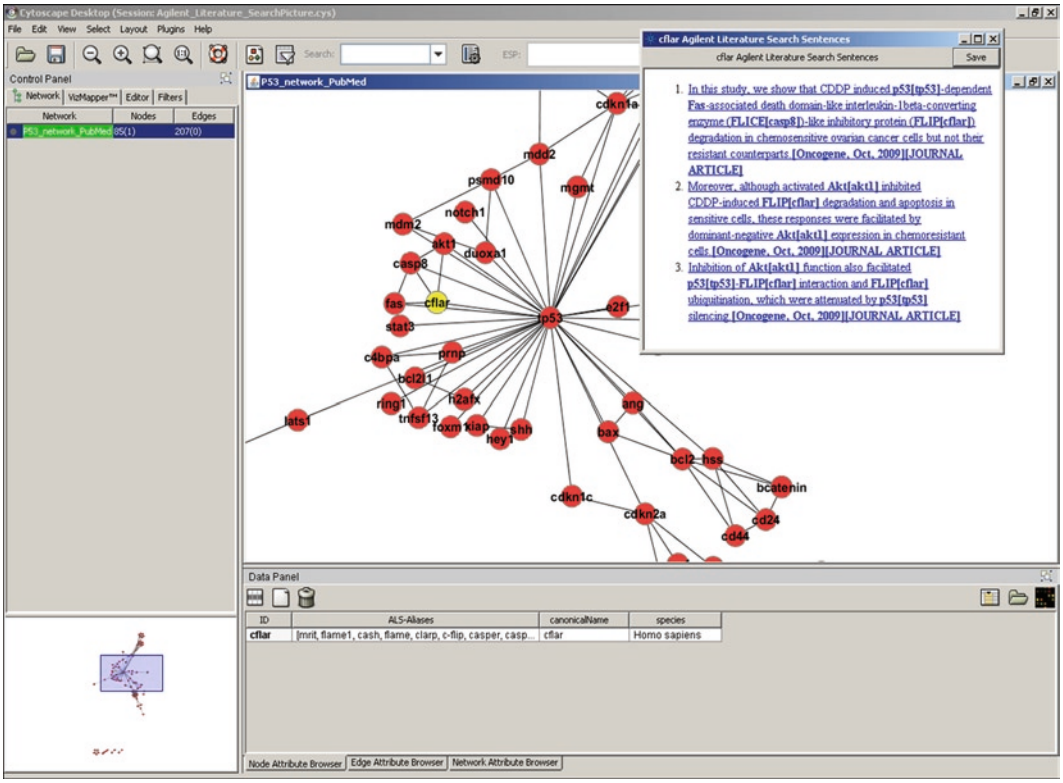


Fig. 3. Screenshot of the Cytoscape GUI. The main canvas shows a detailed view of a network generated with the Agilent Literature Search plug-in. Entries of the PubMed database were searched using "P53" and both "*Homo sapiens*" and "human" as search parameters. Analysis of reported associations was restricted to a maximum of 50 articles. The smaller window in the *upper right part* of the figure shows the sentences of the search results that were used to extract putative associations for a CASP8 and FADD-like apoptosis regulator (cflar). Note that this protein is selected as indicated by the *yellow* node color. The data panel in the *bottom* shows aliases of the selected protein. The small panel in the *bottom left* is used for adjusting the view of the central window showing a cut-out of the network.

serves as an example for the data analysis capabilities of Cytoscape and its plug-ins.

This analysis requires the application of both the MCODE (5) and the BiNGO (6) plug-in. It is assumed that clustered nodes (i.e., intensively connected parts of the network) indicate molecular complexes. MCODE implements a graph theoretic clustering algorithm that searches for closely connected parts of the network. The algorithm enables investigation of cluster interconnectivity. To this end, the algorithm ranks the proposed complex identification. Higher ranks are assigned to the larger and the more intensively connected complexes. Results of MCODE can then be passed to BiNGO for further processing.

BiNGO detects over- or underrepresentation of Gene Ontology (GO) categories within a group of genes or within a biological network. It is assumed that proteins comprising a specific molecular complex participate in the same biological

processes. BiNGO analysis should, therefore, result in a significant enrichment of certain GO terms for nodes that form a network module or a molecular complex. If both BiNGO and MCODE indicate functional association, there is further evidence that these nodes form a unit on a higher biological scale.

3.1.2.3. Using Cytoscape for Performing Advanced Bioinformatic Analysis with Respect to Molecular Interaction Networks

Integration of several plug-ins into the Cytoscape core application was suggested in order to perform a comprehensive and standardized bioinformatic data analysis workflow (7).

Cytoscape can be combined with other existing popular software tools to establish a larger software environment, which aims at analyzing proteomic data on a larger scale. This environment is named *Integrative Proteomics Data Analysis Pipeline* (IPDAP (8)). It includes software for data conversion, protein identification, and quantification as well as a framework of linked tools providing capabilities for systems biology analysis. Cytoscape is part of this systems biology framework, in which different software tools and web resources (e.g., public databases such as KEGG (9, 10), STRING (11, 12), and BioCyc (13)) interact via the Gaggle software system (14). The core application of Gaggle is a server program ("Gaggle Boss") that facilitates program interaction by passing messages between software registered in the Gaggle framework. Amongst others, Gaggle links software developed for microarray analysis (TIGR Multiexperiment Viewer (15)), statistics (R/Bioconductor (16, 17)), navigating and plotting of experimental data (DataMatrixViewer), data exchange between web resources and Gaggle (Firegoose (18)), and visualization of biomolecular interactions (Cytoscape).

3.2. Special Features of Other Data Visualization Software

In this paragraph, further visualization software tools are discussed and their particular strengths and fields of application are highlighted.

The software VANTED (19, 20) features dynamic graph layout. This means that the map of the biological network changes dynamically with respect to pathway-associated data (e.g., the results of time series experiments). Furthermore, VANTED includes statistical functionalities enabling the simultaneous comparison and analysis of multiple data-sets.

CellDesigner (21, 22) is a valuable tool for researchers with a strong interest not only in representation but also in the modeling of biological processes. It uses a diagrammatic network editing software, which supports the graphical notation system developed by Kitano (23). Process diagrams generated with CellDesigner can be translated into the Systems Biology Markup Language (24), a standard format widely used for encoding biological network models. Because CellDesigner can be connected with the Systems Biology Workbench SBW, (25), the software provides modeling opportunities using a SBW integrated simulation

engine for SBML files such as Jarnac (26). Similar to CellDesigner, the MetaReg software (27) also combines visualization and modeling tasks. The software further includes advanced algorithms for refinement of the model.

ProViz (28) is potentially a good choice for researchers interested in the processing of very large networks due to the fact that it applies a powerful graph-rendering engine, enabling the handling of millions of nodes and edges.

Biological Networks (29) is a free Systems Biology software platform intended for both the analysis and the visualization of biological pathways. The software is a user interface built on top of PathSys (30). The BiologicalNetworks/PathSys platform involves a large number of public biochemical data resources (including resources of genomic, transcriptomic, and proteomic data). A special feature of the software package is the integration of a powerful sql-like query language, which provides a flexible opportunity for network analysis.

4. Notes

1. Cytoscape has a very active community, whose members are constantly developing new plug-ins and are also eager to help both new and experienced users with any problem they might encounter.
2. Some problems may occur concerning "out of memory" errors. Application of Cytoscape to larger networks may require adjustment of the maximum memory size. There are several options available for allocating additional memory for Cytoscape. Detailed description is given in the Cytoscape manual, which is available from http://cytoscape.wodaklab.org/wiki/Cytoscape_User_Manual. However, when dealing routinely with larger networks, the use of software startup from the command line is not preferable as it does not change the default memory size allocated for the software.
3. For data import, the user should place the interaction source (e.g., the bait protein) in one column and the interaction target (e.g., a co-purified protein) in another column of an MS Excel sheet. As edges have a direction in Cytoscape, it is necessary to pay attention to the position of the respective proteins. Next, the user should use File\Import\Network-from-table to import the generated file. Edge attributes can be imported in parallel by marking the respective column in the Import-GUI. Node attributes have to be imported separately.
4. The user should ensure the completeness of the data to be visualized prior to changing the layout of the network.

While single nodes or edges can be added easily to an existing network without altering it, merging two networks will lead to a new default network, which automatically replaces the previous layout.

5. The Edge-weighted Spring Embedded Layout is very powerful for the unbiased portrayal of networks. Nodes with several interconnecting edges are clustered, while Nodes with only a single linkage are distributed further apart.
6. Although Cytoscape is generally able to visualize large networks, the use of layout algorithms, and in particular, the use of the highly popular spring embedded layout may be hampered due to long processing times. Due to restrictions in memory space as stated above, the user may rather focus on subsets of a larger network.
7. The use of the same node/edge property to visualize different attributes in two different visual styles is possible but should be avoided. Saving and reopening of the document may result in the disruption of one of the visual styles. This restriction may be overcome in later versions of Cytoscape.
8. Cytoscape supports the definition of logical combinations (using AND, OR, and XOR as logical operators) of existing filters, yielding a so-called "Boolean Meta-Filter." This modular concept facilitates the application of complex filtering methods. Boolean Meta-Filters are efficient tools for the advanced analysis of larger biological networks.
9. A major focus of Cytoscape extensions is the analysis of networks (32 plug-ins). A number of plug-ins add features for the import of networks and the related annotations (20 plug-ins). Other extensions mainly enable new layout capabilities or provide support for different file formats and connections to databases. The list of plug-ins viewed at the Cytoscape plug-in page includes a description and information about the compatibility with existing Cytoscape versions. The webpage also includes a tutorial with information about writing and the integration of individual plug-in solutions.

   In most cases, installation of plug-ins requires copying of a .jar-file into the *plug-ins* directory, which is located directly below the Cytoscape program root. Additional steps may be required: Restart of the Cytoscape GUI and choose *Plug-ins – Manage plug-ins* from the menu. This yields a categorized list of plug-ins that are prepared for installation. Integration of the plug-in of interest into the GUI can then be performed simply by selecting the plug-in and clicking the install button.
10. The plug-in system supports installation of themes, which are bundles of related plug-ins. These plug-in combinations are designed to solve frequently occurring tasks such as providing

access to common public resources of biological network data. The themes concept will also lead to customized versions of Cytoscape tailored to the necessities of specific users. However, currently there are only few themes available.

Acknowledgments

This work is funded by the Bundesministerium für Bildung und Forschung (BMBF, grant 01 GS 08143) and by the Deutsche Forschungsgemeinschaft within the SFB 642. Thanks are also due to the Agilent Laboratories and Annette Adler for the financial support for the color versions of the figures.

References

1. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D et al (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res 13:2498–2504
2. Harris MA, Clark J, Ireland A, Lomax J, Ashburner M, Foulger R et al (2004) The gene ontology (GO) database and informatics resource. Nucleic Acids Res 32:D258–D261
3. Ashburner M, Ball CA, Blake JA, Butler H, Cherry JM, Corradi J et al (2001) Creating the gene ontology resource: design and implementation. Genome Res 11:1425–1433
4. Vailaya A, Bluvas P, Kincaid R, Kuchinsky A, Creech M, Adler A (2005) An architecture for biological information extraction and representation. Bioinformatics 21:430–438
5. Bader GD, Hogue CW (2003) An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinform 4:2
6. Maere S, Heymans K, Kuiper M (2005) BiNGO: a cytoscape plugin to assess overrepresentation of gene ontology categories in biological networks. Bioinformatics 21:3448–3449
7. Cline MS, Smoot M, Cerami E, Kuchinsky A, Landys N, Workman C et al (2007) Integration of biological networks and gene expression data using cytoscape. Nat Protoc 2:2366–2382
8. Rho S, You S, Kim Y, Hwang D (2008) From proteomics toward systems biology: integration of different types of proteomics data into network models. BMB Rep 41:184–193
9. Ogata H, Goto S, Sato K, Fujibuchi W, Bono H, Kanehisa M (1999) KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res 27:29–34
10. Kanehisa M, Goto S (2000) KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res 28:27–30
11. von Mering C, Jensen LJ, Snel B, Hooper SD, Krupp M, Foglierini M et al (2005) STRING: known and predicted protein–protein associations, integrated and transferred across organisms. Nucleic Acids Res 33:D433–D437
12. Jensen LJ, Kuhn M, Stark M, Chaffron S, Creevey C, Muller J et al (2009) STRING 8: a global view on proteins and their functional interactions in 630 organisms. Nucleic Acids Res 37:D412–D416
13. Krummenacker M, Paley S, Mueller L, Yan T, Karp PD (2005) Querying and computing with BioCyc databases. Bioinformatics 21:3454–3455
14. Shannon PT, Reiss DJ, Bonneau R, Baliga NS (2006) The Gaggle: an open-source software system for integrating bioinformatics software and data sources. BMC Bioinform 7:176
15. Saeed AI, Sharov V, White J, Li J, Liang W, Bhagabati N et al (2003) TM4: a free, open-source system for microarray data management and analysis. Biotechniques 34:374–378
16. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S et al (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5:R80
17. Team RDC (2005) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria
18. Bare JC, Shannon PT, Schmid AK, Baliga NS (2007) The Firegoose: two-way integration of diverse data from different bioinformatics

web resources with desktop applications. BMC Bioinform 8:456

19. Klukas C, Junker BH, Schreiber F (2006) The VANTED software system for transcriptomics, proteomics and metabolomics analysis. J Pestic Sci 31:289–292
20. Junker BH, Klukas C, Schreiber F (2006) VANTED: a system for advanced data analysis and visualization in the context of biological networks. BMC Bioinform 7:109
21. Funahashi A, Matsuoka Y, Jouraku A, Morohashi M, Kikuchi N, Kitano H (2008) Cell Designer 3.5: a versatile modeling tool for biochemical networks. Proc IEEE 96:1254–1265
22. Funahashi A, Tanimura N, Morohashi M, Kitano H (2003) Cell Designer: a process diagram editor for gene-regulatory and biochemical networks. BioSilico 1:159–162
23. Kitano H, Funahashi A, Matsuoka Y, Oda K (2005) Using process diagrams for the graphical representation of biological networks. Nat Biotechnol 23:961–966
24. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H et al (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19:524–531
25. Hucka M, Finney A, Sauro H, Kovitz B, Keating S, Matthews J et al. (2003) Introduction to the systems biology workbench. [http://www.sys-bio.org/caltechSBW/sbwDocs/docs/index.html]
26. Sauro HM, Hucka M, Finney A, Wellock C, Bolouri H, Doyle J et al (2003) Next generation simulation tools: the systems biology workbench and BioSPICE integration. OMICS 7:355–372
27. Ulitsky I, Gat-Viks I, Shamir R (2008) MetaReg: a platform for modeling, analysis and visualization of biological systems using large-scale experimental data. Genome Biol 9:R1
28. Iragne F, Nikolski M, Mathieu B, Auber D, Sherman D (2005) ProViz: protein interaction visualization and exploration. Bioinformatics 21:272–274
29. Baitaluk M, Sedova M, Ray A, Gupta A (2006) Biological networks: visualization and analysis tool for systems biology. Nucleic Acids Res 34:W466–W471
30. Baitaluk M, Qian XF, Godbole S, Raval A, Ray A, Gupta A (2006) PathSys: integrating molecular interaction graphs for systems biology. BMC Bioinform 7:55
31. Rual JF, Venkatesan K, Hao T, Hirozane-Kishikawa T, Dricot A, Li N et al (2005) Towards a proteome-scale map of the human protein–protein interaction network. Nature 437:1173–1178
32. Agne B, Meindl NM, Niederhoff K, Einwachter H, Rehling P, Sickmann A et al (2003) Pex8p: an intraperoxisomal organizer of the peroxisomal import machinery. Mol Cell 11:635–646

Chapter 19

Text Mining for Systems Modeling

Axel Kowald and Sebastian Schmeier

Abstract

The yearly output of scientific papers is constantly rising and makes it often impossible for the individual researcher to keep up. Text mining of scientific publications is, therefore, an interesting method to automate knowledge and data retrieval from the literature. In this chapter, we discuss specific tasks required for text mining, including their problems and limitations. The second half of the chapter demonstrates the various aspects of text mining using a practical example. Publications are transformed into a vector space representation and then support vector machines are used to classify papers depending on their content of kinetic parameters, which are required for model building in systems biology.

1. Introduction

Since the advent of written language scientific advances are communicated in the form of text-based scientific publications. One of the major aims of text mining (TM) in the life sciences is to transfer the text based information into databases for storage, easy accessibility, and further processing. Up to now, this information transfer is heavily dependent on human experts who curate biological information in the text and further map it onto database entities utilizing ontologies or controlled vocabularies. Despite the endless number of biological databases, most information is still contained within the wealth of scientific publications. The sheer volume of the documents makes automated systems for searching and indexing the contained information indispensable to aid the human curation effort.

Two terms often encountered in TM are "Information Retrieval" and "Information Extraction." Information retrieval relates to the task of finding documents with relevance to a prespecified search query. The query can be of arbitrary complexity (e.g., all documents related to systems biology, all documents

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_19,

that contain the terms "polymerase" and "DNA," etc.). As one can already deduce, information retrieval has a huge impact on all forms of information technology, e.g., search engines for the World Wide Web where a document would be considered a web page. Information extraction, on the contrary, is a type of information retrieval with the task of automatically extracting structured information from within unstructured documents. The structured information to be extracted has a well-defined domain (e.g., protein names, gene names, numbers, etc.).

In a perfect world scenario, an automated computerized system would combine the concepts of information retrieval and information extraction. A collection of scientific documents is searched regarding pre-defined criteria (e.g., all documents relevant to systems biology). Each scientific text of the sub-collection is parsed by the system and analyzed toward identification of biological entities (e.g., proteins, genes, chemicals, drugs, species, etc.), physicochemical entities (e.g., constants, rates, etc.), numerical entities (e.g., numbers), and relationships between them (e.g., reactions, interactions, processes, etc.). The found relationships are mapped onto existing database entities and accompanying information of the relationships is stored (e.g. binding constants, half-life data, reaction velocities, etc.). No human interaction would be necessary to extract these relationships and the system is able to populate such a database for any volume of documents (e.g., the whole of Medline.). Unfortunately, up to now several problems influence the quality of a system that would fulfill all these requirements without any human interaction. For a better understanding of the encountered obstacles, the following sections contain a closer look at specific tasks that are required for the implementation of such a system.

2. Specific Tasks Within Text Mining, Their Problems, and Limitations

2.1. Format Conversion

The first problem that one encounters when dealing with text documents is the digital format of the text itself. Automated TM almost always requires the underlying text to be in ASCII or related format (e.g., Unicode for more complex encoding). The biggest resource of biological knowledge is scientific publications. Full-text articles of these publications are often only distributed in PDF format. A straight-forward conversion from PDF to ASCII text is currently not possible without the loss of at least some information, which could prove critical in the assessment of the information contained within the document itself. Research in this field is currently conducted outside the field of life sciences which is natural, given that the roots of TM lie within the field of information technology. Apart from PDF documents, several

projects such as, for example, PubMed Central (http://www.pubmedcentral.nih.gov) focus on gathering full-text articles that do not violate copyright restrictions in XML format. XML format is ASCII based and offers on top a simple annotated structure within a document that can be easily parsed and further processed by a computer program (e.g., specific tags for titles, sections, etc.). With more and more publishers (e.g., BioMed Central (http://www.biomedcentral.com), Public Library of Science (http://www.plos.org), etc.) adopting an open access policy of their content, collections of full-text scientific articles, such as PubMed Central, are steadily growing, but the mass of information is still only available in formats that are not easily translated into a machine readable encoding.

2.2. Identification of Word and Sentence Boundaries

Despite the rudimentary structure within the XML formats that e.g., PubMed Central offers it is still a challenge for automated computerised systems to identify single word and sentence boundaries (often denoted to as *Tokenization*). The former is of importance while identifying entities of interest within the text, while the latter influences more the semantic and syntactic ambiguity while establishing relationships between identified entities. For example, an interaction of two proteins is most likely but not exclusively conveyed within the same sentence:

> " … and it could be shown that *protein A* and *protein B* interact."
>
> vs.
>
> " … as could be shown for *protein A*. An interacting partner, *protein B*, is similarly …"

2.3. Part-of-Speech Tagging

Automated *part-of-speech (POS)* tagging of words in sentences is another field of research conducted in information technology. Here, the aim is to annotate words in a sentence or phrase with its corresponding part of speech (e.g., verb, noun, adjective, etc.). This annotation is of help while identifying entities and establishing relationships between them (e.g., identify nouns in the sentence, identify verbs that connect nouns, etc.). Tasks related and often processed together with *POS* tagging are *stemming* and *lemmatization*. While either are closely related, *stemming* reduces words to their word stem or root (e.g., "interacted" and "interaction" are reduced to "interact"), whereas *lemmatization* maps words to their lemma or base form (e.g., "good" is a lemma of "better").

2.4. Named Entity Recognition and Word Sense Disambiguation

The task of identifying entities in text is often denoted to as "Named entity recognition" (NER). Word sense disambiguation plays an important role in NER. Especially in the life sciences, words with several meanings often appear disproportional. Examples are words used for genes or proteins that resemble

English words in natural speech, such as, for example the Drosophila genes "decay," "off," "blue," etc., which might relate to a property of the gene but which would not be easily identified by an automated system as a gene. Another example would be the denomination of a gene that resembles another biological entity, e.g., a protein. These problems could be avoided with the establishing of a formal naming convention for all biological entities. Despite efforts in this direction (1, 2), it is still far from being complete, commonly accepted, or utilized (3). Automated systems for word sense disambiguation try to overcome such problems by taking, for example, *POS* tags into consideration to identify nouns in text, which does not necessarily help in mapping the found text entities onto existing database entities.

2.5. Identification of Relationships Between Entities

The aforementioned tasks and their individual limitations influence the identification of relationships between entities. Automated systems still struggle with semantic ambiguity that is often encountered in the English language. A sentence read by a human reader can have several different meanings, depending on where the reader puts the stress within the sentence. Such sentences are generally difficult for computer software to analyze. It becomes even more difficult when relationships among entities span through several sentences. Even trained human curators with a sufficient biological background are not able to fulfill the task with a 100% accuracy.

3. Biomedical Ontologies `and Text Mining

Ontologies are foremost conceptual models. They try to establish a unifying representation and systematical order for entities, concepts, and relationships between them in a hierarchical manner for unambiguous and consistent sharing of knowledge over different domains. The Open Biomedical Ontologies (OBO, www.obofoundry.org) initiative is a collaborative effort to create guidelines for the development of biomedical ontologies. In addition, it gives an overview of biomedical ontologies currently under development. An example for a biomedical ontology is the well-known and studied Gene Ontology (2) (GO, www.geneontology.org). An example of an early TM system that focuses on the GO is GoPubMed (4) (www.gopubmed.org), which categorizes results of a PubMed (www.ncbi.nlm.gov/pubmed) search based on GO terms and concepts, thus letting a possible user navigate abstracts through these categories rather than through a list of, e.g., authors or publication titles.

One of the main criticisms of ontologies and their application in the biomedical domain is that an ontology will always be an

unfinished product that can be improved and that they often do not follow stringent standards (5, 6). In addition, the creation and the research of ontologies were not driven by the need of controlled vocabularies with hindsight to biomedical TM. The main obstacles for the application of ontologies within the scope of biomedical TM are the nonstandardized ontology language, the earlier mentioned inconsistency in naming convention for biological entities and concepts, and the incompleteness of ontologies (7). Nevertheless, research into ontologies and their application within biomedical TM is currently of a huge interest and more and more TM systems are developed that rely on ontologies.

4. Examples of General Text Mining Systems in the Biomedical Domain

TM systems in the biomedical field can be categorized broadly into two categories: (1) general TM systems and (2) specialized TM systems. The former systems do not focus on a specialized field of biology and are capable of retrieving either documents or co-occurrences to a variety of biological questions. The main aims of a researcher in utilizing such a tool are twofold. First, to filter out documents of interest to a particular search question (e.g., "Retrieve all documents that contain a particular gene or protein" "Retrieve all documents where *protein A* occurs with another protein in the same sentence" etc.) and, second to find literature evidence for testing a hypothesis (e.g., "Does evidence in the scientific literature exist that *protein A* and *protein B* interact?" "Does evidence in the literature exist that the *drug X* is related to the *disease* Υ?" etc.). Examples of such tools are manifold. For example, *iHOP* (8) uses gene/protein names as hyperlinks between sentences and abstracts of the PubMed database. The TM system is gene/protein centered, which means that the starting point for utilizing the system is a gene/protein name. Based on the name, *iHOP* finds sentences in the literature that contain the gene/protein name with other genes/proteins, thus creating an easily searchable network around the input gene/protein linked to the underlying literature. *EBIMed* (9) also, based on PubMed abstracts, has the goal to present information about UniProtKB/Swiss-Prot proteins, GO annotations, drugs, and species found in the abstracts in the form of an easy accessible table. The advantage of this TM system is that the input query into the system can be of arbitrary complexity. Standard search queries that would be utilized to query PubMed directly can be used. The resulting set of abstracts is then analyzed toward the former mentioned biological concepts and the results are presented in the form of a table, where each entry is linked to the

underlying sentence and abstract where the information was found, as well as to biological databases for more information on the biological entity/concept. This table highlights all co-occurrences of biological entities and concepts found in the corpus of abstracts that was retrieved by the search query. The table can be ordered in manifold ways to satisfy the user needs. Another similar approach is the TM system *AliBaba* (10) that also works on PubMed abstracts. Based on a protein or disease, it creates a network in the form of a graph, which visualizes interacting concepts such as cells, compounds, diseases, drugs, enzymes, proteins, genes, species, and tissues mined from the PubMed abstracts. The extracted information is again linked to the underlying text source, which is made readily accessible to provide the means for the user to confirm the accuracy of the extracted associations by hand.

All these systems provide in essence a method to query a literature corpus and retrieve abstracts/sentences that match a prespecified search query. The results are presented in different formats, while the focus is on different biological entities. It is a quick way to find fast information about a biological entity of interest. The extracted information is linked to the text source, and in most cases, to other biological databases, which enables a user to verify by hand how much confidence he gives to certain extracted information.

The biggest downside of these TM systems is that they only work on PubMed abstracts. The wealth of information buried in the full-text articles is thus not considered at all. Many of the problems and limitation in TM systems mentioned above play a role for disregarding the full-text articles in the first place. The main reason for considering only the abstracts is their easy accessibility, which in case of PubMed can be obtained free of charge in XML format for public institutions.

5. Measuring Success

After a system for information retrieval or extraction has been developed, its performance has to be measured. This can, for instance, be done by calculating sensitivity and specificity, while in the field of information retrieval, more often recall and precision are used. To make things even more confusing, sometimes also the positive, and, respectively negative predictive values are used to characterize a classifier.

The connection between all these terms is displayed in Fig. 1 for a binary classification problem. An example can in reality be true or false and the classifier can give a positive or a negative result, leading to four possible outcomes. In two cases

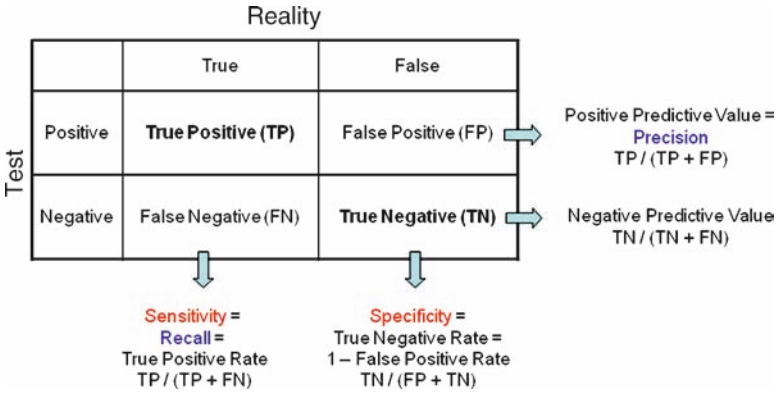


Fig. 1. Possible outcomes of a two-class classification problem. In the different scientific communities, different measures for classification success are used. Examples are the sensitivity and specificity system or the recall and precision system. For further details, see text.

(true positive and true negative), the prediction was correct, whereas in the other two cases (false positive and false negative), it was wrong. Sensitivity is now defined as the number of true positive predictions divided by all positive examples and specificity is the number of the true negative predictions divided by all negative examples. Thus, sensitivity measures how well a classifier recognizes true examples and specificity measures how well false examples are recognized. The term recall is actually identical to sensitivity, while precision is identical to the positive predictive value. Thus, precision is the fraction of positive predictions that are correct. A noteworthy feature of the sensitivity/specificity system is its independence of the ratio of true to false examples. Precision, in contrast, does vary with sample composition.

A specific pair of sensitivity and specificity values often depends on the discrimination threshold used by the classifier, and thus a single classifier can produce a whole range of sensitivity/specificity pairs. Consider, for example, the PSA level that is used in prostate cancer diagnostics. If the threshold level used for positive classification is low, the test will generate many positive predictions but with a high error rate, i.e., sensitivity is high, but specificity is low. If, however, a high threshold is used, there will be only few positive predictions but most of them will be correct. This means sensitivity is low, but specificity is high. To compare classifiers, it is, therefore, not sufficient to compare single sensitivity/specificity values, but instead the whole range of generated values has to be considered. A convenient way to do this is the use of a receiver operator curve (ROC), which displays true positive rate (sensitivity) as function of the false positive rate (1-specificity). The area under the receiver operator curve (AUC) ranges from 0 to 1 and is a popular measure for the quality of a classifier. For a more in-depth discussion on the use of ROCs see (11).

6. An Example of Text Mining for Systems Biology

As a specific example, the problem of finding scientific publications that contain kinetic parameters is now described. Biochemical reaction systems are usually modeled by a set of ordinary differential equations (ODEs) that describe the changes in the concentration of a biochemical species. The rate of a reaction is a function of the concentrations of the substrates, products, and of kinetic parameters that are part of the kinetic law. The irreversible Michaelis–Menten kinetics is a simple kinetic for the case that one substrate, with concentration c_S, is irreversibly converted into a product:

$$v = \frac{V_{\max} \cdot c_S}{K_M + c_S}$$

$V_{\max}$ denotes the maximal rate for high substrate concentrations and K_M is the half-saturation concentration (Michaelis–Menten constant). Other, more complicated, kinetic laws exist that depend on further parameters such as half-life and activation, respectively, inhibition constants. For the quantitative modeling of biochemical reaction networks, it is important to know the values of the various parameters and to know to which kinetic type they belong. Whereas most reaction networks are well described qualitatively, detailed quantitative values are missing or scattered in various scientific publications.

The aim was, therefore, to build a classifier that could separate few publications that contain values for kinetic parameters from those that do not (see also (12)). For this purpose, 4,582 randomly chosen full-text documents were downloaded from 12 different journals. From the full set, a keyword search generated 791 candidate articles. The keywords consisted of names and identifiers of constants (such as "Michaelis–Menten" or "Km") and words describing functions (such as "degradation," and "activation") or components ("enzyme"). Reading those 791 documents revealed that only 155 actually contained kinetic parameters, corresponding to a precision of 20% of this method. However, this first selection step was necessary, because it would have been a prohibitive amount of work to read all 4,582 articles.

6.1. Document Representation

The representation most often used for the application of certain machine learning techniques is the vector space model (VSM) (13). This model describes each document as a set of properties called features. This leads to a comparable representation of texts, regardless of their prior format, size, or structure (book, journal, article, and paragraph). It becomes irrelevant whether the information is presented in the Results or the Methods section of a

research article, or what the exact content is (e.g. differences in nomenclature usage or spelling variants). Another advantage is the suitability of such vector formats for machine learning techniques, which can easily gather hints on the importance and influence of a particular fact (a feature) or certain nonlinear combinations of those.

Representing documents using the VSM, a fixed vector of features observed in the entire document collection (a feature vector) is calculated. Next, for each single document, an instance of this feature vector is filled with values describing the relevance of each feature for this particular document. Some features or properties might be present (to some degree) in one document, but absent in others. A single document can contain a certain term, with a certain number of occurrences, or not. The corresponding coordinate in the document vector, an instance of the feature vector, is assigned a value reflecting this occurrence, that is, the term frequency (*tf*). After tokenization and stemming of the texts, a fixed feature vector can be extracted consisting of every word stem encountered. Instances of the feature vector are then filled with the corresponding occurrences of each term for this particular document, resulting in one document vector per publication. The underlying approach is called a bag-of-words, as all words are represented by their frequency only, regardless of co-occurrences, collocations, and context. Additionally, one might think of different weighting schemes to represent the significance of a term for describing a certain document. Most weighting schemes (14, 15) comprise a combination of a term's local weight (i.e. within the document) and its global weight (i.e. in the document collection). However, in this study, only *tf* was used to construct the feature vector. Processing of the complete corpus (791 documents) resulted in approximately 44,000 different features.

6.2. Feature Ranking and Dimensionality Reduction

The described way to represent documents leads to a very high dimensional feature vector. These extreme dimensionalities can negatively affect the classification performance. On the contrary, one can argue, that the more information is used to describe the documents, the better will be the classification model generated by the machine learning algorithms. It is, therefore, an important step to find an appropriate balance between these opposing effects. To pick the most relevant features of a document (or the whole document collection), different ideas were applied. In every language, there are a lot of so-called stop-words, common terms which do not provide any information toward discriminating documents, as they tend to appear with the same frequency in every kind of text (e.g. and, are, it, and with). These words can be removed, as well as very rare words, appearing in only a few (or a single) documents. A pruning of such words helps to reduce the dimensionality of the vector space.

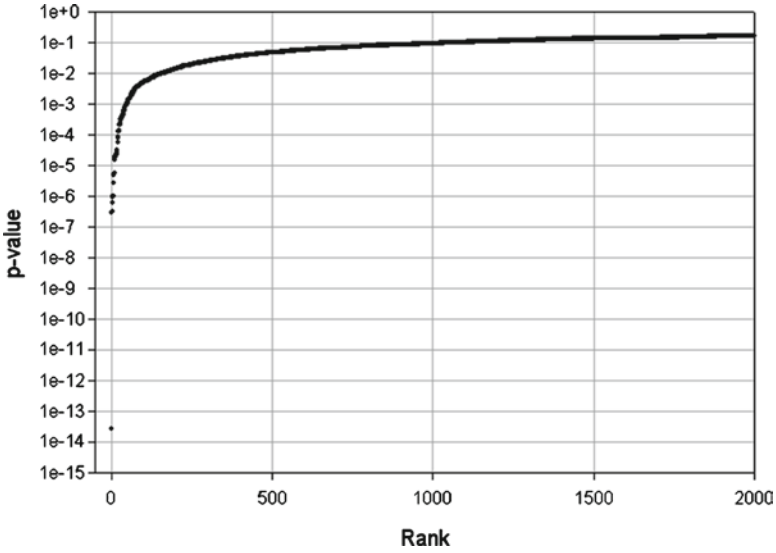


Fig. 2. Sorted results of the nonparametric Mann–Whitney test used to rank all words in the feature list obtained from the analysis of the document corpus (796 scientific papers). From the 44,000 features, only 532 have a *p*-value smaller than 0.05.

Furthermore, the remaining features can be ranked according to their importance by some appropriate statistical test and then only the most important terms are used for the classification algorithm. To calculate such a ranking, the non-parametric Mann–Whitney test was used, which does not rely on special assumptions about the data distribution (such as normality). The test calculates for each of the approximately 44,000 different features a *p*-value, indicating how important this feature is for separating the two classes. Figure 2 shows the *p*-values for the 2,000 most significant features. There are only relatively few features with small values, while the large majority of terms seems to be evenly distributed between the two classes of documents (resulting in large *p*-values). Although we perform multiple tests (namely, 44,000), corrections for multiple testing are not required since we are only interested in the relative ranking and not the absolute significance of each term.

6.3. Classification Performance and Feature Number

Several classification runs were performed to study the dependency between feature number and classification performance. For this purpose, only a certain number of top ranked (Mann–Whitney) features were included in the support vector used by a support vector machine (SVM) (16). Classification was performed with RBF (radial basis function) kernel and tenfold cross-validation to avoid over-fitting. Figure 3 shows the connection between the area under the receiver operator curve (AUC) and the number of used features. As can be seen, the AUC rapidly increases

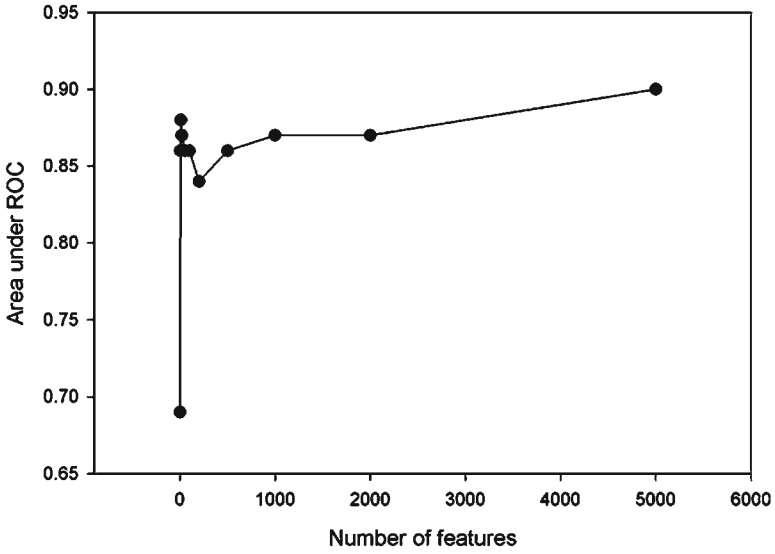


Fig. 3. Diagram showing the dependence between the area under the ROC curve (*AUC*) and the number of best ranked features used for classification. The features were ranked using a Mann–Whitney test (see Fig. 2).

with increasing feature number and then approaches a maximum at 5,000 features (which was the number of features given as input to the SVM). Thus, in this case, already a small number of top-ranked features are sufficient to give a good classification performance. Furthermore, the classification ability of the SVM does not degrade with feature number (it even seems to increase slightly). This confirms the well-known observation that the performance of SVMs is quite robust against a surplus of features.

6.4. Classification Performance with 5,000 Features

Finally, the classification performance is examined when using a feature vector with 5,000 features, which gave the best AUC value of the studied cases. Figure 4 shows the ROC curve for this situation with an AUC of 0.90.

Support vector machines can provide a probability estimate on how likely it is that an example belongs to one class or the other. By using different probabilities as threshold for the classification (normally 0.5 is used), different combinations of sensitivity (true positive rate) and specificity (1-false positive rate) can be obtained. All points on the surface of the ROC can be reached by an appropriate choice of the classification threshold.

Another way to visualize this connection is displayed in Fig. 5. The diagram shows directly how sensitivity and specificity vary with the used threshold. In general, there is a trade-off between sensitivity and specificity. However, depending on the problem, it might not be necessary to have high values for both measurements. In our case, sensitivity is not as important. Since a potentially

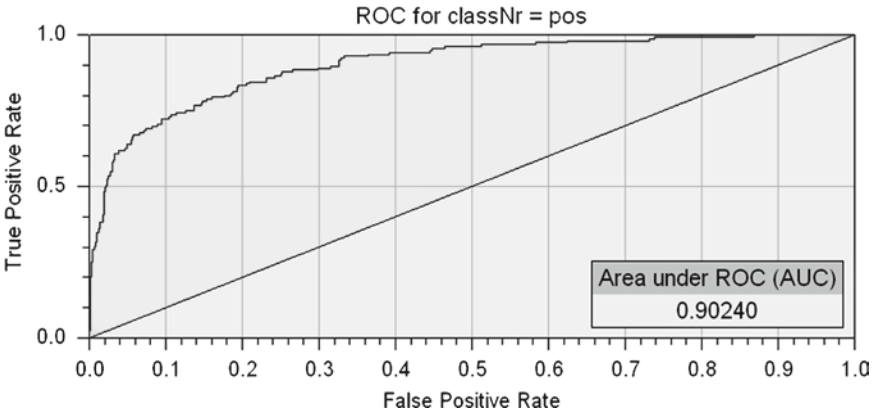


Fig. 4. Receiver operator characteristic (*ROC*) curve for a support vector machine classification using a feature vector with the 5,000 top-ranked features.

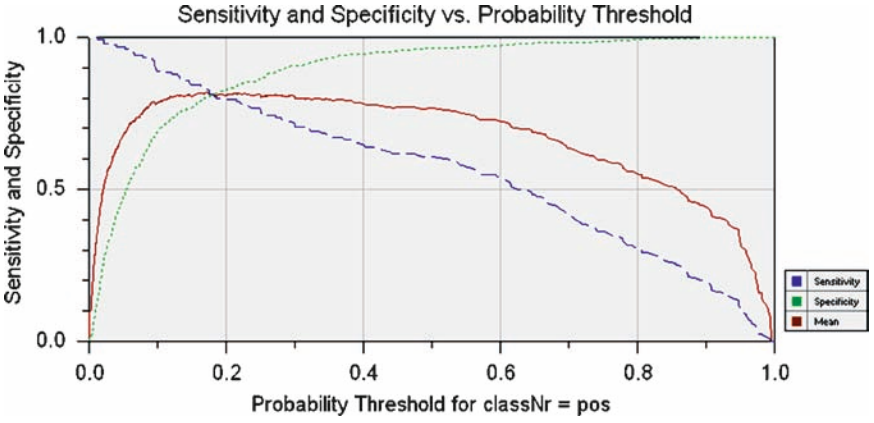


Fig. 5. Achieved sensitivity and specificity (and geometric mean of both) as a function of the used probability threshold. Support vector machines can calculate a probability value indicating how "sure" the classifier is that the example belongs to the predicted class. If different probability thresholds are used for classification, different combinations of sensitivity and specificity are obtained.

very large number of publications with kinetic parameters does exist in the literature, it is not so important if one is not found (false negative). But false positives are very costly, because those papers have to be inspected manually before the error is detected (labor costs). Therefore, a high specificity is desirable. That means a large threshold value will be chosen to obtain a high specificity.

7. Conclusions

Interest and research in biomedical TM has increased greatly over the last decade. Currently, information retrieval and extraction provide the means to support a variety of biomedical studies.

An example study of a TM approach set in the field of systems biology has been described. The aim was to train a machine learning classifier to distinguish relevant from irrelevant scientific publications. The relevance is defined by their content of kinetic parameters that are necessary for the in silico modeling of biological pathways. Several TM sub-tasks such as format conversion, POS tagging, stemming, feature representation with the help of the vector space model approach, and machine learning. have been discussed during the analysis. It could be shown that with the help of TM techniques it was possible to fulfill the task with an acceptable performance. However, several difficulties were encountered during the course of the study. The automatic conversion from PDF documents to plain ASCII text was imperfect. The used software was not able to resolve all words and symbols encountered in the PDF documents correctly. Future advances in conversion technology and optical character recognition (*OCR*) software will definitively improve PDF-based TM. Shifting the focus away from PDF documents toward full-text publications in HTML or XML format would solve this problem. An example of such a format is ePub, which has in 2007 been endorsed by the International Digital Publishing Forum (www.idpf.org) as a new standard for electronic publishing. Furthermore, other feature representation schema or machine learning algorithms might lead to improvements as well. However, even though the created system for the automatic classification of documents from a specialized biological domain is not perfect, it could be demonstrated that such a system can already now be of great value for scientists seeking kinetic information from text sources.

References

1. White J, Wain H, Bruford E, Povey S (1999) Promoting a standard nomenclature for genes and proteins. Nature 402(6760):347
2. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM et al (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet 25(1):25–29
3. Chen L, Liu H, Friedman C (2005) Gene name ambiguity of eukaryotic nomenclatures. Bioinformatics 21(2):248–256
4. Doms A, Schroeder M (2005) GoPubMed: exploring PubMed with the gene ontology. Nucleic Acids Res 33:W783–W786 (Web Server issue)
5. Soldatova LN, King RD (2005) Are the current ontologies in biology good ontologies? Nat Biotechnol 23(9):1095–1098
6. Smith B, Ashburner M, Rosse C, Bard J, Bug W, Ceusters W et al (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nat Biotechnol 25(11):1251–1255
7. Spasic I, Ananiadou S, McNaught J, Kumar A (2005) Text mining and ontologies in biomedicine: making sense of raw text. Brief Bioinform 6(3):239–251
8. Hoffmann R, Valencia A (2004) A gene network for navigating the literature. Nat Genet 36(7):664
9. Rebholz-Schuhmann D, Kirsch H, Arregui M, Gaudan S, Riethoven M, Stoehr P (2007) EBIMed-text crunching to gather facts for proteins from Medline. Bioinformatics 23(2):e237–e244

10. Plake C, Schiemann T, Pankalla M, Hakenberg J, Leser U (2006) AliBaba: PubMed as a graph. Bioinformatics 22(19):2444–2445
11. Fawcett T (2006) An introduction to ROC analysis. Pattern Recognit Lett 27(8):861–874
12. Hakenberg J, Schmeier S, Kowald A, Klipp E, Leser U (2004) Finding kinetic parameters using text mining. OMICS 8(2):131–152
13. Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620
14. Strasberg HR, Manning CD, Rindfleisch TC, Melmon KL (2000) What's related? Generalizing approaches to related articles in medicine. Proc AMIA Symp 838–842
15. Glenisson P, Antal P, Mathys J, Moreau Y, De Moor B (2003) Evaluation of the vector space representation in text-based gene clustering. Pac Symp Biocomput 391–402
16. Vapnik VN (1995) The nature of statistical learning theory. Springer, Berlin

Chapter 20

Identification of Alternatively Spliced Transcripts Using a Proteomic Informatics Approach

Rajasree Menon and Gilbert S. Omenn

Abstract

We present the protocol for the identification of alternatively spliced peptide sequences from tandem mass spectrometry datasets searched using X!Tandem against our modified ECgene resource with all potential translation products and then matched with the Michigan Peptide to Protein Integration (MPPI) scheme. This approach is suitable for human and mouse datasets. Application of the method is illustrated with a study of the Kras activation-Ink4/Arf deletion mouse model of human pancreatic ductal adenocarcinoma.

1. Introduction

By means of alternative splicing and posttranslational modifications, one gene can generate a variety of proteins. Alternative splice events that affect the protein coding region of the mRNA will give rise to proteins which differ in their sequence and activities. Alternative splicing within the noncoding regions of the RNA can result in changes in regulatory elements, such as translation enhancers or RNA stability domains, which may dramatically influence protein expression (1).

Alternative splicing has been associated with such diseases as growth hormone deficiency, Fraser syndrome, cystic fibrosis, spinal muscular atrophy, and myotonic dystrophy (2, 3). In cancers, there are examples of every kind of alternative splicing, including alternative individual splice sites, alternative exons, and alternative introns (4). A number of public alternative splice databases have recently become available, including ASD, HOLLYWOOD, and ASAP II. Each of these repositories contains transcript models that have been constructed from either expression data

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_20,

(ESTs and mRNA) or previous annotations of known proteins. The databases vary in their annotation methods and their overall size. One of the larger of these databases is the ECgene database developed by Kim et al. (5). Entries in the database are scored as high, medium, or low confidence reflecting the amount of amassed evidence in support of the existence of a particular alternatively spliced sequence. Evidence is collected from clustering of ESTs, mRNA sequences, and gene model predictions.

We have devised a proteomic informatics approach to identify known and novel alternative splice variants. Briefly, we search mass spectrometric data against a custom-built, nonredundant human or mouse database created with translation products using all three reading frames from cDNA sequences taken from ECgene and Ensembl databases (6). The peptide sequences identified are analyzed using Blast and Blat searches and integrated to distinct proteins.

2. Methods

2.1. Database of Translated Alternatively Spliced Sequences: The Modified ECgene Database

The target alternative splice variant protein database, the modified ECgene database, was constructed and can be updated by combining the latest Ensembl and ECgene databases for the mouse; the analogous combination generates a modified ECgene database for human studies. Taking alternative splicing events into specific consideration, ECgene combines genome-based EST clustering and the transcript assembly procedure to construct gene models that encompass all alternative splicing events (5). The reliability of each isoform is assessed from the nature of cluster members and from the minimum number of clones required to reconstruct all exons in the transcript.

cDNA sequences from the ECgene database and from the Ensembl database were obtained in FASTA format. Each sequence set was translated separately in three reading frames and the first instance of every protein sequence longer than 14 amino acids was recorded. The cDNA sequences are translated in three reading frames instead of the six frames which are used in translation of genomic double-stranded DNA; cDNAs are made from single-stranded mRNA sequences. Following the three-frame translation, the resulting sequences from each data source were combined and then filtered for redundancy. Preference was given to protein sequences originating from an Ensembl transcript. A collection of common protein contaminant sequences (http://www.thegpm.org/crap) was added to this set. Lastly, all sequences were reversed and appended to the set of forward sequences as an internal control for false identifications. This last step resulted in doubling the total number of entries in the modified ECgene database.

For examples, the mouse ECgene database (mm8, build 1) contains a total of 417,643 splice variants; the Ensembl version 40 database (each Ensembl release has an integer version number associated with it which is used to identify the correct versions of API, Web code and databases for that release) has 21,839 mouse genes with 28,110 transcripts, of which there are 10,922 alternative transcripts derived from 4,651 genes. The modified mouse ECgene database contains 10.4 million protein sequences. Similarly, the modified human ECgene database which contains Ensembl version 53 contains 14.2 million protein entries.

2.2. Searching Mass Spectral Data Against Alternative Splice Database

The mzXML files containing the spectral information were extracted from mass spectrometric RAW files using ReAdW.exe program (http://tools.proteomecenter.org). The mzXML files were then searched against the modified ECgene database using X!Tandem software (7).

2.3. Postsearch Analyses

Figure 1 summarizes the analytical work flow of the X!Tandem search results. In brief, the peptides are first integrated to a list of proteins using the Michigan Peptide to Protein Integration (MPPI) approach described below. Thereafter, all peptides identified from the integrated protein list are searched against the latest protein databases. This step is required due to frequent updates of protein databases. If a match occurs, the peptide is referred to as a known peptide; if not, it is considered as a novel peptide.

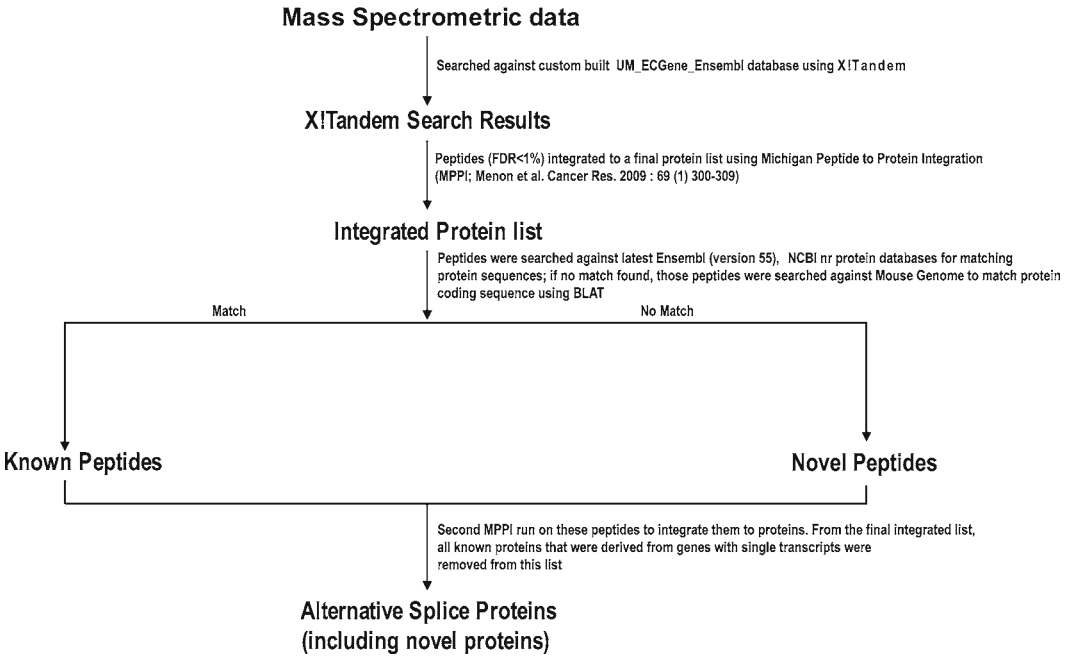


Fig. 1. The *Flow chart* displaying the analytical work flow of the X!Tandem search results for the identification of Alternative Splice Variant proteins.

The protein sequences from which these peptides are identified are aligned to the genome to determine the location of the peptides. Next, the known peptides with protein annotations from the latest databases and the novel peptides identified from modified ECgene variants undergo another round of MPPI. A threshold is applied to keep the FDR <1%. All known protein identifications that were derived from genes with multiple transcripts and the novel identifications from ECgene entries are retained, and the other proteins are removed. Hence, the final integrated protein list contains the known and novel alternative splice variants.

2.4. Michigan Peptide to Protein Integration

The peptide identifications from the X!Tandem searches were integrated to a final list of proteins using MPPI. Only the peptide identifications within the FDR < 1% limit were used in the MPPI analysis. The MMPI algorithm is as follows:

1. List all peptide matches that fall within an FDR <1% (based on X!Tandem expect value).
2. Order peptides by the number of spectra matching each peptide.
3. Select peptide with the largest number of matching spectra.
4. List all proteins containing this peptide, ranked by decreasing number of total distinct peptides identified, decreasing number of total spectra, increasing expect value, and then increasing protein length.
5. Select the highest ranking protein to be included in the final integrated protein list; if a tie, give preference to Ensembl protein over ECgene protein.
6. Remove all other peptides contained within this protein from the peptide list.
7. Repeat steps 3–6 until no peptides remain in the peptide list.

2.5. Sequence Analyses

The peptide identifications from the proteins after the first MPPI analysis were searched against the mouse genome using BLAT (8) and against latest Ensembl and NR databases using NCBI blastp (9). In the case of a novel peptide, the translated splice variant sequence is aligned against the mouse genome. Thus, the location of the peptide within the gene is determined. One can deduce the splice mechanism which has generated the novel peptide, including the deletion or switch of exons, intron retention, alternate splice site, and translation in an alternative reading frame. In rare cases, the ECgene variant sequence from which the novel peptide was identified matches to a conserved chromosomal region with no known genes; if so, the identification is retained only if the novel peptide is from multiple good quality spectra.

2.6. Validation of Novel Peptides

If total mRNA of the sample used in the mass spectrometric study is available, an independent validation of the novel splice variant peptides by reverse transcription polymerase chain reaction (RT-PCR) or quantitative RT-PCR can be performed. Specific primers can be designed to amplify precisely the novel peptide sequence using free online applications, including Primer3 (http://frodo.wi.mit.edu/primer3/). In a comparative study, for example, of tumor versus normal tissue specimens, the qRT-PCR enables assay of differential mRNA expression related to the novel peptide and comparison with the evidence of differential expression at the protein level.

2.7. Differential Expression of Alternative Splice Variants

In studies where samples under different conditions are analyzed, knowledge of the differential expression of the unique peptides by which an alternative splice variant protein is identified would be very useful. This information would indicate whether the particular variant might be functionally involved in the phenotype associated with the specimen. If the samples are labeled by heavy isotopes or molecular tags, proteomics tools such as LIBRA (10) or XPRESS (10, 11), which are embedded in Trans Proteomic Pipeline (TPP, Institute for Systems Biology, Seattle) (10, 11), can help determine the relative expression of the unique peptide. In addition, spectral counting is a label-free method to estimate protein quantification using peptide identification results from tandem mass spectrometry; no isotopic labeling is required to perform spectral counting.

2.8. Annotation of Novel Peptides

To characterize the novel peptides identified, online tools including InterProScan or Motif Scan can be used. InterProScan combines different protein signature recognition methods from the InterPro consortium member databases into one resource (12). Motif Scan scans a sequence against protein profile databases (http://myhits.isb-sib.ch/cgi-bin/motif_scan). The Berkeley Drosophila Genome Project Splice Site Prediction by Neural Network (http://www.fruitfly.org/seq_tools/splice.html) can be used for predicting alternative splice sites which may have generated these novel peptides.

The interactions of the alternative splice variants can be displayed by the Cytoscape MiMI plugin (13) using parent gene symbols as the input genes. Michigan Molecular Interactions (MiMI) gathers data from well-known protein interaction databases and deep merges the information. Utilizing an identity function, molecules that may have different identifiers but represent the same real-world object are merged. The Cytoscape MiMI plugin enables one to connect to the MiMI database and view the interactions (http://portal.ncibi.org/gateway/mimiplugin.html).

2.9. Alternative Splice Variant Analysis of a Pancreatic Tumor Dataset

To assess the potential of tumor-associated alternatively spliced gene products as a source of biomarkers in biological fluids, a large dataset of mass spectra derived from the plasma proteome of a mouse model of human pancreatic ductal adenocarcinoma was analyzed (14). MS/MS spectra were interrogated for novel splice isoforms using the nonredundant modified ECgene database described above. Among 1,278 distinct proteins, this integrated analysis identified 420 distinct splice isoforms, of which 92 did not match any previously annotated mouse protein sequence. Novel variants of muscle pyruvate kinase, malate dehydrogenase 1, glyceraldehyde-3-phosphate dehydrogenase, proteoglycan 4, minichromosome maintenance complex component 9, high mobility group box 2 and hepatocyte growth factor activator are of particular interest for pancreatic cancer. Isotopic labeling of cysteine-containing peptides from tumor-bearing mice and wild type controls enabled relative quantification of identified proteins having cysteine-labeled peptides. Statistically significant differential expression between tumor-bearing and control mice was noted for peptides from nine novel alternative splice variant proteins. We validated a subset of 7 of the 92 novel peptide sequences, all of which had multiple spectra, with appropriate primers for the corresponding mRNAs, using qRT-PCR of the tissues (14).

These results, in this mouse model for pancreatic cancers, show that novel and differentially expressed alternative splice isoforms are detectable in plasma. Such alternatively spliced protein variants may be clues to cancer progression and cancer biology and may become a source of candidate biomarkers.

3. Notes

The proteomic informatics approach presented here is intended to identify specific alternative splice variants, including novel proteins with differential expression under different conditions. Different organs, tissues, and biofluids may have different splicing propensities and different responses to external or internal stimuli, which will lead to interesting comparative analyses.

A major limitation of the protocol is the large size of the database. Our modified ECgene nonredundant translation product database for the human species contains 14.2 million records; the corresponding database for the mouse species contains 10.4 million records. Searching the mass spectral data against this database takes several days to complete. In addition, when the experimental protocol includes deep fractionation of the proteins (e.g., 162 fractions in the case of the pancreatic cancer-associated plasma sample described above), the computer search time is

multiplied to weeks. With very large datasets, sufficient memory is essential and may become apparent only when the server freezes and stops, requiring restart on the search. Dividing the database into subgroups and searching the mass spectral data against these databases in parallel can reduce the search time appreciably. An alternative is to run the forward and reverse sequences separately.

Another complication is the frequent updating of the Ensembl database. The novel peptide identifications have to be searched against the latest protein sequence database in order to be annotated as novel. Of course, as soon as a novel variant is published and made available to repository users, the novel variants became known splice variants for the next study. We have a series of studies of mouse and human cancers and cell lines in progress using this protocol.

4. Conclusions

The combined proteomic bioinformatics approach of the modified ECgene database and X!Tandem-MPPI search tools can identify specific known and novel splice variants in tissue and plasma specimens. The study of MS/MS data from the mouse plasma proteome of pancreatic tumor-bearing mice showed many specific known and novel alternative splice variants, some with differential expression between tumor-bearing and wild type mouse. Differentially-expressed splice variant proteins may influence many yet-to-be-identified cancer-related mechanisms. The data suggest that alternative splice variant proteins are a potentially rich source of candidate biomarkers for cancers and probably for other diseases, as well.

References

1. Bracco L, Kearsey J (2003) The relevance of alternative RNA splicing to pharmacogenomics. Trends Biotechnol 21:346–353
2. Faustino NA, Cooper TA (2003) Pre-mRNA splicing and human disease. Genes Dev 17:419–437
3. Garcia-Blanco MA, Baraniak AP, Lasda EL (2004) Alternative splicing in disease and therapy. Nat Biotechnol 22:535–546
4. Venables JP (2004) Aberrant and alternative splicing in cancer. Cancer Res 64:7647–7654
5. Kim N, Shin S, Lee S (2005) ECgene: genome-based EST clustering and gene modeling for alternative splicing. Genome Res 15:566–576
6. Fermin D, Allen B, Blackwell T, Menon R, Adamski M, Xu Y, Ulintz P, Omenn G, States D (2006) Novel gene and gene model detection using a whole genome open reading frame analysis in proteomics. Genome Biol 7:R35
7. Craig R, Beavis RC (2004) TANDEM: matching proteins with tandem mass spectra. Bioinformatics 20:1466–1467
8. Kent WJ (2002) BLAT – the BLAST-like alignment tool. Genome Res 12:656–664
9. McGinnis S, Madden TL (2004) BLAST: at the core of a powerful and diverse set of sequence analysis tools. Nucleic Acids Res 32:W20–W25

10. Pedrioli PG (2009) Trans-proteomic pipeline: a pipeline for proteomic analysis. Meth Mol Biol 604:213–238
11. Han DK, Eng J, Zhou H, Aebersold R (2001) Quantitative profiling of differentiation-induced microsomal proteins using isotope-coded affinity tags and mass spectrometry. Nat Biotechnol 19:946–951
12. Quevillon E, Silventoinen V, Pillai S, Harte N, Mulder N, Apweiler R, Lopez R (2005) InterProScan: protein domains identifier. Nucleic Acids Res 33:W116–W120
13. Gao J, Ade AS, Tarcea VG et al (2009) Integrating and annotating the interactome using the MiMI plugin for Cytoscape. Bioinformatics 25:137–138
14. Menon R, Zhang Q, Zhang Y, Fermin D, Bardeesy N, DePinho RA, Lu C, Hanash SM, Omenn GS, States DJ (2009) Identification of novel alternative splice isoforms of circulating proteins in a mouse model of human pancreatic cancer. Cancer Res 69:300–309

Chapter 21

Distributions of Ion Series in ETD and CID Spectra: Making a Comparison

Sarah R. Hart, King Wai Lau, Simon J. Gaskell, and Simon J. Hubbard

Abstract

Databases which capture proteomic data for subsequent interrogation can be extremely useful for our understanding of peptide ion behaviour in the mass spectrometer, leading to novel hypotheses and mechanistic understanding of the underlying mechanisms determining peptide fragmentation behaviour. These, in turn, can be used to improve database searching algorithms for use in automated and unbiased interpretation of peptide product ion spectra.

Here, we examine a previously published dataset using our established methods, in order to discover differences in the observation of product ions of different types, following ion activation and unimolecular dissociation either by collisional dissociation or the ion/ion reaction, electron transfer dissociation. Using a target-decoy database searching strategy, a large data set of precursor ions, were confidently predicted as peptide sequence matches (PSMs) at either a 1% or 5% peptide false discovery rate, as reported in our previous study. Using these high quality PSMs, we have conducted a more detailed and novel analysis of the global trends in observed product ions present/absent in these spectra, examining both CID and ETD data. We uncovered underlying trends for an increased propensity for the observation of higher members of the ion series in ETD product ion spectra in comparison to their CID counterparts. Such data-mining efforts will prove useful in the generation of new database searching algorithms which are well suited to the analysis of ETD product ion spectra.

1. Introduction

Tandem mass spectrometry has long been used in the analysis of unknown compounds, to enable rapid elucidation of structural information, exploiting spectral libraries and our understanding of natural isotopic abundances. More recently, collisional dissociation has been routinely applied in the determination of peptide primary structure. The use of tandem mass spectrometry in polypeptide analysis relies on the reproducible and predictable behaviour of peptides following gas-phase collision and subsequent

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_21,

unimolecular dissociation (1). Peptide bond fragmentation is the primary dissociation pathway, and the observed fragments, termed b ions if derived from the N-terminus, and y ions if C-terminal in origin, are detected, and their masses submitted for unbiased database searching (2, 3). While global fragmentation behaviour of peptides is relatively predictable, and mechanistic hypotheses explaining sequence-specific fragmentation patterns have been derived, in part from statistical analyses of large proteomic datasets (4), much remains to be understood about the fragmentation behaviour of peptides. Specifically, the relative abundance of fragment ions is still poorly predictable, and the presence of additional non-canonical fragmentation products following cyclisation and similar processes remain subjects of active research (5, 6).

More recent arrivals within the tandem mass spectrometry field have included the use of ion/electron and ion/ion reactions, which also enable structural characterisation of polypeptides in vacuo. Following pioneering work by McLuckey's group in gas-phase ion/ion reactions (7), and that of McLafferty, Zubarev, and colleagues in electron capture dissociation (8), Syka and colleagues described the use of cationic species which could act as electron donors, effecting electron capture-type dissociation within simple ion traps (9). This method, termed electron transfer dissociation (ETD), overcomes some of the difficulties of collisional dissociation, in terms of the ability to observe peptide sequence-specific product ions from peptides bearing labile post-translational modifications, and from larger, highly-charged polypeptides bearing many internal degrees of freedom (10, 11). While strong evidence exists to indicate a highly similar mechanism for ETD with that of electron capture dissociation (12), little work has thus far been performed to investigate the occurrence of minor pathways, and dissection of the influence of ion heating effects from electron transfer-induced events is in its infancy (13).

Similarly, to date, little effort has been made to characterise or investigate the global differences between CID spectra and their ETD counterparts. This has particular significance in proteomics, where database search algorithms, such as Sequest (14) and Mascot (15), are used to make qualitative decisions regarding the assignment of product ion spectra to putative sequence counterparts. Principally, these algorithms have been developed using sequence-specific rules primarily derived from collisional dissociation data (16). While several published studies perform comparative assessment of the two dissociation techniques, little or no comparison of the global patterns of observed peptide dissociation properties as characterised by a large data set of peptide sequence matches have hitherto been performed (17).

We have previously interrogated peptide product ion data repositories (18, 19) to examine the occurrence of instrument-specific

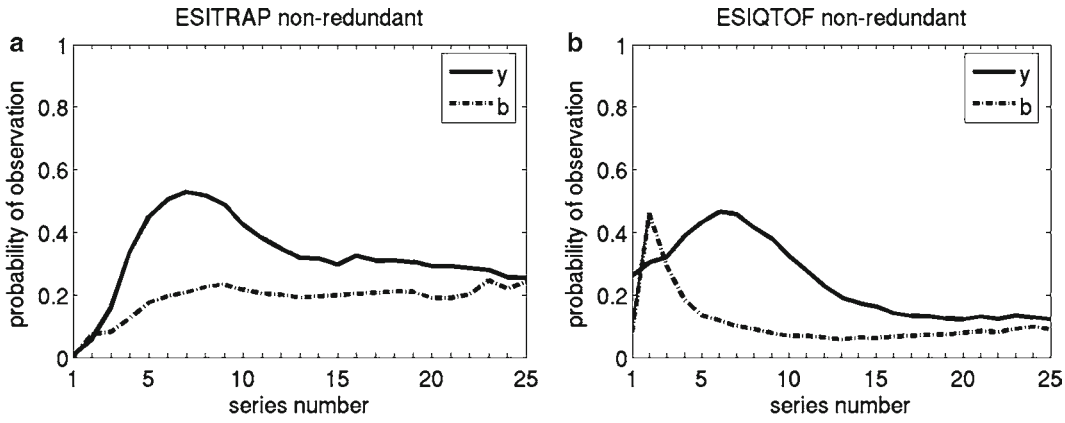


Fig. 1. Comparison of global patterns in CID data generated using quadrupole ion trap (resonant excitation CID) vs. quadrupole time-of-flight (transmission CID). Adapted from Ref. (20) with permission.

observations in collisional dissociation spectra (20). Our previous work focused on the investigation of patterns in the relative stability of b- and y-type product ions in large, non-redundant datasets, selected to avoid any bias from over-representation of common peptide sequences identified multiple times in different experiments. We revealed disparities between their relative frequencies, and identified global patterns in dissociation trends within CID datasets generated using CID via tandem-in-space, performed using hybrid QqTof instruments vs. the same species subjected to tandem-in-time, resonant excitation CID upon quadrupole ion traps (see Fig. 1) (20).

We have also recently published our findings within a large-scale analysis of an organelle proteome using ETD in addition to collisional dissociation (21). This analysis included a brief qualitative investigation of differences in the nature of precursor ions which were best suited to each dissociation strategy. We now apply a similar interrogation method to that in our tandem-in-space/tandem-in-time study to investigate our ETD/CID proteomic dataset of high-quality peptide sequence matches. We have used this approach to examine underlying patterns and differences between data generated for the *same* precursor ion population on the *same* mass spectrometer, using *different* ion activation methods. To this end, we re-examined data from our previous study (21) using custom-written perl scripts, to examine the relative frequency of observation of N-terminus-derived (b-type for CID and c-type for ETD) and C-terminal (y and z-type respectively) product ions. Data interrogation tools such as these can be applied to the investigation of multiple hypotheses in datasets. Their utility is not limited to the analysis of unmodified peptides; indeed investigation of the influence of site-specific post-translational modifications or derivatisation chemistries upon peptide dissociation are obvious potential applications of our method.

Examination of proteomic datasets in this manner enables the observation of patterns in terms of fragmentation, and hence will aid in the derivation of new information regarding the mechanisms underlying peptide fragmentation and improvements in the accuracy and precision of database searching methods.

2. Materials and Methods

2.1. Preparation of Proteolytic Digests for Mass Spectrometry

1. Extracted flagellar pellets from *Trypanosoma brucei* (~12 μg) were dissolved in 500 mM triethylammonium bicarbonate containing 0.1% sodium dodecyl sulphate (w/v) (both Sigma, Poole, Dorset), and subjected to reduction and alkylation using 1 mM triscarboxyethyl phosphine (Sigma) and 1 mM methyl methanethiosulfonate (Pierce, Cramlington, Northumberland), respectively.
2. Digestion was performed overnight at room temperature using endoproteinase LysC and trypsin at enzyme: substrate ratios of 1:100 (w/w) (Sigma).
3. Peptides were separated by strong cation exchange on a 2.1 mm id, 20 cm polysulfoethyl A column (Hichrom, Theale, Berkshire), using a linear gradient from 0 to 200 mM potassium chloride in 10 mM monobasic potassium phosphate, 20% acetonitrile (Sigma).

2.2. Tandem Mass Spectrometry

Tandem mass spectrometric data of many types can be examined using an approach, such as the one described here, indeed we have utilised this general approach previously in the comparison of transmission and resonant excitation mode CID data (see Fig. 1). This figure is reproduced here for later comparative purposes with ETD-based data. Further, this method is not subject to bias in terms of the search algorithm used in the assignment of product ion spectra, exemplified by the use of Mascot in our previous investigations, and Sequest in the examples provided here, which both produce the same overall patterns. The examination of widespread phenomena in tandem mass spectrometric datasets has importance in the development of widespread mechanistic understanding of peptide fragmentation in vacuo, and in the improvement of interpretation and search algorithms for unbiased assignment of product ion spectra.

1. SCX fractions were subjected to LC-MS/MS with switching between CID and ETD on a linear ion trap instrument equipped with an external CI source for fluoranthene anion generation (LTQ, Thermo Fisher Scientific, San Jose, CA). Xcalibur was used to collect mass spectrometric data (Thermo). MS/MS experiments were performed according to the methods

previously described (21). All selected precursors were subjected to both CID and ETD to enable direct comparison of product ion spectra (see Fig. 2).

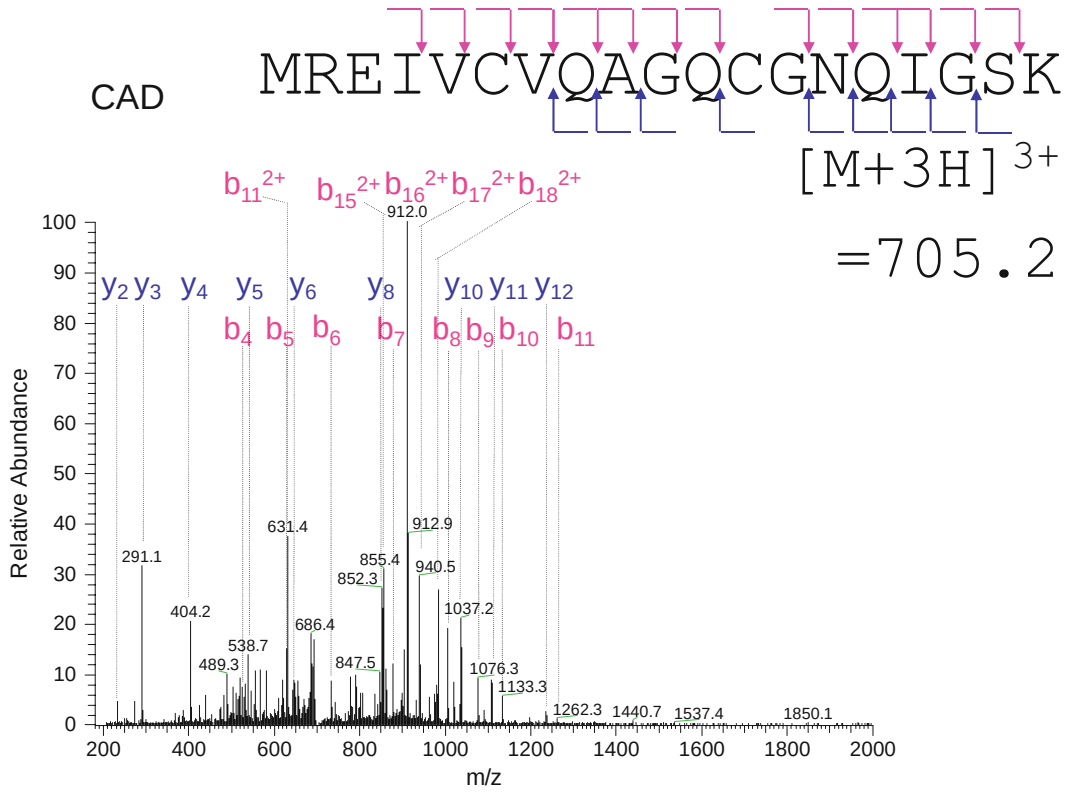
2. RP-HPLC gradients were run from 0 to 40% acetonitrile, 0.1% formic acid over 40 min, with the eluent being directly connected to the nanoelectrospray source of the mass spectrometer.

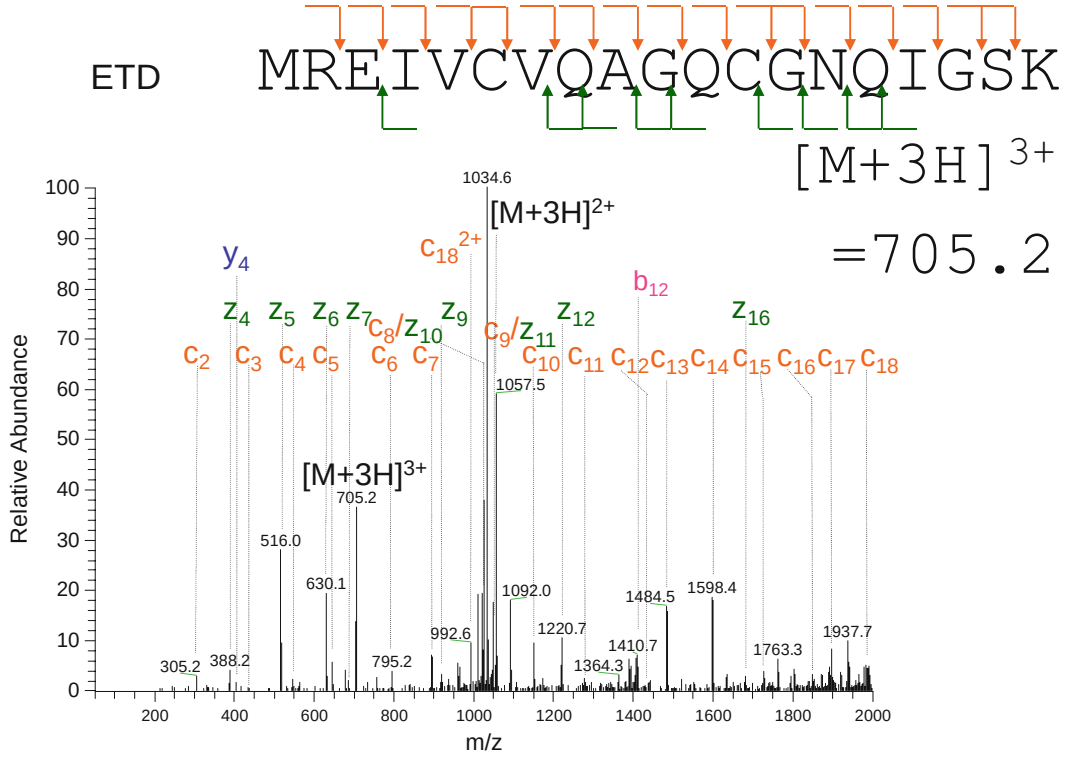
2.3. Data Processing and Database Searching

1. Product ion spectral data were extracted from raw data files using TurboSequest and separated into text files within two folders, according to fragmentation method, using a batch script.
2. ETD and CID product ion spectra represented as .dta text files were independently subjected to database searching using Sequest (14). Search parameters: ≤2 missed cleavages, fully enzymatic products (trypsin), fixed cysteine modification by methyl methanethiosulfonate (45.99 Da), variable modification of methionine residues by oxidation (15.99 Da), peptide mass tolerances 2 Da, product ion tolerance 1 Da. For CID data, allowed ion types were b and y, while for ETD data, ion types were c and z. Approximately 22,000 product ion spectra per dissociation method were collected, filtered, and subjected to database search against a concatenated target-decoy database of proteins translated from the *Trypanosoma brucei* genome, containing 9,210 forward (target) and 9,210 reverse (decoy) entries.

2.4. Estimation of False Discovery Rates and Generation of Product Ion Frequency Diagrams

1. Data generated using tandem mass spectrometry, and peak extraction and database searching were performed to create a list of peptide candidates associated with product ion spectra matching to both forward and reversed database entries. Reverse entries within the concatenated database were tagged by appending "_r" after their accession number, thus facilitating ready false discovery rate (FDR) estimation.
2. Product ion identification data were filtered using in-house perl scripts (see Note 1), which enable user-defined thresholds to be set in terms of peptide FDR and number of non-redundant peptides per candidate protein to be imposed for inclusion (20). To maximise the number of peptides reported while maintaining a given FDR, the correlation between experimental and matched theoretical spectra (XCorr) and differential between top and next match (ΔCn) are varied. This takes into account the strong qualitative differences between ETD and CID data (see Fig. 2, (21)), while maintaining a similar quality of match. FDR is defined as previously reported, adapted from (22), where:


CAD
MREIVCVQAGQCGNQIGSK
[M+3H]3+
=705.2
b11 2+ b15 2+ b16 2+ b17 2+ b18 2+
912.0
y2 y3 y4 y5 y6 y8 y10 y11 y12
b4 b5 b6 b7 b8 b9 b10 b11
631.4
912.9
291.1
855.4
852.3
940.5
1037.2
404.2
686.4
538.7
489.3
847.5
1076.3
1133.3
1262.3
1440.7
1537.4
1850.1
Relative Abundance
m/z


ETD
MREIVCVQAGQCGNQIGSK
[M+3H]3+
=705.2
1034.6
[M+3H]2+
c18 2+
y4
b12
z16
z4 z5 z6 z7 c8/z10 z9 c9/z11 z12
c2 c3 c4 c5 c6 c7 c10 c11 c12 c13 c14 c15 c16 c17 c18
1057.5
[M+3H]3+
705.2
516.0
630.1
1092.0
1484.5
1598.4
992.6
1220.7
1410.7
1937.7
305.2
388.2
795.2
1364.3
1763.3
Relative Abundance
m/z

$$\text{FDR} = \frac{2 \times \text{False Positives}}{(\text{True Positives} + \text{False Positives})}$$

Defining True Positives as all forward database PSMs exceeding the acceptance criteria in XCorr and ΔCn, and False Positives similarly but with respect to the reverse/decoy database. Typical FDR values reported within the general literature are between 0.5 and 5%.

3. Data were compiled using our in-house perl program, which supports the setting of a user-defined FDR. In this case, we selected a FDR of 0.01 although identical trends were observed with FDR=0.05. Our FDR was based, as before, on a combination of Xcorr and ΔCn scores which yielded the maximal number of PSMs for a given FDR with a minimum threshold set for each parameter (20).
4. Data from the candidate peptide sequence identifications passing the FDR criteria were processed to assign product ion types to individual peaks, based on the Sequest assignment. In this case, no further refinement from the Sequest ion assignments was undertaken although it is possible that a very small number of peaks might be misassigned (see Note 2). Output files plotted on similar axes may be compared manually to identify global differences in patterns of observation, for instance, the difference in observation of high members of c/z ion series from highly charged peptide precursors (see Figs. 3 and 4). Filtering of specific classes of peptides (e.g. cysteine-containing peptides or peptides bearing post-translational modifications, where included in database searching strategies) could be performed at this stage to examine differences between global patterns of observation and those for specific classes of analyte.

3. Discussion of Results

Electron capture and transfer dissociation methods are renowned for generating significantly higher quality product ion spectra from large, multiply charged precursor ion signals (8, 9, 11). This

Fig. 2. Linear ion trap CID and ETD product ion spectra example. A triply charged precursor ion at *m/z* 705.2 was selected and subjected to CID (*upper panel*) and ETD (*lower panel*). Observed products from primary fragmentation pathways are marked on the sequence (b and y products for CID, c and z for ETD), and peaks are annotated upon the spectrum with their series name and number (number of residues from terminus). Reprinted from Ref. (21) licence 2284810299358, with permission from Elsevier.

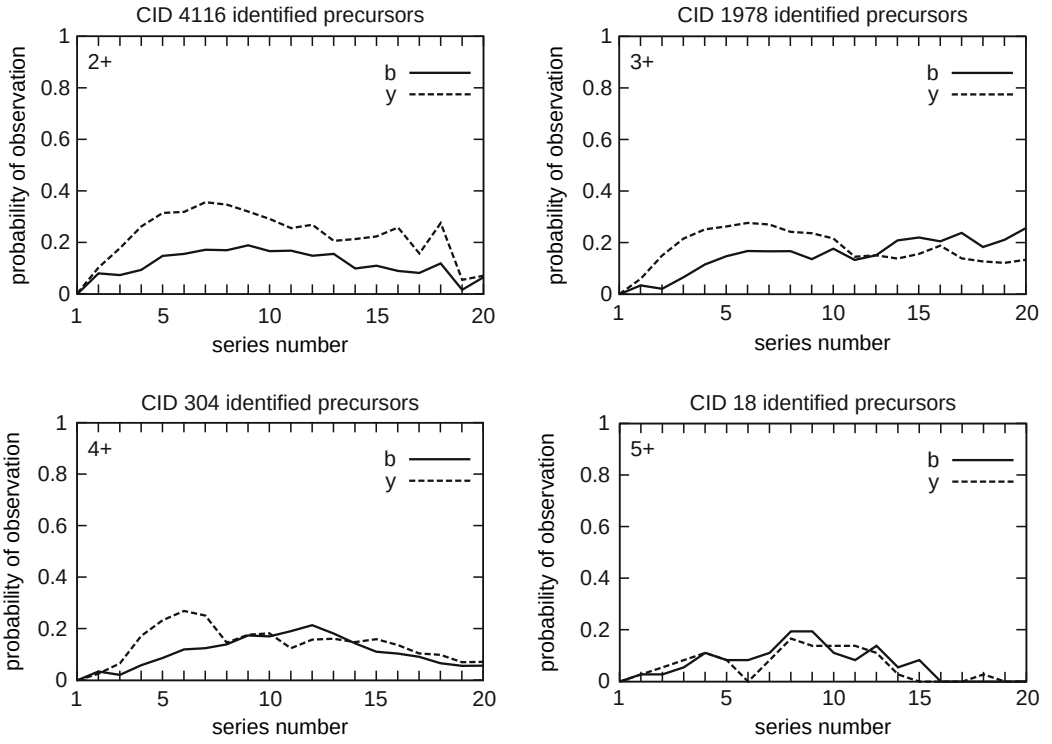


Fig. 3. Patterns of ion observation from CID product ion spectra. Product ion spectral identification data, collected over the *m/z* range between the automated cutoff for the LTQ and 2,000, roughly representing ion series members 1/2–20, are filtered according to a 1% peptide FDR, and matched product ions extracted and plotted using custom perl scripts. The observation frequency of a particular member of a given ion series (b, y,) is plotted, calculated for both singly and doubly charged ions, upto and including the 20th member of an ion series. Data for ion series beyond this are not shown as the frequency drops off and becomes noisy, in part due to the decreased data in our dataset and in part owing to the cutoff used on our LTQ instrument (<2,000).

trend is also apparent within our flagellar proteome data, as indicated by the number of confidently identified precursor ions of each charge state observed for the different dissociation methods (see Figs. 3 and 4). The number of confidently identified peptide precursors for the same precursor ion pool shows enormous variance between the CID data, where the vast majority of identified precursors are doubly protonated, vs. ETD, where triply- and quadruply protonated precursors predominate.

The extended nature of sequence coverage for ETD spectra has previously been identified as a specific advantage of the technique, as has the stochastic, "even" nature of bond cleavage and the related lack of specific sites showing high propensity for N-Cα bond cleavage (9, 12). This means that for the longer, larger peptides which form prime substrates for ETD analyses, the likelihood of generating extended sequence-related product ion series is high, increasing the probability of making a confident and correct sequence assignment using de novo assignment or database

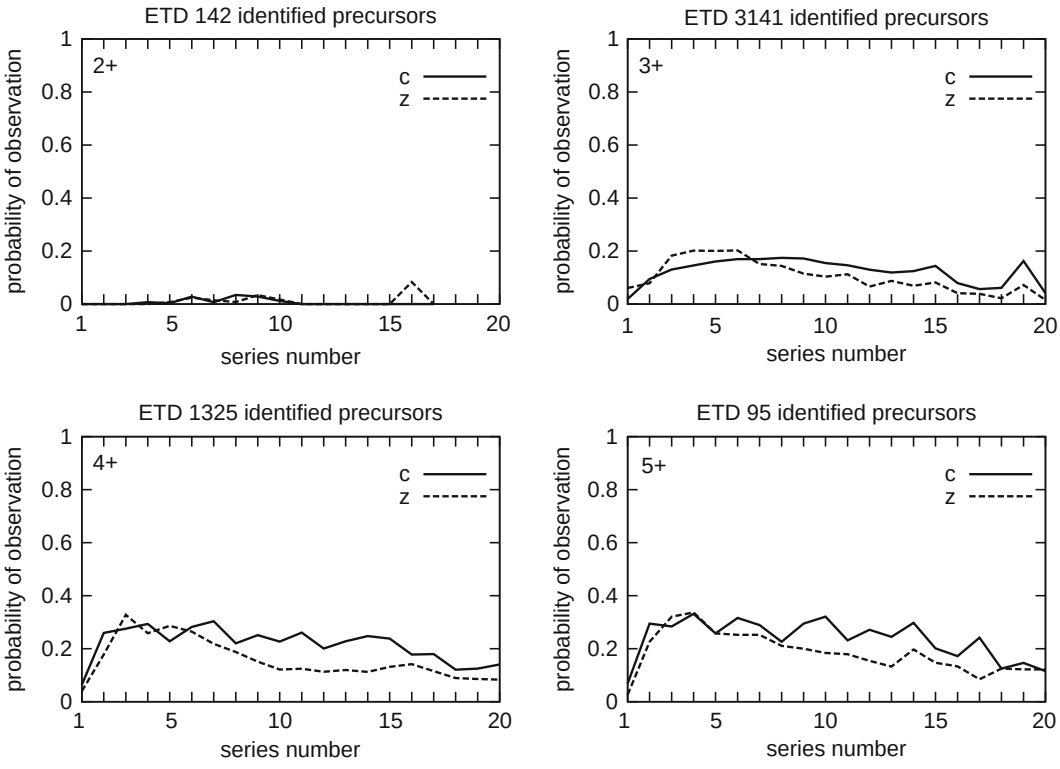


Fig. 4. Patterns of ion observation from ETD product ion spectra. Product ion spectral identification data, collected over the *m*/*z* range between 50 and 2,000, roughly representing ion series members 1–20, are filtered according to a 1% peptide FDR, and matched product ions extracted and plotted using custom perl scripts. The observation frequency of a particular member of a given ion series (c, z) is plotted.

matching software tools. Our ETD data show extended sequence coverage for both c- and z-series products over a range of charge states, with roughly equivalent frequency of observation for most c- and z-series products, while the CID data indicate maximal frequency of product ion observation for ~b2–9 and ~y4–8.

Looking at the observation frequency data in more detail, the propensity for N- or C-terminal retention of charge (and hence observation by MS/MS), influenced by relative stability of the product ions thus formed (see our previous study (20) for more detail) shows some differences. ETD product ion spectra show reduced bias towards C-terminal CID data, where there is a strong bias in favour of the formation of C-terminal y-type ions. This disparity may partially reflect the inherent charge state distribution of the precursor ions identified by each method, as the overall trends for CID data for triply- and higher-order-protonated precursors show greater similarity to ETD data than the CID of doubly charged precursors. In fact, a mild bias towards N-terminal c-type ions is apparent in ETD data for all charge states $z > 2$,

presumably resulting from the presence of additional protonated sites within the N-terminal region which stabilise charge, and hence, product ions within the mass spectrometer.

4. Notes

1. Product ion spectra in standard formats (e.g. .dta or .mgf standard output formats) are readily amenable to this type of interrogation.
2. We have used the native Sequest and Mascot assignments for peaks in our work, as well as assigning b,y and c,z ions ourselves. We find a very strong agreement in general between the various methods with over 90% agreement between any two. For the purposes of the data presented here, we have retained the native Sequest assignments for simplicity.

References

1. Sadygov RG, Cociorva D, Yates JR III (2004) Large-scale database searching using tandem mass spectra: looking up the answer in the back of the book. Nat Methods 1:195–202
2. Biemann K (1990) Nomenclature for peptide fragment ions (positive-ions). Meth Enzymol 193:886–887
3. Roepstorff P, Fohlman J (1984) Proposal for a common nomenclature for sequence ions in mass-spectra of peptides. Biomed Mass Spectrom 11:601
4. Huang YY, Triscari JM, Tseng GC, Pasa-Tolic L, Lipton MS, Smith RD et al (2005) Statistical characterization of the charge state and residue dependence of low-energy CID peptide dissociation patterns. Anal Chem 77:5800–5813
5. Harrison AG (2008) Peptide sequence scrambling through cyclization of b(5) ions. J Am Soc Mass Spectrom 19:1776–1780
6. Harrison AG (2009) To b or not to b: the ongoing saga of peptide b ions. Mass Spectrom Rev 28:640–654
7. McLuckey SA, Huang TY (2009) Ion/Ion reactions: new chemistry for analytical MS. Anal Chem 81(21):8669–8676
8. Zubarev RA, Kelleher NL, McLafferty FW (1998) Electron capture dissociation of multiply charged protein cations. A nonergodic process. J Am Chem Soc 120:3265–3266
9. Syka JE, Coon JJ, Schroeder MJ, Shabanowitz J, Hunt DF (2004) Peptide and protein sequence analysis by electron transfer dissociation mass spectrometry. Proc Natl Acad Sci USA 101:9528–9533
10. Bunger MK, Cargile BJ, Ngunjiri A, Bundy JL, Stephenson JL Jr (2008) Automated proteomics of E. coli via top-down electron-transfer dissociation mass spectrometry. Anal Chem 80:1459–1467
11. Mikesh LM, Ueberheide B, Chi A, Coon JJ, Syka JE, Shabanowitz J et al (2006) The utility of ETD mass spectrometry in proteomic analysis. Biochim Biophys Acta 1764:1811–1822
12. Ben Hamidane H, Chiappe D, Hartmer R, Vorobyev A, Moniatte M, Tsybin YO (2009) Electron capture and transfer dissociation: peptide structure analysis at different ion internal energy levels. J Am Soc Mass Spectrom 20:567–575
13. Rand KD, Zehl M, Jensen ON, Jorgensen TJ (2009) Protein hydrogen exchange measured at single-residue resolution by electron transfer dissociation mass spectrometry. Anal Chem 81:5577–5584
14. Lundgren DH, Han DK, Eng JK (2005) Protein identification using TurboSEQUEST. Curr Protoc Bioinformatics. Chapter 13, Unit 13.3
15. Perkins DN, Pappin DJ, Creasy DM, Cottrell JS (1999) Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis 20:3551–3567

16. Steen H, Mann M (2004) The abc's (and xyz's) of peptide sequencing. Nat Rev Mol Cell Biol 5:699–711
17. Vorobyev A, Hamidane HB, Tsybin YO (2009) Electron capture dissociation product ion abundances at the X amino acid in RAAAA-X-AAAAK peptides correlate with amino acid polarity and radical stability. J Am Soc Mass Spectrom 20(12):2273–2283
18. McLaughlin T, Siepen JA, Selley J, Lynch JA, Lau KW, Yin HJ et al (2006) PepSeeker: a database of proteome peptide identifications for investigating fragmentation patterns. Nucleic Acids Res 34:D649–D654
19. Siepen JA, Swainston N, Jones AR, Hart SR, Hermjakob H, Jones P et al (2007) An informatic pipeline for the data capture and submission of quantitative proteomic data using iTRAQ. Proteome Sci 5:4
20. Lau KW, Hart SR, Lynch JA, Wong SC, Hubbard SJ, Gaskell SJ (2009) Observations on the detection of b- and y-type ions in the collisionally activated decomposition spectra of protonated peptides. Rapid Commun Mass Spectrom 23:1508–1514
21. Hart SR, Lau KW, Hao Z, Broadhead R, Portman N, Huhmer A et al (2009) Analysis of the trypanosome flagellar proteome using a combined electron transfer/collisionally activated dissociation strategy. J Am Soc Mass Spectrom 20:167–175
22. Peng J, Elias JE, Thoreen CC, Licklider LJ, Gygi SP (2003) Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (LC/LC-MS/MS) for large-scale protein analysis: the yeast proteome. J Proteome Res 2:43–50

Part V

Tools

Chapter 22

Evaluation of Peak-Picking Algorithms for Protein Mass Spectrometry

Chris Bauer, Rainer Cramer, and Johannes Schuchhardt

Abstract

Peak picking is an early key step in MS data analysis. We compare three commonly used approaches to peak picking and discuss their merits by means of statistical analysis. Methods investigated encompass signal-to-noise ratio, continuous wavelet transform, and a correlation-based approach using a Gaussian template.

Functionality of the three methods is illustrated and discussed in a practical context using a mass spectral data set created with MALDI-TOF technology. Sensitivity and specificity are investigated using a manually defined reference set of peaks. As an additional criterion, the robustness of the three methods is assessed by a perturbation analysis and illustrated using ROC curves.

1. Introduction

Peak picking is an early key step in mass spectrometry (MS)-based proteomics and is crucial for data analysis. It goes hand in hand with smoothing, baseline correction, and peak alignment within a general workflow of preprocessing steps that allows for subsequent statistical data analysis and biological interpretation (see Note 1). Preprocessing of MS data aims at transforming a big amount of raw spectral data (usually >30K data points) into a much smaller, statistically manageable set of peaks (see Note 3). Subsequent data analysis will typically aim at biomarker discovery or sample classification. Comprising tens of thousands of data points in each spectrum, MS data are inherently noisy. The main sources of noise are chemical in nature, such as interference from matrix material and sample contamination, or electrical, which is dependent on the analytical set-up employed (1). As a result, various algorithms differing in principle, implementation, and performance have been proposed to address these problems (see Note 4).

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_22,

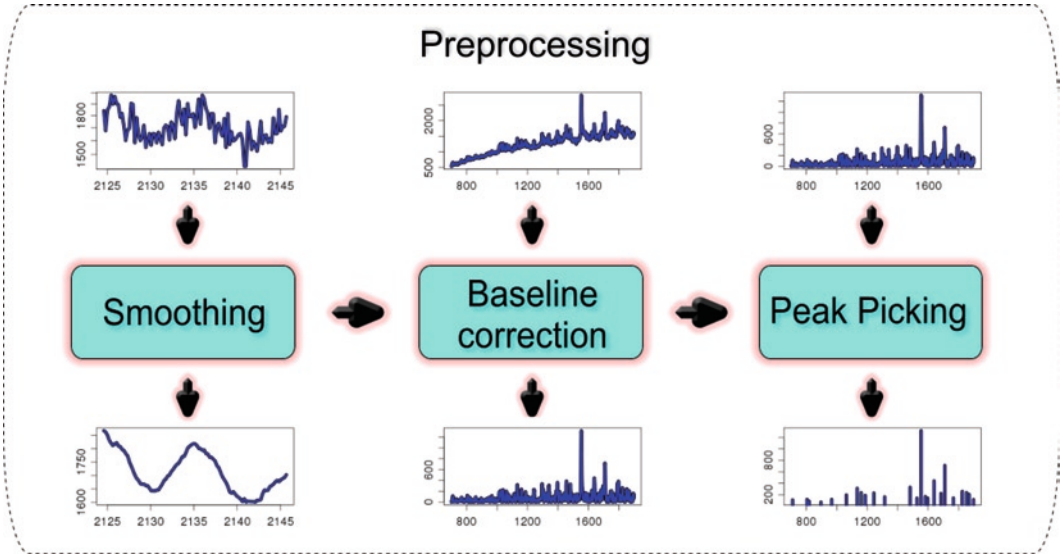


Fig. 1. The preprocessing workflow is typically composed of the steps: smoothing, baseline correction, and peak picking. In the course of preprocessing, a raw spectrum is transformed into a peak list suitable for further statistical analysis.

A typical preprocessing workflow comprises the following three steps (see Note 2): (See Fig. 1 for a schematic illustration and exemplary visualization of each step)

- *Data smoothing*: Smoothing mainly aims at removing high-frequency noise. Besides traditional signal processing techniques such as Savitzky–Golay filter (2), Mean/Median filter, or Gaussian filters, wavelet-based techniques are also employed for data smoothing (1, 3)
- *Baseline correction*: Baseline correction intends to remove low-frequency noise and thus eliminates the correlation of nearby features. Typically, methods such as Top Hat filter (4), Loess derivative filters (5), or linear splines are applied to estimate the baseline.
- *Peak picking*: The number of proposed methods for peak detection is immense. Most common algorithms make use of signal-to-noise ratio (SNR), continuous wavelet transform (CWT) (6, 7), or model functions such as Gaussian function used as templates for peak detection (see Subheading 2 for more details).

A large variety of software packages implementing the complete workflow is available. Common software tools are R or Bioconductor packages msProcess or PROcess (8), Matlab packages LIMPIC (9) or Cromwell (3), the comprehensive C++ library OpenMS (10, 11), and, of course, the proprietary software packages that come with the analytical equipment (see Note 5).

In this chapter, we will focus on three different peak-detection algorithms (SNR, CWT, and Template-based approach), illustrate their principles in an intuitive manner, and compare them in terms of sensitivity and specificity using ROC curves. We have selected these three algorithms since they are very popular and widely used. Furthermore, all three of them are very different and derived from distinctive theoretical motivations. Many extensions or combinations for these algorithms have emerged over the last years. For a more comprehensive overview including different techniques for smoothing and baseline correction, see Yang et al. (12). While Yang et al. give a comprehensive overview on publicly available software by briefly describing the applied methods, our interest in this article is rather the demonstration of the working principle of the algorithms employed in these public software packages. Following up the evaluation of available peak-detection algorithms by Yang et al, Liu et al. (13) compared different feature selection and classification algorithms in a similar way.

1.1. Data Set

To evaluate the different algorithms, we used data obtained by MALDI-TOF-MS analysis of 259 blood plasma samples from 56 different mice taken at five different time points. Plasma MS profiles were obtained using an Ultraflex MALDI-TOF/TOF mass spectrometer (Bruker Daltonics, Bremen, Germany). Spectra were acquired automatically for the m/z range of 700–10,000. The amount of plasma obtained for each sample varied between 0 and 12 µl. Since 5 µl of plasma was needed for each sample preparation, it was possible to perform up to two sample preparations. In a few cases, only one or no sample preparation could be performed. From each sample preparation, four replicate MALDI spectra were acquired, resulting in a total of up to eight technical replicates per sample.

The total number of mass spectra acquired was more than 2,100. Prior to any data processing described in this article, technical replications are averaged, reducing the number of spectra to 258. For averaging multiple spectra, we applied a peak alignment strategy (14).

2. Peak Picking

2.1. Algorithms

The three common peak-detection algorithms we will focus on are SNR, template-based peak detection, and CWT. We have selected these three algorithms since they are very popular and widely used. For the SNR and template-based approach, we used an in-house implementation, while for CWT, we used the R package msProcess.

1. *Signal-to-noise ratio*: This is a very general approach. The essential part of this algorithm lies in the definition of noise.

In statistics, noise is often defined as variance or median absolute deviation (MAD) along different samples. In signal processing, noise is often defined as the estimated background. For instance, in the Bioconductor package PROcess (8), MAD of points within a window is used. For this analysis, we follow the second approach defining noise as background of the spectrum. We estimated the background using Top Hat filter (4) with small window size. Having defined the noise, we calculated the SNR. Peaks were then identified by searching a local maximum of points within a certain neighborhood (e.g., about expected peak width) having an SNR bigger than a given threshold.

2. *Template-based peak detection*: This algorithm assumes that the peaks to be detected are shaped like some model function, e.g., a Gaussian function. With a running window, the algorithm scans along the mass spectra and calculates the correlation (Pearson correlation coefficient) to a template Gaussian function with predefined parameters. Thus the mass spectrum is transformed into a vector of correlation coefficients. Peak identification is done by searching for correlation values above a certain threshold.
3. *Continuous wavelet transform* : CWT (6, 7) is a more sophisticated approach that is used to split the signal into different frequency ranges. Regarding the *m/z* scale as generalized time scale, CWT constructs a time–frequency representation of the spectrum by mapping it from the time domain to the time-scale domain. The essential part of CWT is the mother wavelet whose translated and scaled versions are used to generate daughter wavelets. The mother function we used for this evaluation is the second derivative of a Gaussian function (Mexican Hat Wavelet). Peak picking typically includes the inspection of multiple scales. For peak detection (using R package msProcess), the peak candidate has to be clearly distinguishable from the background (parameter: *snr.min*) and visible across at least seven scale domains (parameter: *length.min*) excluding the first three high-frequency wavelet scales (parameter: *scale.min*). Excluding high-frequency wavelets acts as a filter for high-frequency noise.

2.2. Reference Peaks

In order to evaluate the peak-picking algorithms, we defined a set of reference peaks. A peak-picking algorithm can then be evaluated in terms of sensitivity (how many of the reference peaks are found) and specificity (how many of the found peaks are part of the reference set). An optimal algorithm has high sensitivity and high specificity.

The reference set was created in a semi-automatic process. To this end, we initially picked peaks manually and subsequently

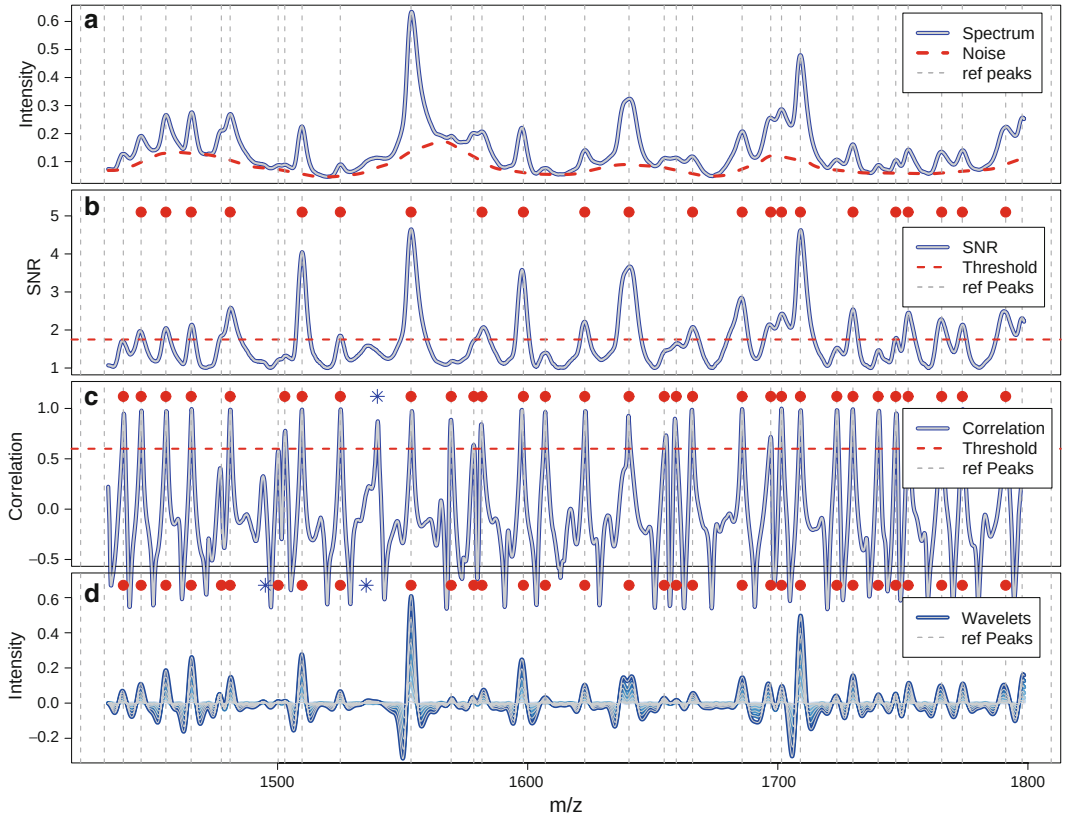


Fig. 2. Comparison of the three different peak-picking algorithms for the *m*/*z*-range of 1,400–1,800. (**a**) Mean spectrum and noise/background (*dashed line*); (**b**) SNR and threshold used for peak picking (*horizontal dashed line*); (**c**) correlation coefficient and threshold (*horizontal dashed line*); (**d**) first seven wavelets. The *vertical dashed lines* are reference peaks. *Marks above the plot* indicate identified peaks (*dot*, contained in the reference set; *asterisk*, not in reference set – false positives).

optimized peak positions automatically. This procedure ensures a high-quality reference set containing very prominent peaks as well as peaks situated in the rising or falling edge of another peak or peaks with poor signal intensities. All in all, the reference set contained a total of 381 peaks.

2.3. Comparing Peak-Picking Algorithms

Figure 2 gives a graphical impression of how the different algorithms are working. The first box shows the mean intensity spectrum of the complete data set in a mass window of 1,400–1,800 Da. The noise level was defined as baseline calculated using Top Hat filter (see dashed line). The 33 peaks from the reference set within this mass window (see Subheading 2.2) are indicated as vertical dashed lines.

The second part of Fig. 2 shows the SNR along the mass window of the mean spectrum. The SNR threshold used for peak identification was 1.75 and is indicated as a horizontal dashed

line. Using SNR, we identified 22 peaks in this mass range, whereas we found 69% of our reference peaks (with the SNR threshold of 1.75). With this threshold, we did not find any peak that was not part of the reference set.

The third part in Fig. 2 visualizes the performance of template-based peak detection. The correlation coefficients along the spectrum are shown. The correlation threshold of 0.6 is indicated as a horizontal dashed line. All in all, we found 31 of the 33 reference peaks (94%), indicated as dots above the peaks. We also found one peak that is not within the reference set (false positive), shown as asterisk above the peak.

The last part of Fig. 2 demonstrates the peak picking using wavelet transform. The first seven daughter wavelets are shown. Compared to the other two methods, peak picking using wavelet transform is complicated by the fact that information from different time-scale domains has to be combined (see Chapter 2.1 for more details). The reference peaks again are indicated as vertical dashed lines and the picked peaks are marked above the peaks. Using CWT, we identified 97% of the peaks but also got two false-positive hits (marked with asterisks above the peaks).

2.4. Evaluating Peak-Picking Algorithms

As already mentioned in Subheading 2.2, the reference peak set can be used to calculate values for sensitivity and specificity. These, in turn, can be used to generate ROC curves (see Fig. 3). ROC curves are calculated by scanning the threshold values of the different algorithms, e.g., changing the correlation threshold in the template-based approach (for an illustration of the threshold operation, see Fig. 2).

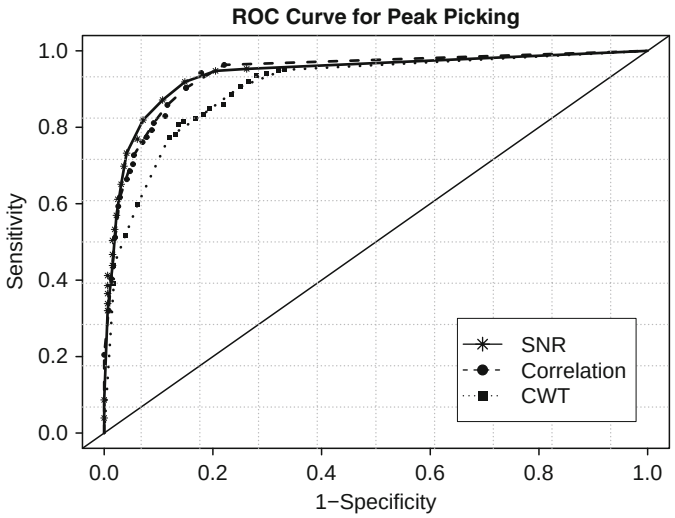


Fig. 3. ROC curves presenting the performance of the three different peak-picking algorithms: SNR (*full line*), Correlations with Gaussian function (*I*) and continuous wavelet transform (CWT) (*pointed line*). The ROC curves are calculated by scanning the threshold values of the different algorithms.

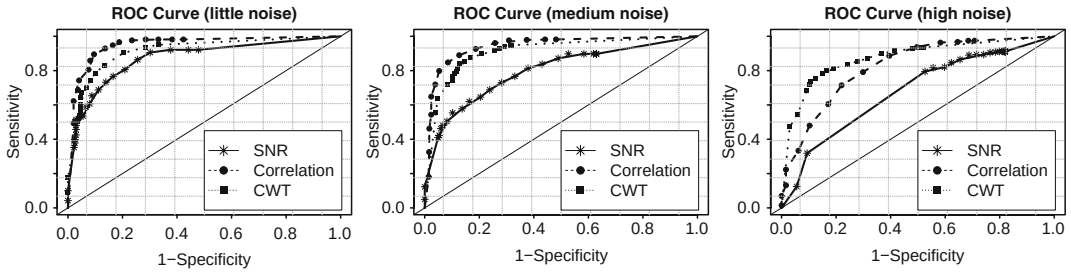


Fig. 4. ROC curves for the three different peak-picking algorithms on noisy data: SNR (*full lines*), correlations with Gaussian function (*dashed lines*), and CWT (*pointed lines*).

2.4.1. Stability

While baseline correction removes low frequency noise, smoothing rather aimes at filtering high frequency noise, and hence, application of both, baseline correction and smoothing, defines a band-pass filter. The combination of smoothing and baseline correction defines a bandpass filter removing high- and low-frequency noise. Parameter tuning of the preprocessing steps affects the signal and noise that the peak-picking algorithms have to deal with. In order to evaluate the sensitivity to noise, we added different quantities of high-frequency noise (white noise). Since the observed error behavior for MS spectra indicates a multiplicative error behavior on log scale data (not shown), we added a normally distributed noise with mean = 0 and an error of 2, 4, and 10%. The performances of the three methods are affected to a very different degree (see Fig. 4 for the ROC curves). SNR is very sensitive to noise and the ROC curve worsens dramatically. The other two algorithms are much more robust. While on perfectly smoothed data the template correlation approach seems to be the method of choice, for noisy data, the advantage of the template-based approach decreases and CWT shows the best performance. In conclusion, the three presented peak-picking algorithms show a different sensitivity to noise and, therefore, to the number of spectra and the choice of parameters for preprocessing steps.

In order to get more insight into how noise influences the peak detection, Fig. 5 gives a demonstration of the algorithm's performances on noisy data. The first row shows the raw spectra and the baseline. SNR is depicted in the second row of Fig. 5. Here, the noise is even amplified due to the ratio calculation, leading to an increased number of false-positive peaks (see the asterisks). The template correlation approach is more robust but since this method assesses only the shape and not the intensity, even small fluctuations may result in high correlation coefficients. Thus peaks are not clearly distinguishable from background noise any more. For noisy data, CWT outperforms the other methods since CWT intrinsically acts like a smoothing filter on the data. Even if the first wavelets are noisy, the lower frequency scales are very smooth (see lower row of Fig. 5). Hence high-frequency noise does not affect the algorithm's performance strongly as high-frequency wavelets that include most of the noise are filtered out.

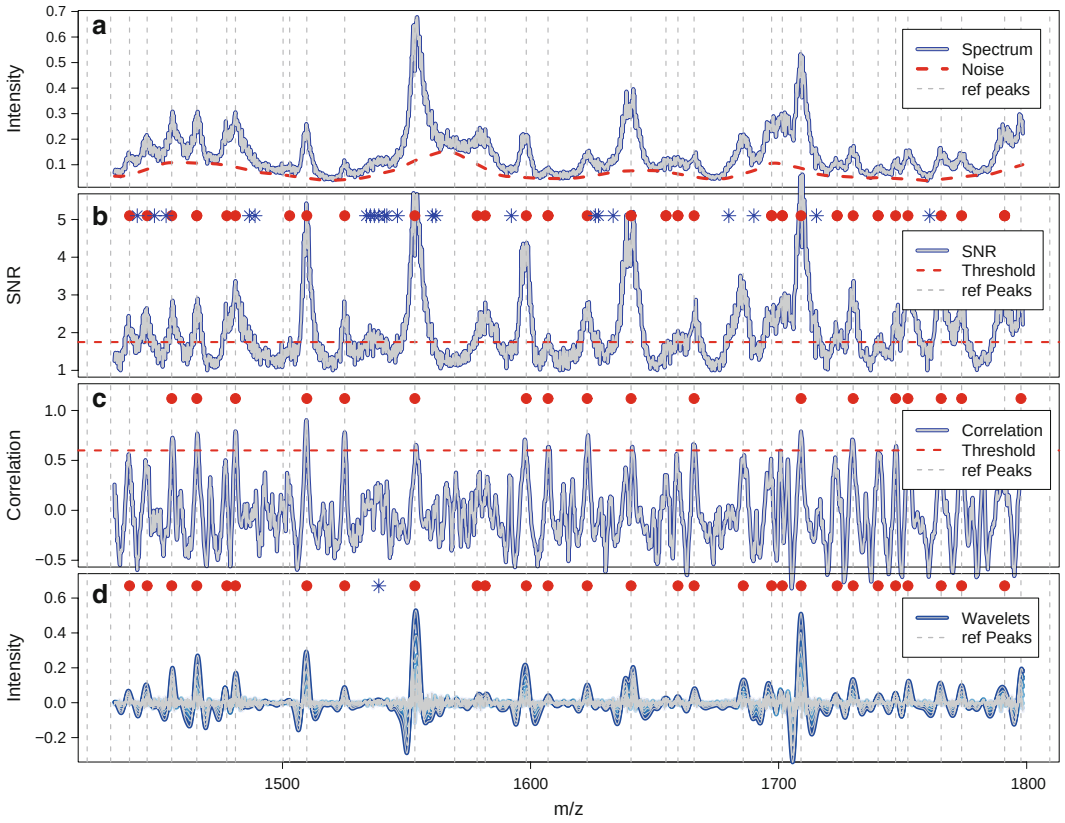


Fig. 5. Comparison of the three different peak-picking algorithms on noisy data for the *m/z* range of 1,400–1,800. High-frequency noise was added to the spectrum as described in the text. (**a**) Mean spectrum and background (*dashed line*); (**b**) SNR and threshold used for peak picking (*horizontal dashed line*); (**c**) correlation coefficient and threshold (*horizontal dashed line*); (**d**) first seven wavelets. The *vertical dashed lines* are reference peaks. *Marks above the plot* indicate peaks identified with the different algorithms (*dot*, contained in the reference set; *asterisk*, not in the reference set).

3. Discussion

The three different peak-picking algorithms investigated here are distinct in terms of complexity, performance, and stability. But all the three methods have a common parameter: the estimated peak width. There are different ways to estimate the optimal peak width. For instance, OpenMS (10, 11) as a freely available MS-processing library offers the possibility to measure the peak width manually using graphical interface, or the peak width can be estimated by the CWT algorithm itself.

For an overview of the advantages and disadvantages of the algorithms, see Table 1. SNR as a universally used signal-processing technique is computationally fast, easy to implement, and shows good performance on smoothed data. However, it is not very

Table 1
Summary of advantages and disadvantages of the three presented peak-picking algorithms

Method	PRO	CONTRA
SNR	Simple – easy to implement Fast performance Only few parameters	Depends on the definition of noise Unstable – very sensitive to noise Ignoring peak shape
Template correlation	Simple – easy to implement Only few parameters Stable for small noise	Detection favors Gaussian-shaped peaks Sensitive to high noise
CWT	Stable even for massive noise Internal data smoothing Flexible – tuneable	Complicated algorithm Slow performance Difficult to tune – high number of parameters

specific for this task as it ignores the shape of the peak. Since the noise is an integral part of the algorithm, it is very sensitive to noise and, therefore, strongly depends on the quality of the data and on the performance of previously performed smoothing and baseline correction steps (see Note 6).

The template-based approach is much more specific for the peak-picking task assuming peaks to be shaped like a Gaussian function. This assumption, however, might often not be exactly applicable because peaks may show a considerable asymmetry. Depending on the experimental parameters, particularly laser energy, significant deviation from a Gaussian peak shape can be globally obtained. Although this method has only a few parameters, it appears rather robust for lower levels of noise. However, for high levels of noise, the performance decreases (see Note 7).

CWT is like SNR a very universal signal-processing technique used for many different tasks. Contrary to SNR, the algorithm is complex and computationally expensive. The large number of parameters allows for tuning CWT to be very specific for this task taking into account the shape of the peak. As smoothing is an intrinsic part of the algorithm, CWT is very robust even for substantial amounts of noise. On the contrary, tuning of the algorithm is difficult due to the large set of parameters (see Note 8).

For perfectly smoothed data, all the three methods show good performances but CWT seems to be little worse than the other two. For data including a substantial amount of noise, CWT clearly outperforms the other methods in terms of sensitivity and specificity.

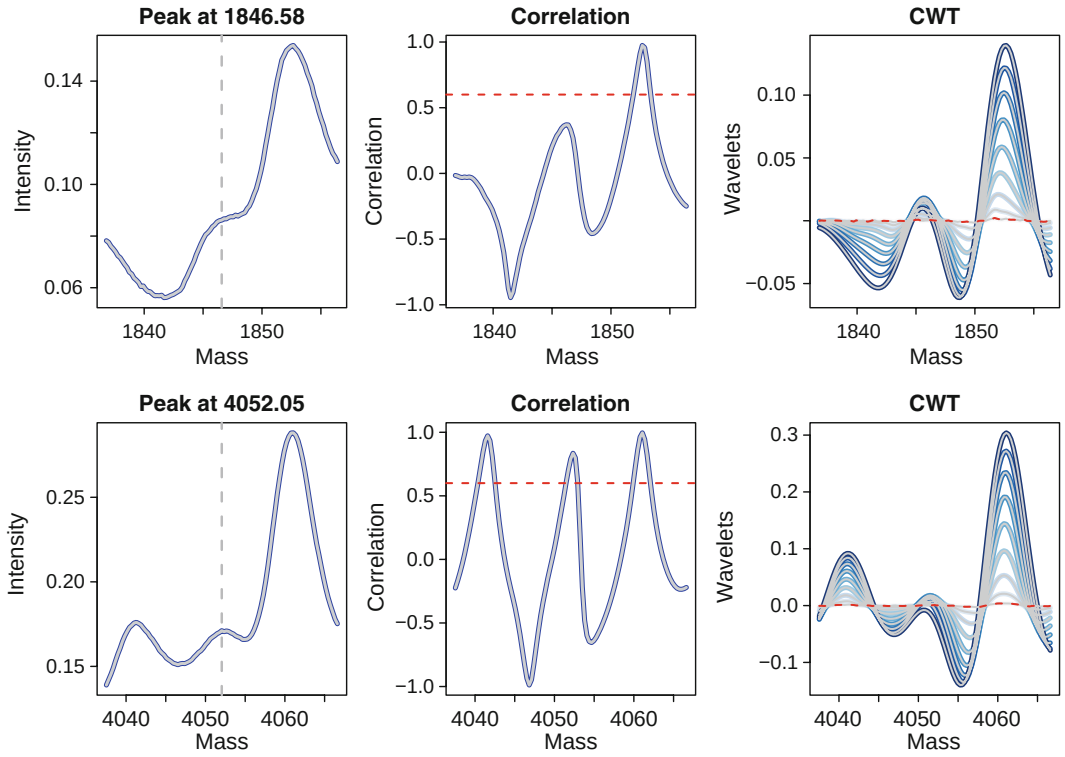


Fig. 6. Peaks found with CWT and not with template-based approach (*upper part*) and vice versa (*lower part*), first column: spectrum, second column: correlation coefficient and correlation threshold, third column: first nine wavelets and noise (*dashed line*).

Both the template-based approach and CWT show good performances, including a robustness for noise. Figure 6 shows two example peaks for the different peak detection using these two algorithms. In the upper row, the peak at m/z 1,846 was identified only with CWT, while in the lower row, the peak at m/z 4,052 was detected only with the template-based method (see Note 9). The shortcoming of the template-based approach is clearly visible since the peak at m/z 1,846 is not shaped like a Gaussian function, resulting in lower correlation coefficients. Hence this peak could not be detected using a Gaussian function as template. In contrast, the peak at m/z 4,052 shows a good matching Gaussian shape, facilitating peak detection by correlation. CWT does not find this peak since there are not enough wavelets above threshold (in this case, there are only five wavelets above the noise level but the algorithm requires at least seven) (see Note 10).

The reference set we used for evaluation was manually created assuming the human eye to be a good peak detector. With this procedure, we assure that the reference set is constructed without giving preference to any algorithm. Looking at the spectra and the visualization of the three algorithms (Fig. 2), we see

that there is one peak identified with correlation-based approach and CWT (indicated as asterisks), and even with a higher SNR that was not classified as a peak using the human eye. However, remarkably, the three algorithms differ in exactly this peak, underlining that in general peak picking is a non-trivial task (see Note 11).

4. Notes

1. For MALDI-TOF data, adequate preprocessing is required in order to allow subsequent statistical data analysis, such as biomarker discovery or sample classification.
2. Preprocessing workflow typically comprises algorithms for data smoothing such as Mean filter or Savitzki–Golay filter, baseline correction such as Top Hat filter or Loess derived filters, and peak picking such as SNR, CWT, or template-based approaches.
3. The main objective is to transform the big amount of raw spectral data into a much smaller, statistically manageable set of peaks.
4. The number of algorithms implementing peak picking is large. The various algorithms differ in performance, implementation, and theoretical motivation.
5. Various common software tools designed to address the preprocessing workflow are available. They are based on different platforms including R and Matlab packages as well as stand-alone C++ applications.
6. Approaches based on SNR are not only simple, easy to use, and fast but also sensitive for noise. Moreover, the shape of the peak is ignored completely.
7. Template-based approaches are simple, easy to use, and robust to limited noise. But they can only detect peaks shaped like the used template function and they are vulnerable to strong noise.
8. CWT shows good performances and is stable even for strong noise but more complicated, difficult to tune, and, therefore, harder to use and understand.
9. Every algorithm has pros and cons as it fails in finding certain types of peaks.
10. Template-based approach fails to detect peaks differing in shape from those of the used template. CWT tends to miss thin peaks surrounded by higher ones.
11. The definition of the reference peak set is a crucial step for evaluating the different algorithms. Neither the human eye

nor some automatic peak detection algorithm can guarantee to detect all peaks. Still, regarding a sensitivity and specificity of 0.9, the majority of the peak show good agreement of the used algorithms and the human eye.

References

1. Kwon D, Vannucci M, Song JJ, Jeong J, Pfeiffer RM (2008) A novel wavelet-based thresholding method for the pre-processing of mass spectrometry data that accounts for heterogeneous noise. Proteomics 8: 3019–3029
2. Savitzky A, Golay MJE (1964) Smoothing and differentiation of data by simplified least squares procedures. Anal Chem 36(8):1627–1639
3. Coombes KR, Tsavachidis S, Morris JS, Baggerly KA, Hung MC, Kuerer HM (2005) Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform. Proteomics 5:4107–4117
4. Sauve AC, Speed TP (2004) Normalization, baseline correction and alignment of high-throughput mass spectrometry data. Proceedings of the genomic signal processing and statistics, 2004
5. Cleveland WS, Grosse E, Shyu WM (1992) Local regression models. In: Chambers JM, Hastie T (eds) Statistical models in S. Wadsworth & Brooks/Cole, Pacific Grove, CA, pp 309–376
6. Lange E, Gropl C, Reinert K, Kohlbacher O, Hildebrandt A (2006) High-accuracy peak picking of proteomics data using wavelet techniques. Pac Symp Biocomput 11:243–254
7. Du P, Kibbe WA, Lin SM (2006) Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. Bioinformatics 22: 2059–2065
8. Gentleman R, Carey V, Huber W, Irizarry R, Dudoit S (eds) (2005) Bioinformatics and computational biology solutions using r and bioconductor. Springer, New York
9. Mantini D, Petrucci F, Pieragostino D, Del Boccio P, Di Nicola M, Di Ilio C, Federici G, Sacchetta P, Comani S, Urbani A (2007) LFMPIC: a computational method for the separation of protein MALDI-TOF-MS signals from noise. BMC Bioinform 8:101
10. Kohlbacher O, Reinert K, Gröpl C, Lange E, Pfeifer N, Schulz-Trieglaff O, Sturm M (2007) TOPP-the OpenMS proteomics pipeline. Bioinformatics 23:e191–197
11. Sturm M, Bertsch A, Gröpl C, Hildebrandt A, Hussong R, Lange E, Pfeifer N, Schulz-Trieglaff O, Zerck A, Reinert K, Kohlbacher O (2008) OpenMS – an open-source software framework for mass spectrometry. BMC Bioinform 9:163
12. Yang C, He Z, Yu W (2009) Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis. BMC Bioinform 10:4
13. Liu Q, Sung AH, Qiao M, Chen Z, Yang JY, Yang MQ, Huang X, Deng Y (2009) Comparison of feature selection and classification for MALDI-MS data. BMC Genomics 10(Suppl 1):S3
14. Jeffries N (2005) Algorithms for alignment of mass spectrometry proteomic data. Bioinformatics 21:3066–3073

Chapter 23

OpenMS and TOPP: Open Source Software for LC-MS Data Analysis

Andreas Bertsch, Clemens Gröpl, Knut Reinert, and Oliver Kohlbacher

Abstract

Proteomics experiments based on state-of-the-art mass spectrometry produce vast amounts of data, which cannot be analyzed manually. Hence, software is needed which is able to analyze the data in an automated fashion. The need for robust and reusable software tools triggered the development of libraries implementing different algorithms for the various analysis steps. OpenMS is such a software library and provides a wealth of data structures and algorithms for the analysis of mass spectrometric data. For users unfamiliar with programming, TOPP ("The OpenMS Proteomics Pipeline") offers a wide range of already implemented tools sharing the same interface and designed for a specific analysis task each. TOPP thus makes the sophisticated algorithms of OpenMS accessible to nonprogrammers. The individual TOPP tools can be strung together into pipelines for analyzing mass spectrometry-based experiments starting from the raw output of the mass spectrometer. These analysis pipelines can be constructed using a graphical editor. Even complex analytical workflows can thus be analyzed with ease.

1. Introduction

Over the last several decades, mass spectrometry has become a key technology in analytical chemistry for the analysis of proteins. High-throughput analysis using mass spectrometry-based methods has led to significant progress in proteomics and, more recently, in metabolomics. In addition to the growing amount of data, the development of new instruments is a very active field resulting in new and improved hardware within short time intervals. The combination of high data volume and new hardware leads to a short lifetime of software tools, which are often designed in a monolithic fashion combining several analyses in one program. Data analysis of high-throughput experiments has thus become the major bottleneck. Therefore, it is beneficial to have individual software components that are designed for one small,

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_23,

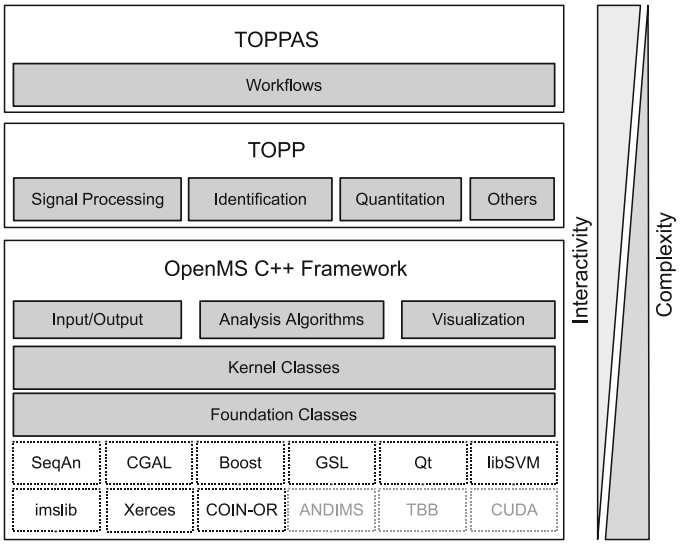


Fig. 1. The whole OpenMS project is subdivided into three main categories. First, there is the OpenMS C++ framework. It uses a number of other projects to provide all the functionality implemented in the library. Using the library requires C++ programming knowledge. On top of OpenMS, TOPP was implemented. It consists of different tools, each of which implements a specific task, e.g., noise filtering of mass spectrometry datasets. Users who are familiar with command line handling are able to use TOPP. The third part is TOPPAS, the TOPP pipeline assistant. It provides a graphical user interface and conveniently allows the creation of analysis workflows using the TOPP tools. No programming skills or command line knowledge is needed to analyze datasets with TOPPAS.

specific analysis step only. Although such a component is limited to one specific task, it can be combined with other light-weight tools into powerful analysis workflows.

In this chapter, we first describe the OpenMS framework, its design goals, the core functionality, and its use for rapid application development. This might be of special interest for software developers experienced in implementing applications using C++. Second, we address in detail how TOPP can be used to analyze data from mass spectrometry-based experiments using workflows, e.g., in a scripting language. The third part describes how TOPP can be used to conveniently create analysis pipelines and create powerful workflows through a graphical user interface, TOPPAS. See Fig. 1 for an overview.

2. Materials

2.1. The OpenMS Framework

OpenMS (1) is a C++ framework for computational mass spectrometry. It provides a large collection of data structures and algorithms to process and analyze data from mass spectrometry-based

experiments. The kernel data structures, which implement the main functionality regarding the handling of basic mass spectrometry data, are built on top of a few common basic data structures, e.g., classes for mass spectra, chromatograms, and whole LC-MS experiments. The *chemistry module* of the framework offers a large functionality like support for empirical formulas, amino acids, amino acid modifications, and calculation of isotope distributions. Metainformation, such as instrument settings for a whole experiment, settings for a specific spectrum, sample processing information, contact information, and all other information required by the MIAPE guidelines (2) are implemented in the *metadata module*. The described data structures and functionality are also used to implement file format adapters. Most of the mass spectrometry data produced today is stored in proprietary file formats that are specific to a vendor or even a specific type of instrument. To overcome the disadvantages of proprietary file formats, OpenMS implements the file formats developed by the HUPO Proteomics Standards Initiative (HUPO-PSI). Additionally, the widely used spectral data formats mzML, mzXML, and mzData as well as several others are supported.

The framework implements a large number of sophisticated algorithms, which can be roughly subdivided into four groups: *signal processing, map alignment, quantitation,* and *identification* algorithms. Signal processing of profile mass spectrometry data includes baseline filtering, noise filtering, and centroiding. Quantitation algorithms include feature finding and linking of features, either based on their label in labeled experiments or across different HPLC-runs in label-free experiments.

Another module implements classes for the map simulator (3), which allows one to simulate different kinds of experiments. This simulator is very helpful, in estimating the performance of algorithms, especially when reliable ground truth is hard to obtain. Visualization classes implement the basis for viewer applications and provide visual access to whole HPLC-MS experiments, either in 2D or 3D view, or scanwise as single spectra. Also, visualizations for parameter editing and metainformation visualization and editing are available.

Robustness, portability, rich functionality, and ease-of-use were the key design goals during the development of OpenMS. The resulting software framework allows one to prototype most applications in computational mass spectrometry with comparative ease and in a short time. A simple example of OpenMS code is shown in Fig. 2. The code illustrates how a few lines of C++ code allow the signal processing and peak picking on mass spectrometric data and introduces some of the core data structures of OpenMS.

The OpenMS framework also implements support for features only used by a small number of algorithms or tools.

```
01 PeakMap exp_raw;
02 PeakMap exp_picked;
03
04 MzMLFile mzml_file;
05 mzml_file.load("data/Tutorial_PeakPickerCWT.mzML", exp_raw);
06
07 PeakPickerCWT pp;
08 Param param;
09 param.setValue("peak_width", 0.1);
10 pp.setParameters(param);
11
12 pp.pickExperiment(exp_raw, exp_picked);
```

Fig. 2. An HPLC-MS dataset is read using the MzMLFile file adapter into a PeakMap object (lines 1–5). Then, in line 7 an instance of the peak picker is created. To be able to handover parameters to the peak picker, a Param object is initialized in line 8. The peak_with parameter is set to 0.1 Th and in line 10 the parameters are finally assigned to the peak picker. By calling the member function pickExperiment in line 12, the peak picker transforms the raw data from the file into picked data stored in exp_picked. The complete code example can be found in the OpenMS tutorial.

For example, several parallelization concepts are readily available within the framework. OpenMP (4) was used to parallelize the peak picker code and is automatically enabled if the platform supports it. OpenMP can be used to parallelize existing code with minimal effort. Intel Threading Building Blocks (TBB) (5) provide a sophisticated concept of parallelization of C++ code, however, requires restructuring of existing algorithms. An even more advanced concept is CUDA (6), which allows the execution of parallelized code on Nvidia graphics cards. Although CUDA requires completely specialized implementations, its usefulness has been demonstrated, in particular, by an implementation of the isotope wavelet feature finder using TBB and CUDA, which executes about 200 times faster than the single-threaded version (7).

OpenMS also provides support for different statistical models. It provides different fitters built upon the GSL library (8), uses geometric algorithms provided by the CGAL library (9), and support vector machine-based machine learning models can also be used by including the LIBSVM library (10). Several methods have been successfully implemented using the SVM technique (3, 11). Sequence-based computations and data structures, such as suffix trees and sequence alignments, are provided using the SeqAn library (12). Imslib (13), which provides efficient decomposition of masses and mass differences, has been successfully used in CompNovo, a de novo search tool (14). Finally, Boost (15) completes the set of external libraries used by OpenMS. The libraries are included in the source package as a so-called contrib package, and building can be done automatically on the different platforms. The graphical user interface, database support, network communication, among others, are implemented using the Qt framework (16).

Since OpenMS is meant to be a rich framework for *software development* in mass spectrometry, it does not provide the actual

executable programs. That is the purpose of TOPP, The OpenMS Proteomics Pipeline (17). Writing new programs using OpenMS requires C++ programming skills and at least a certain level of familiarity with the internal concepts and implementation details. For users lacking those skills, most of the functionality of OpenMS is readily available as applications in TOPP, which will be described in the next section.

2.2. TOPP: The OpenMS Proteomics Pipeline

Virtually, all complex data processing tasks can be decomposed into a series of applications of basic building blocks, such as peak picking, map alignment, peptide identification, quantitation, etc. Each TOPP tool is designed to solve a single, basic task from such a data analysis workflow. It encapsulates a minimal useful set of functionality from OpenMS into one program. The idea is to have one program for one small task, to maximize the reusability of the individual workflow components.

This approach has been shown to be very fruitful since the early days of UNIX, and the design goals of TOPP were similar to those of the OpenMS framework. TOPP is available on the same platforms as OpenMS (Unix/Linux, MacOS X, and Windows); it is robust against failures or corrupt data, and easy to use by non-programmers. Splitting the functionality among different tools has some big advantages over monolithic tools: the modular concept is very flexible and can be rearranged to suit novel experimental setups. For example, it might turn out that most of the protocol can be realized with standard components of TOPP, and only a single piece has to be modified or implemented from scratch. On the other hand, steps not needed using a specific experimental setup can be easily skipped. In particular, quantitation can be done in various fashions using the same building blocks to create many different quantitation protocols.

All TOPP tools share a common interface that is extended by the tool-specific options. For example, each TOPP tool can write its default parameters by calling it with the *-write_ini* parameter and specifying a file that the options should be written to. Also the log file can be specified in each tool using the *-log* parameter. The *-threads* parameter specifies the number of threads, which can be used by the TOPP tool. This can significantly speed up the computation if the algorithm implemented in the TOPP tool supports the use of more than one CPU. Of course, different tools must have different parameters; however, each of the tools reads some input and writes some output. Therefore, the *-in* and *-out* parameters are shared among all the different TOPP tools.

TOPP also provides a convenient visual editor for the parameters. Figure 3 shows this graphical user interface, the "INIFileEditor". The parameters have descriptive names, given in the first column of the parameter table. By clicking on a line, a parameter is highlighted and its descriptive name is complemented

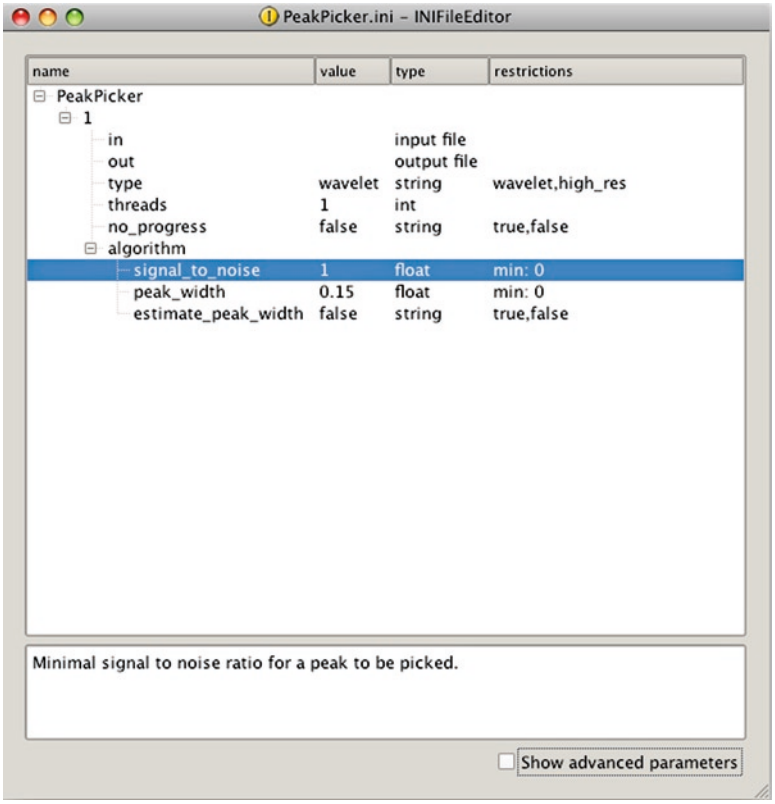


Fig. 3. INIFileEditor provides a convenient graphical interface to parameters used by the TOPP tools. The parameter ini file can be written from the TOPP tool itself and edited by the INIFileEditor. The figure shows several parameters, along with their default values and restrictions if available. A description of the selected parameter is shown *in the box at the bottom* of the window.

by a description shown at the bottom of the parameter editor. The second column shows the current value of the parameter. The third column gives the type of the parameter, which can be an integer value, a floating point value, a string, or a list. The fourth column shows restrictions, e.g., whether there is a minimal value or a maximal value that must not be exceeded. If only a limited list of values is allowed, a drop-down list appears for the user to select from.

In order to analyze the results and data visually, TOPP provides TOPPView, a graphical viewer for mass spectrometric data. TOPPView can display mass spectrometry experiments in many different ways. It allows the user to inspect the data, the results of different algorithms and provides an interface to the metadata and annotations of the data.

Figure 4 shows a screenshot of TOPPView in 2D mode. The m/z axis is plotted from left to right, the retention time axis is drawn from the bottom to top. The peaks are color-coded according to

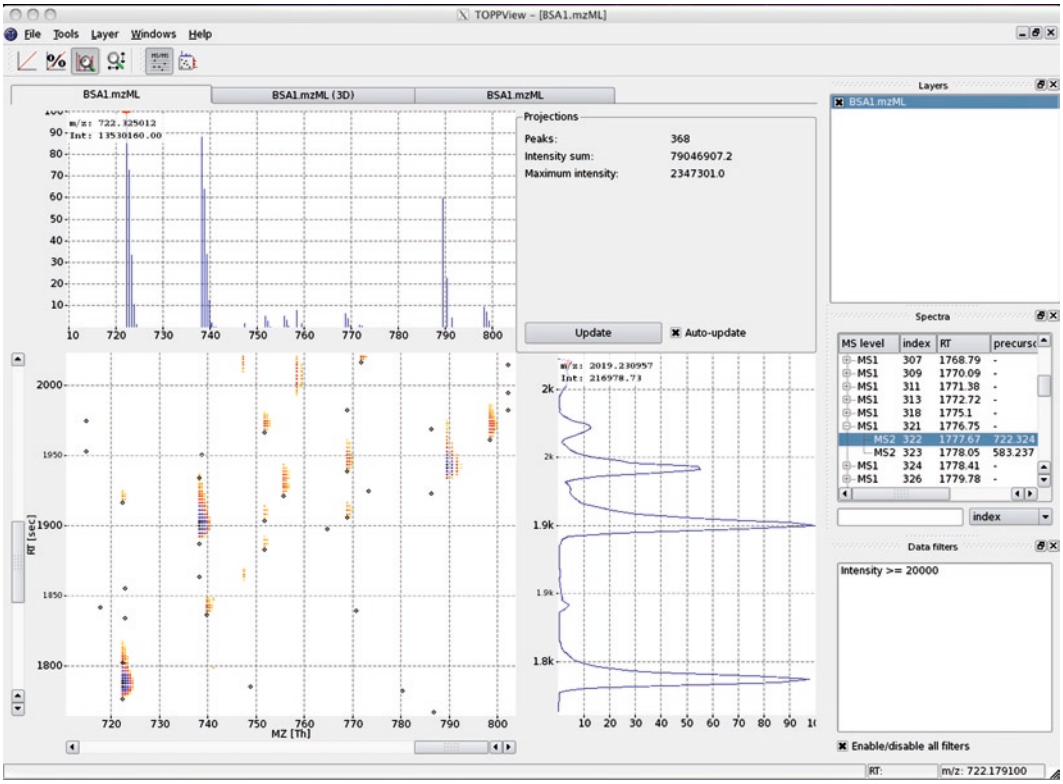


Fig. 4. TOPPView 2D view from the top. An example of HPLC-MS dataset is shown. *m/z* is from *left to right*, the retention time of the different MS scans are shown from bottom to top. The peaks are color-coded by their respective intensities. *Circles* in the 2D view indicate tandem MS scans.

their intensities. TOPPView also provides a 3D view (Fig. 5a), which allows a rapid inspection of the whole dataset and a 1D view (Fig. 5b) suitable for detailed inspection at the level of individual spectra. TOPPView was recently described in a separate publication (18) and an in-depth tutorial is available as well.

Most of the functionality provided by TOPP is dedicated to algorithms for various analysis steps. Database search engines are among the key software tools in proteomics. They assign peptide sequences to spectra based on a protein sequence database. Both commercial and open source tools are currently available for this task. Unfortunately, all tools have different interfaces, require distinct parameter sets, and even require different input and output formats. To simplify the use of these tools in the context of larger analysis pipelines, TOPP offers several search engine wrappers. These wrappers offer a common interface with common input and output formats and thus allows the seamless integration of the different search engines. The files and parameters are translated into the native formats of the respective search engines. The output is then read and presented to the user in a common format for different search engines. Wrappers currently available in

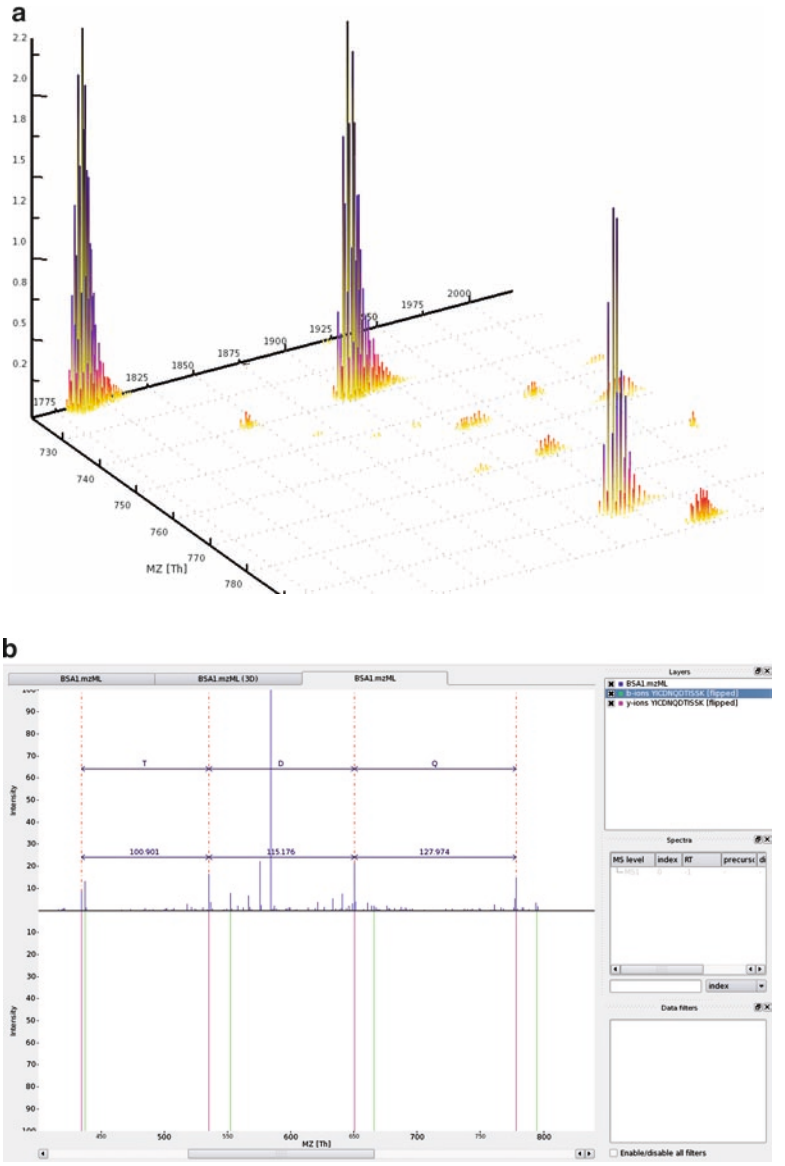


Fig. 5. Different views to inspect mass spectral datasets. (**a**) 3D view, which is very useful to get a quick overview of the data. Retention profiles and feature densities can be quickly estimated using this view. (**b**) 1D spectrum viewer, which can be used to inspect individual spectra. Either MS spectra or tandem MS spectra can be viewed. The spectrum viewer provides several features to annotate or analyze individual spectra.

TOPP include Mascot (19), Sequest (20), OMSSA (21), X!Tandem (22), and InsPecT (23).

The common interface of all TOPP tools makes it easy to string together a pipeline of different tools, which are executed

```
NoiseFilter -ini  NoiseFilter.ini -in ds.mzML -out ds_nf.mzML

BaselineFilter -ini BaselineFilter.ini -in ds_nf.mzML -out ds_bl.mzML

PeakPicker -ini PeakPicker.ini -in ds_bl.mzML -out ds_picked.mzML
```

sequentially. In most of the cases, output of the preceding tool is used as input of the one following. For example, after preparing the parameter files for NoiseFilter, BaselineFilter, and PeakPicker for a specific instrument, applying these tools to a data file is straightforward:

In this example, it is obvious how the output of the first tool (NoiseFilter) provides the input for the second tool (BaselineFilter). The output of the second tool is then used as input for the third tool. If the user wants to apply this pipeline to a number of files, a batch script (*.bat) under Microsoft Windows or a shell script under GNU/Linux or MacOS X can be used. In this way, one can effortlessly construct simple, linear data analysis pipelines.

3. Methods

3.1. TOPP Workflows

Constructing workflows in the manner just described is convenient for simple, linear pipelines. Experimental setups, however, are becoming increasingly complex and data analysis workflows have to keep up with this. For these complex, often nonlinear, workflows, we provide a convenient graphical workflow editor, TOPPAS, the OpenMS Proteomics Pipeline Assistant.

TOPPAS allows the convenient interactive creation and editing of complex analytical workflows without any programming skills. Workflows can be constructed by dragging analysis components on the screen and connecting these components in the desired order. The resulting workflows can be stored, including all parameters of the individual, and applied to arbitrary datasets. Workflows are platform-independent and can thus be transferred to other computers. A data analysis pipeline can be developed and tested on a laptop and later run on a larger compute cluster in a high-throughput setting.

All TOPPAS workflows start with an *Input files* node. There the files, which should be used in the pipeline, are selected. The last node of each TOPPAS workflow is an *Output files* node. It defines where the results of the analysis should be stored. In between the input and the output node, each TOPP tool can be

used by dragging the tool from the left bar to the workflow screen and dropping it there. Parameters of TOPP tools can be edited or loaded from files by double-clicking on a node in the workflow. Connections between tools are represented by arrows between the boxes representing the tools. An arrow between two boxes means that the output of the first tool is used as input of the second tool. By double-clicking on an arrow, the output and input file parameters can be changed. If several arrows leave from a tool, the data files are sent to each of the subsequent tools. The results of each intermediate step can be viewed in TOPPView from the context menu of the node. Once the pipeline has been built, it can be executed directly from within TOPPAS. Colored icons indicate the state of each tool (yellow: waiting; green: completed; red: error).

The use of TOPPAS for the creation of complex analysis workflows is now illustrated using two examples. The first example is an identification pipeline using different search engines and combining the result into a list of identified peptides. The second example performs a label-free quantitation of several HPLC-MS runs and uses the identification results to annotate the quantitative data.

The data files can be found in the examples directory of the OpenMS distribution, which is accessible via the *Open example file* menu entry in the *File* menu of TOPPView. The dataset consists of three test runs of a Bovine Serum Albumin (BSA) protein standard sample, recorded on different days. The datasets were generated on a Thermo Orbitrap XL.

3.2. Example: Peptide Identification Pipeline

The identification pipeline as it is seen in TOPPAS is shown in Fig. 6. First, an *Input files* node must be defined, by dragging it from the bar at the left side onto the canvas. By double-clicking on the node, the files for the pipeline can be selected. The nodes of the different search engines can be added, OMSSA (using *OMSSAAdapter*) and X!Tandem (*XTandemAdapter*). By adding arrows from the input files node to the search engines, TOPPAS runs each of the search engines on each input file. Parameters of the search engines can be modified by double-clicking the search engine nodes. The *PeptideIndexer* is required to recreate the protein references from the peptides to the proteins and to add information, such as whether a peptide is a decoy peptide. The *FalseDiscoveryRate* nodes then calculate false discovery rates and add the scores to the results. Then, the *IDFilter* can be used to cut at a specific false discovery rate (FDR), in our case 5%. The merge node merges the output of the two *IDFilter* nodes into a list of files, which is then merged into a single idXML file. The *ConsensusID* node is able to calculate the average FDR for each spectrum and accepts only spectrum identifications when both search engines agree on the identification. All intermediate steps

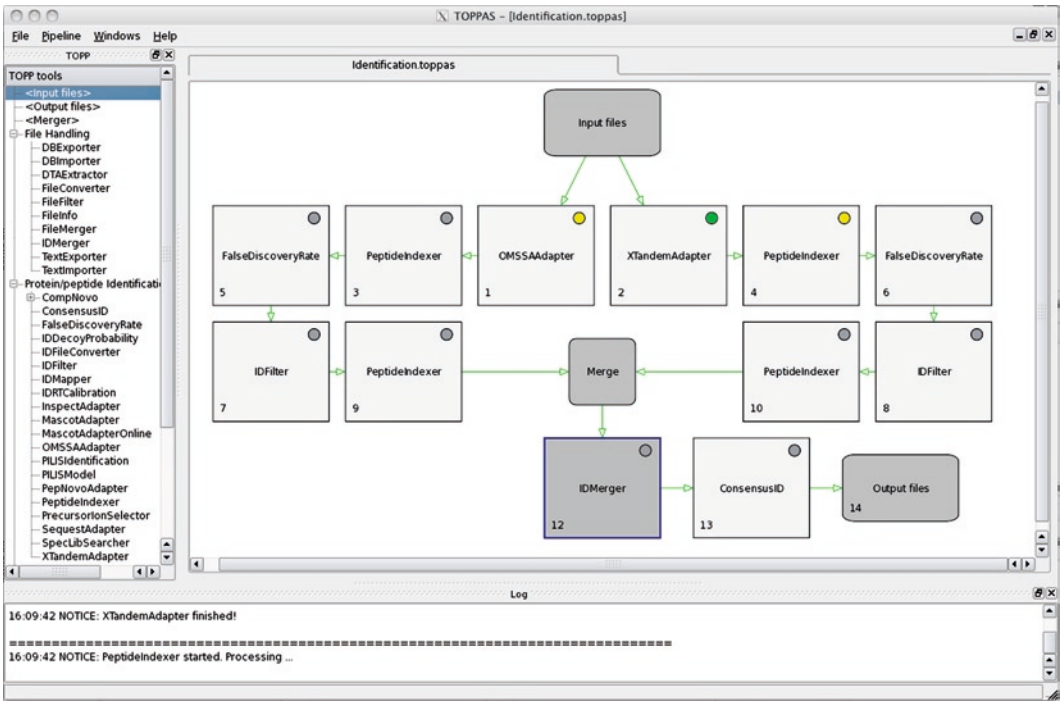


Fig. 6. An identification workflow using several search engines and combining the search results at the end. The *green circle* in the upper right corner indicates that the tool has finished. *Yellow means* that it is currently running and *gray* indicates that it has not started yet.

are stored, and the consensus identification results in idXML format will be used later on in the quantitation pipeline.

3.3. Example: Quantitation Pipeline

The quantitation pipeline also starts with an *Input files* node, as shown in Fig. 7. The next step is to detect all features in the different maps. The *FeatureFinder*, in our case the centroided feature finder, extracts regions of the measurements that contain signals that may be caused by different peptides of our target protein. To annotate the features with identification information, the *IDMapper* tool is used. The input of this tool, consisting of the spectrum data and the identification data, is provided by introducing an additional *Input files* node. The last step is to assign features present in different maps, but representing the same peptide into a group of features (also called consensus features). In our case, this group is distributed among the different measurements. After linking the features into groups, the results can also be exported to text-based formats for further analysis, e.g., with a spreadsheet application.

The final result of this analysis can be inspected using TOPPView. A small region of the dataset is shown in Fig. 8a. It contains the tryptic BSA peptide LSSPATLNSR, which was identified and quantified in each of the samples.

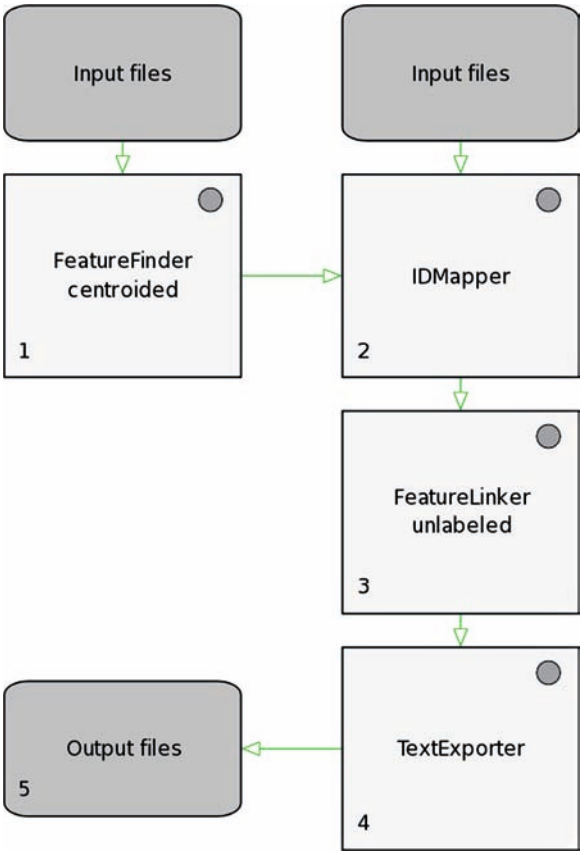
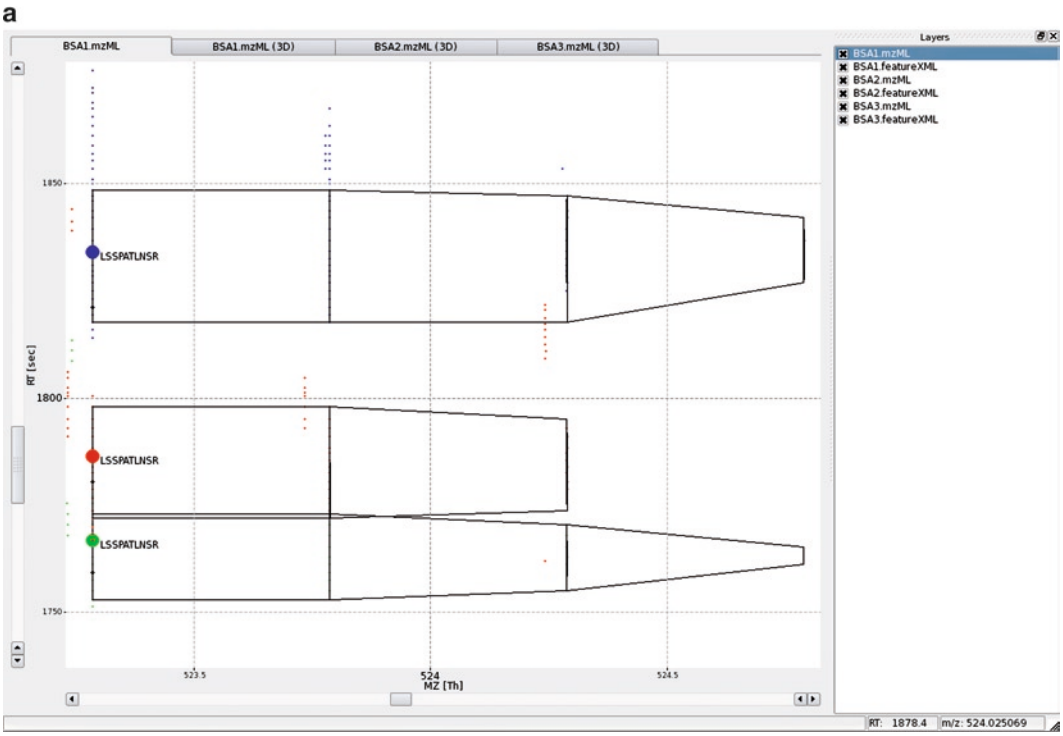


Fig. 7. This figure shows a quantitation workflow using several HPLC-MS experiments. The feature finder quantifies the different peptide features in the different HPLC runs. The IDMapper maps the identification results from the identification pipeline to the features and the feature linker finally links corresponding features to each other. At the end, the results are exported into a text-based format for further analysis.

4. Conclusions

OpenMS and TOPPView open source software packages for the analysis of high-throughput mass spectrometric datasets. Both are freely available on various platforms, including Microsoft Windows, MacOS X, and Linux. Binary packages are available for convenient installation. Additionally, a platform-independent source package is available, which is required to implement one's own code using the OpenMS C++ framework. OpenMS and TOPP are distributed under the GNU lesser general public license (LGPL).

The documentation of the current release is shipped with the binary packages as well as with the source package. Additionally,



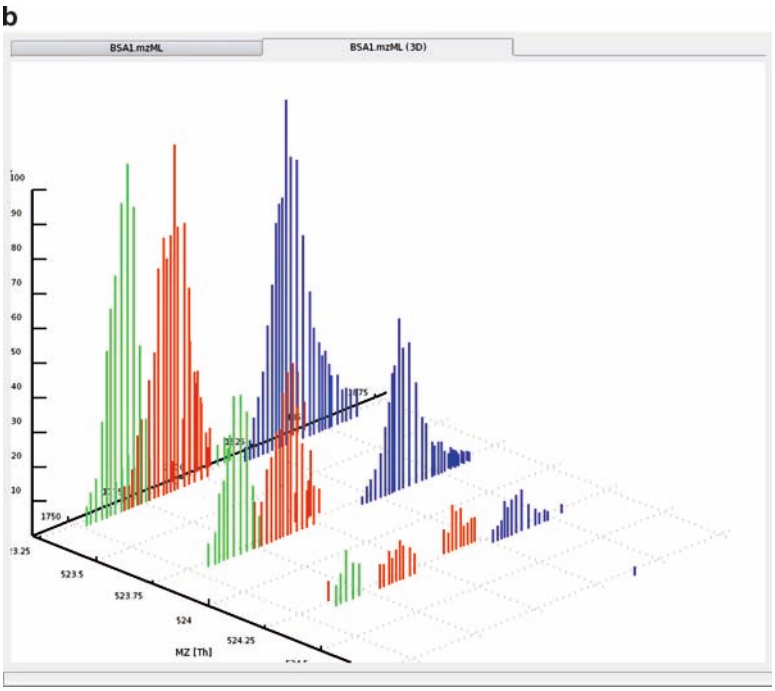


Fig. 8. A small part of the final results. (**a**) A 2D plot of the datasets with the features of the peptide LSSPATLNSR. (**b**) The same data as in the 2D plot shown in 3D. The different colors indicate the different measurements. The retention times are slightly shifted in the different runs.

the documentation can be found on our Web site at www.OpenMS.de. All the pipelines and examples described in this chapter as well as other code and pipeline examples are described in the tutorials contained in the documentation.

5. Notes

TOPP provides a collection of different tools for specific steps for the analysis of mass spectrometry-based proteomics experiments. The two analysis workflows described in the methods section are of course only examples. Depending on the experimental design, instrumentation used, and the desired output, the workflows might be very different. For example, labeled experiments can be analyzed using a similar workflow, skipping the steps needed to map the different HPLC runs onto each other. The provided examples just show how one can create analysis pipelines very efficiently.

Acknowledgments

The authors would like to thank Johannes Junker for most of the implementation work of TOPPAS and the whole OpenMS team for their contributions to this project. We would also like to thank the Proteome Center Tübingen for providing the BSA measurements of the example dataset.

References

1. Sturm M, Bertsch A, Gröpl C et al (2008) OpenMS – an open-source software framework for mass spectrometry. BMC Bioinform 9:163
2. Taylor CF, Paton NW, Lilley KS et al (2007) The minimum information about a proteomics experiment (MIAPE). Nat Biotechnol 25:887–893
3. Schulz-Trieglaff O, Pfeifer N, Gröpl C et al (2008) LC-MSsim – a simulation software for liquid chromatography mass spectrometry data. BMC Bioinform 9:423
4. http://www.OpenMP.org
5. http://www.threadingbuildingblocks.org
6. http://developer.nvidia.com/object/cuda.html
7. Hussong R, Gregorius B, Tholey A et al (2009) Highly accelerated feature detection in proteomics data sets using modern graphics processing units. Bioinformatics 25:1937–1943
8. http://www.gnu.org/software/gsl/
9. http://www.cgal.org
10. http://www.csie.ntu.edu.tw/~cjlin/libsvm/
11. Pfeifer N, Leinenbach A, Huber CG et al (2007) Statistical learning of peptide retention behavior in chromatographic separations: a new kernel-based approach for computational proteomics. BMC Bioinform 8:468
12. Döring A, Weese D, Rausch T et al (2008) SeqAn an efficient, generic C++ library for sequence analysis. BMC Bioinform 9:11
13. Bocker S, Letzel M, Liptak Z et al (2009) SIRIUS: decomposing isotope patterns for metabolite identification. Bioinformatics 25:218–224

14. Bertsch A, Leinenbach A, Pervukhin A et al (2009) De novo peptide sequencing by tandem MS using complementary CID and electron transfer dissociation. Electrophoresis 30:3736–3747
15. http://www.boost.org
16. http://qt.nokia.com/products/
17. Kohlbacher O, Reinert K, Gröpl C et al (2007) TOPP – the OpenMS proteomics pipeline. Bioinformatics 23:191–197
18. Sturm M, Kohlbacher O (2009) TOPPView: an open-source viewer for mass spectrometry data. J Proteome Res 8:3760–3763
19. Perkins DN, Pappin DJ, Creasy DM et al (1999) Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis 20:3551–3567
20. Yates JR, Eng JK, McCormack AL et al (1995) Method to correlate tandem mass spectra of modified peptides to amino acid sequences in the protein database. Anal Chem 67:1426–1436
21. Geer LY, Markey SP, Kowalak JA et al (2004) Open mass spectrometry search algorithm. J Proteome Res 3:958–964
22. Craig R, Beavis RC (2004) TANDEM: matching proteins with tandem mass spectra. Bioinformatics 20:1466–1467
23. Tanner S, Shu H, Frank A et al (2005) InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. Anal Chem 77:4626–4639

Chapter 24

LC/MS Data Processing for Label-Free Quantitative Analysis

Patricia M. Palagi, Markus Müller, Daniel Walther, and Frédérique Lisacek

Abstract

In this chapter, we describe the use of SuperHirn and MSight, two complementary tools developed to the processing of label-free LC/MS data in view of the quantitation of proteomics samples. While MSight is mainly dedicated to the visualisation and navigation into LC/MS data, SuperHirn is specialised in peak detection, normalisation and alignment of LC/MS runs. These two tools can be used in a complementary way and one of the possible usages is described here.

1. Introduction

Peak intensities in a mass spectrum are no reliable indicators of the amount of a protein in a sample, due to current shortcomings of ionisation methods of mass spectrometers. However, variations in peak intensity of the same protein in different samples can accurately reflect differences in its abundance. MS-based proteomics is thus used to quantify protein relative abundance across samples.

MS-based quantitative proteomics relies on differential analysis: two or more LC/MS samples are compared; common peptides and/or proteins are detected and relatively quantified. Practically, differential analysis can be achieved by labelling the sample with a stable isotope, which will lead to mass shifts in the produced mass spectra. Differentially labelled samples are then mixed together and analysed by MS. Differences in peak intensities of the isotope pairs may accurately reflect differences in the abundance of the corresponding proteins. Some of these labelling techniques include ICAT, iTRAQ, TMT, SILAC, etc (1). Alternatively, label-free quantification that does not include stable isotopes can

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_24,

produce differential results. Both strategies require bioinformatics tools for detecting relevant peaks and assessing the significance of observed differences.

We focus here on the label-free strategy. There are currently two main label-free computational quantitative approaches: one measures and compares the MS signal intensity of peptide precursor ions belonging to a specific protein and the other counts and compares the MS/MS spectra identifying a specific protein. The software described in this chapter belongs to the first category. In a typical analytical workflow of this type of label-free procedure, the proteins of different samples are digested with an enzyme (usually trypsin) and aliquots of individual samples are injected into the LC/MS system. The remaining aliquots of the sample digests are used to produce replicates as well as LC/MS/MS acquisition. The LC/MS data are acquired with a high frequency of MS spectra and preferentially with high resolution mass spectrometers. The principal computational steps of data processing are the detection of peptide peaks, the alignment of the LC/MS runs with the retention time values, and the matching across samples (2, 3). Many different algorithms and software have been proposed to tackle these three issues, and they have been extensively reviewed by Muller et al. (4) among others.

In this chapter, we describe two complementary bioinformatics tools to display, process LC/MS data and quantify corresponding proteins. On the one hand, MSight (5), based on the Melanie 2-D gel image analysis system ((6), www.expasy.org/melanie), is specialised in two-dimensional representation, as well as visual analysis and comparison of datasets obtained from LC/MS. MSight displays and browses, as an image, any portion of the collected mass spectra, without transition from a global overview of all spectra to selected isotopic peaks. It is useful for navigating through large volumes of data generated by LC/MS runs and discriminating peptides or proteins from noise. MSight performs the first steps of the differential analysis by detecting peaks, aligning and matching sample sets. MSight targets users with little background in computer science, and its visualisation functionalities are suited to the analysis of low to medium number of LC/MS runs.

On the other hand, SuperHirn (7) is adapted to high-throughput runs. SuperHirn does not include an interface and thus requires some basic knowledge of Unix/Linux commands. It is specialised in the extraction of MS features (the equivalent to peak detection, see Note 1), in the alignment and normalisation of LC/MS runs. It is based on the detection of the precursor ion signal intensities on the MS level, the tracking of corresponding isotopic pattern in the retention time level and the reconstruction of a chromatographic elution profile of the monoisotopic peptide mass in a MasterMap data format. The SuperHirn method is

better adapted to data acquired on mass spectrometers equipped with the new generation of time-of-flight (Tof), Fourier transform-ion cyclotron resonance (FT-LTQ), or OrbiTrap mass analyzers. Optionally, SuperHirn can annotate the extracted features with available MS/MS peptide identifications and quantify the protein/peptide profiles in the generated MasterMaps.

2. Materials

2.1. Software and Hardware

The MSight image analysis application runs on the latest Windows operating systems and is freely available at the ExPASy server (www.expasy.org/msight). It can be simply downloaded and installed on a PC.

The SuperHirn software is programmed in C++, and the source code together with detailed documentation material is freely available on http://tools.proteomecenter.org/wiki/index.php?title=Software:SuperHirn. SuperHirn runs on Linux and Mac OS X platforms. The compilation procedure is usually straightforward and the compilation instructions are detailed in the manual given in the Web site above.

2.2. Data Format

MSight accepts LC/MS data generated from the majority of mass spectrometers supplied by Applied Biosystems, Bruker, Waters, ABI-SCIEX, or ThermoFinnigan, for example, and the mzXML format. SuperHirn reads LC/MS runs in mzXML format acquired in PROFILE mode and MS2 peptide identifications in pepXML format.

3. Methods

3.1. Displaying LC/MS Images with MSight

MSight benefits from the redundancy in consecutive mass spectra to visualise all spectra together in one single image. In an MSight image, the vertical dimension (y-axis) represents the retention time from LC, while the horizontal dimension (x-axis) represents the mass-to-charge (m/z) values from MS. The intensity (grey levels) of the images corresponds to the MS signal intensities.

Before displaying the images, you will need to import the LC/MS data in MSight.

1. Open MSight by double-clicking on its icon.
2. Import the LC/MS runs by choosing File → Import → MS Data. Locate the LC/MS runs in your disk and select the appropriate file format. Click on the green arrow of the Import MS window to start.

3. Once MSight has finished importing, you can display the LC/MS images, placed automatically by MSight in the Image Pool, by clicking in their icons.
4. Create a new Project by choosing File → New Project and give a sounding name (see Note 2). Right click in the name of the newly created project, choose Create a MatchSet by giving a sounding name (see Note 3).
5. Drag the imported images from the Image Pool to the newly created MatchSet (see Note 4).

3.2. Adapting LC/MS Image Analysis with MSight

Several features of MSight should be modified to improve image visibility and help the navigation across images. In MSight, mass spectra intensities are represented as grey levels. Frequently, the resulting brightness values do not make full use of the available dynamic range, especially when only few *m/z* values have very high intensities. Weak intensities are in this case hardly visible, and the image is almost completely white. This problem can be overcome and faint values accentuated, by stretching the histogram over the available dynamic range.

1. Choose the menu View → MS-Runs → Adjust contrast.
2. In the Adjust contrast window, click as many times as necessary in the different big grey squares to adjust the grey level mapping for the selected images.

Some other useful tools are available in MSight related to data display. For example, MS spectra and chromatograms can be displayed in a 1-D view and regions of the 2-D image can be viewed in a 3-D landscape view. Data can be shown at various resolutions with no information loss using the zoom-in and zoom-out feature. For each desired zoom factor, the image is recalculated on the fly for an optimal display of the data given the available window size. These examples are illustrated in Fig. 1.

3.3. Comparing Data with MSight

A typical workflow to achieve the comparison of MS runs in MSight goes necessarily through a matching procedure with the following steps:

1. Right click on a MatchSet and choose Display in the contextual menu to open the MS-Runs.
2. Click on the menu Edit → Peaks → Detect. A few parameters are available to fine tune the detection and the deisotoping (see Note 5). To preview the effect of the detection parameters, select a region on the image with the rectangle tool.
3. In the case the images show large variations in the retention time axis, select the Landmark tool, and define a few landmarks in common peaks to all images. Then, choose View → Sheet → Align images (see Note 6).

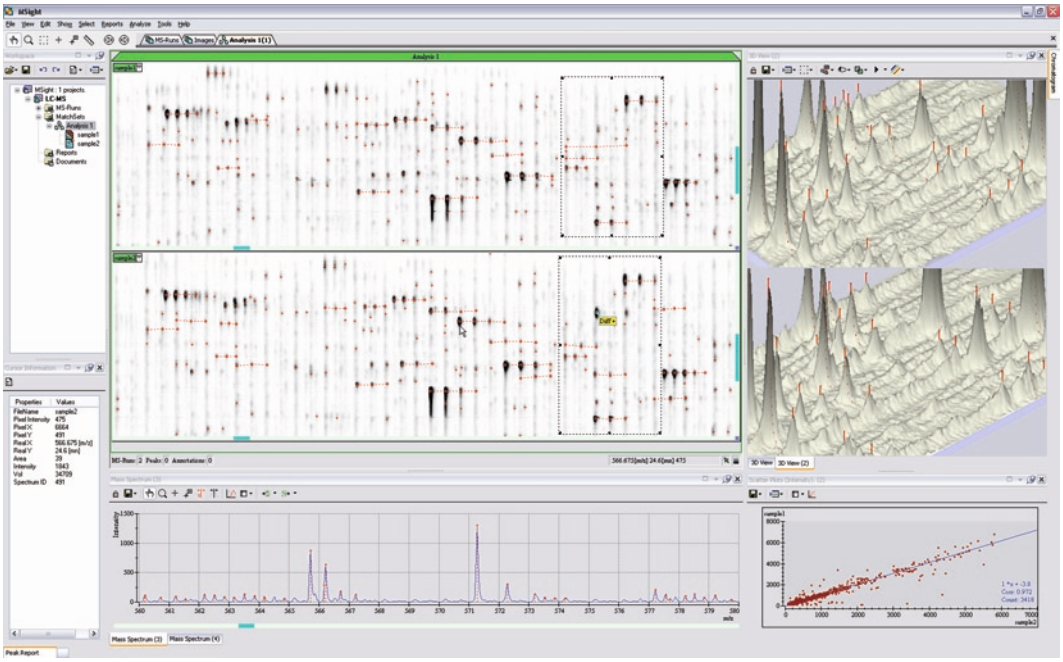


Fig. 1. Illustration of MSight.

4. Finally, to find the corresponding peaks through the runs, click on the menu Edit → Matches → Compute...

The next logical step, following the matching of LC/MS runs, is the differential analysis, i.e. visual or automatic comparison of the runs to determine the differentially expressed peptides and proteins. In the current version of MSight, the variations among runs can be explored through histograms, reports, and scatter plots. The statistical analyses necessary to validate these variations and available within MSight are currently under development and testing.

3.4. Processing LC/MS Data with SuperHirn

SuperHirn processes the typical computational tasks involved in LC/MS data analysis: the detection of peptide features in the mass spectra, the alignment of samples by correcting for shifts in retention time and normalisation of the data. SuperHirn contains some compulsory modules, which include these critical steps, and some other optional modules. The compulsory modules have to be executed in a specific order:

1. First, the MS features are extracted from the LC/MS runs with the command: SuperHirn –FE (see Note 1).

2. Then, the available MS/MS peptide identifications are associated to the MS extracted features, and the pairwise LC/MS similarity analysis is performed to construct a similarity tree of the LC/MS data with the command: SuperHirn-BT.
3. Based on the extracted features, the input LC/MS runs are combined into a MasterMap (see Note 7) by a multidimensional LC/MS alignment process with the command: SuperHirn–CM.
4. And finally, all MS features intensities across all LC/MS runs are normalised and stored, in text format, in a normalised MasterMap with the command: SuperHirn–IN.

At the end of these steps, SuperHirn creates a file containing the normalised MasterMap, i.e. containing the MS feature profiles (see Note 8) present in one or more LC/MS runs.

A MasterMap can be subsequently exploited:

1. by a home-made quantitation tool which will calculate the ratios of the matched peptides and will define those differentially expressed;
2. by continuing the analysis with SuperHirn to cluster profiles and find trends in the present proteins. For example, Rinner et al. (8) have used the MasterMaps to analyse changes in complexes of proteins and find specific partners in networks of interactions.

3.5. Clustering Profiles with SuperHirn

SuperHirn uses the K-means clustering method to group all constructed feature profiles. The starting K cluster centres are randomly chosen from the input feature profiles, and the clustering cycle is repeated until all cluster centres K reach convergence or a maximal number of iterations is achieved (for example, 500 iterations). Each built cluster is stored and subsequently used for targeted profiling analysis.

1. The unsupervised Kmeans clustering analysis of MS feature profiles is performed on the MasterMap with the command: SuperHirn–DP (see Note 9).
2. From the constructed Kmeans clusters, the feature members of the closest cluster to a user-defined target profile are selected (see Note 10). These features are assembled into peptides and proteins and their consensus profile correlation to the target profile is evaluated with the command: SuperHirn–EME (see Note 11).
3. The MasterMap can be further updated by MS2 information in order to assign peptide identifications to MS1 features which have not been annotated in the current LC-MS experiment. Additional MS2 peptide identification data, for

example from other MS instruments or from a user-defined inclusion list, are integrated into the MasterMap by searching for every new MS2 peptide identification its corresponding MS1 feature in the MasterMap. This is done with the command: SuperHirn–ILA.

At the end of each step, SuperHirn produces a text file, that contain the obtained results and can be easily read by end-users and scripting programmes.

3.6. Combining MSight and SuperHirn

In a label-free proteomics analysis, the ultimate objective is to quantify the differentially expressed proteins across multiple LC/MS runs. The current versions of MSight and SuperHirn do not allow performing at once all the necessary steps to reach this objective in a simple and fast way. Ideally, the combination of these two tools would decrease the number of data processing steps, while making the most of the qualities of both tools. For example, the features detected by the precise algorithms of SuperHirn could be integrated into corresponding MSight images to ease the visualisation and analysis of variations among the different runs. Especially in high-throughput settings, quality control is of paramount importance, and visual inspection remains one of the most effective methods. It can only be achieved by a tool, such as MSight, capable of rendering aligned features in different LC/MS runs simultaneously. We strongly advocate its use to check the quality of the final results and to create visual reports with these results. The combination of these two tools is in the pipeline of the current developments and will be delivered to the proteomics community in the near future.

4. Notes

1. A feature of SuperHirn is the collection of *m/z* peaks that derive from the same molecular ion, as a result of 13C isotope distribution and multiple charge states distributed over multiple consecutive MS scans as a result of chromatographic elution.
2. A project allows you to organise your MS-Runs in a logical way (a study or research project for example), to specify how the MS-Runs are to be matched together, and to define your classes (or groups) for statistical analysis. It includes all MS-Runs, peaks, matches, annotations, and other information produced and analysed during the course of a specific LC/MS run study. You can create or add many projects in the Workspace.
3. A project can include one or more match hierarchies, each of which contains a Match folder and a Classes folder. The

Match folder describes how MS-Runs or populations of MS-Runs, called match sets, should be matched together. The Classes folder is where the biological question is stated, through the definition of classes of MS-Runs to be compared.

4. Within each match set, the MS-Run image or match set that has a red marker and appears first in the list is used as the reference in the matching process. To change the match reference, drag the desired MS-Run image or match set onto the name of its parent match set so that it moves into the first position. The reference for each match set must be carefully chosen. This is because automatic matching compares the peaks in the reference to those in the other images. If a peak is absent from the reference, it cannot be matched automatically (although it can be matched manually).
5. The peak detection algorithm looks for areas of high intensity peaks to delineate their shapes. The deisotoping step then looks for the monoisotopic peaks of the same molecule, links them together (dashed lines connect isotopes), and determines ion charge states.
6. To align images, the software needs to know which positions in the different images correspond to each other, that is, represent the same peak. This is done by defining landmarks. The alignment algorithm then deforms the images to superimpose the landmarks.
7. The MasterMap is a proprietary format of SuperHirn which contains a consensus of all matched peaks across all LC/MS runs. It is formatted as a table with *m/z* values (in a given retention time with a given charge) as rows and the LC/MS runs as columns. The intersections of rows and columns are the intensity values.
8. An MS feature profile is a vector containing the intensities of a given *m/z* (in a given retention time with a given charge) in all LC/MS where this *m/z* was detected. For example, the *m/z* 402.2348 with retention time 8.52 and charge +1 could have this <1927500.38 8734537.00 0 490350.91 4125343.50 622670.44> as an MS feature profile. It means that it was detected in 5 out of 6 LC/MS runs.
9. A crucial factor in K-means clustering is the number of start cluster centres. When the number of differentially expressed proteins in the sample is known, which was the case in the samples used to describe SuperHirn (7), the number of start clusters will be minimum equal or higher than this number, and these features are called target profiles. When the number of differentially expressed proteins in the sample is not known, it has to be estimated or guessed by the user.

10. A user-defined target profile is a known protein which would typically have different abundances in different LC/MS runs.
11. For the profiles similar to the target profiles, the *m/z* values and retention times of the corresponding features can be written into a so-called inclusion list (9), which allows further targeted MS2 measurements in order to identify the peptides behind these features.

Acknowledgment

MSight was partially funded by EU project LOCCANDIA (FP6-2004-IST-5 # 034202).

References

1. Bantscheff M, Schirle M, Sweetman G, Rick J, Kuster B (2007) Quantitative mass spectrometry in proteomics: a critical review. Anal Bioanal Chem 389:1017–1031
2. America AH, Cordewener JH (2008) Comparative LC-MS: a landscape of peaks and valleys. Proteomics 8:731–749
3. Schulz-Trieglaff O, Pfeifer N, Gropl C, Kohlbacher O, Reinert K (2008) LC-MSsim – a simulation software for liquid chromatography mass spectrometry data. BMC Bioinform 9:423
4. Mueller LN, Brusniak MY, Mani DR, Aebersold R (2008) An assessment of software solutions for the analysis of mass spectrometry based quantitative proteomics data. J Proteome Res 7:51–61
5. Palagi PM, Walther D, Quadroni M, Catherinet S, Burgess J, Zimmermann-Ivol CG, Sanchez JC, Binz PA, Hochstrasser DF, Appel RD (2005) MSight: an image analysis software for liquid chromatography-mass spectrometry. Proteomics 5:2381–2384
6. Appel RD, Palagi PM, Walther D, Vargas JR, Sanchez JC, Ravier F, Pasquali C, Hochstrasser DF (1997) Melanie II – a third-generation software package for analysis of two-dimensional electrophoresis images: I. Features and user interface. Electrophoresis 18:2724–2734
7. Mueller LN, Rinner O, Schmidt A, Letarte S, Bodenmiller B, Brusniak MY, Vitek O, Aebersold R, Muller M (2007) SuperHirn – a novel tool for high resolution LC-MS-based peptide/protein profiling. Proteomics 7:3470–3480
8. Rinner O, Mueller LN, Hubalek M, Muller M, Gstaiger M, Aebersold R (2007) An integrated mass spectrometric and computational framework for the analysis of protein interaction networks. Nat Biotechnol 25:345–352
9. Schmidt A, Gehlenborg N, Bodenmiller B, Mueller LN, Campbell D, Mueller M, Aebersold R, Domon B (2008) An integrated, directed mass spectrometric approach for in-depth characterization of complex peptide mixtures. Mol Cell Proteomics 7:2138–2150

Part VI

Modelling and Systems Biology

Chapter 25

Spectral Properties of Correlation Matrices – Towards Enhanced Spectral Clustering

Daniel Fulger and Enrico Scalas

Abstract

This chapter compiles some properties of eigenvalues and eigenvectors of correlation and other matrices constructed from uncorrelated as well as systematically correlated Gaussian noise. All results are based on simulations. The situations depicted in the settings are found in time series analysis as one extreme variant and in gene/protein profile analysis with micro-arrays as the other extreme variant of the possible scenarios for correlation analysis and clustering where random matrix theory might contribute. The main difference between both is the number of variables versus the number of observations. To what extent the results can be transferred is yet unclear. While random matrix theory as such makes statements about the statistical properties of eigenvalues and eigenvectors, the expectation is that these statements, if used in a proper way, will improve the clustering of genes for the detection of functional groups. In the course of the scenarios, the relation and interchangeability between the concepts of time, experiment, and realisations of random variables play an important role. The mapping between a classical random matrix ensemble and the micro-array scenario is not yet obvious. In any case, we can make statements about pitfalls and sources of false conclusions. We also develop an improved spectral clustering algorithm that is based on the properties of eigenvalues and eigenvectors of correlation matrices. We found it necessary to rehearse and analyse these properties from the bottom up starting at one extreme end of scenarios and moving to the micro-array scenario.

> *As far as the laws of mathematics refer to reality, they are not certain, and as far as they are certain, they do not refer to reality.*
>
> *A. Einstein*

1. Introduction

In principle, the statements made here on the micro-array data scenario and functional gene groups also apply to protein-arrays. We generally take a more abstract point of view to separate experimental and statistical issues that often get mixed up.

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_25, © Springer Science+Business Media, LLC 2011

The main two situations treated here are (a) more measurements than variables and (b) less measurements than variables. A third (nuisance) scenario of missing measurements which appears in the context of proteins is treated separately in Ref. (3) since not of concern here.

The term *spectral* indicates the calculation and exploitation of matrix spectra, i.e. eigenvalues and eigenvectors. We review spectral properties of correlation-like matrices from a general point of view. One question to answer: What meaning do the eigenvalues and eigenvectors *exactly* have, what information can be extracted that can be used to improve clustering, for example? The established association of *the* large eigenvalue and respective eigenvector with some kind of "dominant" mode (4) in the underlying data seems to be just half of the story.

From a mathematically abstract point of view, the situation and task could be interpreted as the following: We are presented with stochastic variables ξ_n, $n = 1,\ldots, N$. The values taken by these variables are indexed by $t = 1,\ldots, T$ by writing $\xi_n(t)$. This allows the association with time series while this labelling may refer to the experiment number or any other label that expresses meaningfully that variables $\xi_1(t),\ldots, \xi_N(t)$ belong to one measurement or experiment. This pedantry is necessary because the interpretation and the choice of methods crucially depend on the mappability between the mathematical object and the real world. In the former, there is no concept of time and its introduction must be well defined and justified. The entire data can be arranged in a matrix $\mathbf{M}$ of dimension $N \times T$. Assuming that the average is zero, the Pearson estimator for the covariance matrix (C_{ij}) is given by

$$c_{ij} = \frac{1}{T}\sum_{t=1}^{T} \xi_i(t)\xi_j(t), \tag{1}$$

The covariances of all pairs can be collected in a symmetric matrix

$$\mathbf{C} = \frac{1}{T}\mathrm{M}\mathrm{M}^{\mathrm{T}}. \tag{2}$$

The covariance or correlation matrix $\mathbf{C}$ is often associated with the Wishart matrix for which Marčenko and Pastur derived an analytic spectrum in the large size limit if the variables $\xi_n(t)$ are independent and identically distributed with the condition of finite moments (8). The equivalence between the (realisation of a) correlation matrix and the Wishart matrix cannot always be taken for granted as we show later.

A typical task is to extract sets of variables that form correlated groups, or rather groups that have something in common. The above-mentioned correlation coefficient is just one of many possible "linkages" between (real valued) random variables or even other random objects. It is to view clustering as a special case

of spectral reconstruction (approximation) of matrices or related networks (1). The notion of correlation can be extended to any coefficient that measures a link between two random objects in terms of a real number for which a suitable pair-wise distance can be defined. The definition of a correlated group is therefore somewhat arbitrary, likewise is the resulting clustering of different methods more or less different. In real world data, there is usually no correlation in the mathematical sense, but possibly something very similar and interpretable as correlation. Many methods act on matrix **C** to extract information. Specialised methods make model assumptions on the type of correlation (or link value) and are thus *empirically optimised* to cluster the random variables that work best for the given source of the random variables.

Additionally, the mathematically abstract context of realisations of random variables at equidistant points in time to which correlation measurement is often mapped to is not justified in some cases. It does not hold, for example, if not all random variables provide a realised value for each time index. This is the case in protein data and also in high-frequency financial data where waiting times produce zero increments in the stock value between samplings. Neither are the (differential) expression values extracted from micro-arrays easily justified to be interpreted as standard random variables. For very similar experimental settings, very similar values must be expected, up to some noise, while at certain levels of cell stress or whatever the index t stands for, jumps may occur.

Consider the situation shown schematically in Fig. 1. The time series change only once and the same "time index" in panel

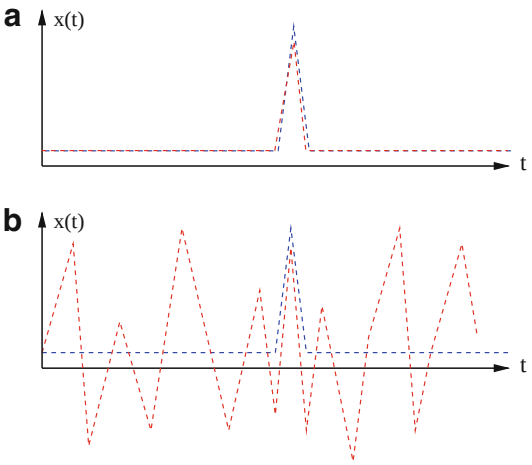


Fig. 1. (*Colour online*) Schematic time series with straight lines between the data pairs. In (**a**), both time series show a peak at the same time. In (**b**), the peaks are still identical while one time series seemingly fluctuates at random. The correlation coefficients calculated from both situations are identical if the increments lie on the same grid. The coefficients are likely to be very similar even with continuous-time random walks and meaningful interpolation schemes.

A. In panel B, time series Red fluctuates a lot and has, by coincidence or not, an identical spike together with Blue. A correlation measurement combined with some typical interpolation technique would produce identical or at least very similar correlation coefficients for both situations. It depends strongly on the system, application, and questions asked if it is appropriate to call either situation correlated or not. It is most likely A that depicts a significant connection between the two time series. Note that this likelihood increases with the total duration or number of data points! We must therefore realise that the regions in time having zero increments do contain information, in particular if their time scale is of the similar order of magnitude as other time scales in the respective situation, for example, the total duration of the measurement. Ergo:

> Any post-processing that only considers the correlation coefficients disregards time and produces in such a case a joint probability density in the (dis-)similarity matrix that does not exhaustively reflect the connection between the time series, i.e. looses information.

It is demonstrated later that with time series with behaviour as described in Fig. 1, or respective stylised facts, the data should not be ignored but used in the reconstruction of "modes" and separation of correlated clusters that are otherwise not separable. *Note, that the term "mode" is not mathematical and mostly intuitive if used in real world data measurement.*

Clustering with matrices and their eigenvalues and vectors is well established in graph theory for a long time under the term "spectral clustering". A good tutorial is Ref. (12). In short, it is based on a certain dissimilarity measure matrix **L**, while definitions sometimes disagree, and it uses the eigenvectors of the smallest k eigenvalues. This value has the same meaning as in k-means clustering, i.e. the a priori estimate on the number of clusters. **L** is symmetric and called Laplacian and one frequent definition is

$$\mathbf{L} = \mathrm{diag}(d_i) - \mathbf{C}, \tag{3}$$

where **C** is the unweighted (positive) adjacency matrix. $\mathrm{diag}(d_i)$ is the diagonal matrix of vertex degree

$$d_i \sum_{j=1}^{N} C_{ij}, \tag{4}$$

i.e. the sum over all correlations with node i. Graph theory denotes the "objects of interest" with the term nodes. It can

be replaced with time series or gene/protein accordingly. One can consider $\mathbf{L}$ to contain a dissimilarity measure via the negated $\mathbf{C}$ which in turn is analogous to the absolute value $|\mathbf{C}|$ sometimes used in a distance measure. For example, the comparison of clustering methods in Ref. (9) uses a dissimilarity measure that is close to the one used later. In the end, it is unlikely that mathematical reasoning leads to the best choice of distance measure, as explained above. In the following examples and figures, it is demonstrated that with some (dis-) similarity measure it makes sense to consider also other eigenvectors than the large eigenvalues's eigenvectors. In spectral clustering, one is free to choose a suitable clustering method, for example k-means, which then performs the clustering using these eigenvectors. This also means that the number of clusters must be guessed beforehand. The motivation to consider here also a dissimilarity measure is to keep track of what type of matrix other non-spectral methods as k-means or PAM (9) use. The latter two work directly on the coefficients of the matrix.

The improved spectral clustering uses the correlation matrix as a similarity measure since it contains no less information than any dissimilarity matrix. Moreover, a theory exists on its random case spectrum. It seems that the use of the Laplacian matrix $\mathbf{L}$ in standard spectral clustering is mostly to achieve plausibility since it matches with a mathematical construction in graph theory. Furthermore, for a "mode" carrying a correlation information, we also have small eigenvalues leaking out of the Marčenko–Pastur law of uncorrelated data (8). This is true for the similarity (correlation) matrix and, in an analogous way, also for the dissimilarity matrix used here and defined later. Since we assume that we are faced with a "noisy" situation, we must use all information we can extract. Since these small eigenvalues and associated eigenvectors are likely to contain redundant information about the correlated cluster, it is appealing not to ignore this information. The naive mapping of a real world situation to simultaneous realisations of random variables is often not easy to be justified and is mostly argued for because of reasonable results. An example is liquid together with illiquid stocks in finance. There, the data itself can be used in the reconstruction of the correlations.

The following sections construct artificial situations that are "extreme" for didactic purposes in the sense that they are not realistic but allow to recognise features in the eigenvalue and eigenvector spectra that could be used in the better exploitation of the information content given in a more noisy and more realistic data set.

2. Scenario 1: Correlated Noise with Many Variables and Many Measurements per Variable

2.1. One Correlated Cluster

The scenario demonstrated here mimics synchronous financial data analysis, i.e. at least as many measurements as variables:

- $T = 200$ number of realisations per random variable
- $N = 200$ number of random variables
- $N_c = 1$ number of independently correlated groups of variables
- $N_1 = 20$ number of correlated variables in group 1 (only one here)
- $\xi_n(t)$ Gaussian noise data set n where $t = 1, \ldots, T$.
- Type of correlation within group i:

$$\xi_n(t) = \xi_n(1-c) + \Xi(t)c \qquad (5)$$

  Ξ is a prefixed "parent" noise vector specific for the correlated group. $c \in [0, 1]$ is a correlation coefficient.
- $c = 0.93$ (very high correlation)

The choice of $Q = T/N = 1$ is to avoid any factor Q if it appears in some normalisation. The mathematical/numerical construction of the artificial correlation is not so relevant since the realistic case does not provide a mathematical correlation coefficient either. Figure 2 shows the correlation matrix created from the series ξ_1 to ξ_N. For identification, the first 20 are correlated. Also shown is the more realistic disordered situation if the correlated data sets

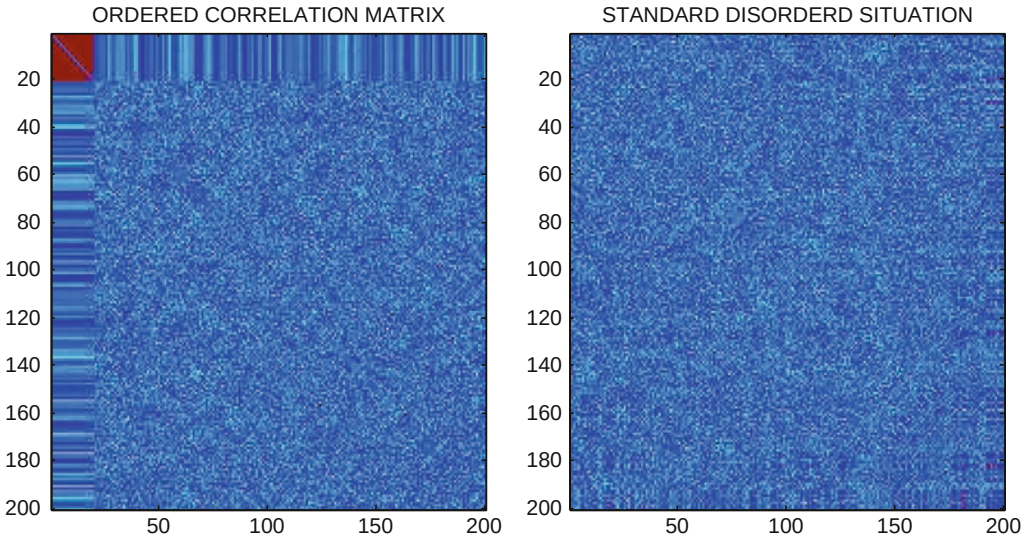


Fig. 2. (*Colour online*) Correlation matrix of uncorrelated noise with one cluster of 20 artificially correlated variables. The right panel is reshuffled to imitate the standard disordered situation in reality, where the *dark (red) dots* of high coefficients are randomly distributed.

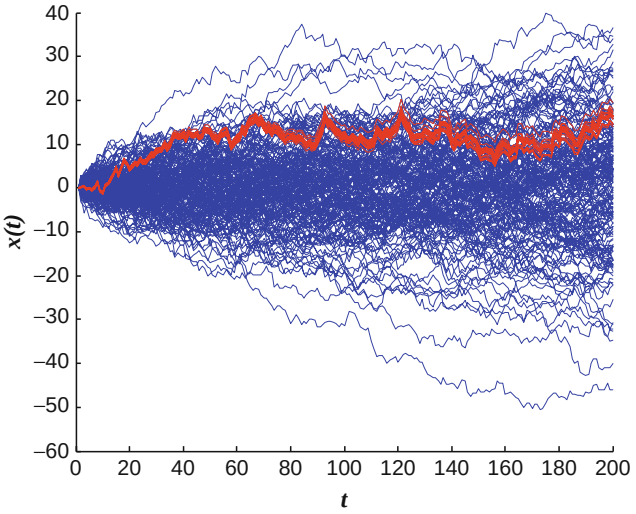


Fig. 3. (*Colour online*) Random walks created from the noise in vectors ξ_1 to ξ_N. The correlated group is marked in *grey* (*red*).

are unknown, i.e. shuffled. Figure 3 shows the respective equal-time random walks (RWs) $x_i(t)$, $t \in \mathrm{N}_+$ on a regular grid that are created from the realisation of the random variables. For the purposes of this chapter, this is entirely sufficient and the data points $x(t_1)$, $x(t_2)$,... are connected with straight lines. In Fig. 3, the correlated group is drawn red.

In addition to the correlation matrix **C**, the following figures also show the results using a dissimilarity matrix **D**. The definition of dissimilarity is a bit arbitrary. The measure used here is

$$\mathbf{D} = 1 - |\mathbf{C}|. \tag{6}$$

Figure 4 shows the main part of the eigenvalue spectra of both matrices. The numbering of eigenvalues is by size, i.e.:

$$\lambda_1 < \ldots < \lambda_N. \tag{7}$$

Some features are outside the plot range. Note that λ_-^C denotes the lower bound of the Marčenko–Pastur domain which is zero in this case with $Q = 1$. In the finite size situation with low correlation parameter, the classification of eigenvalues as belonging to the correlated group is not unique due to the overlap of the distributions with the informationless bulk of the spectrum. Likewise, the expected size of eigenvalues fluctuates. We therefore use the order notation with $O(\cdot)$ to indicate that an eigenvalue is expected to have the value $O(x)$ or the number of eigenvalues in a distinct group is expected to be $O(N)$. This is not to be confused with the usual meaning of order notation. For larger values of Q, N and

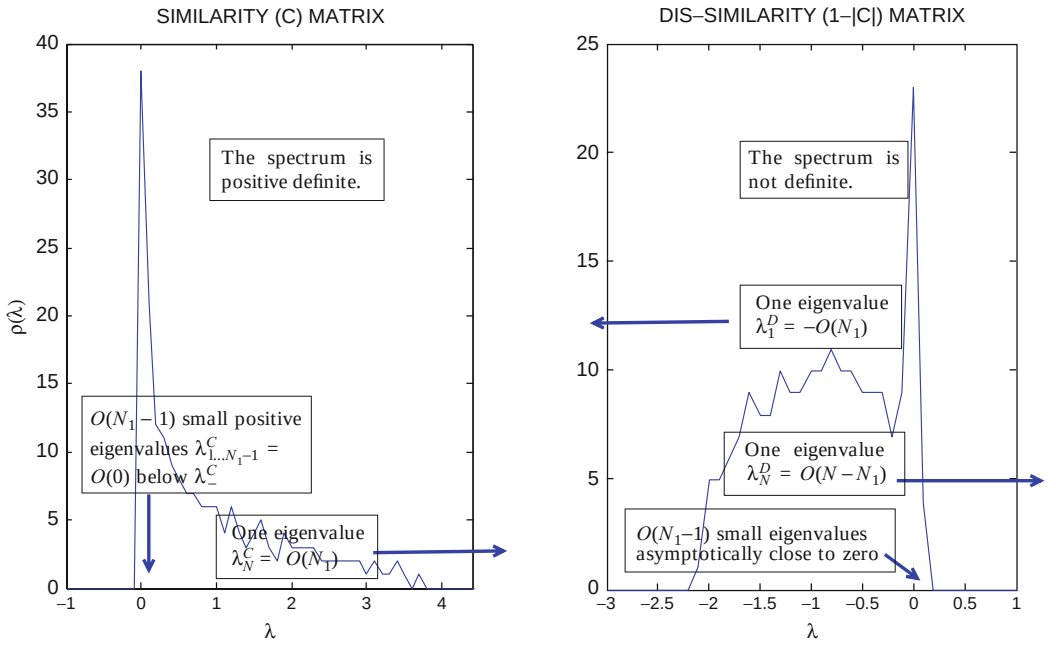


Fig. 4. (*Colour online*) The features of the spectra of the similarity matrix and dissimilarity matrix are similar in their content of information on the number of correlated data sets. Due to "conservation of weight", we have the above (approximate) relations for the extreme values and number of eigenvalues that are close to zero. In the limit of high correlation and $T \ll N$, these become equalities. In this example, we have $N_1 = 20$ correlated among $N = 200$ random variables. For the dissimilarity matrix, we find $\lambda_N^D = 186.15$ and $\lambda_1^D = -17.57$ only approximately correspond to 200 and 19, respectively. $\lambda_N^D + \lambda_1^D - N \approx 0$ holds well, however. The histograms are not normalised on purpose to convey the absolute counts.

correlation coefficient the groups of, e.g. zero eigenvalues, can be distinguished very well. The histograms in the eigenvalue figures are again deliberately not normalised to convey the absolute counts. The Laplacian's **L** smallest eigenvalue is always $\lambda_1^L = 0$ by construction (1).

Figure 5 shows the eigenvector matrices of the correlation matrix as well as of the dissimilarity matrix **D**. The realistic (shuffled) situation is also shown. Columns are eigenvectors with column number corresponding to eigenvalue index. The ordering is entirely arbitrary as long as the pairs of eigenvalue and eigenvector are maintained well.

In the situation created here with one correlated cluster, we find the following (partly empirical) properties.

A. Properties of the eigenvalues and vectors of similarity (correlation) matrix **C:**

1. The spectrum is strictly positive definite with a lower Marčenko–Pastur bound λ_-^C.
2. Conservation law in the limit of high correlation:

$$\lambda_N^C - N_1 = 0. \tag{8}$$

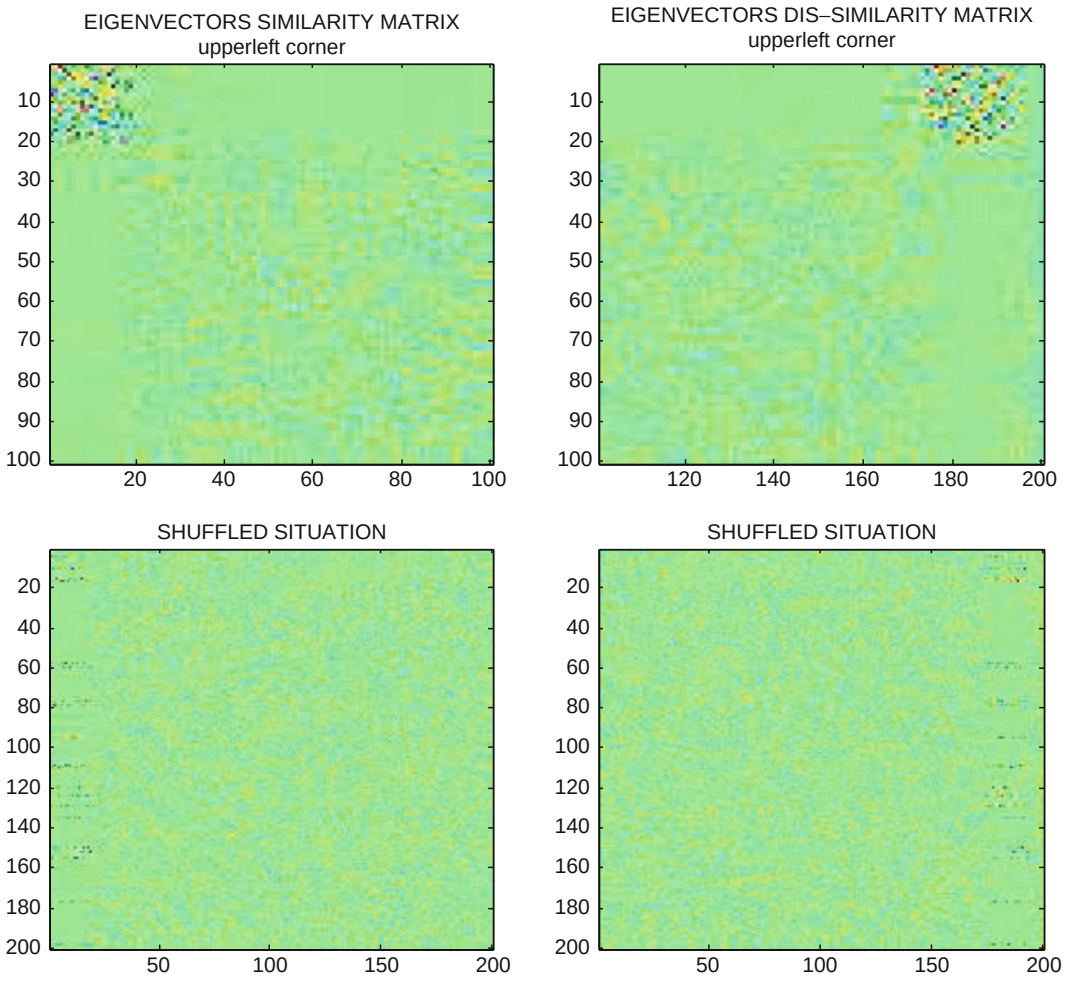


Fig. 5. (*Colour online*) Colour (*greyscale*) coded eigenvector column matrix of the correlation matrix **C** and of the dissimilarity matrix **D**. Shown are the magnified left and right corners. The $N_1 - 1$ eigenvectors belonging to the small group of eigenvalues evoked by the correlated cluster can be identified as a square of $N_1 \times N_1 - 1$ strongly fluctuating elements as compared to the informationless part of the eigenvectors. Due to the normalisation of length 1, the remaining elements in these vectors are close to zero (*column of uniform grey/green area*). Also note that all other eigenvectors are essentially zero in the first $N_1 - 1$ rows as well. The shuffled situation is also shown below.

This not only holds if $1 \ll N_1 \ll N$ but as long as $N \ll T$ and $c \approx 1$.

3. One large eigenvalue $\lambda_N^C = O(N_1)$.
4. The eigenvector V^C belonging to the only large eigenvalue λ_N^C contains N_1 relevant large elements in the indices of the correlated cluster proportional to their contribution to the respective "mode".
5. A group of $O(N_1 - 1)$ small positive eigenvalues λ_1^C to $\lambda_{N_1-1}^C$ below λ_-^C. This group is well distinguishable due to a clear gap to the bulk above λ_-^C.

6. The eigenvectors $v_1^C,\ldots,v_{N_1-1}^C$ belonging to this group of small eigenvalues are well distinguishable by N_1 elements of higher variance in the element indices that belong to the correlated cluster. See Fig. 6. With increasing cluster correlation and matrix size, the other elements of these vectors get asymptotically close to zero.
7. Due to conservation of weight and with increasing cluster correlation and matrix size all other eigenvectors $v_{N_1}^C,\ldots,v_{N-1}^C$ are asymptotically close to zero in the indices that *do not* belong to the correlated cluster. These small elements fluctuate below the variance of the random majority of the eigenvector matrix. Again see Fig. 6.
8. The random bulk of the eigenvector matrix may indeed fluctuate beyond the magnitude of the elements in V^C. This can be observed in Fig. 6.

B. Properties of the eigenvalues and vectors of dissimilarity matrix **D**:

1. The uncorrelated spectrum is not definite due to one large positive eigenvalue. However, all other eigenvalues have a strictly negative upper bound $\lambda_+^D < 0$.
2. Conservation law in the limit of high correlation:

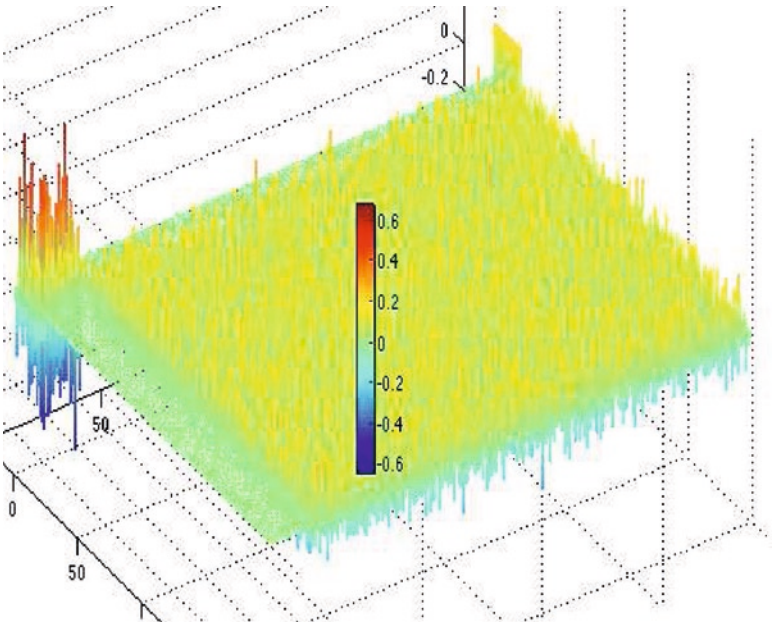


Fig. 6. (*Colour online*) Impressionistic view of the eigenvector matrix of the similarity matrix **C** from Fig. 5. This perspective gives an impression of the overall structure and magnitude of the eigenvectors. Vector V^C can be recognised by the ridge of equally large positive elements at the right upper corner.

$$\lambda_N + \lambda_1 - N = 0. \tag{9}$$

This not only holds if $1 \ll N_1 \ll N$ but as long as $N \ll T$ and $c \approx 1$.

3. One left large eigenvalue $\lambda_1^D = -O(N_1)$.
4. The eigenvector V_{left}^D belonging to the left large eigenvalue $\lambda_1^D = -O(N_1)$ contains N_1 large elements in the indices of the correlated variables.
5. A group of $O(N_1 - 1)$ small eigenvalues λ_1^D to $\lambda_{N_1-1}^D$ asymptotically close to zero with increasing cluster correlation. This group is well distinguishable in the large size limit due to a clear gap to the bulk below λ_-^D.
6. The eigenvectors $v_1^D, \ldots, v_{N_1-1}^D$ belonging to this group of small eigenvalues are well distinguishable by elements of high variance in the indices that belong to the correlated cluster and low variance below the random bulk variance. With increasing cluster correlation and matrix size, the other elements are asymptotically close to zero. See Fig. 6.
7. One right large eigenvalue $\lambda_N^D = O(N - N_1)$
8. The eigenvector V_{right}^D belonging to v right large eigenvalue $\lambda_N^D = O(N - N_1)$ is $(1, \ldots, 1)/\sqrt{N}$ plus some partly systematic fluctuation due to finite size.
9. As opposed to the similarity matrix, the eigenvectors with large fluctuations in the group indices are not situated at the far left end opposite to the vector V_{right}^D but a few columns closer. This property has no explanation yet.
10. The gap between the informationless bulk of the eigenvalues and the group of small eigenvalues around zero is wider than in the spectrum of the similarity matrix. It remains to be seen if this property is an advantage.
11. The random bulk of the eigenvector matrix may indeed fluctuate beyond the magnitude of the elements in VC. This can be observed in Fig. 6.

The list of properties is not finished with the above items. Of particular interest is the linear combination of RWs created from the correlated noise and eigenvector information since the resulting "modes" allow recognition of similarities by eye that otherwise remains hidden in the series of increments. The artificially correlated cluster is produced via a fixed set of increments, called parent noise, that is reused for the production of all members of the cluster by adding more or less noise depending on the correlation parameter c, see Eq. 6. Figure 7 shows in black the RW obtained by this "parent" noise that lies within the correlated cluster. The following notation, that is maintained later, assumes for simplicity's sake vectors if the entire range $t = 1, \ldots, T$ is referenced,

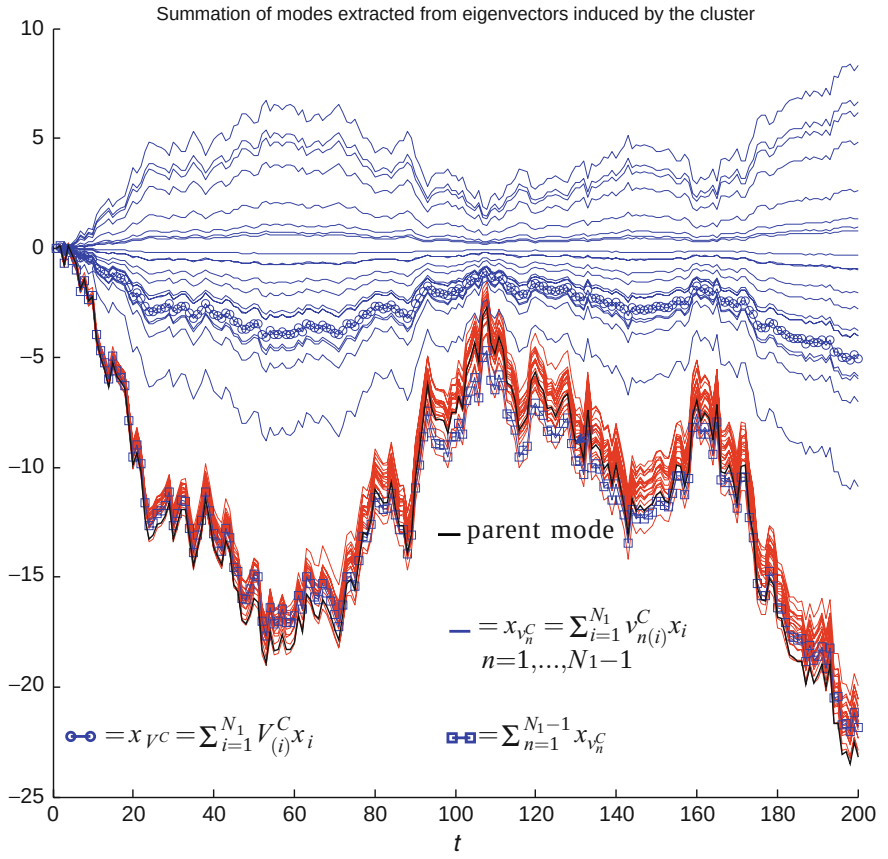


Fig. 7. (*Colour online*) From the coefficients of eigenvectors that belong to eigenvalues outside the random bulk of the spectrum several "modes" can be reconstructed that are identical up to some statistical fluctuations. The parent mode is created with one pre-fixed set of noise used for generation of the correlated bunch (*red*), see Eq. 6. The legend gives the respective summation formulas. The notation uses the short cut $x=(x(1),\ldots,x(T))$ the subscript (i) denotes the ith vector element.

i.e. $x=(x(1),\ldots,x(T))$. We start with the similarity matrix **C** and its two main objects of interest:

- The linear combination of RWs using the (in this case first) N_1 relevant large elements of eigenvector V^C as coefficients:

$$x_{V^C} = \sum_{n=1}^{N_1} V_{(n)}^C x_n, \tag{10}$$

where $V_{(n)}^C$ denotes vector element n which is also the index of the random variable. x_n is the RW-vector constructed from the respective increments:

$$x_n(t) = \sum_{\tau=1}^{t} \xi_n(\tau). \tag{11}$$

Even though the sum runs over the first N_1 vector elements of V^C, the correlated mode is reproduced well. Figure 7 shows the curve obtained via Eq. 10 as blue circles.

- The linear combination of RWs using the N_1 relevant elements of the $N_1 - 1$ eigenvectors v_n^C belonging to the small group of eigenvalues:

$$x_{V_n^C} = \sum_{i=1}^{N_1} v_{n(i)}^C x_i, \; n = 1, \ldots, N_1 - 1 \tag{12}$$

This gives $N_1 - 1$ different "modes". Figure 7 shows that these as several blue lines. Even though the sum uses only the first N_1 vector elements and RWs, the modes are reproduced well.

- Of possibly greatest interest could be the sum of all modes according to Eqs. 25.10 and 25.12:

$$\sum_{n=1}^{N_1 - 1} x_{v_n^C} \text{ blue squares in Fig. 7} \tag{13}$$

and also the sum Eq. 12 plus Eq. 10:

$$\sum_{n=1}^{N_1 - 1} x_{v_n^C} + x_{V^C} \tag{14}$$

- Finally, it appears that one of the modes $v_n^C, n = 1, \ldots, N_1$ contains the "zero-mode". Empirical examples indicate that up to some fluctuation that is decreasing with increasing T one possibly special v_n^C is essentially zero on the entire axis.

The entire exercise on linear combination of modes can be repeated with the eigenvectors of the dissimilarity matrix, but this has to be skipped for now since a detailed comparison requires separate study.

2.2. Two Correlated Clusters

Figures 8 and 9 demonstrate how the situation of two correlated clusters is coded in the eigenvectors of the respective correlation matrix. With two clusters, an additional feature appears: *both* eigenvectors belonging to the two large eigenvalues code the mode within their elements. The clusters are chosen such that the first (1–20) and the last (180–200) noise vectors are correlated to simplify identification by eye in the eigenvector matrix. First, observe the following characteristics of this particular realisation of the respective RWs in Fig. 9:

1. We have two eigenvectors that code the two modes. As expected, they are located at the right of the eigenvector matrix because they belong to two large eigenvalues of essentially equal value up to some noise due to the finite size of the situation.

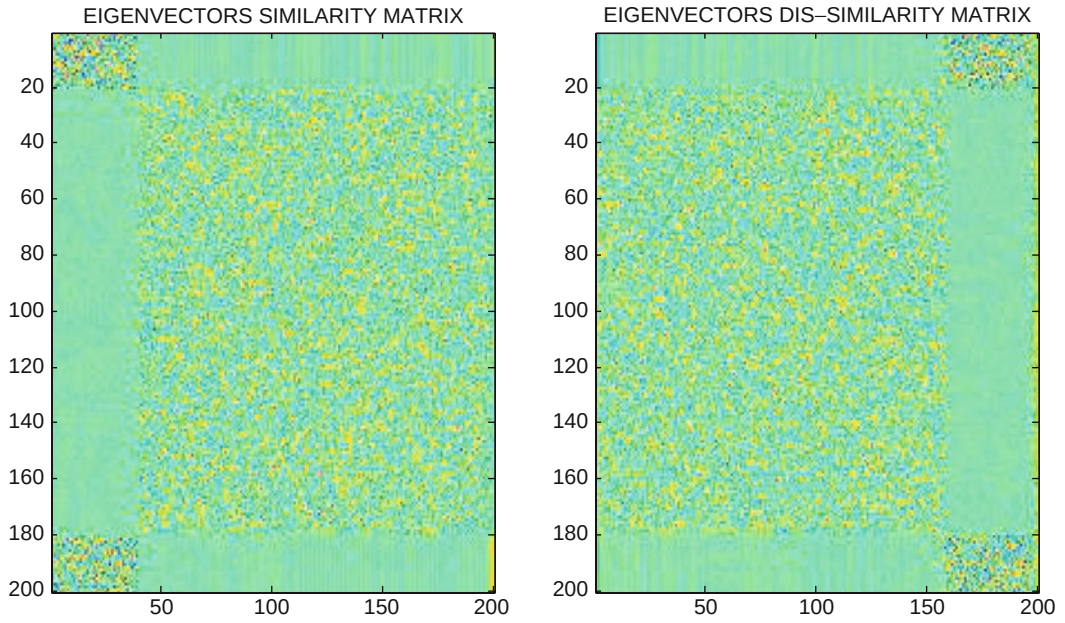


Fig. 8. (*Colour online*) Colour (*greyscale*) coded eigenvector column matrix of the correlation matrix C and of the dissimilarity matrix D.

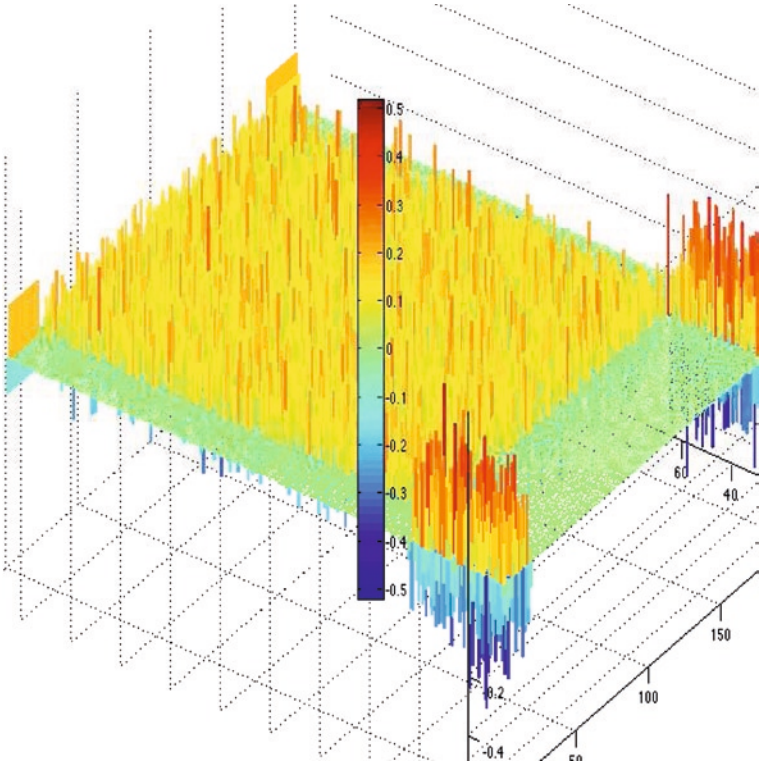


Fig. 9. (*Colour online*) Colour (*greyscale*) coded eigenvector matrix of the correlation matrix **C**. This perspective shows that the two eigenvectors belonging to the two large eigenvalues are non-zero in the indices of both correlated random variables. The regions at the "other end" of the matrix also "code" the respective modes.

2. Both eigenvectors are non-zero in the indices of *both* of the correlated clusters.
3. The sign of these non-zero elements is either entirely positive or entirely negative. The sign depends on the realisation, yet not all combinations are possible. Two signs in one group, e.g. 1–20, must be equal, the other in 180–200 two are then opposed.

Figure 10 demonstrates in detail which eigenvector matrix elements recover the mode. The matrix elements are used for linear combinations of the corresponding RWs according to Eq. 13 or Eq. 14. The elements are marked in the schematic overview Fig. 11 and can be recognised in Fig. 9 as columns. It is apparent that the respective recovered modes are clusterwise numerically identical up to an overall factor. The reconstructions approximate the parent cluster mode Eq. 6. With correlation coefficient $c_1 = c_2 = 0.8$, i.e. equal for both clusters, we do not expect a perfect recovery of the parent mode.

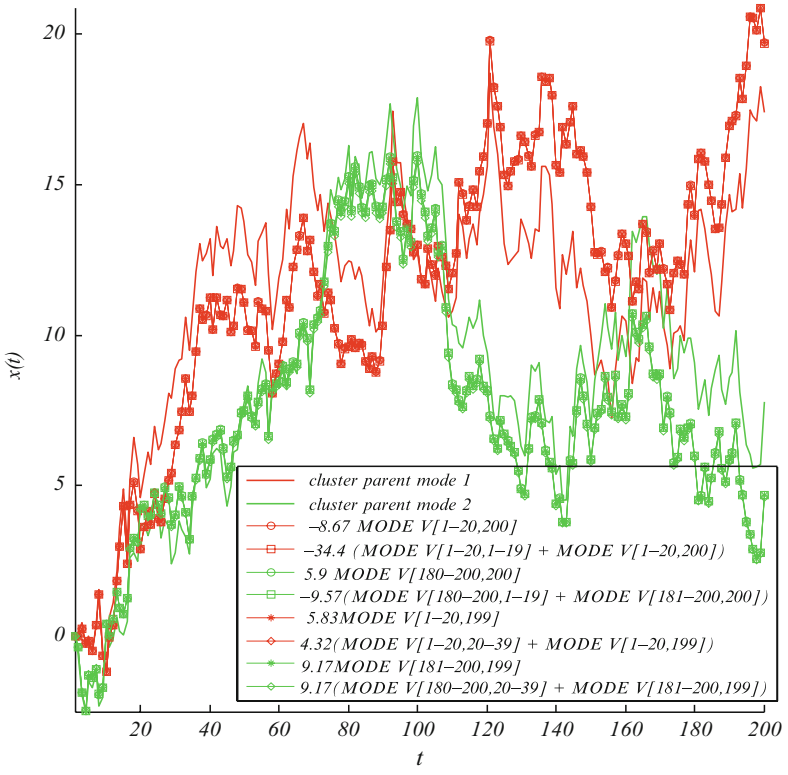


Fig. 10. (*Colour online*) The two "parent" modes used for the construction of two artificial clusters and the reconstructed modes are shown as *grey and light-grey* (*red and green*) continuous lines. The notation MODE V[$i-j$, $k-l$] denotes the eigenvector matrix elements used in the mode reconstruction by linear combination of the respective RWs according to Eq. 13 or Eq. 14. The factors are empirical. In this example, we have 20 possibilities to reconstruct exactly the same mode up to a linear factor. These reconstructions are identical but only approximate the parent mode.

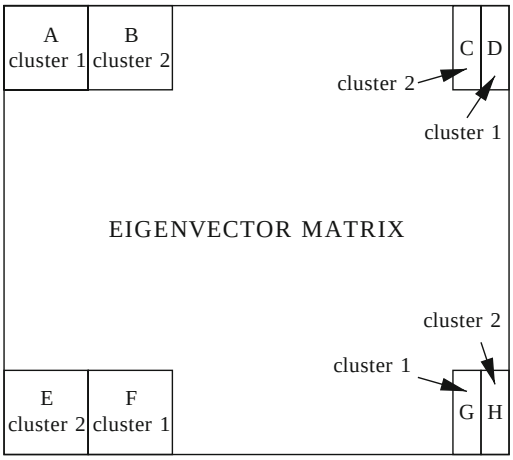


Fig. 11. (*Colour online*) In the case of two correlated clusters of random variables in the indices (1–20) and (180–200), the regions in the eigenvector matrix of the correlation matrix whose elements code the respective mode are known beforehand up to an arbitrary ordering of clusters 1 and 2. All eigenvectors that stem from a large eigenvalue contain all modes. In the realistic, or shuffled, situation all rows would be reordered in a random way, but the overall order within the columns remains.

3. Improved Spectral Clustering

We make use of the fact that the small eigenvalues belonging to a mode also code the mode up to a factor. In contrast to, for example, *k*-means the following procedure has the advantage of estimating the number of clusters, and it also allows for cluster overlaps.

Algorithm:

1. Decide on the number of significant modes (or clusters) by counting the number k of "large" eigenvalues, number of significant eigenvectors, etc. using a criterion of choice.
2. Take the corresponding k eigenvectors. For all k vectors do:
3. "Plot" the mode coded in V_k via Eq. 10.
4. Decide for V_k which η_k largest elements are significant, e.g. by comparing with the Porter-Thomas law or via contribution to the resulting mode, etc.
5. Find among all vectors the corresponding set $\{v_{ki}\}$ of size $\eta_k - 1$ (region A, B, etc.) There are different ways: E.g. try to recover the mode coded in V_k doing many fits with linear combinations as in Eq. 12.
6. Test if a different number than $\eta_k - 1$ gives the best fit by repeating step 5 trying an additional v^*_{ki} (consequently also an additional vector element in all v_{ki} as well as v^*_{ki}). Then, η_k, the number of significant elements in the large eigenvalue's eigenvector, can be adjusted.
7. Remember these eigenvector element indices as the cluster $\{N_k\}$.
8. Repeat step 2.
9. We are left with k sets of numbers. Some may overlap.

The first, main, improvement is located in steps 5 and 6. We use η_k eigenvectors to determine the number of correlated random variables in cluster k instead of only the one eigenvector V_k belonging to the kth-largest eigenvalue. Furthermore, random matrix theory suggests a threshold and other criteria above which eigenvalues are significant, thus giving an estimate of the contained clusters. Third, the above classification which allows overlapping clusters is meaningful.

Remark on step 1. One can also select different complementary criteria. For example, a measure that tells how significant an eigenvector that belongs to a potentially large significant eigenvalue departs from the Porter-Thomas law. Another criterion checks for any structure in the eigenvector even though the distribution of elements is Porter-Thomas. That this criterion is not redundant has been stated above. Furthermore, we have seen there that *within* the Marčenko–Pastur bound close to the right edge the eigenvectors are not necessarily informationless.

Remark on step 5. There are two pieces of information in the eigenvector matrix that help in this task:

(a) We can use the η_k significant elements determined above of each vector v_{ki} (of which there are $\eta_k - 1$). Of great help is the fact that the relevant elements of *all* the to-be-found v_{ki} are located in the *same index*, i.e. row, of the relevant (=large) elements in V_k. This fact can be observed also in reality as ridges of high (colour) variance in the reshuffled version of the eigenvector matrices in Fig. 5. As shown in examples with more clusters, there is no ambiguity. So, in the end, the found v_{ki} arranged next to each other form a rectangle of size $\eta_k \times (\eta_k - 1)$ of high (colour) variance. The remainder is made up of only small elements.

(b) The number η_k of relevant (not necessarily large!) elements in the regions A, B, E, F, etc. is equal to the number of significant vectors η_k minus one! The best fit must be in accordance with this, otherwise the algorithm has to switch to step 6 again.

Remark on step 6. In other words, try to add an additional vector element to the set of relevant vector elements in V_k, and then see what happens. This implies that an additional vector v_{ki}^* has to be selected as well as an additional relevant vector element in *all* other v_{ki} so far selected. (This of course includes the additional v_{ki}^*). So we use an increased rectangle A, B, etc. If the fit of all coded modes does not improve, then the best clustering will be reached. Furthermore, a "quality" value can be chosen that associates the best clustering with the smallest overlap between the clusters or anything else that is suitable for the application.

With the above procedure, the accidental clustering of the three time series in Fig. 1 does not happen. The fluctuating (red) time series in panel B is not included in the cluster with the other two.

Note that in this field the widely used Kernighan-Lin algorithm (6) is the best example of a purely empirical clustering method that is justified by achieving satisfactory results in practice.

4. Scenario 2: Uncorrelated Noise with More Variables than Measurements per Variable

The Marčenko–Pastur theory for uncorrelated independent Gaussian noise in the Wishart matrix ensemble was developed for $T > N$ in the large size limit, i.e. more realisations per random variable than variables. It has been recognised only recently by Lehmann (5) that for $T < N$ the theory persists essentially unchanged.

In particular, for $T \ll N$ the limiting distribution is the Wigner semicircle law. The difference is a shift of variables and a delta-contribution of zero eigenvalues. With $m = N/T$ the eigenvalue density can be expressed as

$$\rho(\lambda) = \frac{1}{2\pi\lambda}\sqrt{4m - (1+m-m\lambda)^2} + \delta(\lambda)(1-m),\ \lambda \in (\lambda_-, \lambda_+), \quad (15)$$

$$\lambda_\pm = (1 \pm \sqrt{m})^2/m. \quad (16)$$

This law is intuitively understandable since the number of non-zero eigenvalues in the product of two iid random matrices is the rank given by $\min(T, N)$. The non-zero eigenvalues resulting from the matrix products $\mathbf{A}^T\mathbf{A}$ or $\mathbf{A}\mathbf{A}^T$ with rectangular $\mathbf{A}$ made of independent random variables are even numerically identical up to a global normalisation factor. Eq. 15 contains a delta function that represents the zero eigenvalues. Since the eigenvalue density of a GOE random matrix is independent of the matrix size beyond ca. N, $T > 50$, the matrix products $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$ are equivalent in the non-zero part of the spectrum.

Formula 15 is independent of N and T, and in this section we deal with $T \ll N$. To minimise finite size effects in obtaining the Marčenko–Pastur density in the non-zero part of the spectrum, N has to be chosen quite large to allow T to be sufficiently large to achieve $T \ll N$. Equation 15 can be reconstructed in numerical experiments; the spectra are shown in Fig. 12 with $T = 3$ and $N = 900$. The logarithmic y-scale allows the delta function at $\lambda = 0$ to be observable together with the non-zero part of the spectrum. In the linear plot, the shape of the Marčenko–Pastur law can be recognised, but it lies very far out as compared to the standard case with $T \geq N$. In the following, most examples are based on the choice $m = N/T = 900/3$ according to the example presented in Lehmann (5). Larger values of T are better calculated on a parallel machine or with a lot of time because many realisations are necessary to obtain a reasonable accuracy in the histograms.

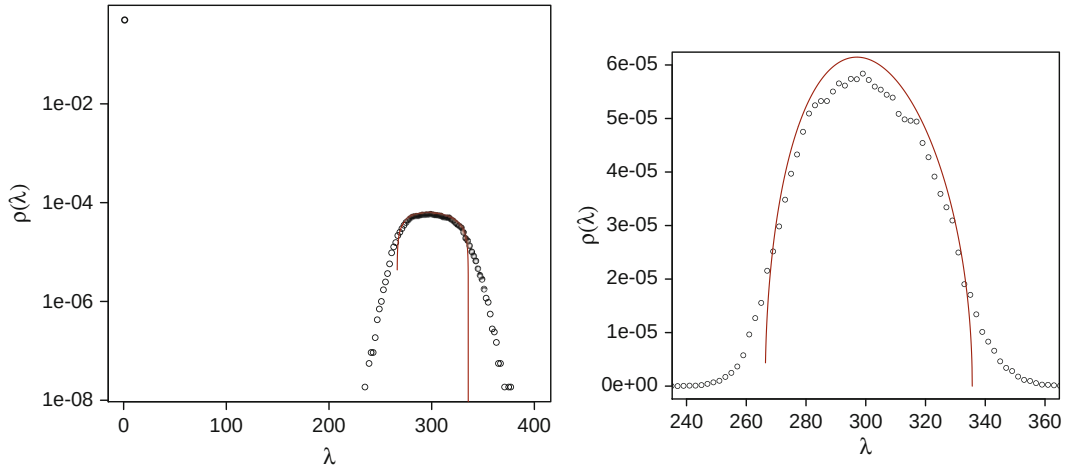


Fig. 12. Spectrum averaged from 12,000 Wishart matrices $\mathbf{AA}^T$ where $\mathbf{A}$ is $N \times T = 900 \times 3$ with uncorrelated Gaussian noise. In the limit $N \to \infty$, the histogram approaches the shifted and rescaled GOE spectral density of 3×3 matrices, i.e. the Wigner semicircle (*continuous line*). The single data point at $\lambda = 0$ marks the delta function in Eq. 15 representing many zero eigenvalues.

In respective literature, several fundamental results in random matrix theory are presented as candidates for a correspondence with some real world situation or some mathematical/statistical object that cannot easily be calculated. The same is the case with the above result by Lehmann. The point of view is simply carried over from applications in finance, where the Wishart matrix ensemble is considered to be sufficiently close to the respective correlation matrix. However,

> The matrix of *sample* correlation coefficients is an inappropriate approximation of the Wishart matrix ensemble if T is small such that the true and realised mean and variance of iid noise differ significantly.

The statement above can be restated as follows: Assume $\xi_1(t),\ldots,\xi_N(t)$ to be iid and uncorrected random variables with zero mean and standard deviation $\sigma = 1$ and realisation index t. Then, the matrix of sample correlation coefficients

$$C_{ij} = \frac{\Sigma_t \left(\xi_i(t) - \bar{\xi}_i\right)\left(\xi_j(t) - \bar{\xi}_j\right)}{(T-1)\mathrm{sd}(\xi_i)\mathrm{sd}(\xi_j)} \tag{17}$$

with sample standard deviation $\mathrm{sd}(\xi)$ and sample mean $\bar{\xi}$ is significantly different from the Wishart ensemble

$$C = \frac{1}{T} MM^{\mathrm{T}} \tag{18}$$

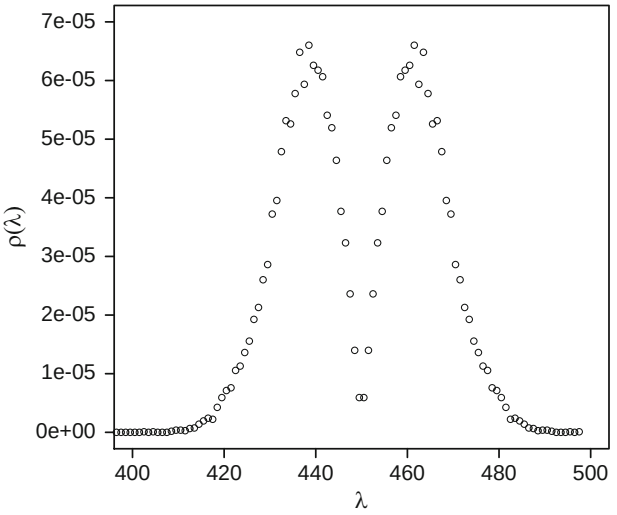


Fig. 13. Non-zero part of the spectrum averaged from 12,000 sample correlation coefficient matrices with uncorrelated noise; $N=900$, $T=3$.

according to the definition of **M** as in Eq. 2. This is simply due to the insufficient number of realisations to obtain a good estimate of mean and variance. The spectrum produced via the sample correlations for the same data as in Fig. 12 is shown in Fig. 13. The differences are peculiar:

1. The approximate middle spectrum is shifted to a higher value of $\lambda = 300 \rightarrow \lambda = 450$.
2. The spectrum consists of two disconnected parts.

One can show, at least empirically by analysing the eigenvalues directly, that there is no eigenvalue falling into the point region between the two supports of the halves in this example.

How can this be explained? We first observe that for $N \nearrow T$ up to $N=T$ the matrix rank of the sample correlation coefficient matrix is not full but exactly one less than the number of independent rows or columns in the Wishart ensemble. Looking at the list of eigenvalues, we discover one numerically zero eigenvalue below the (empirical) Marčenko–Pastur spectrum, which is strictly positive definite. This is due to the Bessel correction for degrees of freedom; (11), appendix 8. This issue becomes noteworthy if the number of non-zero eigenvalues is very small, say around five. This is a realistic case in the analysis of functional groups.

We can also measure the rank against the number of random variable realisations T. See Table 1. The last three rows of the table correspond to Figs. 12, 14, and 15. These three spectra are calculated for fixed $m = N/T = 900/3 = 1{,}200/4 = 1{,}500/5$. The rank of the matrix is reflected in the number of independent rows and columns as the QR-decomposition used for these

Table 1
(*Colour online*) With the transition $T > N$ to $T \leq N$, the rank of the sample correlation matrix drops earlier and persistently by one than the rank of the Wishart matrix

N	T	Rank sample correlation matrix	Rank $C = \frac{1}{T} MM^T$
900	902	900	900
900	901	900	900
900	900	899	900
900	899	898	899
900	898	897	898
1,500	5	4	5
1,200	4	3	4
900	3	2	3

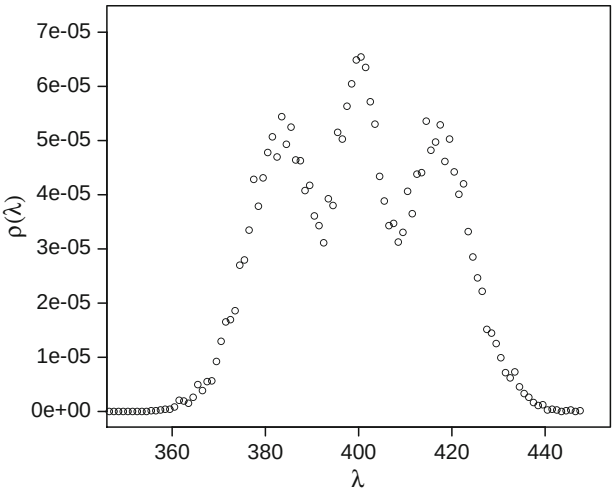


Fig. 14. Non-zero part of the spectrum averaged from 6,000 sample correlation coefficient matrices with uncorrelated noise; $N = 1{,}200$, $T = 4$.

calculations is numerically able to tell. The output is consistent with the observed eigenvalues. With increasing T and constant m the number of maxima increases, and we can expect the spectrum to converge against the analytic prediction for the Wishart ensemble because the position of the spectrum wanders closer to the position of the analytic curve and the number of maxima increases.

So far, these are empirical facts that seem to be relevant. An intuitive explanation of the overall shift of the spectrum is that the

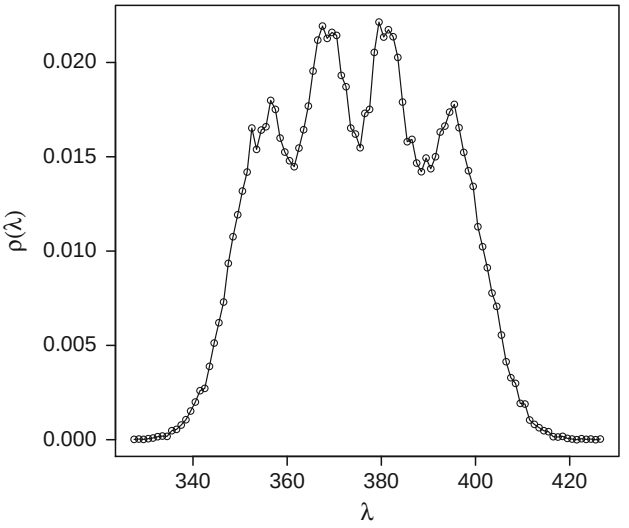


Fig. 15. Non-zero part of the spectrum averaged from 15,000 sample correlation coefficient matrices with uncorrelated noise; $N = 1{,}500$, $T = 5$. The data points are connected with lines to make the series of maxima more apparent.

normalisation by realised means and variances introduces correlation, on the average, in all pairs of variables.

Whether micro-array experiments with many more genes/proteins than expression values can benefit from the results of random matrix theory has to be seen. Since we have at best a few micro-arrays, each giving about ten expression values we get very few non-zero eigenvalues for comparison with the null-hypothesis spectrum above. The nuisances of micro-array experiments and data analysis is an epic in itself. What can be claimed already is that no single experiment with two orders of magnitude more genes/proteins than experiments can contain the information to separate more than very few functional groups. And even if the data was perfect in the sense of no technical and biological variance and with only three to four functional groups that reveal themselves perfectly in significant up (and down?) of the genes/proteins, which would be biologically quite a luxury, then the groups could be identified by eye in the data already.

5. Scenario 3: Correlated Noise with More Variables than Measurements per Variable

The time series data from Subheading 2.2 can be used as a starting point for making T smaller than N. The demonstration will use $T = 100$ keeping $N = 200$. This scenario is still by far not as extreme as with micro-array data. We now show the eigenvector

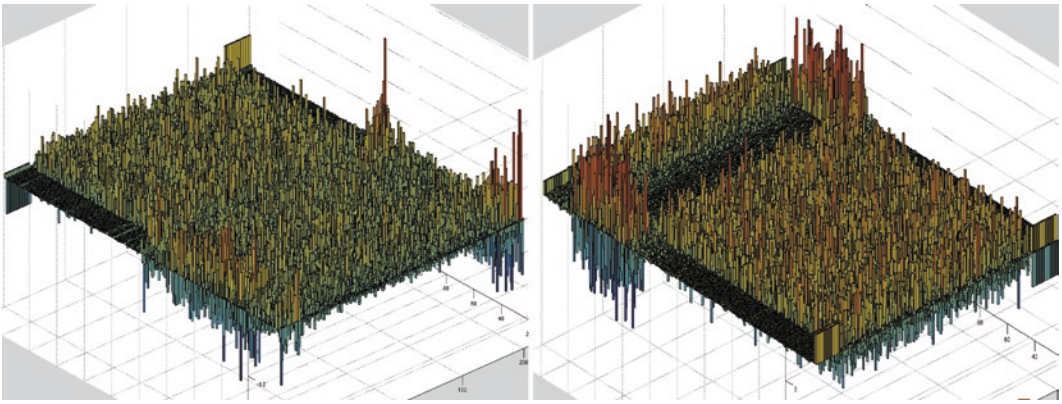


Fig. 16. (*Colour online*) Eigenvector matrices of the similarity (correlation) matrix **C** (*left*) and the dissimilarity matrix $1-|\mathbf{C}|$ (*right*). With $N=200$ and $T=100$ the two correlated clusters cannot be easily recovered in the vectors anymore. The behaviour of both eigenvectors is rather different, too. In the right picture, the clusters are not even expressed symmetrically.

matrix of the correlation (similarity) matrix and of the dissimilarity matrix $D=1-|C|$, see Fig. 16 showing both from the same perspective. First note one eigenvector $(1,...,1)/\sqrt{N}$ in the highest index of the dissimilarity matrix. The explanation is analogous to why the Laplacian also always has $(1,...,1)$ as an eigenvector, see Ref. (10). At this point, it also becomes apparent that for $T<N$ the behaviour of the two matrices departs more significantly and the detailed analysis of the vectors is left open for future work.

Whether the equality $T=N$ is a special barrier, at least in the Wishart ensemble, depends on the questions asked. We have seen in the previous section that sudden changes occur in certain measures like the matrix rank. The less information with decreasing T, the less eigenvalues and eigenvectors are able to capture the behaviour. With $T \ll N$ the data can hardly be distinguished from randomness. We saw in Fig. 16 that the "disturbance" of randomness caused by correlations spreads out to more eigenvectors than originally contained in the artificial clusters. This can also be observed with $T=N$ and more so in the dissimilarity matrix: A closer look at the eigenvector matrices in the regions A, B, E, and F as defined in Fig. 11 reveals that the fluctuations decay slowly towards increasing column index. This is coincidental misinterpretation by the correlation estimator of a random walk as correlated with a cluster or vice versa. Neither are the edges of the Marčenko–Pastur spectrum hard. The dissimilarity matrix seems less prone to such false-positive and false-negative measurement because it disregards information due to the equal treatment of correlation and anti-correlation.

6. Genetic Profile Scenario of Micro-array Data on Differential Expressions

In the previous sections, the problem of extracting clusters from correlation matrices was discussed. In micro-array experiments, it is common to deal with values of differential expressions

$$x_i = \log\left(\frac{I_{\text{red}}}{I_{\text{green}}}\right) \tag{19}$$

for each gene i. The values I are dye intensities on the array wafer that code the number of detected RNA/protein molecules. One of the two usually contains the "control experiment" as a reference base. Any deviation from zero indicates that the gene is differentially expressed with respect to the reference value. This is the extremely idealised situation that is actually far from reality. Such is the point of view of random matrix theory, though. Likewise, we assume that the deviations around zero are linear and symmetric plus other idealisations.

It is common in practice to work only on a subset of approximately 300–500 genes that are expected to contain one or more functionally related genes that are "provoked" to express differentially, i.e. to change expression levels, under certain possibly changing experimental conditions. The term "experiment" is, in this case, used on a more general level and can refer to different spots on the wafer or different wafers containing replicas or data from different biological conditions. In both cases, the types of systematic errors are different. Commonly, the number of experiments under different experimental conditions is often restricted below ten, mostly due to financial limits. There are three scenarios that can be considered:

0. The true null-situation. All I, where colour $\in$ {red, green}, contain the same experimental condition. Any extracted pattern is some systematic error. This test is non-trivial since the different dyes behave differently. The house-keeping genes in particular, which inadvertently could also be regulated, must pass the test.
1. We are given the values $x_i(t)$ where $t = 1,\ldots, T$ indexes the experiments with the *same* experimental condition, i.e. replicas. One may assume that the values fluctuate randomly and independently around the same (*expectation?*) value due to technical or biological variance. This scenario can be considered as a measure to extract *at least some* possible systematic errors that are introduced technically or biologically and influence the measurement within the same experiment index t. This situation should correspond to the null-hypothesis of uncorrelated random variables. A grand unprovable theory of

experimental physics states that in a chain of errors the resulting final error is most likely Gaussian. Therefore, we have a good chance that this is also the case here, and we get the scenario of Bessel-distributed random numbers if X, $\Upsilon \sim N(0,\sigma)$ and we deal with one of the above null-situations. The product of independent Gaussian random numbers is distributed according to the Bessel function of the first kind index zero.

2. We are given the values $x_i(t)$, where $t = 1,\ldots, T$ indexes the experiment with *different* experimental condition, for example, increasing cell stress or anything biologically sensible that is hoped to provoke expression changes in functional groups. One may assume that the expression values carry information if they are biologically expected to do so. The abstractional step from micro-arrays to realisations of random variables is debatable in this case. We expect the differential expression $x_i(t)$ to have a functional relationship with t. If t indexes the experiment number with increasing cell stress level or any other meaningful condition, then subsequent values $x(t_j)$, $x(t_{j+1})$ are highly dependent, probably monotonic and possibly even nearly linear in t if the cell stress is increased slowly. The latter would implicate that $\Delta x(t) = \text{const}$, leading to scenario 1 after normalisation to $\langle \Delta x(t) \rangle = 0$. This experimental setting of "small" changes is likely to achieve the initial goal of controlled differential expression of the same set of functional groups best. Yet since this ideal situation is not to be expected, this second scenario is probably still a distinct case.

In the light of the previous section by considering scenario 2, it can be questioned in how far it might be sensible to normalise the variance and the means of the expressions for one gene with the sample values or any other value. In addition, it is debatable whether to take the increase of the differential expression level as the to-be-correlated variable or maybe $x_i(t)$ directly. As we have seen, the realised variance, mean and probably other statistical measures of interest, lose their meaning with extremely small number of realisations. The ideal genetic profiling draws no information from additional experiments (e.g. increased cell stress) if the relation between $x_i(t_j)$ and $x_i(t_{j+1})$ is (ideally) nearly linear. This poses a paradox since the micro-array business considers many experiments as beneficial. This line of reasoning however, leads to fundamental debates about the current view on micro-array experiments and the information they can contain as well as which statistical prerequisites/algorithms to use. The information contained in the data set $x_i(t_j = 1,\ldots, T)$ is then essentially the slope independent of T. Thus, the information contained in the data set can be coded entirely into one matrix element.

Random matrix theory only accounts for equal-time or "equal-experiment-index" correlation. This statement clearly points to the previous section on the reconstruction of modes while it is yet unclear how to deal with the situation of reduced matrix rank. The following statement may arise from the reasoning above:

> Scenario 2 provides highly dependent realisations, possibly even linear, of variable $x_i(t)$ with index $t_j \to t_{j+1}$. As opposed to time, the difference $t_{j+1} - t_j$, e.g. coding cell stress level, is not meaningless. For such a highly systematic situation, the ordering sequence $t_1,\ldots, t_T$ cannot be disregarded. It would be inappropriate for any analysis to ignore this dependency, thus to stick to the random matrix point of view alone.

Nevertheless, for the null-situations depicted above, we can still perform some sand box simulations with the luxury of nearly infinite sand in order to be able to calculate densities for the sample correlation matrix ensemble. The spectral density for the null-situation with six experiments is shown in Fig. 17 with unknown variances and means, i.e. we are dealing with the sample correlation matrix ensemble. Note that this density is obtained from averaging over many histograms. If variances and means are known (somehow), we get the Wishart ensemble back. Figure 18 shows an artificial experiment to create a possible scenario in

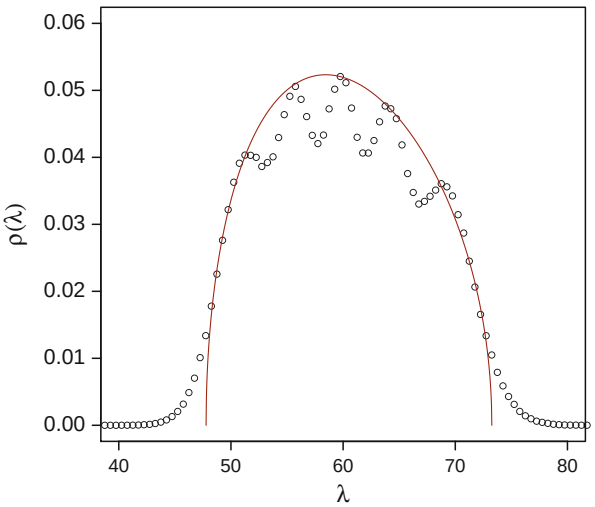


Fig. 17. Non-zero part of the spectrum averaged from 250,000 sample correlation coefficient matrices with uncorrelated noise; $N = 300$, $T = 6$. The normalisation in this picture is to the total number of non-zero eigenvalues. Also shown is the scaled and shifted analytic prediction for the Wishart ensemble to allow comparison of the shapes.

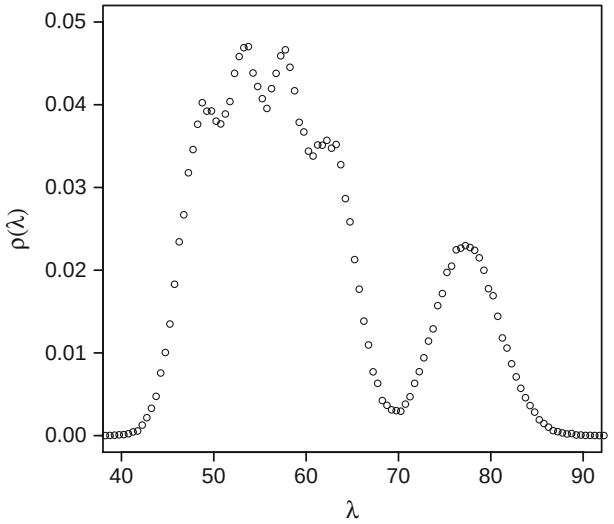


Fig. 18. Non-zero part of the spectrum averaged from 50,000 sample correlation coefficient matrices; $N=300$, $T=6$. The first 20 variables are artificially correlated with each other with a coefficient $c=0.9$ according to Eq. 25.20. The normalisation in this picture is to the total number of non-zero eigenvalues which is the only significant part of the spectrum.

micro-array data analysis with differential expression of a subset of replicas with high correlation:

- $T=6$ number of realisations per random variable
- $N=300$ number of random variables
- $N_c=1$ number of independently correlated groups of variables
- $N_1=20$ number of correlated variables
- $\xi_n(t)$ Gaussian noise data set n where $t=1,\ldots, T$, $n=1,\ldots, N$.
- Type of correlation within the group:

$$\xi_n(t)=\xi_n(1-c)+\Xi(t)c \qquad (20)$$

  Ξ is a prefixed "parent" noise vector specific for the correlated group. $c\in[0, 1]$ is a correlation coefficient. Ξ is identical for all realisations!
- $c=0.9$ (very high correlation)

The correlated cluster induces a bump of eigenvalues on the right of the null-hypothesis spectrum. Despite the extremely high correlation coefficient, the additional bump does not lie far outside the uncorrelated null-spectrum. We conclude from this that the correlation measurement, whether via Wishart or sample correlation matrix ensemble, is rather insensitive. In practice, we would have only one single eigenvalue that has to be judged by its position with respect to the null-spectrum. And since it turned

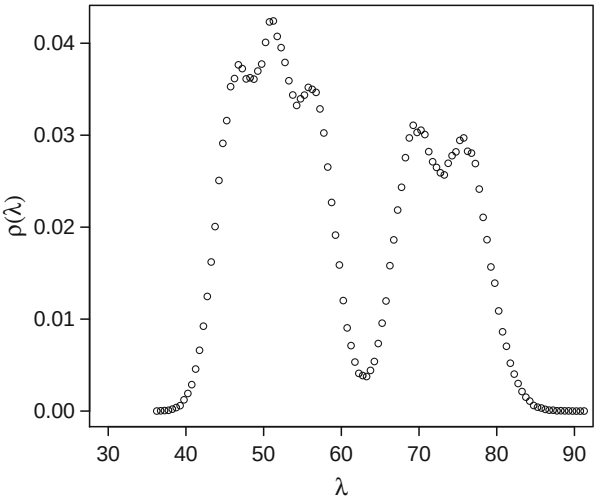


Fig. 19. Non-zero part of the spectrum averaged from 80,000 sample correlation coefficient matrices; $N=300$, $T=6$. The first 20 and the second 20 variables are artificially correlated clusters independently of each other with a coefficient $c=0.9$ according to Eq. 20. The normalisation in this picture is to the total number of non-zero eigenvalues which is the only significant part of the spectrum.

out to be a non-trivial extension of one correlated cluster Fig. 19 shows the following scenario with two independent and equally correlated clusters:

- $T=6$ number of realisations per random variable
- $N=300$ number of random variables
- $N_c=2$ number of independently correlated groups of variables
- $N_1=20$, $N_2=20$ numbers of correlated variables in each group
- $\xi_n(t)$ Gaussian noise data set n where $t=1,\ldots,T$.
- Type of correlation within the group i:

$$\xi_n(t)=\xi_n(1-c_i)+\Xi_i(t)c_i \qquad (21)$$

Ξ_i is a prefixed "parent" noise vector specific for group i and $c_i\in[0,1]$ is a correlation coefficient. Ξ_1 and Ξ_2 are identical for all realisations!

- $c_1=c_2=0.9$ (equal and very high correlation)

In spite of equal number of variables and magnitude of correlation in both groups we get two clearly separable maxima in the distribution of eigenvalues that are pushed out of the null-spectrum. The distance between the maxima is coincidence and likely not to be zero since the number of realisations is small. The cause of the separation is the mutual correlations between the

members of the two groups which is very likely to be biased (highly non-zero) with such a small number of realisations.

Note again that the histograms are produced by averaging over many realisations, thus we cannot compare two curves in the application but can do a confidence test using the previously calculated expected hypothesis density. In reality, we only have six eigenvalues that have to be judged by their likelihood of appearance with respect to the informationless bulk of the spectrum. Furthermore, the data for the above toy examples is perfect in the sense that there are

1. no outliers,
2. no (systematic) measurement problems,
3. no technical variance (usually also systematic).

Such errors have to be modelled and obtained from experiments. The null-hypothesis requires the a priori calculation of the null-hypothesis density by averaging over many realisations. The data must contain the above error model in a parameterised fashion.

This is not a sort of nit-picking since the impact of some error or outlier within the series of only six experiments for a gene on the resulting six non-zero eigenvalues is very large. There are articles devoted entirely to error modelling and treatment in micro-array experiments and data analysis (2).

> The creation of the null-hypothesis spectrum would have to include these problems under the assumption that they are stationary and reproducible *during the experiment* in a parameterisable fashion.

Note that in principle this is the same procedure as in the null-hypothesis in financial data analysis that creates uncorrelated time series with empirically plausible increments.

7. Summary and Conclusion

We have shown so far to what extent the information content in the eigenvalues and eigenvectors of a correlation matrix is redundant as long as we have more measurements than variables. Spectral clustering ignores this redundancy which is justified as long as the data is perfect. Moreover, clustering methods that disregard the data and only consider the correlation matrix make errors due to ambiguity. In case of having far less measurements than variables, we have demonstrated how the correlation matrix spectrum differs to the Wishart ensemble, using either the sample

mean and covariance or the expectation values. In the scenario of micro-arrays, the application of correlation matrix analysis for the extraction of functional groups turns out to be questionable. It can serve, however, in the analysis and modelling of errors. We also conclude that the choice of the correlation coefficient as an estimator of "connection" between two genes/proteins is completely arbitrary and mostly the best guess. The indicator function that takes up values –1 and 1 depending on positive or negative co-expression is a priori not less suitable.

8. Notes

Despite being a theoretical contribution, we can give practical guidelines.

1. The interpretation of a statistical analysis in the context of micro/protein-arrays requires error modelling and a null-hypothesis test to quantify the information content of the results. Note that the random and informationless situation produces statistical patterns that can go for significant information. The error model must include all effects that also produce patterns, in particular the systematic errors.
2. Missing values as in the context of proteins do not bias the estimator *only if the values miss in a random and independent fashion*. This has to be controlled and judged beforehand. See Ref. (3) for a very comprehensive study. However, already without missing values as in typical micro-array scenarios the information content is overcritically low. The limit of minimum information (or maximum percentage of missing values) can only be found in the context of the questions asked and precise error model by creating a suitable null-hypothesis test.
3. The emergence of one clear functional group as the dream scenario in cluster analysis is likely to be visible in the direct inspection of correlation coefficients.
4. The emergence of more than one clear functional group with similar strengths of co-expression as another dream scenario in cluster analysis requires principle component analysis, clustering or other suitable methods to distinguish the clusters.
5. A large eigenvalue that does not clearly stem from a sub-cluster of correlated genes but involves a large majority of all genes indicates a co-expression of all genes. This should evoke suspicion towards the experiment.
6. Only weak but *very consistent and reproducible* co-expression of a particular gene/protein may not show up in the direct

inspection of correlation coefficients, it may show up in a principle component analysis, and it is likely to show up best if using the above-mentioned indicator function.

7. If the sign of the correlation coefficients for different experiments with sufficiently slowly increasing, e.g. cell stress, is not consistent, then this should raise alarm about the experimental conditions. The cause for suddenly negated/reverted expression must then be explained biologically.
8. In continuation of the above note: The disposal of weak correlation coefficients is not appropriate since a small value is not a sign of low information content. This is possibly a debatable issue in Ref. (7) where this is practised.

References

1. Comellas F, Diaz-Lopez J (2008) Spectral reconstruction of complex networks. Physica A 387:6436–6442
2. Fang Y, Brass A, Hoyle DC, Hayes A, Bashein A, Oliver SG, Waddington D, Rattray M (2003) A model-based analysis of microarray experimental error and normalisation. Nucleic Acids Res 31(16):e96
3. Fulger D, Politi M, Germano G, Iori G (2009) The pearson and fourier pearson correlation estimators in the context of spectral correlation matrix analysis with continuous-time random walks
4. Laloux L, Cizeau P, Bouchaud J.-P, Potters M. (1999) Noise Dressing of Financial Correlation Matrices. Phys Rev Lett 83(3) 1467–1470. American Physical Society. DOI 10.1103/Phys Rev Lett 83.1467
5. Lehmann N (2006) Principal components selection given extensively many variables. Stat Probab Lett 74:51–58
6. Lin S, Kernighan BW (1973) An effective heuristic algorithm for the traveling-salesman problem. Operations Res 21: 498–516
7. Luo F, Zhong J, Yang Y, Scheuermann RH, Zhou J (2006) Application of random matrix theory to biological networks. Phys Lett A 357:420–423
8. Marčhenko VA, Pastur LA (1967) Distribution for some sets of random matrices. Math USSR-Sb 1:457–483
9. Minicozzi P, Rapallo F, Scalas E, Dondero F (2008) Accuracy and robustness of clustering algorithms for small-size applications in bioinformatics. Physica A 387:6310–6318
10. Newman MEJ (2004) Detecting community structure in networks. Eur Phys J B Condens Matter Phys 38:321–330
11. Reichmann WJ (1961) Use and abuse of statistics. Methuen. Reprinted 1964–1970 by Pelican
12. Luxburg U (2007) A tutorial on spectral clustering. Statistics and Computing 17(4) 395–416. Kluwer Academic Publishers. http://dx.doi.org/10.1007/s11222-007-9033-z

Chapter 26

Standards, Databases, and Modeling Tools in Systems Biology

Michael Kohl

Abstract

Modeling is a means for integrating the results from Genomics, Transcriptomics, Proteomics, and Metabolomics experiments and for gaining insights into the interaction of the constituents of biological systems. However, sharing such large amounts of frequently heterogeneous and distributed experimental data needs both standard data formats and public repositories. Standardization and a public storage system are also important for modeling due to the possibility of sharing models irrespective of the used software tools. Furthermore, rapid model development strongly benefits from available software packages that relieve the modeler of recurring tasks like numerical integration of rate equations or parameter estimation.

In this chapter, the most common standard formats used for model encoding and some of the major public databases in this scientific field are presented. The main features of currently available modeling software are discussed and proposals for the application of such tools are given.

1. Introduction

Systems biology is an emerging field of study, which aims on quantitative analysis of the interdependencies of components within a biological system. Generally, such an approach needs acquisition of both temporal and spatial data obtained at high resolution. The rise of several "omics" research areas is the basis for acquiring such comprehensive data sets. Therefore, modeling is a means for integrating the results from Genomics, Transcriptomics, Proteomics, and Metabolomics experiments and to gain insight into the interaction of the constituents of a biological system.

Data collection for large-scale modeling attempts usually surpasses a single laboratory's forces. Therefore, a joint effort of different institutions is promising. However, sharing such large

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696,
DOI 10.1007/978-1-60761-987-1_26,

amounts of frequently heterogeneous and distributed information needs both standard data formats and public repositories. The efforts made for the standardization of DNA microarray experiments yielded the *Minimal Information About a Microarray Experiment* (MIAME, (1)) and the *MicroArray Gene Expression Markup Language* (MAGE-ML (2)), which constituted a paradigm that was applied to other biological techniques. In the field of Proteomics, for example, several standard formats are available like mzML (3) and mzIdentML (http://www.psidev.info/index.php?q=node/319), see Subheading 3 of this volume. Public repositories like PRIDE (4) and others along with a data submission pipeline (5), which includes several converters into the standard data formats, were established in order to facilitate data sharing and rapid publication. Standardization is also important for ensuring compliance with quality standards. In the field of Proteomics, the *Minimal Information About a Proteomics Experiment* (MIAPE, (6)) addresses this issue and aims on specifying the mandatory information needed for the interpretation of Proteomics surveys.

However, standardization and a public storage system are not only advantageous for the experimental part of systems biology, but also for modeling. Standardization enables storing of models irrespective of the used software tools.

Concerning this, several standard formats (e.g., Systems Biology Markup Language (SBML), http://sbml.org/Main_Page (7–9); CellML, http://www.cellml.org/ (10–12); BioPAX (13); NeuroML (14)) have been designed for model encoding. The *Minimum Information Requested in the Annotation of biochemical Models* (MIRIAM, (15)) is a set of guidelines dealing with annotation and curation of computational models in this field of research. Finally, public repositories exist in order to store and to access quantitative models, which target on the simulation of biological processes. In particular, these efforts strongly support a modular developing concept, where comprehensive models can be assembled from small reusable software modules.

Furthermore, a modular modeling design benefits from software packages that relieve the modeler of recurring tasks like numerical integration of rate equations or parameter estimation. Such a modeling environment facilitates focusing on the implementation of the actual biological process.

The following paragraph deals with the XML-based formats SBML and CellML, probably the most common standard formats used for model encoding, and presents some of the major public databases. Then, the mean features of currently available software packages are discussed. Finally, the “notes” paragraph deals with the adoption of SBML and CellML and gives some suggestion on typical application and the ease of use for the reviewed modeling software packages.

2. Requirements for the Exchange of Quantitative Biological Models

This paragraph compares major features of CellML and SBML, the current standard model interchange languages used for the sharing of biochemical models within the community (see Note 1).

2.1. CellML and SBML: Standard Formats for the Annotation of Biological Models

2.1.1. The Systems Biology Markup Language (SBML)

SBML is a data format for encoding mathematical models that reproduce biological processes. Special focus is given on supporting the modeling of biochemical reaction networks, gene regulation, metabolism, and cell signaling pathways.

SBML uses the terms "level" and "version" for denoting modifications of the format, where major releases are called levels. Versions are utilized to denote the minor modifications of SBML. SBML Level 2 Version 4 is the current (June 2010) final release of SBML. SBML is well supported within the systems biology community. On September first, 2009 the Web site of the SBML project listed 171 software tools that use SBML (http://sbml.org/SBML_Software_Guide/SBML_Software_Matrix). Moreover and probably more important, there are hundreds of SBML-encoded models available from public repositories. This highlights the importance of a standard exchange format for sharing and reusability of biological models.

There are several converters available for SBML including conversion to CellML, to XPPAUT ".ode" files (see http://www.math.pitt.edu/~bard/xpp/xpp.html), and to BioPAX (see Note 1).

2.1.2. CellML

In comparison with SBML, CellML is designed as a generic framework with a broader scope of application. CellML is capable of reproducing any kind of mathematical models and represents a viewpoint originating from engineering sciences. Consequently, the language was used not only for simulation of systems biology issues, but also for, e.g., multiscale models in the field of synthetic biology (16).

2.1.3. Comparison of SBML and CellML (see Notes 2 and 3)

Like other standards, the development of a generic data format for systems biology is subject to continuous modification. This appraisal is therefore rather temporary and focuses on the main differences between the two languages.

There is difference concerning the encoding of biological information. SBML uses the annotation tag (<annotation>). In CellML, only the structure of the model and the mathematics applied is specified in the language elements. For all other relevant information like annotations, CellML uses metadata for storing. The CellML language applies the Resource Description

Framework (RDF, http://www.w3.org/RDF/) for embedding metadata (e.g., references to model-related publications or to experimental data used for model calibration) in an arbitrary location of the CellML document.

In comparison with SBML, CellML has less third-party support.

Currently, neither SBML nor CellML support models that apply partial differential equations (PDEs) (see Note 4).

2.2. Minimum Information Requested in the Annotation of Biochemical Models (MIRIAM) and the Systems Biology Ontology (SBO)

The need for model reuse and modular assembly of already available submodels, which enables the creation of comprehensive models, requires the definition of a basic quality standard for model encoding. Both MIRIAM and SBO address this issue and incorporate a semantic domain into the model encoding process.

2.2.1. MIRIAM

MIRIAM (15, 17) provides a set of guidelines supporting standardized annotation and curation of models in the field of systems biology. MIRIAM consists of two parts. The first part of MIRIAM proposes a standard for reference correspondence. Here, among others both the relation of the model to a single reference description and encoding in a public, machine-readable format is a prerequisite for MIRIAM compliance. MIRIAM compliant models must also include all quantitative attributes (e.g., initial conditions and parameter values) in order to enable the reimplementation of the model in an adequate software environment. Furthermore, the output of the model must be in agreement with the results given in the reference description. The second part of MIRIAM relates each model component unambiguously to an external bioinformatics data resource by the use of Unique Resource Identifiers (URIs). For a comprehensive exemplification of the guidelines, please refer to the paper of Le Novere et al. (15).

2.2.2. Systems Biology Ontology (SBO)

The SBO (http://www.ebi.ac.uk/sbo/) (17, 18) comprises a set of controlled vocabularies (CVs) that organizes systems biology related terms in a hierarchical structure. Currently, the SBO consists of six different vocabularies. For example, the branch denoted *mathematical expressions* classifies calculi used in biochemical modeling.

Important components of SBO entries are a stable unique identifier, a name, a definition, synonyms, one or several comments, and in case of mathematical expressions an equation given in both the MathML (http://www.w3.org/Math/) notation and as graphical display.

Usage of the controlled vocabularies given by the SBO is an important prerequisite for browsing the public repositories of biochemical models in an efficient and comprehensive way.

Usage of SBO terms is helpful for both understanding and evaluating models because of the unambiguous way of model description.

2.3. Systems Biology-Related Databases

This paragraph deals with some important databases for publicly available biochemical models. In particular, databases that provide SBML and CellML-encoded models are considered.

A main resource for annotated published SBML models is the BioModels database (http://www.ebi.ac.uk/biomodels-main/) (19). The SBML team contributes to the development of this database. Therefore, the default data format for storing models is SBML. However, the models can be downloaded in other formats as well (CellML, XPPAUT ".ode" files, SciLab (http://www.scilab.org/)). In the beginning of September, 2009, the database contained 231 curated and 198 noncurated models. Curated models in the BioModels database are MIRIAM compliant. Furthermore, acceptance in the curated branch additionally requires publication of the model in a peer-reviewed scientific journal.

JWS online (http://jjj.biochem.sun.ac.za/index.html, (20)) is a curated database providing download of kinetic models in SBML. Additionally, JWS online includes a simulation tool, enabling both simulation and manipulation of the models hosted in the database directly within the Web browser. There is strong collaboration between the BioModels database and JWS online, including model exchange between both initiatives.

Within the Kyoto Encyclopedia of Genes and Genomes (KEGG, (21, 22)) resource, the information of various biological networks is stored in the KEGG PATHWAY database. KEGG pathways integrate the knowledge of several cellular processes like molecular interaction or processing of genetic information and consider both normal and perturbed cellular stages. Each pathway can be regarded as a representation of a biological model. As a consequence, KEGG pathways can be encoded in SBML via the KEGG2SBML tool (http://sbml.org/Software/KEGG2SBML) and used for modeling purposes.

A large set of CellML-encoded models are available directly from the Website of the CellML project (http://models.cellml.org/, (23)). This database covers a wide range of biological processes, including biochemical processes, physiology, and synthetic biology.

There are several databases targeting on special scientific domains. The Database of Quantitative Cellular Signaling (DOQCS, http://doqcs.ncbs.res.in/) stores mathematical models of signaling pathways. However, the model description is given in the GENESIS scripting language. Usage is therefore limited to simulation with Kinetikit/GENESIS (http://www.ncbs.res.in/index.php?option=com_content&task=view&id=304, (24)).

ModelDB (http://senselab.med.yale.edu/modeldb/, (25)) stores computational models in the field of neurosciences.

The SigPath system (http://www.sigpath.org, (26)) stores data related to cell signaling pathways and networks both qualitatively and quantitatively. Furthermore, SigPath provides an interesting alternative concept for the sharing of models and public management of quantitative data. Within SigPath, users can assemble quantitative models using quantitative data stored in a so-called information management systems. These models can then be exported in several formats, including SBML.

3. Software Packages for Systems Biology

This paragraph deals with available software tools designed for the modeling of biochemical networks. Because the number of such software packages is growing, a selection was made, including both relatively simple modeling tools and comprehensive modeling environments (Table 1). Simple tools may be adequate for small projects enabling a rapid and intuitive model development, where more complex tasks require the usage of a sophisticated modeling environment.

The given selection while intended to cover currently available software in this field is somewhat subjective and the reader should consider available literature (27) for further reading.

In the first part of this paragraph, several requirements frequently involved in the modeling of biochemical networks are addressed. Then, basic features of the considered software packages are discussed. Special attention is given to features that are unique for a given software tool.

3.1. Basic Requirements for Modeling Biochemical Networks

1. Systems biology requires mathematical approaches for modeling the dynamics of biological processes. Three major approaches may be differentiated. One of those approaches is frequently called “stochastic” and tries to simulate the behavior of each existing molecule within the system. This enables the consideration of statistical variation. However, such a comprehensive technique leads to high computational costs especially when performing large-scale projects.

   Another approach is often denoted as “deterministic” and assumes that the stochastic variation is negligible due to the high amount of molecules in a certain cell compartment. Generally, deterministic models apply a set of ordinary or partial differential equations. Because analytical solutions are frequently not available or limited to specific conditions of the considered biological system, numerical algorithms are necessary for integration. Some software tools employ a “hybrid” strategy where

Table 1
Global information about the reviewed modeling tools and the supported import and export capabilities

Name of the program	Version	Resource	Operating system	Input formats	Output formats	License/terms of use	Open source
Copasi	4.5	http://www.copasi.org/tiki-index.php	Windows, OS X, Linux Solaris (SPARC)	SBML (Level 1&2) Copasi Gepasi	SBML (Level 1&2) Copasi	free (noncommercial use)	+
CellWare	3.0.1	http://www.bii.a-star.edu.sg/achievements/applications/cellware/index.php	JRE 1.4[a] or later	SBML (Level 1&2)	SBML (Level 1&2)	free	–
Dizzy	1.11.4	http://magnet.systemsbiology.net/software/Dizzy/	JRE 1.4[a] or later	CMDL[b] SBML (Level 1)	CMDL[b] SBML (Level 1)	LGPL	+
Virtual Cell	4.6	http://www.nrcam.uchc.edu/	JRE 1.5[a] or later	VCML[c], Matlab, SBML, CellML[b]	VCML[c], Matlab, SBML, CellML[d]	free	–
JDesigner/Jarnac[e]	2.1c/2.29j	http://www.sys-bio.org/index.htm	Windows	SBML (Level 1&2)	SBML (Level 1&2)	BSD	+
Dynetica	1.2	http://www.duke.edu/~you/Dynetica_page.htm	JRE 1.3[a] or later	.dyn[c]	.dyn[c]	free	+
E-Cell	3.1.105	http://www.e-cell.org/ecell	Windows Linux	SBML (Level 1&2)	SBML (Level 1&2)	GPL	+

[a]All Java written software is executable on operating systems that are supported by the Java virtual machine (JVM)
[b]CMDL is the Chemical Model Definition Language, the native language of the Dizzy scripting engine
[c]Native file format of the software
[d]Virtual Cell exports both SBML and CellML formats. However, only the native VCML format support all Virtual Cell features
[e]JDesigner and Jarnac are part of the Sytems Biology Workbench (SBW)

both stochastic and deterministic approaches are mixed even within a single model.

2. A deterministic modeling scheme requires the implementation of numerical algorithms. However, numerical integration leads to discretization errors and the applied techniques vary with respect to runtime, stability, and robustness. Therefore, different biological models may need different numerical methods. A generic modeling tool benefits from the implementation of a multitude of such algorithms.
3. A cellular system is very complex and often shows a behavior that is far from linearity. However, this complexity can be attributed to a rather small set of kinetic rate laws. Therefore, constructing the model can be eased when the software supports selection from a library of predefined rate equations.
4. Eukaryotes are much larger than prokaryotes and exhibit a much higher degree of organization: Cellular space is divided in several compartments characterized by both different physicochemical properties and biological processes. Considering the multicompartmental structure is critical for an advanced understanding of the eukaryotic cell's functioning. Therefore, the reproduction of cellular compartments is an important feature for a comprehensive cellular modeling software.
5. Effective sharing of models implicates the need for public repositories and usage of widely accepted standard formats for model encoding. Therefore, an ideal modeling environment supports import and export in standard formats and permits options for searching public repositories along with a download feature of selected models.
6. Shared model development can be assisted from a Web-based software architecture or by the use of a client – server architecture, where simulation is carried out on a central server. This supports cooperation of researchers working at different locations.
7. Complex biological models usually comprise a huge number of parameters, especially when aiming on a mechanistic mathematical description of biochemical processes. However, it is often difficult or even impossible to measure all of these parameters. Implementation of some kind of parameter estimation is a crucial feature when calibrating such comprehensive biological models. Furthermore, sensitivity analysis is an important tool for evaluating the impact of a specific parameter or the used initial values of the state variables on changes within the considered biological system. Such an analysis gives valuable insight whether or not a specific parameter or initial value is negligible. Additionally, the results of the sensitivity analysis give useful decision guidance for scheduling of further measurement and the experimental design.

8. Biological systems frequently show sudden qualitative changes, which are caused by small changes of the parameter values. For example, such systems may oscillate or show some kind of switching or even chaotic behavior. The initial of a changing model dynamics in the parameter space is a so-called bifurcation point. Bifurcation analysis searches for these bifurcation points and is an important means in order to achieve a better understanding of structurally unstable dynamical systems.
9. In order to evaluate the modeling performance, several statistical tests are needed for calculating the goodness of fit, a measure for the discrepancies between observed and simulated values. Though a basic statistical analysis can be performed using standard spread sheet programs (e.g., Excel, Microsoft Corp., Redmond, WA, USA) or, e.g., the statistics program STATISTICA (StatSoft Europe GmbH, Hamburg, Germany), the ability to perform statistics directly within the modeling software will greatly enhance usability. Because a single statistical criterion may be insufficient, several of these tests should be used for a sound modeling evaluation.
10. Evaluation of simulation results is strongly simplified by graphical representation. Therefore, modeling tools need at least a basic plotting system.
11. Larger biological systems comprise processes that run within time ranges differing in several orders of magnitude. Concurrent integration of very fast and very slow reactions is challenging. The ability to cope with this issue is a key feature for tools aiming on biological simulation.

3.2. Comparison of the Reviewed Modeling Tools

We used a standard PC (Intel Pentium Dual Core, 2.5 GHz, 4 GB RAM, Windows XP Professional, Service Pack 3) for installation of the discussed software. The installation processes went smoothly and quietly for all of the tools. All reviewed software tools are freely available, at least for noncommercial use (Table 1). However, they vary greatly regarding their complexity and the implemented features (Table 2).

For some of the tools (COPASI (28), Dizzy (29), JDesigner/Jarnac (30, 31) and E-Cell (32–34)), source code is available (Table 1, see Note 8).

Each considered software implements both deterministic and stochastic algorithms for modeling interpretation. Furthermore, all programs provide at least two different numerical techniques for solving differential equations. Therefore, providing a set of different simulation techniques may be regarded as standard for modeling of biochemical networks.

Special feature of COPASI is a multitude of predefined rate equations. This facilitates the creation of models for the most common biological modeling applications.

Table 2
Overview over basic features of the reviewed modeling software

Name of the program	COPASI	CellWare	Dizzy	Virtual cell	JDesigner/jarnac	Dynetica	E-Cell & E-Cell IDE
Stochastic/deterministic model interpretation	Both	Both	Both	Both	Both	Both	Both
Multiple numerical algorithms	+	+	+	++	+	+	+
Partial differential equations	–	–	–	+	–	–	–
Predefined rate equations	++	+	–	+	+	+	+
Modeling of cell compartments	+	+	+	++	+	–	+
Interoperability/Extensibility	–	(+)[a]	(+)[b]	–	+[c]	–	(+)[d]
Connection to public repositories	–	+ (KEGG)	–	++	+	–	–
Software architecture	Stand alone	Stand alone	Stand alone	Client – server	Stand alone Web services	Stand alone	Stand alone
Parameter estimation	+	+	–	+	–	–	+
Sensitivity analysis	+	–	–	+	+	+	+
Bifurcation analysis	–	–	–	–	+[e]	–	+
Distributed computing	Batch	Distr. Comp./ Batch	–	Distr. Comp.	Batch	–	Distr. Comp.
Graphical representation	(+)[f]	++	–	+	++	+	+

[a]There is possibility for adding further deterministic and stochastic algorithms
[b]Dizzy integrates interfaces for both the Systems Biology Workbench (SBW) and Cytoscape
[c]Because JDesigner/Jarnac are modules of the Systems Biology Workbench (SBW) direct connection to other systems biology-related tools is available
[d]E-Cell allows dynamically the loading of plug-ins in order to provide alternative algorithms for simulation
[e]Bifurcation analysis can be performed via the connection to the Bifurcation Discovery tool (44), which is a module of the SBW
[f]COPASI is capable of displaying layout information of encoded models. However, creating diagrams within COPASI is not supported

Dizzy and Dynetica (35) are both rather slim solutions for modeling issues, providing the core functionality indispensible for the simulation of biochemical networks (see Note 5). Dynetica also includes a tool for graphical representation of biological processes, which can be used for an intuitive creation of the models. Dizzy lacks this feature, but in exchange it supports modeling of spatial compartments, which is essential when considering large-scale cellular models of eukaryotic cells.

Both JDesigner and Jarnac are modules of the Systems biology workbench (SBW, http://sys-bio.org/). JDesigner enables visual assembly of biochemical models. Jarnac comprises both a scripting language for building models and a simulation engine. Jarnac can be used to run models designed within the JDesigner application. Therefore, in combination of both software tools, JDesigner can be regarded as frontend and Jarnac as backend (see Note 6).

Both CellWare (36, 37) and E-Cell/E-Cell IDE (http://www.e-cell.org/ecell/, http://www.e-cell.org/ide/) provide good tradeoff between ease of use and considerable performance. Special feature of both applications is the support of distributed computing (see Note 7). Some differences between CellWare and E-Cell/E-Cell IDE may be addressed: CellWare offers direct connection to the KEGG database. There is no possibility for a direct query of public model repositories from the E-Cell software. In contrast to CellWare, E-Cell includes better capabilities for sensitivity analysis.

Only E-Cell and JDesigner/Jarnac support bifurcation analysis. Users that are interested in studying the nonlinear behavior of biological systems may benefit from this opportunity.

The most advanced program reviewed is Virtual Cell (38, 39). The software is appropriate to model a wide range of biological tasks, e.g., reactions, membrane transport, and electrical potential. Virtual Cell integrates the complex geometries of cellular compartments into the model. Virtual Cell is the only simulator discussed in this chapter that is able to solve PDEs. PDEs are utilized in order to represent biological processes where concentrations within the compartment are heterogeneously distributed. Such gradients frequently arise in the vicinity of biological membranes, for example caused as a result of the activity of membrane-bound ion pumps. Virtual Cell features download models from the BioModels database, which offers a large resource for curated and annotated biological models. Furthermore, the software allows storage of models within a centralized repository, enabling collaborative work of researchers from different locations.

4. Notes

1. Although efforts were made for the conversion of SBML and CellML ((40); http://www.ebi.ac.uk/compneur-srv/sbml/convertors/SBMLConvertors.html) currently, only translations of simple models are available. Therefore, if submission to public repositories is intended, at present the supported input and output formats is a crucial feature for choosing an adequate modeling software.
2. CellML is characterized by relatively elevated complexity. Therefore, the usage of this interchange language requires an extended training period in comparison with SBML. The SBML development team put much effort on facilitating the development of SBML-encoded models. The libSBML library (41) liberates the software developer from the necessity to care about the adequate release of the language. Furthermore, libSBML supports model validation and MIRIAM compliant annotations. Several features of libSBML are accessible from a comfortable editor software, named SBMLeditor (42). Additionally, the SBMLeditor provides connection to the SBW, a software framework that supports access to several tools designed for quantitative systems biology. Therefore, the SBML protects its user from working with too much technical details of encoding. There is more time available for central issues concerning model development (e.g., determination of rate equations or constituents of the system).
3. CellML provides a more generic modeling framework. However, enhanced flexibility leads to a more complex structure of this language in comparison with SBML. Scientists, who are interested in modeling at the interface of different scientific domains (e.g., systems and synthetic biology) may benefit from the flexibility of CellML. Researchers who are solely interested in systems biology may prefer SBML, which offers a more concise and focused structure.
4. If a modeling task requires the application of PDEs, constraints arise for both selection of the adequate modeling software and the appropriate input or output format. Hopefully, in the future a linkage between CellML and FieldML (43) may allow encoding of PDE models. Therefore, this data format will become attractive for researchers who use this kind of models.
5. For educational purposes or to familiarize oneself with basic modeling concepts a small tool like Dizzy or Dynetica seems preferable. Both tools are shipped with some examples that are adequate for introducing basic modeling concepts. Dynetica includes a graphical user interface. Such a graphical

representation of the model is more intuitive and straightforward in particular for inexperienced users in comparison with the pure textual way of model creation, which is used by Dizzy. However, "direct" encoding forces the studying of differential equations in detail, which results in thorough understanding of the basic concepts involved in systems biology. CellWare has also an intuitive and well-structured user interface, which qualifies this tool for educational purposes.

6. Users of JDesigner/Jarnac may benefit from the integration of these modules in the SBW. This framework includes a set of SBML translators, including Matlab, Fortran, and XPP translators. Therefore, interoperability of JDesigner and Jarnac is strongly increased. Furthermore, JDesigner and Jarnac can cooperate with other SBW modules in order to further analyze the model. Several tasks are addressed by such modules, e.g., frequency analysis, searching for oscillation and switching behavior (bifurcation analysis) or a module designed for the visualization of models in 3D.
7. Several standard modeling tasks (e.g., parameter estimation or sensitivity analysis) require multiple runs of the models and frequently tie up a lot of computational resources. Therefore, both calibration and analysis of model behavior will benefit from the implementation of batch processing or distributed computing. CellWare and E-Cell seem to be the best choice if such features are indispensible.
8. Special problems require special solutions. Though most software environments are capable of modeling most features of biochemical networks, there are specific problems possible that require customization of the modeling system. Therefore, both well-documented open source software and/or a modeling environment that allows the integration of plug-ins is sometimes very important in order to adapt existing software tools to new arising questions. All tools except for CellWare and Virtual Cell are open source.

Acknowledgments

Michael Kohl is funded by the Bundesministerium für Bildung und Forschung (BMBF), grant 01 GS 08143.

References

1. Brazma A, Hingamp P, Quackenbush J, Sherlock G, Spellman P, Stoeckert C, Aach J, Ansorge W, Ball CA, Causton HC, Gaasterland T, Glenisson P, Holstege FCP, Kim IF, Markowitz V, Matese JC, Parkinson H, Robinson A, Sarkans U, Schulze-Kremer S, Stewart J, Taylor R, Vilo J, Vingron M (2001) Minimum information about a microarray experiment (MIAME) - toward standards for microarray data. Nat Genet 29:365–371

2. Spellman P, Miller M, Stewart J, Troup C, Sarkans U, Chervitz S, Bernhart D, Sherlock G, Ball C, Lepage M, Swiatek M, Marks WL, Goncalves J, Markel S, Iordan D, Shojatalab M, Pizarro A, White J, Hubley R, Deutsch E, Senger M, Aronow B, Robinson A, Bassett D, Stoeckert C, Brazma A (2002) Design and implementation of microarray gene expression markup language (MAGE-ML). Genome Biol 3:research0046.1–0046.9
3. Deutsch E (2008) mzML: a single, unifying data format for mass spectrometer output. Proteomics 8:2776–2777
4. Martens L, Hermjakob H, Jones P, Adamski M, Taylor C, States D, Gevaert K, Vandekerckhove J, Apweiler R (2005) PRIDE: the proteomics identifications database. Proteomics 5:3537–3545
5. Eisenacher M, Martens L, Hardt T, Kohl M, Barsnes H, Helsens K, Hakkinen J, Levander F, Aebersold R, Vandekerckhove J, Dunn MJ, Lisacek F, Siepen JA, Hubbard SJ, Binz PA, Bluggel M, Thiele H, Cottrell J, Meyer HE, Apweiler R, Stephan C (2009) Getting a grip on proteomics data - proteomics data collection (ProDaC). Proteomics 9:3928–3933
6. Taylor CF, Paton NW, Lilley KS, Binz PA, Julian RK, Jones AR, Zhu WM, Apweiler R, Aebersold R, Deutsch EW, Dunn MJ, Heck AJR, Leitner A, Macht M, Mann M, Martens L, Neubert TA, Patterson SD, Ping PP, Seymour SL, Souda P, Tsugita A, Vandekerckhove J, Vondriska TM, Whitelegge JP, Wilkins MR, Xenarios I, Yates JR, Hermjakob H (2007) The minimum information about a proteomics experiment (MIAPE). Nat Biotechnol 25:887–893
7. Hucka M, Finney A, Sauro HM, Bolouri H, Doyle JC, Kitano H, Arkin AP, Bornstein BJ, Bray D, Cornish-Bowden A, Cuellar AA, Dronov S, Gilles ED, Ginkel M, Gor V, Goryanin II, Hedley WJ, Hodgman TC, Hofmeyr JH, Hunter PJ, Juty NS, Kasberger JL, Kremling A, Kummer U, Le Novere N, Loew LM, Lucio D, Mendes P, Minch E, Mjolsness ED, Nakayama Y, Nelson MR, Nielsen PF, Sakurada T, Schaff JC, Shapiro BE, Shimizu TS, Spence HD, Stelling J, Takahashi K, Tomita M, Wagner J, Wang J (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19:524–531
8. Hucka M, Finney A, Bornstein BJ, Keating SM, Shapiro BE, Matthews J, Kovitz BL, Schilstra MJ, Funahashi A, Doyle JC, Kitano H (2004) Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup Language (SBML) project. Syst Biol (Stevenage) 1:41–53
9. Finney A, Hucka M (2003) Systems biology markup language: Level 2 and beyond. Biochem Soc T 31:1472–1473
10. Garny A, Nickerson DP, Cooper J, dos Santos RW, Miller AK, McKeever S, Nielsen PMF, Hunter PJ (2008) CellML and associated tools and techniques. Philos T R Soc A 366:3017–3043
11. Lloyd CM, Halstead MD, Nielsen PF (2004) CellML: its future, present and past. Prog Biophys Mol Biol 85:433–450
12. Cuellar AA, Lloyd CM, Nielsen PF, Bullivant DP, Nickerson DP, Hunter PJ (2003) An overview of CellML 1.1, a biological model description language. Simul-T Soc Mod Sim 79:740–747
13. Luciano JS (2005) PAX of mind for pathway researchers. Drug Discov Today 10:937–942
14. Goddard NH, Hucka M, Howell F, Cornelis H, Shankar K, Beeman D (2001) Towards NeuroML: model description methods for collaborative modelling in neuroscience. Philos T Roy Soc B 356:1209–1228
15. Le Novere N, Finney A, Hucka M, Bhalla US, Campagne F, Collado-Vides J, Crampin EJ, Halstead M, Klipp E, Mendes P, Nielsen P, Sauro H, Shapiro B, Snoep JL, Spence HD, Wanner BL (2005) Minimum information requested in the annotation of biochemical models (MIRIAM). Nat Biotechnol 23:1509–1515
16. Serrano L (2007) Synthetic biology: promises and challenges. Mol Syst Biol 3:158
17. Le Novere N, Courtot M, Laibe C (2006) Adding semantics in kinetics models of biochemical pathways. Ruedesheim, Germany, pp 137–153
18. Le Novere N (2006) Model storage, exchange and integration. BMC Neurosci 7:S1
19. Le Novere N, Bornstein B, Broicher A, Courtot M, Donizelli M, Dharuri H, Li L, Sauro H, Schilstra M, Shapiro B, Snoep JL, Hucka M (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. Nucleic Acids Res 34:D689–D691
20. Olivier BG, Snoep JL (2004) Web-based kinetic modelling using JWS online. Bioinformatics 20:2143–2144
21. Kanehisa M, Goto S, Hattori M, Aoki-Kinoshita KF, Itoh M, Kawashima S, Katayama T, Araki M, Hirakawa M (2006) From genomics to chemical genomics: new developments in KEGG. Nucleic Acids Res 34:D354–D357

22. Kanehisa M, Goto S (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res 28:27–30
23. Lloyd CM, Lawson JR, Hunter PJ, Nielsen PF (2008) The CellML model repository. Bioinformatics 24:2122–2123
24. Bhalla US (2002) Use of Kinetikit and GENESIS for modeling signaling pathways. Method Enzymol 345:3–23
25. Hines ML, Morse T, Migliore M, Carnevale NT, Shepherd GM (2004) ModelDB: a database to support computational neuroscience. J Comput Neurosci 17:7–11
26. Campagne F, Neves S, Chang CW, Skrabanek L, Ram PT, Iyengar R, Weinstein H (2004) Quantitative information management for the biochemical computation of cellular networks. Sci STKE 248:pl11
27. Alves R, Antunes F, Salvador A (2006) Tools for kinetic modeling of biochemical networks. Nat Biotechnol 24:667–672
28. Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, Singhal M, Xu L, Mendes P, Kummer U (2006) COPASI- A COmplex PAthway SImulator. Bioinformatics 22:3067–3074
29. Ramsey S, Orrell D, Bolouri H (2005) Dizzy: stochastic simulation of large-scale genetic regulatory networks. J Bioinform Comput Biol 3:415–436
30. Hucka M, Finney A, Sauro H, Kovitz B, Keating S, Matthews J, Bolouri H (2003) Introduction to the Systems Biology Workbench. Available via the World Wide Web at http://sbw.kgi.edu/caltechSBW/sbwDocs/docs/intro/intro.pdf.
31. Sauro HM, Hucka M, Finney A, Wellock C, Bolouri H, Doyle J, Kitano H (2003) Next generation simulation tools: the systems biology workbench and BioSPICE integration. OMICS 7:355–372
32. Takahashi K, Ishikawa N, Sadamoto Y, Sasamoto H, Ohta S, Shiozawa A, Miyoshi F, Naito Y, Nakayama Y, Tomita M (2003) E-cell 2: Multi-platform E-Cell simulation system. Bioinformatics 19:1727–1729
33. Tomita M, Hashimoto K, Takahashi K, Matsuzaki Y, Matsushima R, Saito K, Yugi K, Miyoshi F, Nakano H, Tanida S, Saito Y, Kawase A, Watanabe N, Shimizu TS, Nakayama Y (2000) The E-CELL project: towards integrative simulation of cellular processes. New Generat Comput 18:1–12
34. Tomita M, Hashimoto K, Takahashi K, Shimizu TS, Matsuzaki Y, Miyoshi F, Saito K, Tanida S, Yugi K, Venter JC, Hutchison CA (1999) E-CELL: software environment for whole-cell simulation. Bioinformatics 15:72–84
35. You LC, Hoonlor A, Yin J (2003) Modeling biological systems using Dynetica - a simulator of dynamic networks. Bioinformatics 19:435–436
36. Dhar P, Meng TC, Somani S, Ye L, Sairam A, Chitre M, Hao Z, Sakharkar K (2004) Cellware - a multi-algorithmic software for computational systems biology. Bioinformatics 20:1319–1321
37. Dhar PK, Meng TC, Somani S, Ye L, Sakharkar K, Krishnan A, Ridwan ABM, Wah SHK, Chitre M, Hao Z (2005) Grid Cellware: the first grid-enabled tool for modelling and simulating cellular processes. Bioinformatics 21:1284–1287
38. Slepchenko BM, Schaff JC, Macara I, Loew LM (2003) Quantitative cell biology with the virtual cell. Trends Cell Biol 13:570–576
39. Moraru II, Schaff JC, Slepchenko BM, Blinov ML, Morgan F, Lakshminarayana A, Gao F, Li Y, Loew LM (2008) Virtual Cell modelling and simulation software environment. IET Syst Biol 2:352–362
40. Schilstra MJ, Li L, Matthews J, Finney A, Hucka M, Le Novere N (2006) CellML2SBML: conversion of CellML into SBML. Bioinformatics 22:1018–1020
41. Bornstein BJ, Keating SM, Jouraku A, Hucka M (2008) LibSBML: an API library for SBML. Bioinformatics 24:880–881
42. Nicolas R, Donizelli M, Le Novere N (2007) SBMLeditor: effective creation of models in the Systems Biology Markup Language (SBML). BMC Bioinform 8:79
43. Christie GR, Nielsen PMF, Blackett SA, Bradley CP, Hunter PJ (2009) FieldML: concepts and implementation. Philos T R Soc A 367:1869–1884
44. Chickarmane V, Paladugu SR, Bergmann F, Sauro HM (2005) Bifurcation discovery tool. Bioinformatics 21:3688–3690

Chapter 27

Modeling of Cellular Processes: Methods, Data, and Requirements

Thomas Millat, Olaf Wolkenhauer, Ralf-Jörg Fischer, and Hubert Bahl

Abstract

Systems biology is a comprehensive quantitative analysis how the components of a biological system interact over time which requires an interdisciplinary team of investigators. System-theoretic methods are applied to investigate the system's behavior. Using known information about the considered system, a conceptual model is defined. It is transferred in a mathematical model that can be simulated (analytically or numerically) and analyzed using system-theoretic tools. Finally, simulation results are compared with experimental data. However, assumptions, approximations, and requirements to available experimental data are crucial ingredients of this systems biology workflow. Consequently, the modeling of cellular processes creates special demands on the design of experiments: the quality, the amount, and the completeness of data. The relation between models and data is discussed in this chapter. Thereby, we focus on the requirements on experimental data from the perspective of systems biology projects.

1. Systems Biology: System-Theoretic Studies of Cellular Dynamics

Systems biology is a rapidly growing interdisciplinary field of research that focuses on biological systems using system-theoretic approaches. It investigates the dynamics of complex interactions and networks in and between different levels of biological organization (1). The considered systems span all levels of biological organization from cellular compartments to populations of cells. Systems biology integrates the available biological knowledge and modeling approaches to understand the complexity of biological functions, the fundamental principles of biological construction, and the emergent properties of biological systems (2). Toward this end, it uses information about the molecular interactions of molecules, transcriptomic data, intracellular and extracellular concentrations/particle numbers of proteins, and structural information. These data represent the temporal and spatial behavior of

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1_27, © Springer Science+Business Media, LLC 2011

the considered system. All this information is combined into a specific model. Its analysis and simulation helps unraveling unknown network connections, and to predict the behavior of the investigated system under conditions which were not experimentally measured.

The field of systems biology can reflect upon a history of almost 100 years. Beginning with Lottka in 1925 (3), Volterra in 1926 (4), and later Schrödinger in 1944 (5) this research area was initially called mathematical biology. The system-theoretic view was introduced by Mesarović (6), Haken (7), and Bertalanffy (8) in the late 1960s and the 1970s which eventually introduces the name "Systems Biology." In the wake of sequencing and "omics" programs, the increasing amount of experimental data, the observed complexity, and ability to quantify cellular processes resulted in a strong stimulation for this field over the last decade.

In recent years, the growing interest in the modeling of cellular systems leads to increasing requirements on experimental data to be suitable for systems biology projects. Hence, we want to discuss the demands on experiments, which experimenters should consider to provide the collaborating modeling groups with suitable data. Firstly, we discuss what a model is, what one can expect from a model and, very important, what one cannot expect. We then define what modeling means and the consequences to experiments. This is followed by a section about mathematical approaches, which are used in systems biology. Thereby, we do not focus on their theoretical derivation but sketch the main ideas of the approaches. For more detailed introductions, we refer to the literature.

2. Why Is Modeling Necessary and How Will Experimentalists Profit from It?

Cellular processes are of the greatest complexity. This complexity manifests itself in different features. Cellular systems consist of very large numbers of interacting components; the role and function of the components is often vaguely known and many of them have yet to be identified and characterized. Between the components a multitude of processes exists by which they interact. Many of these interactions are highly nonlinear. Finally, biological systems consist of highly related organizational levels, e.g., transcriptome, proteome, and metabolome, which to this day have been mostly considered in isolation. However, the increasing amount of data shows that one has to integrate these levels in order to understand the system's behavior.

Owing to their complexity, we cannot make reliable predictions about biological systems with intuition and data alone. It is necessary to create and apply mathematical/computational tools

which support us in our efforts (9). These tools must be able to deal with many variables, nonlinearities, hierarchies, functional and spatial organization as found in biology. Here, modeling and simulation of biological systems open new opportunities to improve our knowledge using system-theoretic approaches (9, 10).

Nevertheless, every systems biology project starts with available biological knowledge and data. Using this information, models are established and subsequently analyzed and simulated. The results create the demand for new experiments to validate and refine the models or to test hypotheses emerging from the models. Thus, systems biology projects consist of repeated rounds of modeling, model analysis, and experiments, see Fig. 1. Additionally, system-theoretic methods can also support the experimentalist in experimental design. They provide information about which and when components should be measured and which perturbation maximizes the amount of gathered information.

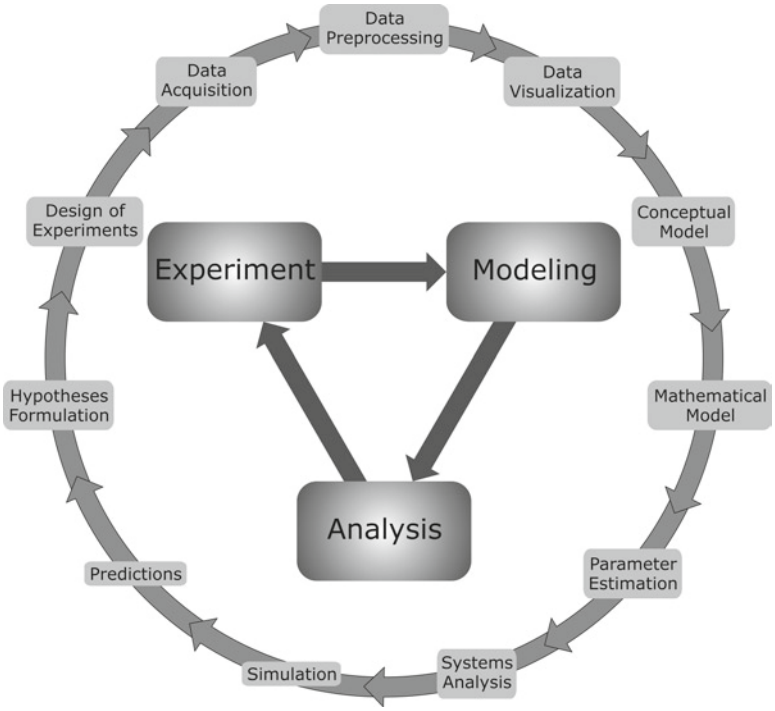


Fig. 1. The iterative process of modeling in systems biology. Meaningful models result from repeated cycles of modeling, model analysis, and experiments. As denoted in the figure, the cycles consist of some consecutive steps. In the following course of this chapter, we focus on conceptual models, mathematical models, and their simulation. Methods for parameter estimation and systems analysis are discussed briefly.

3. Conceptual Models: The Relation Between Cellular Components and Cellular Processes

The modeling of biological systems starts with the definition of a conceptual model. As we see in the following course, this is not as trivial as it sounds. In fact, it is a crucial step which already significantly influences the success of the modeling project. To begin with, we have to define the term "model" because it is used with different meanings across different disciplines. We understand it as a representation of systems components and the processes influencing them. These processes could be interactions between components in the form of biochemical reactions or changes affecting the properties of components such as temperature variation. Considering the famous enzyme kinetic reaction (11, 12) in Fig. 2, the components are substrate S, enzyme E, enzyme–substrate complex C, and product P. They interact via three biochemical reactions. In the course of this biochemical reaction, the substrate is bound to the enzyme resulting in an intermediate complex. The complex can dissociate either into enzyme and substrate or into product and enzyme.

One should be aware that even at this early stage already assumptions were made, which have consequences to the mathematical representation and also the experimental setup generating experimental data. Thus, we assume that we can decompose the catalytic conversion into independent steps. In our example, the enzyme kinetic reaction is separated into three elementary reactions (13, 14):

1. Bimolecular association of substrate and enzyme.
2. Unimolecular dissociation of the enzyme–substrate complex into substrate and enzyme.
3. Unimolecular dissociation of the enzyme–substrate complex into product and enzyme.

It should be noted, that every individual step directly depends on and might be influenced by the considered biosystem under investigation. In our example, it is furthermore assumed that the

$$\mathrm{E} + \mathrm{S} \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \mathrm{C} \xrightarrow{k_3} \mathrm{P} + \mathrm{E}$$

Fig. 2. Chemical equation of the enzyme-facilitated conversion of substrate S to product P. The formation of the enzyme–substrate complex C is assumed to be reversible, whereas the dissociation of C into the product is irreversible (11, 12). Every involved biochemical reaction is determined by a rate coefficient *k* which has to be estimated from experimental data.

enzyme acts as an ideal catalyst which remains unchanged over the net reaction. Additionally, it is assumed that the enzyme can bind immediately to a new substrate molecule after it was released from the enzyme–substrate complex. We assume furthermore that the product is moving very fast away from the enzyme or that a reaction between enzyme and product is very improbable. So we can neglect the reverse reaction between enzyme and product.

It is often useful to merge some reaction steps to reduce and simplify the model and obtained mathematical expressions. Commonly applied techniques include steady-state assumptions (11, 12) or power-law representations (15, 16).

A crucial step in establishing a model is to restrict the system. In the above considered enzyme kinetic reaction, we assumed an isolated system, where no additional processes occur that could also change the substrate, enzyme, or product concentration. Gene expression and enzyme degradation are, for instance, neglected.

Furthermore, we assumed spatial homogeneity for the environment, leaving biochemical properties constant over the observation time. Contrary to this assumption is that physical parameters as pH value and temperature affect the kinetic properties. Therefore, we extend the enzyme kinetic reaction in Fig. 3 considering a pH-dependency (17).

As the example illustrates, a model may include different aspects. However, the more aspects are considered, the more complex the model will be. On the one hand, this complicates the later analysis and interpretation. On the other hand, it requires more information which has to be gathered in experiments. Using our example of a pH-dependent enzyme kinetic reaction, this means that the experimentalist has to measure the cellular pH and the corresponding kinetic properties of the involved enzymes. As illustrated, every process which is significant for the system's behavior has to be considered in the model and should be supported by experimental data. Thus, the experimental design should optimize the controllability of systems parameters and systems evolutions with respect to the demands of modeling. This depends upon a close collaboration between modeler and experimentalist even during the planning phase of experiments.

Current and future systems biology projects require the integration of biological levels, e.g., transcription, protein synthesis, and intracellular turnover of substrate or availability of cofactors. This data have to be complemented by different other aspects, like enzyme activities and regulatory effects like allosteric regulation, or modifications (e.g., phosphorylation state). Generally, no

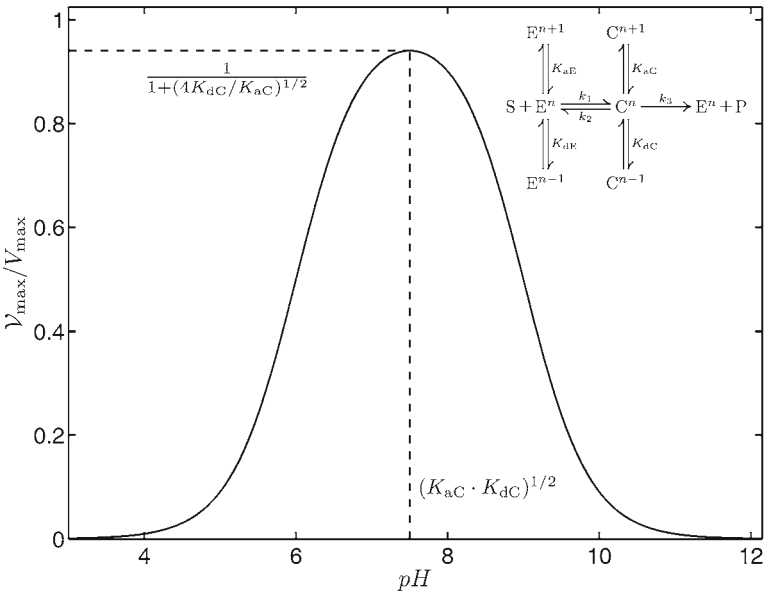


Fig. 3. pH-dependent enzyme kinetic reaction. The conceptual model, shown in the upper right corner, assumes that the enzyme activity is influenced by the hydrons bound to the enzyme and the enzyme–substrate complex, respectively. Only the enzyme with n bound hydrons converts the substrate into the product (17). The association/dissociation of hydrons is assumed to be very fast. Thus, they are determined by their dissociation constants *K*. Further models are discussed in (11). A typical bell-shapep pH-dependent limiting rate (solid line) is derived from the model. Its maximum value is determined by association and dissociation rates of the hydron and the intermediary complex.

single lab has the portfolio to deliver such a diversity of data. That means different specialized labs are necessary to gather the data.

A proper incorporation of experimental data and measured parameters gained by different wet labs at different places requires a maximized identical experimental setup. Although this could be reached by the development of Standard Operation Procedures (SOPs), which should be an integral part of experiments and although many labs already established protocols, the measurement of comparable data using the same experimental setup is to this day more the exception rather than the rule.

We illustrate this requirement using the known temperate dependence of kinetic parameters as an example. From the simple Arrhenius equation (13, 14), it follows that kinetic parameters may vary by a factor 2–3 if the temperature varies by 10°C (We note that the temperature dependence of enzyme activity is much more complex (11, 12).) Even if this sounds not as important, in complex systems such differences can change the whole systems behavior. However, this simple example illustrates how important a common experimental design is for systems biology projects.

4. Requirements on Data: Quantitative, Complete, and Significant Information is Crucial for Systems Biology

As discussed in previous sections, experimental data are crucial for systems biology. First of all, the information gathered is used to establish a conceptual model. However, data are also crucial for later work steps of the systems biology cycle (Fig. 1) where they are required for parameter estimation, model analysis, simulation, and prediction. The demands on the quality of data are different for these two phases of the workflow. To a first conceptual model, qualitative and uncertain data may be sufficient, but later steps require quantitative, complete, and significant experimental information.

Quantitative data are necessary for parameter estimation, comparison of numerical and experimental data (model validation), and to test new hypotheses that emerge from the model. A qualitative model analysis is also part of the systems biology cycle, but one cannot close the cycle using qualitative information alone.

Temporal complete data allow us to distinguish between different dynamic behaviors and provides enough data points to estimate unknown model parameters. Fig. 4a and b show two

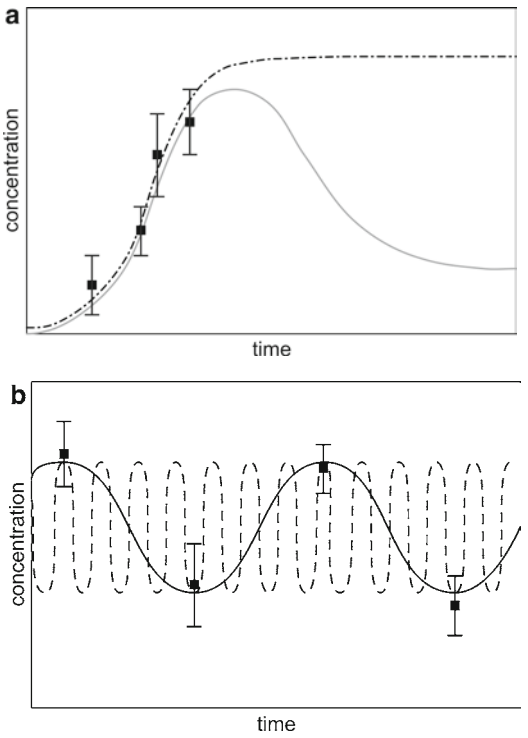


Fig. 4. Two examples of temporal incomplete data. In the *upper plot* (**a**), two models fit the experimental data measured during the initial phase. Alas, the models cannot be distinguished due to too short observation time. The *lower plot* (**b**) shows an oscillating biosystem, where the frequency cannot be uniquely estimated due to insufficient choice of time points to measure.

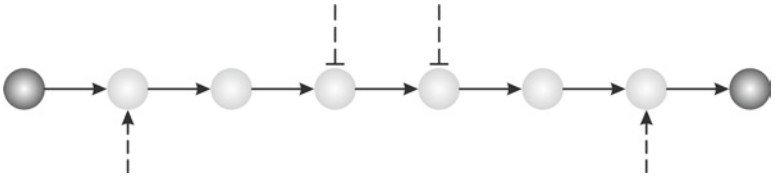


Fig. 5. Structurally incomplete data. In a stimulus–response experiment, the intermediates were not measured. Thus, the system can be only represented as a black box model. Detailed information about time courses and regulations of the intermediates cannot be drawn from the model.

different situations where the measured data are insufficient to falsify different models.

Furthermore, data have to be structurally complete providing information about the time courses of considered cellular components, as shown in Fig. 5. As previously discussed, we consider cellular systems as network of interacting and regulated components. To validate the network structure and to estimate its parameters, we rely on time-resolved data of included components. Otherwise, any plausible substructure may be applied without validation. However, crucial components and measurement strategies can be identified from hypothetical conceptual models. Thus, we again emphasize the importance of a joint experimental design in systems biology projects.

Last but not least, data should be significant. Thereby, different aspects affect the significance of data. First, experimental measurements should be reproducible not only by a single, but also by other labs. This demand is strongly related to common SOPs and experimental setups. Secondly, experimental data should provide additional information about the accuracy of measurements. It is used to distinguish between different models and/or different parameter sets as it is shown in Fig. 6. There, the model represented by a dashed-dotted line is falsified by the experimental data. However, the models shown as a gray solid line and a dashed line, respectively, are both supported by the shown data.

5. Mathematical Modeling: Approaches for Simulation and Analysis of Cellular Systems

In the previous section, we discussed conceptual models representing the components and interactions of cellular systems. Such models present an overview about the ongoing processes. However, the prediction of dynamic behavior, steady states, etc. and finally the comparison with experimental data require a translation into mathematical model which can be simulated and analyzed using system-theoretic tools. Later in this section, we discuss briefly some commonly used methods for systems analysis. At this point,

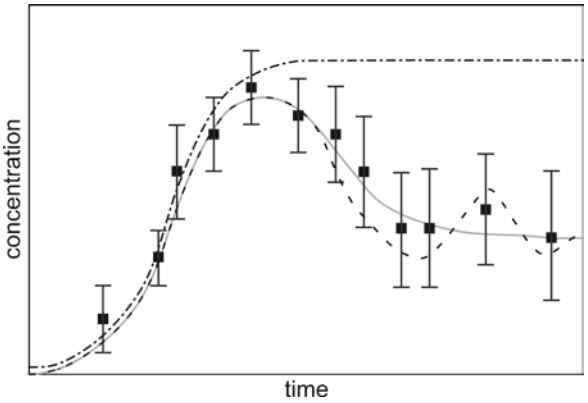


Fig. 6. Comparison of three different models and experimental data. The model represented as a dashed-dotted line does not fit the data points. The other models, shown by *gray solid* and *dashed lines*, reproduce the available experimental data within their uncertainty. Thus, we cannot conclude from the available time course which model is the correct one. We note that the distance to the data points is not a sufficient criterion to distinguish between models. Further information is required to elucidate the network structure.

we focus on the mathematical approaches to model cellular systems and their consequences for experiments. However, we give only a very sketchy introduction to this field. This topic fills whole lectures series and is thus beyond the scope of this chapter. For the interested reader, however, we refer to established textbooks where appropriate.

We should mention that there is no single unified approach to model biological systems. Instead, there are many different representations with different assumptions and different applications. To choose the appropriate approach is a crucial step. The system under consideration, available data, and the question to answer has naturally a high impact on this decision. Furthermore, personal preference, available tools, computer costs, and measurement errors play a role.

In the modeling of biochemical systems, two main classes are used over the last decades: kinetic equations (often called deterministic or conventional approach) and stochastic approaches.

5.1. Kinetic Equations: The Conventional Approach Using Differential Equations

The conventional approach uses kinetic equations of motion to describe the changes in the system as consequence of, e.g., biochemical reactions, transport processes or environmental changes. For an introduction to chemical kinetics we refer to (13, 14). A very detailed derivation, starting at microscopic, molecular properties is presented in (18). In its elementary representation, it is derived from microscopic properties of the reacting molecules. Thus, this approach reflects the mechanistic details of the underlying biochemical reaction (13, 14, 18). This physical basis is the

advantage of the approach. However, it makes strong assumptions concerning the stochastic nature of biochemical reactions and the environmental conditions. It is assumed that the statistical average or expectation value is an appropriate measure for the systems dynamics (18, 19). Fluctuations are assumed to be small in comparison to the average and do not cause new behavior. The first assumption is usually but not always fulfilled in systems with high numbers of particles. If the fluctuations cause new dynamic behavior is a non-trivial question which can often answered only by a systems analysis.

Finally, the considered system is described as a system of coupled nonlinear differential equations (13, 14). In Fig. 7, the results of a numerical simulation are shown using the enzyme kinetic reaction as example. Every component of this biochemical reaction is described by a differential equation which determines the corresponding rate of change.

$$\begin{aligned}
\frac{\mathrm{d}S}{\mathrm{d}t} &= -k_1\,[S][E] + k_2\,[C] \\
\frac{\mathrm{d}E}{\mathrm{d}t} &= -k_1\,[S][E] + (k_2 + k_3)[C] \\
\frac{\mathrm{d}C}{\mathrm{d}t} &= k_1\,[S][E] - (k_2 + k_3)[C] \\
\frac{\mathrm{d}P}{\mathrm{d}t} &= k_3\,[C]
\end{aligned}$$

Positive terms are related to processes producing the component. Negative terms decreases it. The parameters k_1, k_2, and k_3 have to be estimated from experimental data.

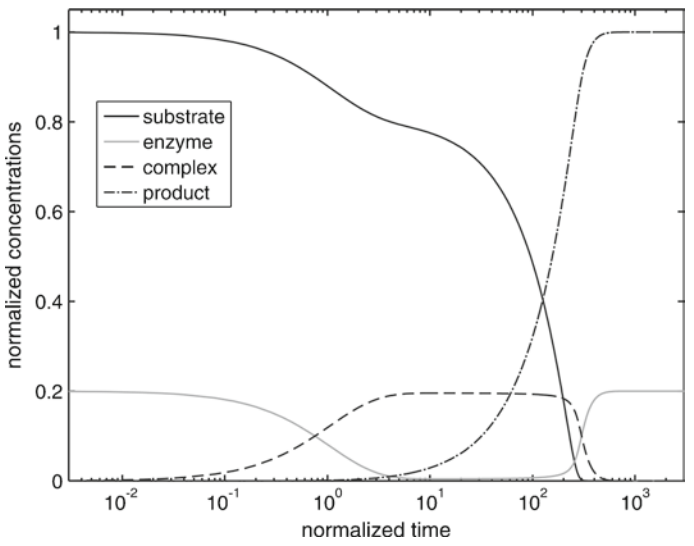


Fig. 7. Numerical simulation plotted in semilogarithmic scales. Every component is described by a differential equation which determines the corresponding rate of change. We normalized the time against the time which is needed to establish a quasi-steady state due to the saturation of the enzyme. The concentrations are plotted as ratio of the initial substrate concentration.

If spatial aspects, e.g., diffusion, can be neglected, the system consists of ordinary differential equations (13–15). Biological compartments can be considered defining spatially isolated areas with specific reactions. Then, additional transport terms describe the flow of matter between the compartments (20, 21). The resulting differential equations determine the rate of change of the systems components. Conventional kinetics (13, 14, 18) requires a detailed knowledge of the mechanism and their properties of biochemical processes and assumes ideal (gaseous) systems which are often unknown and far away from cellular conditions. Hence, approaches were developed which either simplify the system using approximations, e.g., rational reaction rate in enzyme kinetics (11, 12), or merging the systems variable into net-rates describing parts of the network, e.g., Power-laws (15, 16) or S-systems (22).

If spatial aspects have to be considered, the system is represented by a set of coupled partial differential equations describing the spatial propagation and the chemical reactions, e.g., reaction–diffusion equations (23, 24). Because of the complex geometry of cellular systems and increased mathematical demands, simplifications are often used that reduce the systems to ordinary differential equations, e.g., finite-elements methods (25–27).

For the analysis of coupled systems of differential equations, a wide range of well-established analytical and numerical tools was developed in mathematics, physics, and engineering covering different aspects of dynamical systems theory (28–30). Some of the developed methods with special interest to systems biology we discuss briefly. They are often included in numerical tools for systems biology which are introduced in the "Modeling" chapter of this book.

Sensitivity Analysis (31) investigates how sensitive the systems state is to changes in parameters, e.g., kinetic coefficients and enzyme concentrations. It is used to identify crucial steps in pathways which should be measured accurately. On the other hand, such steps are candidates for process optimization and medical applications. Metabolic control analysis (MCA) is commonly used to investigate the dependence of the steady state of metabolic pathways to systems parameters (32–34).

Robustness Analysis investigates the ability of a system to maintain its function against internal and external perturbations (35, 36). In biology, this concept is closely related to "stability" and "homoeostasis" (37).

Bifurcation Analysis studies qualitative changes of the behavior of dynamical systems under parameter variation (38, 39). In biology, it is expected that, e.g., developmental processes as differentiation or apoptosis are governed by multistable networks which show a switch-like behavior if the amount of related signaling proteins exceed a critical value. At such critical points, the systems change their properties in a reversible or irreversible

manner (40). Also bacterial heterogeneity may be explained by multistability (41). Furthermore, bifurcations play an important role in phase transitions of biochemical networks.

Stability Analysis investigates the stability of steady states with respect to perturbations (42). The steady state is stable if the system returns to it after a small perturbation (43). If the system moves to a neighbored steady state, the initial state was unstable or metastable. The stability of steady states can be analyzed by linearization around the fix point (28, 43) or by Lyapunov coefficients (28, 44). It is closely related to bifurcation analysis.

Distinguishability (of states) investigates if two states of a system can be distinguished on the basis of input/output experiments (45).

Observability (of systems) analyzes if every two distinct states of the system are distinguishable (45). Unobservable systems have subsystems that have no influence on the output (46). The distinguishability (of states) and the observability (of systems) are concepts which can be used to optimize experiments for model validation/falsification, parameter identification, and testing model predictions.

Parameter estimation (47) is a procedure to determine the values of model parameters from experimental data. Starting from a randomly distributed set of parameters, the estimator varies the parameters in such a way that an optimum is reached. The obtained parameter set represents the best fit of the proposed model to the experimental data. There are different optimality criteria which can be applied. The best known one is the "Least Square Method" (Regression) (48) which minimizes the distance between experimental data points and a theoretical curve. A successful parameter estimation requires a sufficient temporal (spatial) and structural resolution of data. Additionally, the combination of measured system components makes a difference. Using the enzyme kinetic reaction as example, more accurate information is obtained from an experiment measuring substrate and enzyme–substrate-complex than from the measurement of substrate and product. For model identification, some parameters should be known from experiments. These known parameters restrict the parameters of the still unknown parameters and may discriminate different model structures.

The above (incomplete) list of system-theoretic tools summarizes a further advantage of the kinetic approach. The available tools are sophisticated and implemented in many numerical packages (see another chapter of this book).

5.2. Stochastic Approaches: Fluctuations in Cellular Systems

The randomness of biochemical reactions and biological events can be taken into account using stochastic approaches. An introduction to stochastic processes and their mathematical description can be found in (49–51). The fundamentals of the stochastic modeling were developed simultaneously to the kinetic approaches

described above. Due to the mathematical complexity of stochastic approaches, in the past, only simple systems were considered. With increasing computer power, stochastic simulations became feasible for more complex systems. Stochastic approaches should be used if one expects that fluctuations are important. This is usually the case for systems with low molecule numbers, where the variance (as measure of the fluctuations) becomes comparable to the average or expectation value (52). In a stochastic representation the enzyme kinetic reaction is separated into three reaction channels R_i which results in a discrete change of molecule numbers.

$$\begin{array}{lll} & & \begin{matrix} S & E & C & P \end{matrix} \\ R_1: & \mathrm{E}+\mathrm{S} \xrightarrow{k_1} \mathrm{C} & \begin{bmatrix} -1 & -1 & 1 & 0 \end{bmatrix} \\ R_2: & \mathrm{C} \xrightarrow{k_2} \mathrm{E}+\mathrm{S} & \begin{bmatrix} 1 & 1 & -1 & 0 \end{bmatrix} \\ R_3: & \mathrm{C} \xrightarrow{k_3} \mathrm{P}+\mathrm{E} & \begin{bmatrix} 0 & 1 & -1 & 1 \end{bmatrix} \end{array}$$

The occurence of a reaction is determined by a probability per time, which depends on the number of participating molecules and kinetic parameters. In Fig. 8, a stochastic trajectory for this biochemical reaction is shown.

In systems with small numbers of components even the stochastic and kinetic mean value may differ. In biological systems, some stochastic phenomena are caused by a combination of complexity and nonlinearity. Stochastic resonances are used to amplify weak signals and to improve the processing of information (53, 54). The influence of fluctuations on the mean results also in changes in the sensitivity of signaling pathways. So, fluctuations can make a gradual response mechanism (using kinetic approaches) work like a threshold mechanism (55). Such behavior is called stochastic focusing. However, fluctuations may also decrease the

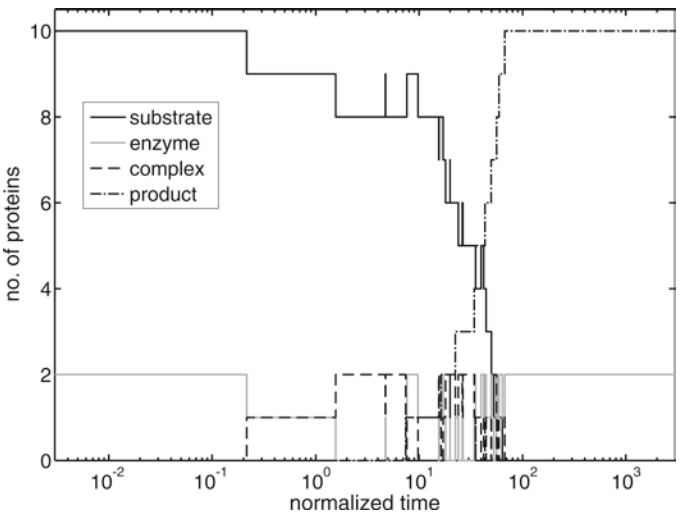


Fig. 8. The trajectory from a single stochastic simulation in semilogarithmic scale. We normalized the time against the time to reach a quasi-steady state due to the saturation of the enzyme.

sensitivity which is called stochastic defocusing (56). In multistable systems, giant fluctuations can populate all existing stable states. This leads to a multimodal distribution of states (49). Such behavior cannot be observed in a kinetic description. The occurrence of giant fluctuations is estimated from the escape time (49, 50). This stochastic switching may cause the development of subpopulations in an initially homogeneous population (56, 57). Further phenomena are noise-induced oscillations and noise-induced multistability, but also the suppression of oscillations or multistability.

Different approaches were developed to simulate the effects of fluctuations. In the framework of master equations (49, 50), the system's dynamics are described as discrete changes of the systems state. From the knowledge of possible states, transitions, and related transition probabilities per time (also called propensities) one can calculate the temporal evolution of the distribution of states. Because the master equation is a mesoscopic representation, the transition probabilities cannot be derived within this framework. Thus, they have to adopt from the kinetic theory. The master equation can be solved analytically only for some simple systems, which is why approximations like the Fokker-Planck equation (49, 50) have been developed.

The Langevin approach (49, 50) combines the kinetic approach with an additive stochastic force term, leading to a stochastic differential equation (50, 58).

Recently, a hierarchical approach was developed which uses coupled moments of the distribution function to calculate the system dynamics. Interestingly, the expressions describing the moments can be derived from kinetic theory. The zero-order moment corresponds to the mean or expectation value. The first moment (variance) is determined by a drift or flux term. Further moments can be included to the hierarchy. In its Two-Moment-Approximation (59, 60), the mean and the linearized variance are considered, only. Nevertheless, due to the coupling of both moments the consequences of fluctuations can be investigated as recently demonstrated on the example of the cell cycle (61).

Much progress has been made in the last decades to develop numerical methods that solve stochastic systems. However, the cost in computing power can still be quite high. Furthermore, such system-theoretic approaches such as sensitivity analysis or bifurcation analysis are either not available or not well developed for stochastic systems. The estimation of parameter values from experimental data in combination with stochastic simulations is also unsolved.

In the 1970s, direct simulation methods were developed to simulate the Chemical Master Equation (CME). Using Monte Carlo methods, these simulations compute the time of the next reaction and which reaction will occur. Subsequently, the system's state is changed according to the computed reaction. Repeating this procedure until a stop criterion is reached, a trajectory is obtained, see also Fig. 8. To get valuable information from such

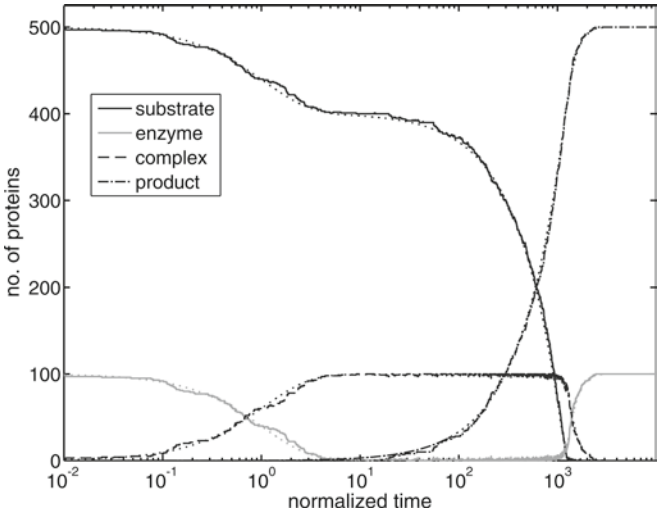


Fig. 9. Stochastic analysis of the enzyme kinetic reaction. The *left figure* shows a comparison between the stochastic mean over 50 simulation runs (*solid lines*) and the kinetic mean (*dotted lines*). The right figure shows a comparison of a single stochastic simulation for high molecule numbers (*solid lines*) and the kinetic mean (*dotted line*). In our example, stochastic mean and high particle number behavior are appropriately described by the kinetic mean. However, we emphasize that this conclusion cannot be generalized.

stochastic simulations, one has to repeat the simulation many times to determine the distribution function. Subsequently, mean, variance, and further moments of the distribution function can be calculated. Hence, this approach is very cost intensive with respect to computing power. The results should be compared to the average of the kinetic approach and experimental data as shown in Fig. 9.

The most commonly applied algorithm to simulate biochemical reaction networks was developed by Gillespie (62). In its original formulation, the algorithm considers every possible reaction. This can be very time-consuming, especially if reactions with different timescale are present in the system. Hence, the algorithm was improved using well-defined jumps, τ-leap (63, 64), and K-leap (65), with respect to the systems time. Additionally, varying system's volume or changes in the system's temperature can be incorporated into the algorithm (66) as well as delays (67).

5.3. Molecular Dynamics: The View Through the Computational Microscope

Scientists interested in systems consisting of many interacting parts dreamed for a time to have the opportunity to investigate and follow every single component in time and space (68). The rapid progress in computing power, especially in parallel computing and supercomputers, allows the investigation of more and more complex systems using such a detailed simulation. Nevertheless, the considered biological systems were restricted to

well-defined subcellular compartments and/or small timescales. In the past, molecular dynamics (MD) simulations in biology mainly focused on structural, functional, and folding properties of proteins, see f.i. (69). Recent projects extend this method to biochemical reactions (70), e.g., ligand–receptor signaling (71) and the influence of spatial aspects (e.g., cellular architecture and diffusion) to signal transduction (72).

Toward this end, MD simulations compute the changes of every single component in the system, say atoms and molecules, with respect to position, momentum, and state (73). Starting from a known system configuration, the next configuration is calculated from the forces resulting from the interactions of the system components. Thus, such simulations require sophisticated methods in numeric computation and data handling (74–76). An introduction to the theoretical and numerical fundamentals of MD simulations can be found in (73–75). See Fig. 10.

Even if the approach of MD simulations seems to be deterministic, it inherently contains stochastic features. First, the initial state of the system is randomly distributed. Thus, also MD simulations have to be repeated to obtain meaningful data. Second, chemical reactions are incorporated in a probabilistic manner because of the uncertainty of quantum-mechanical events. Third, the approximate solution of the equation motion of the many particle system adds artificial noise to the considered biological system.

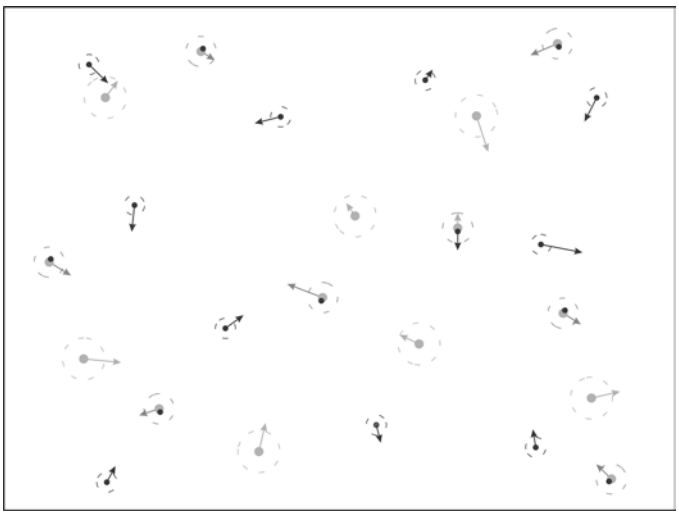


Fig. 10. Schematic drawing of a snapshot from a molecular dynamics (MD) simulation of a reversible bimolecular reaction. Every molecule is represented by its position and momentum. If two molecules collide with enough energy to reach the reaction cross-section which is usually much smaller than the collision cross-section (*dashed circles*), they may bind to each other. Additionally, complexes can dissociate during a time step.

The detailed information of MD simulations with respect to time, space, and state of the system allows a direct visualization and opens new perspectives for the understanding and representation of biological systems. Nevertheless, the very high computing costs restrict the range of applications. Statistical analyzes of the data from MD simulations provide trajectories like those from stochastic simulations, or means and variances which can be compared to the results from differential equations.

System-theoretic tools are not available for MD simulations. Hence, some previous knowledge obtained usually from conventional approaches is required. Parameters used in the simulations have to be transferred from other approaches or measurements. Changes in the system conditions require new rounds of simulation.

6. Conclusions/ Outlook

The modeling of cellular systems is a major research activity related to systems biology. Different approaches, including conventional kinetic equations or stochastic representations, are used to describe the considered system in a mathematical model. Once a set of equations has been established, system-theoretic tools are used to analyze and simulate the system's behavior. These investigations may identify crucial components and biological processes, unravel (so far unknown) regulatory structures, or discover optimization principles. Furthermore, model predictions enable the design of experiments that test hypotheses. This interplay between experiment and theory opens new opportunities for the life sciences.

References

1. Aderem A (2005) Systems biology: its practice and challenges. Cell 121:511–513
2. Ideker T (2004) Systems biology 101 – what you need to know. Nat Biotechnol 22:473–475
3. Lottka AJ (1925) Elements of physical biology. Dover Publications. reprinted by Dover Publications as Elements of mathematical biology (1956)
4. Volterra V (1926) Fluctuations in the abundance of a species considered mathematically. Nature 118:558–60
5. Schrödinger E (1944) What is life. Cambridge University Press, Cambridge
6. Mesarovich M (1968) System theory and biology. Springer, New York
7. Haken H (1983) Synergetics, an introduction: nonequilibrium phase transitions and self-organization in physics, chemistry, and biology, 3rd edn, Springer, New York
8. Bertalanffy LV (1976) General systems theory: foundations, development, applications. Revised Edition, George Braziller, Inc
9. Ferrell JE Jr (2009) Q&A systems biology. J Biol 8:2
10. Williamson MP (2005) Systems biology: will it work. Biochem Soc Trans 33:506–509
11. Segel I (1993) Enzyme kinetics. Wiley, New York
12. Cornish-Bowden A (2004) Fundamentals of enzyme kinetics. Portland Press

13. Wright MR (2004) An introduction to chemical kinetics. Wiley, New York
14. Atkins P, de Paula J (2002) Atkins' physical chemistry. Oxford University Press
15. Voit EO (2000) Computational analysis of biochemical systems. a practical guide for biochemists and molecular biologists. Cambridge University Press, Cambridge
16. Vera J, Balsa-Canto E, Wellstead P, Banga JR, Wolkenhauer O (2007) Power-law models of signal transduction pathways. Cell Signal 19:1531–1541
17. Alberty R, Massey V (1954) On the interpretation of the pH variation of the maximum initial velocity of an enzyme-catalyzed reaction. Biochim Biophys Acta 13:347–353
18. Fowler R, Guggenheim EA (1952) Statistical thermodynamics. Cambridge University Press, Cambridge
19. Rosser WGV (1982) An introduction to statistical physics. Wiley, New York
20. Walter GG, Gontreas M (1999) Compartmental modeling with networks. Birkhäuser
21. Godfrey K (1983) Compartmental models and their application. Academic
22. Savageau MA (1976) Biochemical systems analysis. Addison-Wesley Publishing Company
23. Cussler EL (1997) Diffusion – mass transfer in fluid systems. Cambridge University Press, Cambridge
24. Volpert V, Petrovskii S (2009) Reaction–diffusion waves in biology. Phys Life Rev 6:267–310
25. Farlow SJ (1993) Partial differential equations for scientists and engineers. Dover Publications Inc.
26. Zienkiewics OC, Taylor RL, Zhu JZ (2005) The finite element method. Its basis and fundamentals, 6th edn. Butterworth Heinemann
27. Johnson C (2009) Numerical solution of partial differential equations by the finite element method. Dover Publications Inc.
28. Perko L (2008) Differential equations and dynamical systems, 3rd edn. Springer, New York
29. Jordan DW, Smith P (1999) Nonlinear ordinary differential equations. Oxford University Press
30. Hirsch MW, Smale S, Devaney RL (2004) Differential equations, dynamical systems and an introduction to chaos. Elsevier
31. Saltelli A, Chan K, Scott E (2000) Sensitivity analysis. Wiley, New York
32. Heinrich R, Schuster S (1996) The regulation of cellular systems. Chapman & Hall
33. Fell D (1997) Unterstanding the control of metabolism. Portland Press
34. Hofmeyr J-H (2001) Metabolic control analysis in a nutshell. In: Proceedings on the second international conference on systems biology, pp 291–300
35. Kitano H (2007) Towards a theory of biological robustness. Mol Syst Biol 3:137
36. Hunter P (2009) Robust yet flexible. EMBO Rep 10:949–952
37. Stelling J, Sauer U, Doyle FJ, Doyle J (2004) Robustness of cellular functions. Cell 118: 675–685
38. Kuznetsov YA (1998) Elements of applied bifurcation theory, 2nd edn. Springer, New York
39. Strogatz SH (1994) Nonlinear dynamics and chaos. With applications to physics, biology, chemistry, and engineering. Westview
40. Tyson J, Chen K, Novak B (2003) Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell. Curr Opin Cell Biol 15:221–231
41. Smits WK, Veening JW, Kuipers O (2007) Phenotypic variation and bistable switching in bacteria. In: El-Sharoud W (ed) Bacterial physiology. A molecular approach. Springer, New York
42. Jeffrey A (1993) Linear algebra and ordinary differential equations. CRC Press
43. Schlögl F (1971) On the stability of steady states. Z Phys 243:303–310
44. Sastry S (1999) Nonlinear systems – analysis, stability, and control. Springer, New York
45. Sontag ED (1998) Mathematical control theory. Springer, New York
46. Dutton K, Thompson S, Barraclough B (1997) The art of control engineering. Addison-Wesley Longman Publishing Co., Inc
47. Van den Bos A (2007) Parameter estimation for scientists and engineers. Wiley & Sons
48. Wolberg J (2006) Data analysis using the method of least squares: extracting the most information from experiments. Springer, New York
49. Van Kampen NG (2007) Stochastic processes in physics and chemistry, 3rd edn. North-Holland Personal Library
50. Gardiner CW (2004) Handbook of stochastic methods for physics, chemistry and the natural sciences, 2nd edn. Springer, New York
51. Allen LJS (2002) An introduction to stochastic processes with biology applications. Prentice-Hall
52. Ullah M, Wolkenhauer O (2010) Stochastic approaches in systems biology. Wiley interdisciplinary reviews: systems biology and medicine. (in press) doi: 10.1002/wsbm.78

53. Hänggi P (2002) Stochastic resonance in biology. How noise can enhance detection of weak signals and help improve biological information processing. Chemphyschem 3:285–290
54. Bulsara AR, Gammaitoni L (1996) Tuning in to noise. Phys Today 49:39–45
55. Paulsson J, Berg OG, Ehrenberg M (2000) Stochastic focusing: fluctuation-enhanced sensitivity of intracellular regulation. Proc Natl Acad Sci U S A 97:7148–7153
56. Leisner M, Stingl K, Frey E, Maier B (2008) Stochastic switching to competence. Curr Opin Microbiol 11:553–559
57. Acar M, Mettetal JT, van Oudenaarden A (2008) Stochastic switching as a survival strategy in fluctuating environments. Nat Genet 40:471–475
58. Øksendahl B (2003) Stochastic differential equations: an introduction with applications, 6th edn. Springer, New York
59. Goutsias J (2007) Classical versus stochastic kinetics modeling of biochemical reaction systems. Biophys J 92:2350–2365
60. Gomez-Uribe CA, Verghese GC (2007) Mass fluctuation kinetics: capturing stochastic effects in systems of chemical reactions through coupled mean-variance computations. J Chem Phys 126:024109
61. Ullah M, Wolkenhauer O (2009) Investigating the two-moment characterisation of subcellular biochemical networks. J Theor Biol 260:340–352
62. Gillespie DT (1977) Exact stochastic simulation of coupled chemical reactions. J Phys Chem 81:2340–2361
63. Gillespie DT (2001) Approximate accelerated stochastic simulation of chemically reacting system. J Chem Phys 115:1716–1733
64. Cao Y, Gillespie DT, Petzold LR (2006) Efficient step size selection for the tau-leaping simulation method. J Chem Phys 124:044109
65. Cai X, Xu Z (2007) K-leap method for accelerating stochastic simulation of coupled chemical reactions. J Chem Phys 126: 074102
66. Gibson MA, Bruck J (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. J Phys Chem A 104:1876–1889
67. Cai X (2007) Exact stochastic simulation of coupled chemical reactions with delays. J Chem Phys 126:124108
68. Schlick T (1996) Pursuing Laplace's vision on modern computers. In: Mesirov JP, Schulten K, Summers DW (eds), Mathematical applications to biomolecular structure and dynamics, Springer, New York, pp 218–247
69. Karplus M, McCammon JA (2002) Molecular dynamics simulations of biomolecules. Nat Struct Biol 9:646–652
70. Warshel A (2002) Molecular dynamics simulations of biological reactions. Acc Chem Res 35:385–395
71. Grigera JR (2002) Molecular dynamics simulation for ligand–receptor studies. Carbohydrates interactions in aqueous solutions. Curr Pharm Des 8:1579–1604
72. Klann MT, Lapin A, Reuss M (2009) Stochastic simulation of signal transduction: impact of cellular architecture on diffusion. Biophys J 96:5122–5129
73. Allen MP, Tildesley DJ (1989) Computer simulation of liquids. Oxford University Press
74. Rapaport DC (2004) The art of molecular dynamics simulation, 2nd edn. Cambridge University Press, Cambridge
75. Haile JM (1997) Molecular dynamics simulation: elementary methods. Wiley & Sons Inc
76. Pfalzner S, Gibbon P (1996) Many-body tree methods in physics. Cambridge University Press, Cambridge

INDEX

A

B

Michael Hamacher et al. (eds.), *Data Mining in Proteomics: From Standards to Applications*, Methods in Molecular Biology, vol. 696, DOI 10.1007/978-1-60761-987-1, © Springer Science+Business Media, LLC 2011

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V