Data Mining Techniques for the Life Sciences

Edited by

Oliviero Carugo
Frank Eisenhaber

METHODS IN MOLECULAR BIOLOGY™

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

Data Mining Techniques for the Life Sciences

Edited by

Oliviero Carugo

University of Pavia, Pavia, Italy
Vienna University, Vienna, Austria

Frank Eisenhaber

Bioinformatics Institute, Agency for Science, Technology and Research, Singapore

Humana Press

Editors
Oliviero Carugo
Universität Wien
Max F. Perutz Laboratories
    GmbH
Structural & Computational
    Biology Group
Dr. Bohr-Gasse 9
1030 Wien
Campus-Vienna-Biocenter
Austria
oliviero.carugo@univie.ac.at

Frank Eisenhaber
Bioinformatics Institute (BII)
Agency for Science, Technology and
    Research (A*STAR)
30 Biopolis Street, Singapore 138671
#07-01 Matrix Building
Singapore
franke@bii.a-star.edu.sg

Printed on acid-free paper

springer.com

Preface

Most life science researchers will agree that biology is not a truly theoretical branch of science. The hype around computational biology and bioinformatics beginning in the nineties of the 20th century was to be short lived (1, 2). When almost no value of practical importance such as the optimal dose of a drug or the three-dimensional structure of an orphan protein can be computed from fundamental principles, it is still more straightforward to determine them experimentally. Thus, experiments and observations do generate the overwhelming part of insights into biology and medicine. The extrapolation depth and the prediction power of the theoretical argument in life sciences still have a long way to go.

Yet, two trends have qualitatively changed the way how biological research is done today. The number of researchers has dramatically grown and they, armed with the same protocols, have produced lots of similarly structured data. Finally, high-throughput technologies such as DNA sequencing or array-based expression profiling have been around for just a decade. Nevertheless, with their high level of uniform data generation, they reach the threshold of totally describing a living organism at the biomolecular level for the first time in human history. Whereas getting exact data about living systems and the sophistication of experimental procedures have primarily absorbed the minds of researchers previously, the weight increasingly shifts to the problem of interpreting accumulated data in terms of biological function and biomolecular mechanisms. It is possible now that biological discoveries are the result of computational work, for example, in the area of biomolecular sequence analysis and gene function prediction (2, 3).

Electronically readable biomolecular databases are at the heart of this development. Biological systems consist of a giant number of biomacromolecules, both nucleic acids and proteins together with other compounds, organized in complexes pathways, subcellular structures such as organelles, cells, and the like that is interpreted in a hierarchical manner. Obviously, much remains unknown and not understood. Nevertheless, electronic databases organize the existing body of knowledge and experimental results about the building blocks, their relationships, and the corresponding experimental evidence in a form that enables the retrieval, visualization, comparison, and other sophisticated analyses. The significance of many of the pieces of information might not be understood when they enter databases; yet, they do not get lost and remain stored for the future.

Importantly, databases allow analyses of the data in a continuous workflow detached from any further experimentation itself. In a formal, mathematical framework, researchers can now develop theoretical approaches that may lead to new insights at a meta-analytic level. Indeed, results from many independently planned and executed experiments become coherently accessible with electronic databases. Together, they

can provide an insight that might not be possible from the individual pieces of information in isolation. It is also interesting to see this work in a human perspective: in the framework of such meta-analyses, people of various backgrounds who have never met essentially cooperate for the sake of scientific discoveries via database entries. From the technical viewpoint, because the data are astronomically numerous and the algorithms for their analysis are complex, the computer is the natural tool to help researchers in their task; yet, it is just a tool and not the center of the intellectual concept. The ideas and approaches selected by researchers driven by the goal to achieve biologically relevant discoveries remain the most important factor. Due to the need of computer-assisted data analysis, electronic availability of databases, the possibility of their download for local processing, the uniform structure of all database entries as well as the accuracy of all pieces of information including that for the level of experimental evidence are of utmost importance. To allow curiosity-driven research for as many as possible researchers and to enable the serendipity of discovery, the full public availability of the databases is critical.

Nucleic acid and protein sequence and structure databases were the first biological data collections in this context; the emergence of the sequence homology concept and the successes of gene function prediction are scientific outcomes of working with these data (3). To emphasize, they would be impossible without prior existence of the sequence databases. Thus, biological data mining is going to become the core of biological and biomedical research work in the future, and every member of the community is well advised to keep himself informed about the sources of information and the techniques used for "mining" new insights out of databases. This book is thought as a support for the reader in this endeavor.

The variety of biological databases reflects the complexity of and the hierarchical interpretation we use for the living world as well as the different techniques that are used to study them (4). The first section of the book is dedicated to describing concepts and structures of important groups of databases for biomolecular mechanism research. There are databases for sequences of genomes, nucleic acids such as RNAs and proteins, and biomacromolecular structures. With regard to proteins, databases collect instances of sequence architectural elements, thermodynamic properties, enzymes, complexes, and pathway information. There are many more specialized databases that are beyond the scope of this book; the reader is advised to consult the annual January database supplement of the journal "*Nucleic Acids Research*" for more detail (5).

The second section of this book focuses on formal methods for analyzing biomolecular data. Obviously, biological data are very heterogeneous and there are specific methodologies for the analysis of each type of data. The chapters of this book provide information about approaches that are of general relevance. Most of all, these are methods for comparison (measuring similarity of items and their classification) as well as concepts and tools for automated learning. In all cases, the approaches are described with the view of biological database mining.

The third section provides reviews on concepts for analyzing biomolecular sequence data in context with other experimental results that can be mapped onto genomes. The

topics range from gene structure detection in genomes and analyses of transcript sequences over aspects of protein sequence studies such as conformational disorder, 2D, 3D, and 4D structure prediction, protein crystallizability, recognition of post-translational modification sites or subcellular translocation signals to integrated protein function prediction.

It should be noted that the biological and biomedical scientific literature is the largest and possibly most important source of information. We do not analyze the issue here in this book since there is a lot in the flow. Whereas sources such as PUBMED or the Chemical Abstracts currently provide bibliographic information and abstracts, the trend is towards full-text availability. With the help of the open access movement, this goal might be practically achieved in a medium term. The processing of abstracts and full articles for mining biological facts is an area of actively ongoing research and exciting developments can be expected here.

Creating and maintaining a biological database requires considerable expertise and generates an immense work load. Especially maintaining and updating are expensive. Although future success of research in the life sciences depends on the completeness and quality of the data in databases and of software tools for their usage, this issue does not receive sufficient recognition within the community as well as from the funding agencies. Unfortunately, the many academic groups feel unable to continue the maintenance of databases and software tools because funding might cover only the initial development phase but not the continued maintenance. An exit into commercial development is not a true remedy; typically, the access to the database becomes hidden by a system of fees and its download for local processing is excluded. Likewise, it appears important to assess before the creation of the database whether it will be useful for the scientific community and whether the effort necessary for maintenance is commensurate with the potential benefit for biological discovery (6). For example, maintaining programs that update databases automatically is a vastly more efficient way than cases where all entries need to be curated manually in an individual manner.

We hope that this book is of value for students and researchers in the life sciences who wish to get a condensed introduction to the world of biological databases and their applications. Thanks go to all authors of the chapters who have invested considerable time for preparing their reviews. The support of the Austrian GENAU BIN programs (2003–2009) for the editors of this book is gratefully acknowledged.

Oliviero Carugo
Frank Eisenhaber

References

1. Ouzounis, C.A. (2000) Two or three myths about bioinformatics. *Bioinformatics* 17, 853–854
2. Eisenhaber, F. (2006) Bioinformatics: Mystery, astrology or service technology. Preface for "Discovering Biomolecular Mechanisms with Computational Biology", Eisenhaber, F. (Ed.), 1st edition, pp. pp.1–10. Georgetown, New York: Landes Biosciences, Springer

3. Eisenhaber, F. (2006) Prediction of protein function: Two basic concepts and one practical recipe. In Eisenhaber, F. (Ed.), "Discovering Biomolecular Mechanisms with Computational Biology", 1st edition, pp. 39–54. Georgetown, New York: Landes Biosciences, Springer
4. Carugo, O., Pongor, S. (2002) The evolution of structural databases. *Trends Biotech.* 20, 498–501
5. Galperin, M.Y., Cochrane, G.R. (2009) Nucleic acids research annual database issue and the NAR online molecular biology database collection in 2009. *Nucleic Acids Res.* 37, D1–D4
6. Wren, J.D., Bateman, A. (2008) Databases, data tombs and dust in the wind. *Bioinformatics* 24, 2127–2128

Contents

SECTION III: DATABASE ANNOTATIONS AND PREDICTIONS

Contributors

ASA BEN-HUR • *Department of Computer Science, Colorado State University, Fort Collins, CO, USA*

BRUNO CANARD • *Architecture et Fonction des Macromolécules Biologiques, UMR 6098 CNRS et Universités Aix-Marseille I et II, Marseille, France*

OLIVIERO CARUGO • *Department of General Chemistry, Pavia University, Pavia, Italy; Department of Structural and Computational Biology, MFPL – Vienna University, Vienna, Austria*

YING-LEONG CHAN • *Bioinformatics Institute (BII), Agency for Science, Technology, and Research (A*STAR), Singapore*

BIRGIT EISENHABER • *Experimental Therapeutic Centre (ETC), Bioinformatics Institute (BII), Agency for Science, Technology, and Research (A*STAR), Singapore*

FRANK EISENHABER • *Bioinformatics Institute (BII), Agency for Science, Technology, and Research (A*STAR), Singapore*

DMITRIJ FRISHMAN • *MIPS & Helmholz Institute München, Martinsried, Germany*

ANDREAS FUTSCHIK • *Institute of Statistics, University of Vienna, Vienna, Austria*

M. MICHAEL GROMIHA • *Computational Biology Research Center (CBRC), National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan*

JAAP HERINGA • *Centre for Integrative Bioinformatics VU (IBIVU), VU University, Amsterdam, The Netherlands*

IVO L. HOFACKER • *Department of Theoretical Chemistry, University of Vienna, Wien, Austria*

JOEL JANIN • *Yeast Structural Genomics, IBBMC UMR 8619 CNRS, Université Paris-Sud, Orsay, France*

ROMAN A. LASKOWSKI • *EMBL-European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK*

PHILIPPE LIEUTAUD • *Architecture et Fonction des Macromolécules Biologiques, UMR 6098 CNRS et Universités Aix-Marseille I et II, Marseille, France*

TENG-TING LIM • *Bioinformatics Institute (BII), Agency for Science, Technology, and Research (A*STAR), Singapore*

SONIA LONGHI • *Architecture et Fonction des Macromolécules Biologiques, UMR 6098 CNRS et Universités Aix-Marseille I et II, Marseille, France*

SEBASTIAN MAURER-STROH • *Bioinformatics Institute (BII), Agency for Science, Technology, and Research (A*STAR), Singapore*

NICOLA J. MULDER • *National Bioinformatics Network Node, Institute for Infectious Diseases and Molecular Medicine, Faculty of Health Sciences, University of Cape Town, Cape Town, South Africa*

HONG SAIN OOI • *Bioinformatics Institute (BII), Agency for Science, Technology, and Research (A*STAR), Singapore*

GRAZIANO PESOLE • *Dipartimento di Biochimica e Biologia Molecolare "E. Quagliariello", University of Bari, Bari, Italy*

ERNESTO PICARDI • *Dipartimento di Biochimica e Biologia Molecolare "E. Quagliariello", University of Bari, Bari, Italy*

WALTER PIROVANO • *Centre for Integrative Bioinformatics VU (IBIVU), VU University, Amsterdam, The Netherlands*

ANNE POUPON • *Yeast Structural Genomics, IBBMC UMR 8619 CNRS, Université Paris-Sud, Orsay, France*

MICHAEL REBHAN • *Head Bioinformatics Support, Friedrich Miescher Institute for Biomedical Research, Basel, Switzerland*

AKINORI SARAI • *Department of Bioscience & Bioinformatics, Kyushu Institute of Technology (KIT), Iizuka, Japan*

DIETMAR SCHOMBURG • *Department of Bioinformatics and Biochemistry, Technische Universität Carolo-Wilhelmina zu Braunschweig, Braunschweig, Germany*

IDA SCHOMBURG • *Department of Bioinformatics and Biochemistry, Technische Universität Carolo-Wilhelmina zu Braunschweig, Braunschweig, Germany*

GEORG SCHNEIDER • *Bioinformatics Institute (BII), Agency for Science, Technology, and Research (A*STAR), Singapore*

FERNANDA L. SIROTA • *Bioinformatics Institute (BII), Agency for Science, Technology, and Research (A*STAR), Singapore*

PAWEL SMIALOWSKI • *MIPS & Helmholz Institute München, Martinsried, Germany*

TATIANA TATUSOVA • *National Institute of Heath, Bethesda, MD, USA*

CLAUS VOGL • *Institute of Animal Breeding and Genetics, University of Veterinary Medicine Vienna, Vienna, Austria*

STEFAN WASHIETL • *Department of Theoretical Chemistry, University of Vienna, Wien, Austria; EMBL-European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge, UK*

JASON WESTON • *NEC Labs America, Princeton, NJ, USA*

MICHAEL WILDPANER • *Google Switzerland GmbH, Zürich, Switzerland*

ZHENG RONG YANG • *School of Biosciences, University of Exeter, Exeter, UK*

Section I

Databases

Chapter 1

Nucleic Acid Sequence and Structure Databases

Stefan Washietl and Ivo L. Hofacker

Abstract

This chapter gives an overview of the most commonly used biological databases of nucleic acid sequences and their structures. We cover general sequence databases, databases for specific DNA features, noncoding RNA sequences, and RNA secondary and tertiary structures.

Key words: Sequence repositories, nucleic acids databases, RNA structures.

1. Introduction

Both sequence and structure data have experienced exponential growth during the last two decades, a trend that is most likely to continue in the foreseeable future. As a consequence, there is also a growing number of database resources that try to make these data accessible and help with their analysis. Here we give an overview of existing resources for nucleic acid sequences and structures. In addition to the well-known sequence repositories like GenBank, we also cover databases for various functional and other genomic DNA features. In the second part, we describe databases collecting noncoding RNA sequences and their secondary structures, a topic that has received special attention in the past years. Finally, we cover databases of RNA tertiary structures and motifs. Many of the databases mentioned below were published in the database issue of Nucleic Acids Research, which covers new and updated databases every year.

2. Sequence Databases

An overview including Web addresses for the databases discussed in this section is given in **Tables 1.1** and **1.2**.

Table 1.1
General nucleotide sequence databases and DNA databases

Name	URL	Description	References
General nucleotide databases			
EMBL	www.ebi.ac.uk/embl/	Central sequence repository	(1)
GenBank	http://www.ncbi.nlm.nih.gov/Genbank	Central sequence repository	(2)
DNA databank of Japan (DDBJ)	www.ddbj.nig.ac.jp	Central sequence repository	(3)
RefSeq	http://www.ncbi.nlm.nih.gov/RefSeq/	Nonredundant and curated sequences (DNA, RNA, protein) from GenBank	(4)
Transcript structures and alternative splicing			
Alternative splicing and transcript diversity database (ASTD)	www.ebi.ac.uk/astd	Alternative splicing in human, mouse and rat	(5)
Human-transcriptome DataBase for Alternative Splicing (H-DBAS)	www.h-invitational.jp/h-dbas	Alternative spliced human full length cDNAs	(6)
Repeats and mobile elements			
RepBase	http://www.girinst.org/server/RepBase	Eykaryotic repeat sequences, registration required	(53)
STRBase	www.cstl.nist.gov/biotech/strbase/	Short tandem repeats	(7)
TIGR plant repeat database	www.tigr.org/tdb/e2k1/plant.repeats	Plant repeat sequences	(8)
ACLAME	aclame.ulb.ac.be	Prokaryotic mobile genetic elements	(9)

(continued)

Table 1.1 (continued)

Name	URL	Description	References
ISfinder	http://www-is.biotoul.fr	Insertion sequences from eubacteria and archaea	(10)
MICdb	http://www.cdfd.org.in/micas	Prokaryotic microsatellites	(11)
Islander	http://www.indiana.edu/~islander	Prokaryotic genomic islands	(12)
Promoters and regulation			
TRANSFAC	http://www.gene-regulation.com/	Transcription factor binding sites	(13)
JASPAR		Transcription factor binding sites	(14)
SCPD		Promoter sequences in *S. cerevisiae*	(15)
PlantCARE	http://bioinformatics.psb.ugent.be/webtools/plantcare/html	Plant regulatory elements	(16)
RegulonDB	http://www.cifn.unam.mx/Computational_Genomics/regulondb	Gene regulation in *E. coli*	(17)

Table 1.2
RNA sequence databases

Name	URL	Description	References
Noncoding RNA sequences			
Rfam	www.sanger.ac.uk/Software/Rfam	Structural ncRNAs and regulatory elements	(44)
NONCODE	www.noncode.org	ncRNAs from all species	(19)
RNAdb	research.imb.uq.edu.au/rnadb	Mammalian ncRNAs	(18)
fRNAdb	www.ncrna.org	ncRNA meta-database	(20)
mRNA elements			
UTRdb/UTRsite	www.ba.itb.cnr.it/UTR	Elements in untranslated regions	(21)
ARED	rc.kfshrc.edu.sa/ared	AU-rich elements	(22)

Table 1.2 (continued)

Name	URL	Description	References
PolyA_DB	polya.umdnj.edu	Polyadenylation sites	(23)
IRESdb	www.rangueil.inserm.fr/IRESdatabase	Internal ribosome entry sites	(24)
RNA editing			
REDIdb	biologia.unical.it/py_script/search.html	RNA editing sites	(25)
dbRES	bioinfo.au.tsinghua.edu.cn/dbRES	RNA editing sites	(26)
Specific RNA families			
European Ribosomal Database	bioinformatics.psb.ugent.be/webtools/rRNA	Large and small subunit rRNAs	(27)
Ribosomal Database Project	rdp.cme.msu.edu	Large and small subunit rRNAs	(28)
5S ribosomal RNA database	www.man.poznan.pl/5SData	5S rRNAs	(28)
Sprinzl's tRNA compilation	www.tRNA.uni-bayreuth.de	tRNAs	(30)
Genomic tRNA database (GtRDB)		Predicted tRNAs in completely sequenced genomes	–
SRPDB	rnp.uthct.edu/rnp/SRPDB/SRPDB.html	Signal recognition particle RNA	(31)
tmRDB	rnp.uthct.edu/rnp/tmRDB/tmRDB.html	Transfer/messenger (tm)RNAs	(33)
Group I intron sequence and structure Database (GISSD)	http://www.rna.whu.edu.cn/gissd/	Group I self-splicing introns	(34)
Group II intron database	www.fp.ucalgary.ca/group2introns	Group II self-splicing introns	(35)
mirBase	microrna.sanger.ac.uk	Official miRNA repository	(36)
Argonaute	www.ma.uni-heidelberg.de/apps/zmf/argonaute	miRNA resources	(37)
miRNAmap	mirnamap.mbc.nctu.edu.tw	miRNA resources	(38)
miRNApath	lgmb.fmrp.usp.br/mirnapath	miRNA resources	(39)
miRGen	www.diana.pcbi.upenn.edu/miRGen.html	miRNA resources	(40)

Table 1.2 (continued)

Name	URL	Description	References
snoRNA-LBME-db	www-snorna.biotoul.fr	Human snoRNAs	(41)
Plant snoRNA DB	bioinf.scri.sari.ac.uk/ cgi-bin/plant_snorna/ home	Plant snoRNAs	(42)
Artificially selected RNAs			
Aptamer database	aptamer.icmb.utexas.edu	Artificial nucleic acid aptamers from *in vitro* selection experiments	(43)

2.1. General Nucleotide Sequence Databases

There are three general nucleotide sequence database resources of outstanding importance: The EMBL Nucleotide Sequence Database (1) maintained by the European Bioinformatics Institute, GenBank (2) maintained by the US National Center for Biotechnology Information, and the DNA databank of Japan (DDBJ) (3). All different types of nucleotide sequences are considered by EMBL/GenBank/DDBJ. Entries are typically submitted individually by researchers or come from large-scale genomic projects. In close collaboration, the content of all three databases is synchronized on a daily basis to provide one extensive global collection of nucleotide sequences. Database records submitted to one of these databases are guaranteed to remain permanently accessible through a unique accession number and journals generally require all new sequence data to be deposited to EMBL, GenBank or DDBJ. This explains the central importance of this sequence collection and why many other databases described in this chapter build on and refer to entries from EMBL/GenBank/DDBJ.

All three databases provide a Web interface for searching the database as well as direct access to the data for downloading. The most popular interface is probably provided by the NCBI.

When using EMBL/GenBank/DDBJ one has to bear in mind that the entries directly come from thousands of different researchers worldwide and are not extensively reviewed. This results in many redundant entries and variation in sequence quality. The entries usually also contain annotation information of the sequences. Also here, the quality of annotation can vary considerably and the information given can be misleading or in many cases even simply wrong.

As an effort to provide nonredundant, high-quality sequences and annotation for genomes and transcripts, NCBI has started the RefSeq project (4). GenBank entries are systematically organized and annotated using a combination of automatic procedures and manual curation.

2.2. DNA Databases

2.2.1. Transcript Structures and Alternative Splicing

Annotation of coding regions and transcript structures may be given in EMBL/GenBank/DDBJ entries. If available, RefSeq sequences should be used since their annotation is more consistent. Since alternative splicing is common, there may be several entries of different transcripts for one locus. The Alternative Splicing and Transcript Diversity database (ASTD, (5)) is designed to specifically study alternative splicing in human, mouse, and rat. It contains computationally detected and manually curated data sets of splicing events, isoforms, and regulatory motifs associated with alternative splicing. Also the Human-transcriptome DataBase for Alternative Splicing (H-DBAS, (6)) is a database of alternatively spliced transcripts. It provides alternatively spliced transcripts that correspond to completely sequenced and carefully annotated human full-length cDNAs.

2.2.2. Repeats and Mobile Elements

Apart from genes and transcripts, repeats and mobile elements are also important DNA features shaping eukaryotic and prokaryotic genomes. Repbase is a database of prototypic sequences representing repetitive DNA from various eukaryotic species. It is probably the most commonly used repeat database, in particular for identifying (and masking) repeats in genomes using RepeatMasker. Downloading RepBase requires registration and is only free for academic use. STRBase (7) is a database of short tandem DNA repeats maintained by the Institute of Standards and Technology and aimed specifically at the forensic DNA typing community. The TIGR plant repeat database classifies and provides sequences of repeats from numerous plant genera (8). There are also databases for prokaryotic repeats: ACLAME (9), ISfinder (10), MCdb (11), and Islander (12) provide information and sequence data for transposons, insertion elements, prophages, microsatellites, and pathogenicity islands.

2.2.3. Promoters and Regulation

Regulation at the transcriptional level is crucial for understanding gene function. There are many resources available that specifically collect data of regulatory regions in particular transcription factor binding sites (TFBSs). The most popular database resource for transcriptional regulation is TRANSFAC (13). It provides sequence information for transcription factors, experimentally proven binding sites, and regulated genes. It also provides position specific scoring matrices (PSSM) for prediction of TFBSs. A major drawback of TRANSFAC is that only a limited version (reduced functionality and data) is freely available for academic use. To get full access or use it in a nonacademic environment a paid subscription is required. An alternative resource with open data access is JASPAR (14). It also features TFBSs and PSSMs. The data set is much smaller and currently consists of 123 nonredundant and hand-curated profiles. There are specialized TFBS databases for yeast (SCPD, (15)) and plants (PlantCARE, (16)), which do not seem to be updated any more but are still quite commonly used.

Finally, we want to mention RegulonDB (17) that provides information on *prokaryotic* transcriptional regulation specifically on operons and regulons in *Escherichia. coli*.

2.3. RNA Databases

2.3.1. Noncoding RNA Sequences

The most central resource for noncoding RNA sequences is the Rfam database maintained at the Sanger Institute. It is specifically designed for structured RNAs (including *cis*-acting elements, see **Sect. 3**) and currently contains 607 families. It regularly scans primary sequence databases (EMBL) for new sequences which are added to the families. It also contains structure information as well as annotation for all families.

In the past 3 years, three big database projects on noncoding RNAs were started: RNAdb (18), NONCODE (19), and fRNAdb (20). RNAdb and NONCODE manually collect GenBank entries that correspond to noncoding RNAs. RNAdb is specialized to mammalian noncoding RNAs and also provides additional high-throughput datasets of noncoding transcripts as well as computational predictions. fRNAdb is part of the noncoding RNA portal site www.ncrna.org and is basically a meta-database that collects datasets from other databases (Rfam, NONCODE, RNAdb) and high-throughput experiments.

2.3.2. mRNA Elements

UTRdb/UTRsite (21) are database resources for untranslated regions of mRNAs (UTRs). UTRdb contains curated 3' and 5' UTRs from the EMBL nucleotide database including annotation of regulatory elements. A collection of such regulatory elements (sequence or structural patterns) are available in the UTRsite database. We want to mention three additional, more specialized databases for mRNA elements. ARED (22) is specifically dedicated to AU-rich elements which mediate mRNA turnover. PolyA_DB (23) provides data on polyadenylation sites and their locations with respect to the genomic structure of genes as well as *cis*-elements surrounding polyadenylation sites. IRESdb (24) is a database of internal ribosome entry sites which mediate internal translational initiation in viral and some eukaryotic mRNAs.

2.3.3. RNA Editing

RNA editing is a posttranscriptional modification of RNA that changes the sequence of the transcript compared to the DNA template. There are two dedicated databases gathering examples and additional information on different types of RNA editing: REDIdb (25) and dbRES (26).

2.3.4. Specific RNA Families

Databases of ribosomal RNAs have a long tradition since rRNA sequences have been generated already extensively in the early days of nucleotide sequencing for the purpose of molecular phylogenetics. The European Ribosomal Database (27) collects small-subunit and large-subunit sequences from the EMBL nucleotide database. The entries contain both primary and secondary

structure information as well as other information about the sequences such as literature references and taxonomic data. However, it does not seem to be updated regularly any longer. The Ribosomal Database Project (28) is a novel up-to-date resource for small and large-subunit rRNAs that also provides structure annotation as well as online tools for phylogenetic analysis. The 5S ribosomal RNA database (29) specifically contains the 5S rRNA of the large ribosome subunit that is not covered in the other databases. It also provides alignments and structure annotations.

In addition to rRNAs, there are databases for all well-known "classical" noncoding RNA families: Sprinzl and colleagues have put together a widely used compilation of tRNA genes (30) which was first published in 1980 and is still updated. Systematic computational screens for tRNAs using tRNAscanSE are provided for most available sequenced genomes by the genomic tRNA database (GtRDB). Databases containing sequences and structure annotations for the signal recognition particle RNA (SRPDB, (31)), RNAse P (32), tmRNA (tmRNAdb, (33)) group I (GISSD, (34)) and group II introns (35) are available as well.

In the past few years, abundant classes of small RNAs have been detected, most prominently microRNAs (miRNAs). The official database resource for miRNA sequences is mirBase (36). It stores miRNA sequences and provides a systematic nomenclature for novel miRNAs submitted by researchers. MirBase also features a section for computational target predictions for microRNAs across many species. In addition to mirBase, there are several other online resources with similar features (miRNA sequences, target predictions, genomic tools, pathways) including Argonaute (37), miRNAmap (38), miRNApath (39), and miRGen (40).

Also snoRNAs were found to be a class of small RNAs that is more abundant than previously thought. snoRNAs are contained in the general RNA databases like Rfam or NONCODE. In addition, there are two specific databases for human snoRNAs (snoRNA-LBME-db, (41)) and plants (plant snoRNA DB, (42)) including both subfamilies of C/D box and H/ACA box snoRNAs.

2.3.5. Artificial Selected/ Designed RNAs

The aptamer database (43) is a comprehensive resource of artificially selected nucleic acids from in vitro evolution experiments. It contains RNA/DNA aptamers that specifically bind other nucleic acid sequences, proteins, small organic compounds, or even entire organisms.

3. Secondary Structures

The largest general collection of RNA secondary structures is provided by the Rfam database (44). As mentioned above, it collects families of ncRNAs and cis-acting regulatory elements.

For each family, a so-called seed-alignment is manually created. It contains a subset of sequences from different species and a consensus secondary structure. The consensus secondary structure is either derived from experimental data from literature or computationally predicted using various methods generally including covariance analysis. A relatively new database of RNA secondary structure is the RNA Secondary STRucture and statistical ANalysis Database (RNA SSTRAND). It collects known secondary structures from different sources including Rfam and many of the family-specific databases described in **Sect. 2.3.4**. The secondary structures contained in all these databases may contain pseudoknots and noncanonical base-pairs. There are two specialized databases dealing with these aspects of secondary structures. PseudoBase (45) collects known RNA secondary structures with pseudo-knots. NCIR (46) is a compilation of noncanonical interactions in known secondary structures.

4. Tertiary Structures

In spite of recent advances, the number of known nucleic acid tertiary structures lags far behind protein structures. As with proteins, most tertiary structures can be found in the PDB database (47). For researchers interested in nucleic acids, however, the primary resource for atomic resolution tertiary structures is the Nucleic Acid Database, NDB (48) since it provides a more convenient repository that allows complex searches for structures containing nucleic acid-specific features (such as a hairpin loop). As of May 2008, the NDB contained about 3,800 structures (compared to 51,000 structures in the PDB), about half of them are protein nucleic acid complexes and most contain only relatively short RNA or DNA sequences.

The SCOR (structural classification of RNA) database (49) performs a hierarchical classification of RNA structure motifs extracted from X-ray and NMR structures. It currently contains 579 RNA structures with over 500 internal loops and almost 3,000 hairpin loops. It can be browsed by structural classification (loop types), functional classification (e.g., RNA family), as well as tertiary interactions motifs (e.g., kissing hairpins).

In addition, there are a number of smaller databases dedicated to particular tertiary structure motifs, usually extracted from the known tertiary structures in PDB or NDB. The MeRNA database (50), for example, lists all metal-ion binding sites in known structures. The RNAjunction database (51) has

extracted more than 12,000 multiloop structures and kissing hairpin motifs for use in tertiary structure modelling. Similarly, RNA FRAbase (52) allows to search for fragments of known tertiary structures consistent with an input sequence and secondary structure.

All Web addresses for the databases on secondary and tertiary structures can be found in **Table 1.3**.

Table 1.3
Structure databases

Name	URL	Description	References
Secondary structures			
Rfam	www.sanger.ac.uk/ Software/Rfam	Structural ncRNAs and regulatory elements	(44)
RNA SSTRAND	www.rnasoft.ca/sstrand	Collection of RNA secondary structures from various databases	–
PseudoBase	wwwbio.leidenuniv.nl/ ~Batenburg/PKB.html	Known secondary structures with pseudoknots	(45)
NCIR	prion.bchs.uh.edu/ bp_type/	Noncanonical interactions in RNAs	(46)
Tertiary structures			
Nucleic Acid Database (NDB)	ndbserver.rutgers.edu	Atomic resolution tertiary structures of nucleic acids	(48)
Structural Classification of RNA (SCOR)	scor.lbl.gov	Three-dimensional motifs in RNAs	(49)
MeRNA database	http://merna.lbl.gov	Metal ion binding sites in known structures	(50)
RNAjunction	rnajunction.abcc.ncifcrf.gov	Multiloop structures and kissing hairpin motifs	(51)
FRAbase		Three-dimensional fragments of RNA structures	(52)

Acknowledgements

References

1. Kanz, C., Aldebert, P., Althorpe, N., Baker, W., Baldwin, A., Bates, K., Browne, P., van den Broek, A., Castro, M., Cochrane, G., Duggan, K., Eberhardt, R., Faruque, N., Gamble, J., Diez, F. G., Harte, N., Kulikova, T., Lin, Q., Lombard, V., Lopez, R., Mancuso, R., McHale, M., Nardone, F., Silventoinen, V., Sobhany, S., Stoehr, P., Tuli, M. A., Tzouvara, K., Vaughan, R., Wu, D., Zhu, W. and Apweiler, R. (2005) The EMBL Nucleotide Sequence Database. *Nucleic Acids Res* **33**, D29–33

2. Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J. and Wheeler, D. L. (2008) GenBank. *Nucleic Acids Res* **36**, D25–30

3. Miyazaki, S., Sugawara, H., Ikeo, K., Gojobori, T. and Tateno, Y. (2004) DDBJ in the stream of various biological data. *Nucleic Acids Res* **32**, D31–4

4. Pruitt, K. D., Tatusova, T. and Maglott, D. R. (2005) NCBI Reference Sequence (Ref-Seq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res* **33**, D501–4

5. Stamm, S., Riethoven, J. J., Le Texier, V., Gopalakrishnan, C., Kumanduri, V., Tang, Y., Barbosa-Morais, N. L. and Thanaraj, T. A. (2006) ASD: a bioinformatics resource on alternative splicing. *Nucleic Acids Res* **34**, D46–55

6. Takeda, J., Suzuki, Y., Nakao, M., Kuroda, T., Sugano, S., Gojobori, T. and Imanishi, T. (2007) H-DBAS: alternative splicing database of completely sequenced and manually annotated full-length cDNAs based on H-Invitational. *Nucleic Acids Res* **35**, D104–9

7. Ruitberg, C. M., Reeder, D. J. and Butler, J. M. (2001) STRBase: a short tandem repeat DNA database for the human identity testing community. *Nucleic Acids Res* **29**, 320–2

8. Ouyang, S. and Buell, C. R. (2004) The TIGR Plant Repeat Databases: a collective resource for the identification of repetitive sequences in plants. *Nucleic Acids Res* **32**, D360–3

9. Leplae, R., Hebrant, A., Wodak, S. J. and Toussaint, A. (2004) ACLAME: a CLAssification of Mobile genetic Elements. *Nucleic Acids Res* **32**, D45–9

10. Siguier, P., Perochon, J., Lestrade, L., Mahillon, J. and Chandler, M. (2006) ISfinder: the reference centre for bacterial insertion sequences. *Nucleic Acids Res* **34**, D32–6

11. Sreenu, V. B., Alevoor, V., Nagaraju, J. and Nagarajaram, H. A. (2003) MICdb: database of prokaryotic microsatellites. *Nucleic Acids Res* **31**, 106–8

12. Mantri, Y. and Williams, K. P. (2004) Islander: a database of integrative islands in prokaryotic genomes, the associated integrases and their DNA site specificities. *Nucleic Acids Res* **32**, D55–8

13. Matys, V., Kel-Margoulis, O. V., Fricke, E., Liebich, I., Land, S., Barre-Dirrie, A., Reuter, I., Chekmenev, D., Krull, M., Hornischer, K., Voss, N., Stegmaier, P., Lewicki-Potapov, B., Saxel, H., Kel, A. E. and Wingender, E. (2006) TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res* **34**, D108–10

14. Bryne, J. C., Valen, E., Tang, M. H., Marstrand, T., Winther, O., da Piedade, I., Krogh, A., Lenhard, B. and Sandelin, A. (2008) JASPAR, the open access database of transcription factor-binding profiles: new content and tools in the 2008 update. *Nucleic Acids Res* **36**, D102–6

15. Zhu, J. and Zhang, M. Q. (1999) SCPD: a promoter database of the yeast Saccharomyces cerevisiae. *Bioinformatics* **15**, 607–11

16. Lescot, M., Déhais, P., Thijs, G., Marchal, K., Moreau, Y., Van de Peer, Y., Rouzé, P. and Rombauts, S. (2002) PlantCARE, a database of plant cis-acting regulatory elements and a portal to tools for in silico analysis of promoter sequences. *Nucleic Acids Res* **30**, 325–7

17. Salgado, H., Gama-Castro, S., Peralta-Gil, M., Díaz-Peredo, E., Sánchez-Solano, F., Santos-Zavaleta, A., Martínez-Flores, I., Jiménez-Jacinto, V., Bonavides-Martínez, C., Segura-Salazar, J., Martínez-Antonio, A. and Collado-Vides, J. (2006) RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Res* **34**, D394–7

18. Pang, K. C., Stephen, S., Dinger, M. E., Engström, P. G., Lenhard, B. and Mattick, J. S. (2007) RNAdb 2.0–an expanded database of mammalian non-coding RNAs. *Nucleic Acids Res* **35**, D178–82

19. He, S., Liu, C., Skogerbø, G., Zhao, H., Wang, J., Liu, T., Bai, B., Zhao, Y. and Chen, R. (2008) NONCODE v2.0: decoding the non-coding. *Nucleic Acids Res* **36**, D170–2

20. Kin, T., Yamada, K., Terai, G., Okida, H., Yoshinari, Y., Ono, Y., Kojima, A., Kimura, Y., Komori, T. and Asai, K. (2007) fRNAdb: a platform for mining/annotating functional RNA candidates from non-coding RNA sequences. *Nucleic Acids Res* **35**, D145–8

21. Mignone, F., Grillo, G., Licciulli, F., Iacono, M., Liuni, S., Kersey, P. J., Duarte, J., Saccone, C. and Pesole, G. (2005) UTRdb and UTRsite: a collection of sequences and regulatory motifs of the untranslated regions of eukaryotic mRNAs. *Nucleic Acids Res* **33**, D141–6

22. Bakheet, T., Williams, B. R. and Khabar, K. S. (2006) ARED 3.0: the large and diverse AU-rich transcriptome. *Nucleic Acids Res* **34**, D111–4

23. Lee, J. Y., Yeh, I., Park, J. Y. and Tian, B. (2007) PolyA_DB 2: mRNA polyadenylation sites in vertebrate genes. *Nucleic Acids Res* **35**, D165–8

24. Bonnal, S., Boutonnet, C., Prado-Lourenço, L. and Vagner, S. (2003) IRESdb: the Internal Ribosome Entry Site database. *Nucleic Acids Res* **31**, 427–8

25. Picardi, E., Regina, T. M., Brennicke, A. and Quagliariello, C. (2007) REDIdb: the RNA editing database. *Nucleic Acids Res* **35**, D173–7

26. He, T., Du, P. and Li, Y. (2007) dbRES: a web-oriented database for annotated RNA editing sites. *Nucleic Acids Res* **35**, D141–4

27. Wuyts, J., Perrière, G. and Van De Peer, Y. (2004) The European ribosomal RNA database. *Nucleic Acids Res* **32**, D101–3

28. Cole, J. R., Chai, B., Farris, R. J., Wang, Q., Kulam-Syed-Mohideen, A. S., McGarrell, D. M., Bandela, A. M., Cardenas, E., Garrity, G. M. and Tiedje, J. M. (2007) The ribosomal database project (RDP-II): introducing myRDP space and quality controlled public data. *Nucleic Acids Res* **35**, D169–72

29. Szymanski, M., Barciszewska, M. Z., Erdmann, V. A. and Barciszewski, J. (2002) 5S Ribosomal RNA Database. *Nucleic Acids Res* **30**, 176–8

30. Sprinzl, M. and Vassilenko, K. S. (2005) Compilation of tRNA sequences and sequences of tRNA genes. *Nucleic Acids Res* **33**, D139–40

31. Rosenblad, M. A., Gorodkin, J., Knudsen, B., Zwieb, C. and Samuelsson, T. (2003) SRPDB: Signal Recognition Particle Database. *Nucleic Acids Res* **31**, 363–4

32. Brown, J. W. (1999) The Ribonuclease P Database. *Nucleic Acids Res* **27**, 314

33. Zwieb, C., Larsen, N. and Wower, J. (1998) The tmRNA database (tmRDB). *Nucleic Acids Res* **26**, 166–7

34. Zhou, Y., Lu, C., Wu, Q. J., Wang, Y., Sun, Z. T., Deng, J. C. and Zhang, Y. (2008) GISSD: Group I Intron Sequence and Structure Database. *Nucleic Acids Res* **36**, D31–7

35. Dai, L., Toor, N., Olson, R., Keeping, A. and Zimmerly, S. (2003) Database for mobile group II introns. *Nucleic Acids Res* **31**, 424–6

36. Griffiths-Jones, S., Saini, H. K., van Dongen, S. and Enright, A. J. (2008) miRBase: tools for microRNA genomics. *Nucleic Acids Res* **36**, D154–8

37. Shahi, P., Loukianiouk, S., Bohne-Lang, A., Kenzelmann, M., Küffer, S., Maertens, S., Eils, R., Gröne, H. J., Gretz, N. and Brors, B. (2006) Argonaute–a database for gene regulation by mammalian microRNAs. *Nucleic Acids Res* **34**, D115–8

38. Hsu, S. D., Chu, C. H., Tsou, A. P., Chen, S. J., Chen, H. C., Hsu, P. W., Wong, Y. H., Chen, Y. H., Chen, G. H. and Huang, H. D. (2008) miRNAMap 2.0: genomic maps of microRNAs in metazoan genomes. *Nucleic Acids Res* **36**, D165–9

39. Chiromatzo, A. O., Oliveira, T. Y., Pereira, G., Costa, A. Y., Montesco, C. A., Gras, D. E., Yosetake, F., Vilar, J. B., Cervato, M., Prado, P. R., Cardenas, R. G., Cerri, R., Borges, R. L., Lemos, R. N., Alvarenga, S. M., Perallis, V. R., Pinheiro, D. G., Silva, I. T., Brandäo, R. M., Cunha, M. A., Giuliatti, S. and Silva, W. A., Jr (2007) miRNApath: a database of miRNAs, target genes and metabolic pathways. *Genet Mol Res* **6**, 859–65

40. Megraw, M., Sethupathy, P., Corda, B. and Hatzigeorgiou, A. G. (2007) miRGen: a database for the study of animal microRNA genomic organization and function. *Nucleic Acids Res* **35**, D149–55

41. Lestrade, L. and Weber, M. J. (2006) snoRNA-LBME-db, a comprehensive database of human H/ACA and C/D box snoRNAs. *Nucleic Acids Res* **34**, D158–62

42. Brown, J. W., Echeverria, M., Qu, L. H., Lowe, T. M., Bachellerie, J. P., Hüttenhofer, A., Kastenmayer, J. P., Green, P. J., Shaw, P. and Marshall, D. F. (2003) Plant snoRNA database. *Nucleic Acids Res* **31**, 432–5

43. Lee, J. F., Hesselberth, J. R., Meyers, L. A. and Ellington, A. D. (2004) Aptamer database. *Nucleic Acids Res* **32**, D95–100

44. Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S. R. and Bateman, A.

(2005) Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res* **33**, D121–4

45. van Batenburg, F. H., Gultyaev, A. P. and Pleij, C. W. (2001) PseudoBase: structural information on RNA pseudoknots. *Nucleic Acids Res* **29**, 194–5

46. Nagaswamy, U., Larios-Sanz, M., Hury, J., Collins, S., Zhang, Z., Zhao, Q. and Fox, G. E. (2002) NCIR: a database of non-canonical interactions in known RNA structures. *Nucleic Acids Res* **30**, 395–7

47. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. and Bourne, P. E. (2000) The Protein Data Bank. *Nucleic Acids Res* **28**, 235–42

48. Berman, H., Westbrook, J., Feng, Z., Iype, L., Schneider, B. and Zardecki, C. (2003) The Nucleic Acid Database: A repository of three-dimensional structural information about nucleic acids. *Structural Bioinformatics* , 199–216

49. Tamura, M., Hendrix, D. K., Klosterman, P. S., Schimmelman, N. R., Brenner, S. E. and Holbrook, S. R. (2004) SCOR: Structural Classification of RNA, version 2.0. *Nucleic Acids Res* **32**, D182–4

50. Stefan, L. R., Zhang, R., Levitan, A. G., Hendrix, D. K., Brenner, S. E. and Holbrook, S. R. (2006) MeRNA: a database of metal ion binding sites in RNA structures. *Nucleic Acids Res* **34**, D131–4

51. Bindewald, E., Hayes, R., Yingling, Y. G., Kasprzak, W. and Shapiro, B. A. (2008) RNAJunction: a database of RNA junctions and kissing loops for three-dimensional structural analysis and nanodesign. *Nucleic Acids Res* **36**, D392–7

52. Popenda, M., Blazewicz, M., Szachniuk, M. and Adamiak, R. W. (2008) RNA FRABASE version 1.0: an engine with a database to search for the three-dimensional fragments within RNA structures. *Nucleic Acids Res* **36**, D386–91

53. Jurka, J., Kapitonov, V. V., Pavlicek, A., Klonowski, P., Kohany, O. and Walichiewicz, J. (2005) Repbase Update, a database of eukaryotic repetitive elements. *Cytogenet Genome Res* **110**, 462–7

Chapter 2

Genomic Databases and Resources at the National Center for Biotechnology Information

Tatiana Tatusova

Abstract

The National Center for Biotechnology Information (NCBI), as a primary public repository of genomic sequence data, collects and maintains enormous amounts of heterogeneous data. Data for genomes, genes, gene expressions, gene variation, gene families, proteins, and protein domains are integrated with the analytical, search, and retrieval resources through the NCBI Web site. Entrez, a text-based search and retrieval system, provides a fast and easy way to navigate across diverse biological databases.

Customized genomic BLAST enables sequence similarity searches against a special collection of organism-specific sequence data and viewing the resulting alignments within a genomic context using NCBI's genome browser, Map Viewer.

Comparative genome analysis tools lead to further understanding of evolutionary processes, quickening the pace of discovery.

Key words: bioinformatics, genome, metagenome, database, data management system, sequence analysis.

1. Introduction

Recent advances in biotechnology and bioinformatics led to a flood of genomic data and tremendous growth in the number of associated databases. As of February 2008, NCBI Genome Project collection describes more than 2,000 genome sequencing projects: 1,500 Bacteria and Archaea (631 complete genomes, 462 draft assemblies, and 507 in progress) as listed at the NCBI Genome Project site: http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi and almost 500 eukaryotic genomes (23 complete, 195 draft assemblies, and 221 in progress) as listed at http://www.ncbi.nlm.nih.gov/genomes/leuks.cgi.

Information on complete and ongoing genome projects is also available in Genomes OnLine Database (GOLD) (1), a community-supported World Wide Web resource. Hundreds of thousands of genomic sequences for viruses, organelles, and plasmids are available in the three public databases of the International Nucleotide Sequence Database Collaboration [INSDC, www.insdc.org] – EMBL (2), GenBank (3), and the DNA Data Bank of Japan (4). Additional information on biomedical data is stored in an increasing number of various databases. As published in the 15th annual edition of the journal *Nucleic Acid Research* (NAR), also known as Database Issue, the number of databases in 2008 crossed the 1,000 landmark. This issue listed 1,078 databases, 110 more than in the previous year (5). Navigating through the large number of genomic and other related "omic" resources becomes a great challenge to the average researcher. Understanding the basics of data management systems developed for the maintenance, search, and retrieval of the large volume of genomic sequences will provide necessary assistance in traveling through the information space.

This chapter is focused on the infrastructure developed by the National Center for Biotechnology Information over the last 20 years. NCBI, as a primary public repository of genomic sequence data, collects and maintains enormous amounts of heterogeneous data. The databases vary in size, data types, design, and implementation. They cover most of the genomic biology data types including the project description, project sequence data (genomic, transcript, protein sequences), raw sequence reads, and related bibliographical data (6). More recently, NCBI started to collect the results of studies that have investigated the interaction of genotype and phenotype. Such studies include genome-wide association studies, medical sequencing, molecular diagnostic assays, as well as association between genotype and nonclinical traits (7). All these databases are integrated in a single Entrez system and use a common engine for data search and retrieval. This provides researchers with a common interface and simplifies navigation through the large information space.

There are many different ways of accessing genomic data at NCBI. Depending on the focus and the goal of the research project or the level of interest, the user would select a particular route for accessing the genomic databases and resources. These are (1) text searches, (2) direct genome browsing, and (3) searches by sequence similarity. All of these search types enable navigation through precomputed links to other NCBI resources.

This chapter describes the details of text searching and the retrieval system of three major genomic databases, Entrez Genome and Entrez Genome Project and Entrez Protein Clusters, and also illustrates two other methods of accessing the genomic data.

2. Data Flow and Processing

The National Center for Biotechnology Information was established on November 4, 1988, as a division of the National Library of Medicine (NLM) at the National Institutes of Health (NIH) in order to develop computerized processing methods for biomedical research data. As a national resource for molecular biology information, NCBI's mission is to develop automated systems for storing and analyzing knowledge about molecular biology, biochemistry, and genetics; facilitate the use of such databases and software by the research and medical community; coordinate efforts to gather biotechnology information both nationally and internationally; and perform research into advanced methods of computer-based information processing for analyzing the structure and function of biologically important molecules.

The fundamental sequence data resources at NCBI consist of both primary databases and derived or curated databases. Primary databases such as GenBank (3) archive the original submissions that come from large sequencing centers or individual experimentalists. The database staff organizes the data but do not add additional information. Curated databases such as Reference Sequence Collection (8) provide a curated/expert view by compilation and correction of the data. Records in the primary database are analogous to research articles in a journal, and curated databases to review articles. This difference is not always well understood by the users of NCBI sequence data. In response to the users' inquiries, and more specifically to a request from attendees at a 2006 workshop on microbial genomes held at NCBI, the differences between GenBank, RefSeq, and TPA databases have been recently described in the May 2007 issue of the American Society for Microbiology's journal *Microbe* (http://www.asm.org/microbe/index.asp?bid=50523).

In the same way as a review article can present an expert view or provide a result of computational analysis, the databases can be manually curated and/or computationally derived (**Table 2.1**). For more detailed information on all NCBI and database resources see also (6).

The biological sequence information that builds the foundation of NCBI primary databases and curated resources comes from many sources (**Fig. 2.1**).

This section discusses the flow of sequence data, from the management of data submission to the generation of publicly available data products. An information management system that consists of two major components, the ID database and the IQ database, underlies the submission, storage, and access of

Table 2.1
Primary and derived databases at NCBI

| Database type	Database name	Database description
Primary databases	GenBank/EMBL/DDBJ (core nucleotide)	Author submissions of nucleotide (genomic and cDNA) sequence with conceptual translations as appropriate
Primary	GEO	Gene expression experimental data sets
Primary	dbGSS	Genome Survey Sequences
Primary	dbEST	Expressed Sequence Tags
Primary	dbMHC	DNA and clinical data related to the human major histocompatibility complex
Primary	dbRBC	A resource dedicated to the genetics of red blood cell antigens
Primary	dbSNP	Single nucleotide polymorphism
Primary	dbSTS	Sequence tagged sites
Primary	ProbeDB	Registry of nucleic acid reagents
Primary	Trace Archive	Raw trace data from sequencers
Primary	SRA	Short Read Archive
Primary	GenSAT	Gene expression atlas of mouse central nervous system
Primary	CancerChromosomes	Molecular cytogenetic data in cancer
Primary	dbGAP	Phenotype and genometype database
Primary	ProjectDB	
Derived	RefSeq	Curated representative sequence for major molecules of the central dogma
Derived	Genome	Complete and near-complete genomes, chromosomes, and plasmids
Derived	Gene	Gene-centered information from curated RefSeq transcripts, genome annotation
Derived	Homologene	Clusters of related genes from eukaryotic genomes
Derived	Protein Clusters	A collection of related protein sequences
Derived	Protein Neighbors	Database of precalculated protein BLAST hits
Derived	CDD	Conserved protein domains database
Derived	UniGene	Gene-oriented clusters of transcript sequences
Derived	UniSTS	Markers and mapping data
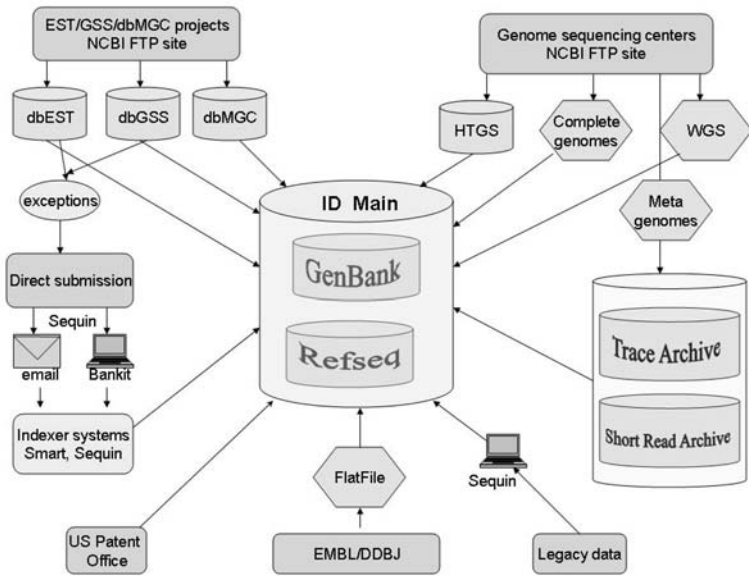
Fig. 2.1. Sources of primary sequence data available at NCBI. Rectangles represent data providers; cylinders represent primary NCBI databases.

GenBank (3), BLAST (9), and other curated data resources [such as the Reference Sequences (8) or Entrez Gene (10)]. Whereas ID handles incoming sequences and feeds other databases with subsets to suit different needs, IQ holds links between sequences stored in ID and between these sequences and other resources.

The data in ID system are stored in Abstract Syntax Notation (ASN.1) format, a standard descriptive language for describing structured information. NCBI has adopted ASN.1 language to describe the biological sequence and all related information (taxonomical, bibliographical) in a structured way. Many NCBI users think of the GenBank flatfile as the archetypal sequence data format. However, within NCBI and especially within the ID internal data flow system, ASN.1 is considered the original format from which reports such as the GenBank flatfile can be generated. As an object-oriented structured language, ASN.1 is easily transformed to other high-level programming languages such as XML, C, and C++. The NCBI Toolkit provides the converters between the data structures. Entrez display options allow to view the data in various text formats including ASN.1, XML, and GenBank flatfiles.

The ID database is a group of standard relational databases that holds both ASN.1 objects and sequence identifier-related information. In the ID database, blobs (binary large objects) are added into a single column of a relational database and are stored and processed as a unit.

Although the columns behave as in a relational database, the information that makes each blob, such as biological features, raw sequence data, and author information, is neither parsed nor split out. In this sense, the ID database can be considered as a hybrid database that stores complex objects.

The IQ database is a Sybase data-warehousing product that preserves its SQL language interface, but which inverts its data by storing them by column, not by row. Its strength is in its ability to increase speed of searches based on anticipated indexing. This nonrelational database holds links between many different objects.

3. Text Search and Retrieval System: Entrez

3.1. Organizing Principles

Entrez is the text-based search and retrieval system used at NCBI for all of the major databases, and it provides an organizing principle for biomedical information. Entrez integrates data from a large number of sources, formats, and databases into a uniform information model and retrieval system. The actual databases from which records are retrieved and on which the Entrez indexes are based have different designs, based on the type of data, and reside on different machines. These will be referred to as the "source databases." A common theme in the implementation of Entrez is that some functions are unique to each source database, whereas others are common to all Entrez databases.

An Entrez "node" is a collection of data that is grouped and indexed together. Some of the common routines and formats for every Entrez node include the term lists and posting files (i.e., the retrieval engine) used for Boolean queries, the links within and between nodes, and the summary format used for listing search results in which each record is called a DocSum. Generally, an Entrez query is a Boolean expression that is evaluated by the common Entrez engine and yields a list of unique ID numbers (UIDs), which identify records in an Entrez node. Given one or more UIDs, Entrez can retrieve the DocSum(s) very quickly.

3.1.1. Query Examples

Each Entrez database ("node") can be searched independently by selecting the database from the main Entrez Web page (http://www.ncbi.nlm.nih.gov/sites/gquery) (*see* **Fig. 2.2**). Typing a query into a text box provided at the top of the Web page and clicking the "Go" button will return a list of DocSum records that match the query in each Entrez category. These include nucleotides, proteins, genomes, publications (PubMed), taxonomy, and many other databases. The numbers of results returned in each category are provided on a single summary page and provide the

Fig. 2.2. Cross database search Web Entrez interface. The counts next to the database description show the number of the records in each database matching the simple text query "mouse."

user with an easily visible view of the results in each of ~35 databases. The results are presented differently in each database but within the same framework, which includes the common elements such as search bar, display options, page formatting, and links.

In processing a query, Entrez parses the query string into a series of tokens separated by spaces and Boolean operators (AND, NOT, OR). An independent search is performed for each term, and the results are then combined according to the Boolean operators.

Query uses the following syntax: term [field] OPERATOR term [field] where "term" refers to the search terms, "field" to the search field defined by specific Entrez database, and "OPERATOR" to the Boolean operators.

More sophisticated searches can be performed by constructing complex search strategies using Boolean operators and the various functions listed below, provided in the Feature Bar:

"Limits" restricts search terms to a specific search field.

"Preview/Index" allows users to view and select terms from search field indexes and to preview the number of search results before displaying citations.

"History" holds previous search strategies and results. The results can be combined to make new searches.

"Clipboard" allows users to save or view selected citations from one search or several searches.

"Details" displays the search strategy as it was translated by query engine, including error messages.

More information about Entrez system can be found from NCBI online Help Manual at http://www.ncbi.nlm.nih.gov/books/bv.fcgi?rid=helpentrez.chapter.EntrezHelp.

The main goals of the information system are reliable data storage and maintenance, and efficient access to the information. The retrieval is considered reliable if the same information that was deposited can be successfully retrieved. The Entrez system goes beyond that by providing the links between the nodes and pre-computing links within the nodes. The links made within or between Entrez nodes from one or more UIDs (Unique IDentifier) are also a function across all Entrez source databases. There are three different linking mechanisms described below.

3.1.2. Links Between the Nodes

The power of Entrez organization lies in the connections between the individual nodes that increase the information space. These links, created during indexing, are reciprocal and stored in a special database, for example, links between genome sequence records and the corresponding genome project. Links can also be provided by the original submitters, for example, links between a nucleotide sequence and a publication (PMID). Links between nucleotide and protein sequences (conceptual translation) of the annotated coding region can also be provided by the original submitters. **Figure 2.3** shows the diagram of the Entrez databases and the connections between them.

3.1.3. Links Within the Nodes

Entrez data can be also integrated by calculating the relationships between the records in a single database. For example, nucleotide and protein sequences can be linked by sequence similarity. The similarity is calculated using BLAST (9), stored in a special database, and made readily available in Entrez via the "Related Sequences" link. In PubMed, the inter-database links are calculated by comparing the frequency terms of the document. The similarity between two documents is based on the number of the
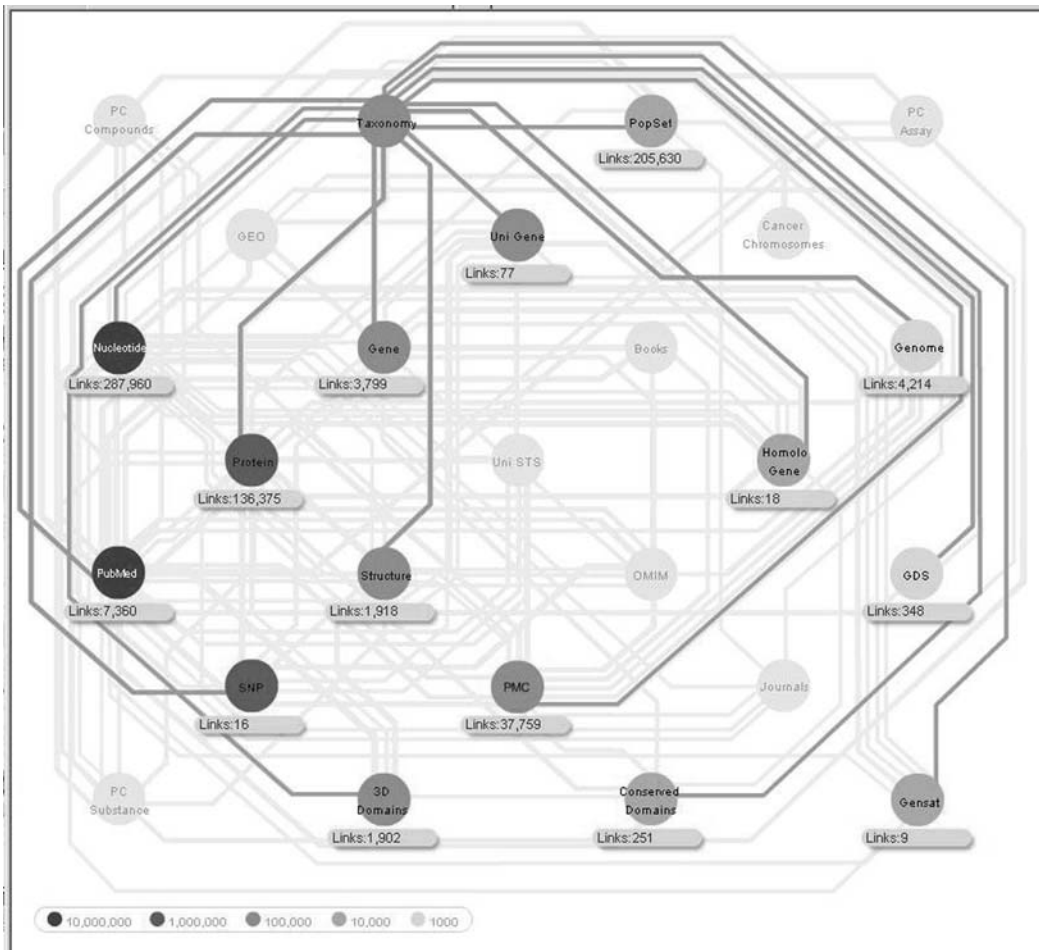
Fig. 2.3. The diagram of the Entrez databases and the connections between them. Each database is represented by a colored circle, where the color indicates the approximate number of records in the database.

weighted terms the two documents have in common. The highest scoring documents can be viewed for each document by selecting Related Articles from the Links menu.

3.1.4. Links Outside the Nodes

Links to outside resources are available through LinkOut, a special service of the Entrez system. It allows relevant outside online resources to link directly to the records in Entrez system. The outside users provide a URL, a resource name, the UID of the record they wish to link to, and a brief description of their Web site in a simple XML format. The request is processed automatically and links are added to the corresponding records in Entrez. This resource gives the end user a central place to look for the information available at NCBI and easily explore the relevant resources.

3.2. Tools for Advanced Users

The Entrez Programming Utilities (eUtils) are a set of eight server-side programs that provide a stable interface to the Entrez query and database system. The eUtils use a fixed URL syntax that translates a standard set of input parameters into the values necessary for various NCBI software components to search for and retrieve data and represent a structured interface to the Entrez system databases.

To access these data, a piece of software first posts an eUtils URL to NCBI, then retrieves the results of this posting, after which it processes the data as required. The software can thus use any computer language that can send a URL to the eUtils server and interpret the XML response, such as Perl, Python, Java, and C++. Combining eUtils components to form customized data pipelines within these applications is a powerful approach to data manipulation. More information and training on this process are available through a course on NCBI Powerscripting: http://www.ncbi.nlm.nih.gov/Class/PowerTools/eutils/course.html.

4. Genomic Databases

The genome sequencing era that started about 20 years ago has brought into being a range of genome resources. Genomic studies of model organisms give insights into understanding of the biology of humans enabling better prevention and treatment of human diseases. Comparative genome analysis leads to further understanding of fundamental concepts of evolutionary biology and genetics. A review on genome resources (11) reports on a selection of genomes of model species – from microbes to human. Species-specific genomic databases comprise a lot of invaluable information on genome biology, phenotype, and genetics. However, primary genomic sequences for all the species are archived in public repositories that provide reliable, free, and stable access to sequence information. In addition, NCBI provides several genomic biology tools and online resources, including group-specific and organism-specific pages that contain links to many relevant Web sites and databases (see **Table 2.2** for the list of available resources and URLs).

4.1. Trace Repositories

Most of the data generated in genome sequencing projects is produced by whole genome shotgun sequencing, resulting in random short fragments (traces).

For many years, the traces (raw sequence reads) remained out of the public domain because the scientific community has focused its attention primarily on the end product: the fully assembled final genome sequence. As the analysis of genomic data progressed, it became necessary to go back to the experimental evidence that underlies the genome sequence to see if there is any ambiguity or uncertainty about the sequence.

Table 2.2
Web genome resources at NCBI

Trace Archive	http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?
Assembly Archive	http://www.ncbi.nlm.nih.gov/Traces/assembly/assmbrowser.cgi?
Short Read Archive (SRA)	http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?
Entrez (cross-database search)	http://www.ncbi.nlm.nih.gov/sites/gquery
Genomic Biology	http://www.ncbi.nlm.nih.gov/Genomes/
Fungal Genome Central	http://www.ncbi.nlm.nih.gov/projects/genome/guide/fungi/
Microbial genomes	http://www.ncbi.nlm.nih.gov/genomes/MICROBES/microbial_taxtree.html
Organelles	http://www.ncbi.nlm.nih.gov/genomes/ORGANELLES/organelles.html
Plant Genome Central	http://www.ncbi.nlm.nih.gov/genomes/PLANTS/PlantList.html
Influenza Virus Resource	http://www.ncbi.nlm.nih.gov/genomes/FLU/FLU.html
Retrovirus Genomes	http://www.ncbi.nlm.nih.gov/retroviruses/
Viral genomes	http://www.ncbi.nlm.nih.gov/genomes/VIRUSES/viruses.html
Genomic BLAST	http://www.ncbi.nlm.nih.gov/sutils/genom_table.cgi
Concise BLAST	http://www.ncbi.nlm.nih.gov/genomes/prokhits.cgi
gMap	http://www.ncbi.nlm.nih.gov/sutils/gmap.cgi
Map Viewer	http://www.ncbi.nlm.nih.gov/projects/mapview/
ProtMap	http://www.ncbi.nlm.nih.gov/sutils/protmap.cgi
TaxPlot	http://www.ncbi.nlm.nih.gov/sutils/taxik2.cgi

4.1.1. Trace Archive

To meet these needs, NCBI and The Wellcome Trust Genome Campus in Hinxton, United Kingdom, created in 2001 a repository of the raw sequence traces generated by large sequencing projects that allows retrieval of both the sequence file and the underlying data that generated the file, including the quality scores. The Assembly Archive (12) created at NCBI in 2004 links the raw sequence information found in the Trace Archive with consensus genomic sequence.

4.1.2. Short Read Archive (SRA)

Trace Archive has successfully served as a repository for the data produced by capillary-based sequencing technologies for many years. New parallel sequencing technologies (e.g., 454, Solexa, Illumina, ABI Solid, Helicos) have started to produce massive amounts of short sequence reads (20–100 kb). Due to the

structure and volume of this data, it is clear that it does not efficiently and effectively fit in the current Trace Archive design, so NCBI has constructed a more appropriate repository, the Short Read Archive. The SRA project is well underway and is being built in collaboration with Ensembl, sequencing centers, and the vendors themselves. SRA Web site has been launched in January 2008: http://www.ncbi.nlm.nih.gov/Traces/sra.

4.1.3. GenBank – Primary Sequence Archive

GenBank is the NIH genetic sequence database, an archival collection of all publicly available DNA sequences (3). GenBank is part of the International Nucleotide Sequence Database Collaboration, which comprises the DNA DataBank of Japan (DDBJ) (4), the European Molecular Biology Laboratory (EMBL) (2), and GenBank at NCBI. These three organizations exchange data on a daily basis. Many journals require submission of sequence information to a database prior to publication to ensure an accession number will be available to appear in the paper. As of February 2008 GenBank release 164.0 (ftp://ftp.ncbi.nih.gov/genbank/release.notes/gb164.release.notes) contains more than 83 billion bases in over 80 million sequence entries. The data come from the large sequencing centers as well as from small experimentalists. These sequences are accessible via Web interface by text queries using Entrez or by sequence queries using BLAST. Quarterly GenBank releases are also downloadable via FTP (see Section 8).

4.2. Entrez Databases

A family of Entrez databases comprise an integrated information system that links together heterogeneous information on biomedical and bibliographical data. The major concepts of Entrez information system are described in Section 3. Below are three examples of Entrez databases containing information on genome projects, genomic sequences, and protein sequence encoded by complete microbial genomes.

4.2.1. Entrez Genome

Entrez Genome (13), the integrated database of genomic information at the NCBI, includes the types of records and formats for major taxonomic groups, as well as the precomputed data and online analytical programs developed to aid investigation. The database was created as part of Entrez in September 1995 for large-scale genome sequencing projects. It was motivated by the release of the first complete microbial genome of *Haemophilus influenzae* sequenced at TIGR (14).

Entrez Genome displays data from small viral and organelle genomes, complete and nearly complete genomes from bacteria, and eukaryotes. An entry in Genomes database represents a single replicon such as a chromosome, organelle, or plasmid. As of February 2008 Entrez Genome houses a collection of 7,850 entries organized in six large taxonomic groups: Archaea, Bacteria, Eukaryota, Viroids, Viruses, and Plasmids. It presents the tools and views

Fig. 2.4. *Haemophilus influenzae* complete genome: single circular chromosome overview. Entrez provides a graphical view of the chromosome with genes color-coded by COG functional categories.

at various levels of detail. For each record, Entrez Genome provides a graphical overview of the chromosome with genes color-coded by COG (clusters of orthologous groups) (15) functional categories (**Fig. 2.4**) as well as other types of text views including flat file, ASN.1, XML, and many others that can be user-selected from a menu. The table provides additional genome information and access to analysis tools.

The available tools include multiple alignments of complete genomes for viruses, precomputed protein clusters from microbial genomes, GenePlot (a genome-scale dotplot generator), TaxPlot (for three-way genome comparisons), gMap, and many others. Some of these tools are described in Section 5 of this chapter. More detailed description of microbial genome resources at NCBI can be found in "In Silico Genomics and Proteomics" (16). Plant genome resources at NCBI have been recently published in a chapter of "Plant Bioinformatics" (17).

Microbial genome sequencing has come a long way since the first *H. influenzae* project. As of February 2008 public collection contains more than 600 complete genomes and close to 500 draft

genome assemblies. The collection represents a very diverse set of organisms; ranging from small (160 kb) endosymbiont Carsonella (18) to the 13-Mb genome of myxobacterium *Sorangium cellulosum* (19). There are organisms isolated from extreme environments such as *Hyperthermus butylicus* (20), an extreme hyperthermophilic, anaerobic archeon, and bacterial species representing deeply branching taxa such as *Rhodopirellula baltica* (21). On the other hand, many projects are aimed toward the comparative analysis of pathogenic bacteria and sequencing multiple strains and isolates of the same organism. For example, *H. influenzae* bacterium is represented in the database by 16 entries including chromosomes and plasmids from different isolated strains. Entrez provides tools that facilitate comparative genome analysis leading into new insights to be gained from genome sequences.

Query examples

Find all the chromosomes of *Haemophilus influenzae*:
Haemophilus influenzae[organism] AND chromosome[replicon type]

4.2.2. Entrez Genome Project

The NCBI Genome Project database is a collection of complete and incomplete (in-progress) large-scale sequencing, assembly, annotation, and mapping projects for cellular organisms. A project is defined by a unique combination of organism name (or metagenomic project name), sequencing center, and sequencing method.

Currently, the database is comprised of projects that have submitted data to NCBI, intend to submit data, or have received public funding. A large eukaryotic genome project usually consists of several components. In the database, projects are organized in a hierarchical, parent–child relationship. A top-level project represents an organism-specific overview and links together all relevant child projects. Each project has its own unique identifier, the Project ID.

The International Nucleotide Sequence Databases Consortium (INSDC) has acknowledged the need to organize genomic and metagenomic data and to capture project metadata. Starting from 2006, the submitters of genome sequence data are required to register their project and obtain a unique project ID. As presented at EMBL guidelines Web site, http://www.ebi.ac.uk/embl/Documentation/project_guidelines.html,

"A project is defined as a collection of INSDC database records originating from a single organization, or from a consortium of coordinated organizations. The collective database records from a project make up a complete genome or metagenome and may contain genomic sequence, EST libraries and any other sequences that contribute to the assembly and annotation of the genome or metagenome. Projects group records either from single organism studies or from metagenomic studies comprising communities of organisms."

NCBI has developed a SOAP (simple object access protocol) compliant Web service, supporting the functions of inserting, updating, deleting, and retrieving of the documents which are used by INSDC collaborators to access/edit the Genome Project database, which in turn controls ProjectIDs and Locus-tag prefixes as well as other project information.

The NCBI Entrez Genome Project database (GenomePrj) is organized into organism-specific overviews that function as portals from which all projects pertaining to that organism can be browsed and retrieved. **Figure 2.5** shows a schematic diagram of a generic eukaryotic genome project.

GenomePrj is integrated into the Entrez search and retrieval system, enabling the use of the same search terms and query structure used in other Entrez databases.

GenomePrj is a companion database to Entrez Genome. Sequence data are stored in Entrez Genome (as complete chromosomes, plasmids, organelles, and viruses) and Entrez Nucleotide (as chromosome or genomic fragments such as contigs). While
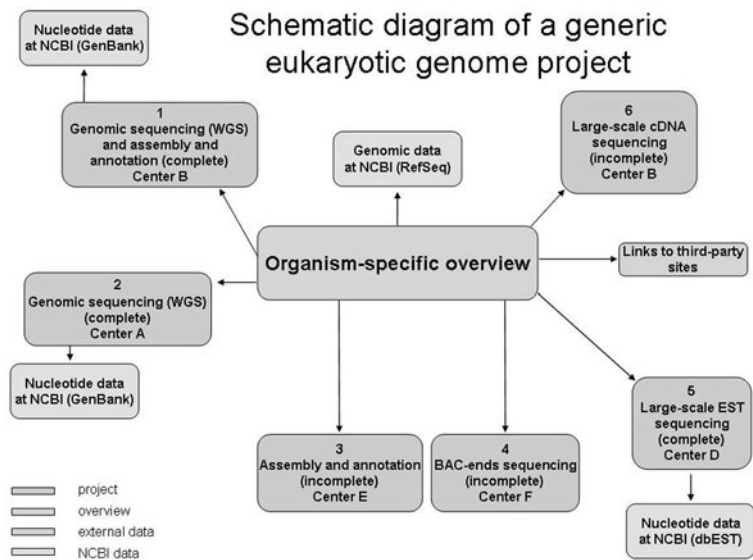
Fig. 2.5. Schematic diagram of a generic eukaryotic genome project. The main overview page shows links to all subprojects, numbered 1–6. Various sequencing centers are associated with each subproject (A–F). These various centers could actually be consortiums consisting of multiple centers. A given center could produce more than one type of project, and data for a given project type could be generated from multiple independent centers. Some of the projects are complete with associated data present in various forms in different Entrez databases at NCBI, while other projects are in progress with no publicly available data at NCBI. A project could be converted over time from containing preliminary data (e.g., WGS) to one where a complete data set is present. RefSeq genomic data are associated with the overview project. Links to third-party sites which contain information of interest regarding the organism are provided.

Entrez Genome does not collect all data for a given organism, GenomePrj provides an umbrella view of the status of each genome project, links to project data in the other Entrez databases and a variety of other NCBI and external resources associated with a given genome project. Sequences associated with a given organism can also be retrieved in the taxonomy browser. However, no distinction is made between GenBank (non-curated) and RefSeq (curated) sequences. There is also no distinction based on which sequencing center submitted the data. Entrez Genome Project also lists projects that are in progress or for which NCBI has not yet received any data. See **Table 2.3** for a comparison of all three databases.

As of January 2008 Genome Project database contains 80 metagenomics project. As shown in **Fig. 2.6**, the database entry contains brief project description, listing of all related subprojects, and project data which include links to genomic data, publication, and Trace data. NCBI Resource Links include an option to BLAST against this particular collection as well as an option to BLAST against all available environmental sequences.

Table 2.3
Comparison of entrez databases

Entrez databases	Organism-specific sequences	Project-specific sequences	Submitter-specific sequences	Complete and in progress	GenBank and RefSeq sequences
Genome	Yes	No	Yes	No	Separated
Taxonomy	Yes	No	No	No	Together
Genome project	Yes	No	Yes	Yes	Separated

Query examples

Find all complete fungal genome projects.

fungi[ORGN] AND complete[SEQSTAT]

Find all projects that correspond to pathogens that can infect humans.

human[HOST]

Find all metagenomic projects

type_environmental[All Fields]

4.2.3. Entrez Protein Clusters

Protein Clusters database is a rich collection of related protein sequences from complete prokaryotic and organelle Reference Sequence (RefSeq) genomes.

Fig. 2.6. Mouse gut metagenome project in Entrez Genome Project database: comparisons of the distal gut microbiota of genetically obese mice and their lean littermates.

Proteins from all complete microbial genomes and plasmids (and separately all chloroplasts) are compared using BLAST all-against-all. Protein clusters are created using a modified BLAST score that takes into account the length of the hit (alignment) versus both the query and the subject. The modified score is then sorted, and all proteins that are contained within the top hits are clustered together. Automatically constructed clusters are then evaluated manually by curators. Based on the sequence alignment information and biological expertise, curators can join or split clusters and add annotation information (protein name, gene name, description) and publication links.

As of January 2008, the database contains 1.4 million proteins that compose 6,043 curated clusters and more than 200,000 automatic clusters. The Entrez Protein Clusters database uses all of the features of other Entrez databases. There are numerous ways to query protein clusters, either with search terms in Entrez or with

a protein or nucleotide sequence. The display for each cluster provides information on cluster accession, cluster name, and gene name, as well as links to protein display tools, external databases, and publications (**Fig. 2.7**). Protein Clusters database can be queried with a protein or nucleotide sequence by using Concise Protein BLAST, a new Web resource developed at NCBI. Concise BLAST is an efficient alternative to standard BLAST. The searchable database is comprised of only one randomly chosen protein from each cluster, and also proteins which are not included in any cluster to assure completeness. This allows rapid searching of the smaller database, but still assures an accurate identification of the query while providing a broader taxonomic view.

Fig. 2.7. Shikimate 5-dehydrogenase overview in Entrez Protein Clusters database. The top part of the page presents text description, some statistics (Cluster Info), direct access to Cluster Tools, cross-references to outside resources, and links to other Entrez databases. The bottom part presents a colored table: clusters are organized into taxonomic groups; cluster position neighbors are shown as well as a summary of alignments and conserved domains. Clicking on alignment summary will open a detailed multiple alignment view (not shown).

Query examples

Retrieve all clusters containing the protein beta galactosidase:

beta galactosidase [Protein Name]

Find all clusters associated with *Escherichia coli*:

Escherichia coli[Organism]

---

5. Analysis of Prokaryotic Genome Data

5.1. gMap – Compare Genomes by Genomic Sequence Similarity

gMap is one of the tools available in Entrez Genome that allows to view and analyze the regions of similarity in closely related genomes. **Figure 2.8** shows closely related strains of *H. influenzae*.

Genomic sequences are compared using BLAST and the resultant hits are filtered out to find the largest syntenic regions. Similar regions are shown color-coded and numbered in each genome with an arrow denoting the 5′–3′ direction of the hit with respect to similar segments in other genomes. Additional sequences can be added by inputting the accession number.
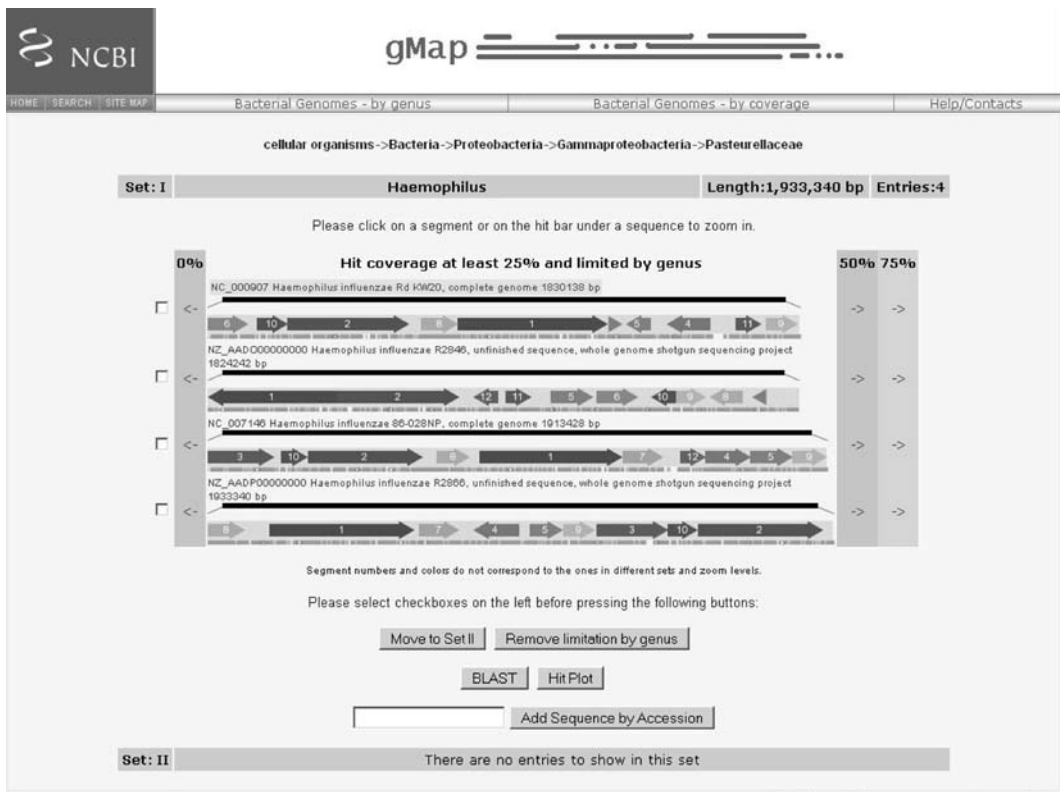
Fig. 2.8. gMap results for four closely related strains of *Haemophilus influenzae* at 25% coverage.

The tool allows navigating from the general overview of domains of life (e.g., bacteria or viruses) down to genome sets with different degrees of mutual similarity. It allows more detailed views of every similarity segment, including the ability to view sequence alignment of two selected similarity regions. Zooming in can be accomplished by clicking on a syntenic group to expand all similar segments. Alternately, a user can click on a hit bar just below the segments to zoom into the surrounding region of the current sequence; this action also displays homologous syntenies from other organisms. After zooming in, all segments are recalculated, recolored, and renumbered, providing a truly dynamic and interactive system with each calculated view presented as a standalone display which is visually easy to comprehend. Pairs of genomic sequences can be selected for output to BLAST, GenePlot, or HitPlot and any number of sequences can be removed from the list by the user to customize the final view to be most appropriate for the user's project. HitPlot shows a dotplot of the two genomes selected based on the magnification level. Precomputed results are available for two categories, one for genomes from the same genus and one for genomes based on the coverage of BLAST hits. Genomes of two or more species from the same genus may not display high levels of synteny, but similar segments in their two genomes can be found at different levels of hit coverage. An example of this would be the *Mycoplasma* genomes. The converse is that organisms from different genera have large syntenic blocks in their genomes such as is found in *Escherichia*, *Salmonella*, and *Shigella*, which are all members of the Enterobacteriaceae family (22). Genomes in both categories are grouped together based on single linkage clustering of coverage level. For example, if genome A has 75% coverage to genome B and genome B has 75% coverage to genome C, then they will all be included in a cluster at the 75% level even though the coverage between A and C may not reach the 75% level.

5.2. Genome ProtMap – Compare Genomes by Protein Sequence Similarity

Genome ProtMap is a comparative display of the genome neighborhoods linked by the orthologous protein sequences. It displays a 10-kb region surrounding either all the proteins in the cluster or, alternately, all the proteins that have the same Cluster of Orthologous Group – COG (15) – or in the case of viruses, VOGs. In the Genome ProtMap display (**Fig. 2.9**), the organism groups are collapsed; clicking the + will expand the group. Clicking the accession number will link to the RefSeq nucleotide record. Mouse over the proteins gives detailed information such as name, cluster ID, and genome location. Clicking on any protein brings up a pop-up menu with links to protein, gene, or cluster. The list of taxa in the ProtMap can be collapsed or expanded by clicking the + or − next to the taxon. "Show Legends" gives the color-coded functional category for the
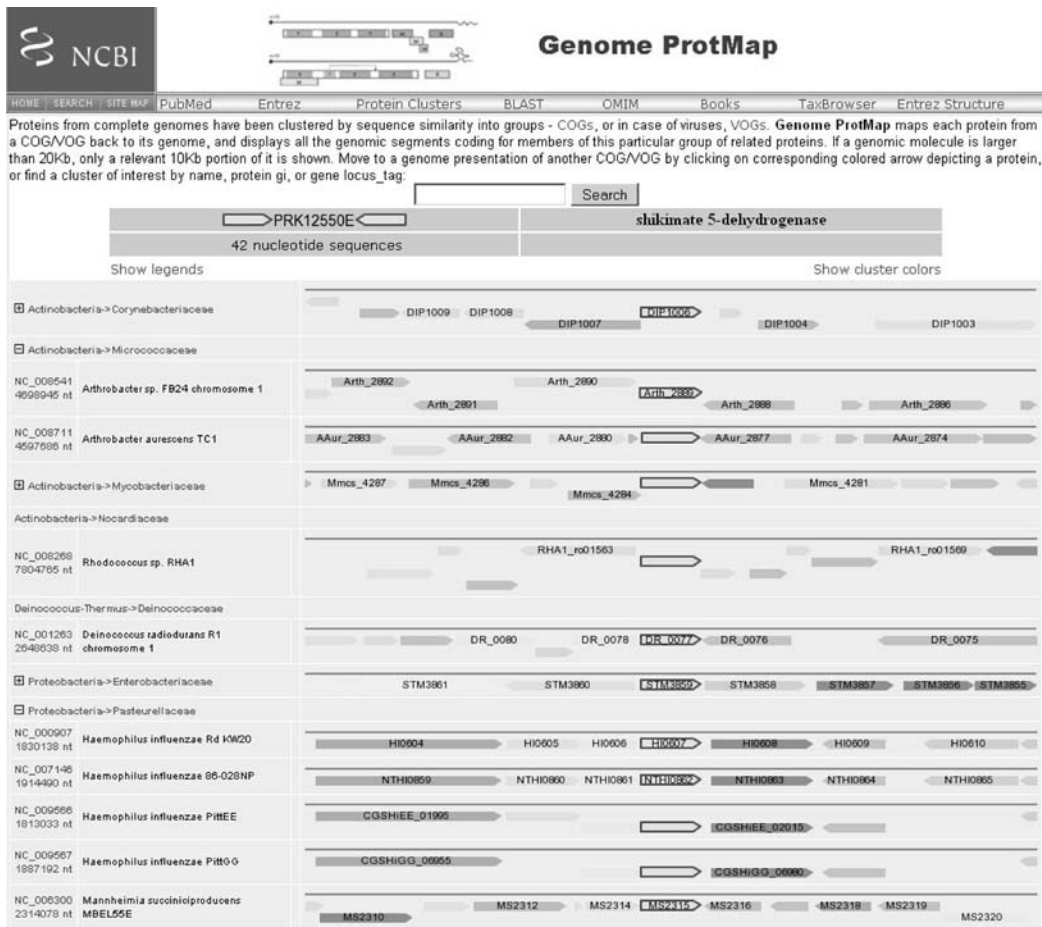
Fig. 2.9. Genome ProtMap shows local genomic neighborhood centered on a set of related genes (via the proteins encoded by them). Related genes are defined by protein clustering algorithms (COG, VOG, or PRK). All proteins in the surrounding area are color-coded by COG category (if applicable) or gray (proteins that do not belong to a COG). ProtMap for PRK12550 – shikimate 5-dehydrogenase is displayed.

proteins while "Show Cluster Colors" lists all the clusters in the ProtMap colored by COG functional category and the name of the cluster.

5.3. Concise BLAST   Concise protein BLAST uses the BLAST engine to allow searching of protein clusters' data sets with a protein or nucleotide sequence query. The database represents protein sequences from complete microbial (prokaryotic) genomes. It uses precalculated clusters of similar proteins at the genus level to represent proteins by groups of related sequences. One representative from each cluster is chosen in order to reduce the data set. The result is reduced search times through the elimination of redundant proteins while providing a broader taxonomic view.

6. Browsing Eukaryotic Genome Data

The main NCBI genome browser Map Viewer provides special browsing capabilities for a subset of organisms in Entrez Genome. The list of organisms available for Map Viewer browsing can be found on the Map Viewer home page (http://www.ncbi.nlm.nih.gov/projects/mapview/).

Map Viewer can display a collection of aligned genetic, physical, or sequence-based maps, with an adjustable focus rangingfrom that of a complete chromosome to that of a portion of a gene. The maps displayed in Map Viewer may be derived from a single organism or from multiple organisms; map alignments are performed on the basis of shared markers. The availability of whole genome sequences means that objects such as genes, markers, clones, sites of variation, and clone boundaries can be positioned by aligning defining sequences from these objects against the genomic sequence. This positional information can then be compared to information on order obtained by other means, such as genetic or physical mapping. The results of sequence-based queries (e.g., BLAST) can also be viewed in genomic context as described in the next section.

Any text search term can be used as a query at the top of the Map Viewer home page. These include, but are not limited to, a GenBank accession number or other sequence-based identifier, a gene symbol or alias, or the name of a genetic marker. For more complex queries, any query can be combined with one of three Boolean operator terms (AND, OR, and NOT). Wild cards, which are denoted by placing a * to the right of the search term, are also supported. Map Viewer uses the Entrez query search engine, described in section 3, to analyze a complex query and perform a search.

Another way of getting to a particular section of a genome is to use a range of positions as a query. First it is necessary to select a particular chromosome for display from a genome-specific Map Viewer page. Once a single chromosome is displayed, position-based queries can be defined by (1) entering a value into the Region Shown box. This could be a numerical range (base pairs are the default if no units are entered), the names of clones, genes, markers, SNPs, or any combination. The screen will be refreshed with only that region shown.

Map Viewer provides an option to simultaneously search physical, genetic, and sequence maps for multiple organisms. This option is currently available for plant and fungal genomes. Multi-organism plant searching is available at http://www.ncbi.nlm.nih.gov/projects/mapview/map_search.cgi?taxid=33090.

Since the early 1990s several researchers have shown that large-scale genome structure is conserved in blocks across the grasses (23–26). Locus nomenclature is organism-specific and is unreliable as a query method between species; however, the regular nomenclature of plasmids (27) is not influenced by how the plasmid or insert is

used. The data for the plant maps available through Map Viewer include the probe–locus relationship for each locus where the allelic state is identified by the probe. This information enables the rendering of the visual connection between those mapped loci in adjacently displayed maps that were identified by the same probe. This locus–probe relationship allows a cross-species text search using the probe name as the query string. **Figure 2.10** shows the result of the search across all plants using "cdo718" as a query.
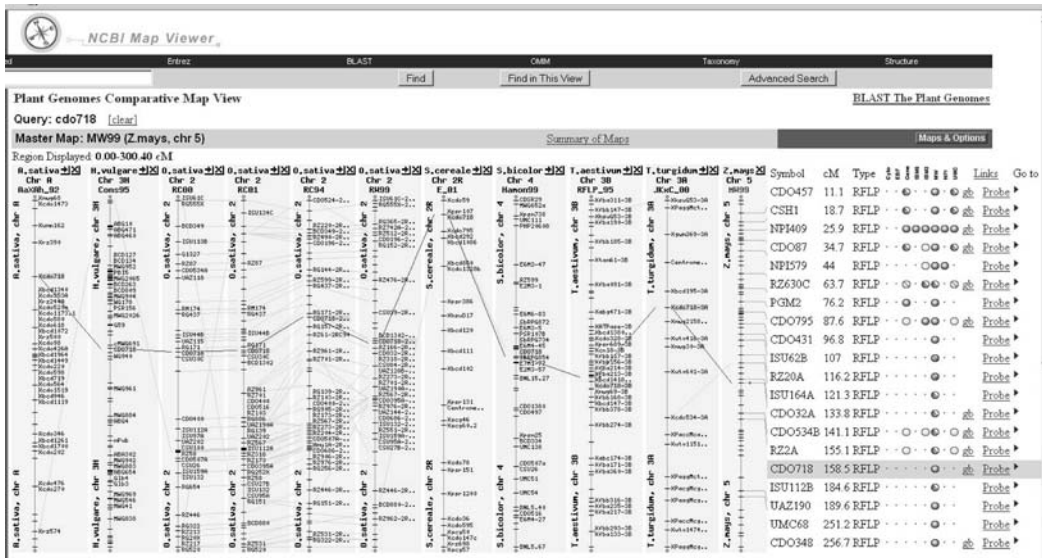
Fig. 2.10. Map Viewer Displays resulting from a search for marker "cdo718" showing aligned maps from several plants. The marker "cdo718" is highlighted on each map with lines between maps connecting the highlighted markers.

"cdo718" is the name of a plasmid with an oat cDNA insert. This probe was used to map loci in nine maps available in Map Viewer: the AaXAh-92 map in *Avena sativa*, the Cons95 map in *Hordeum vulgare*, the RC94, RW99, R, RC00, and RC01 maps in *Oryza sativa*, the E-01 map in Secale cereale, the S-0 map in *Triticum aestivum, JKxC map in Triticum turgidum*, and the RW99 map in *Zea mays*. The dark grey lines between each map connect the loci identified by the probe. The light gray lines connect the other loci in adjacent maps that have been identified by the same probe.

7. Searching Data by Sequence Similarity (BLAST)

The Basic Local Alignment Search Tool (BLAST) (28) finds regions of local similarity between sequences. By finding similarities between sequences, scientists can infer the function of newly sequenced genes, predict new members of gene families, and explore evolutionary relationships.

7.1. Organism-Specific Genomic BLAST

Genome-specific BLAST pages that restrict a search to a specific genome are provided for several organisms and allow the results of the search to be displayed in a genomic context (provided by Map Viewer).

Query sequence (protein or nucleotide) can be compared to genomic, transcript, or protein coded by the genome. **Table 2.4** provides the list of available databases. Not all databases are always available; some projects provide additional data sets such as SNP, traces, and alternative assemblies. If the reference genome (the default) is selected as the database to be searched, the Genome View button (**Fig. 2.11B**) will appear on a diagram showing the chromosomal location of the hits (**Fig. 2.11C**). Each hit links to a Map Viewer display of the region encompassing the sequence alignment.

7.2. Multi-organism Genomic BLAST

Microbial Genomic BLAST (29) provides access to complete genomes and genome assemblies of 940 Bacteria and 48 Archaea and 162 Eukaryota (as of February 2008). Genomic BLAST has been recently extended to include data sets for insects, fungi, nematodes, protozoa, and metagenomes. The genomes can be viewed

Table 2.4
Customized project-specific BLAST databases

DB name	Description
Genome (all assemblies)	Sequences from all available genome assemblies
Genome (reference only)	Sequences from the reference assembly only
RefSeq RNA	RefSeq transcript sequences (NM + XM)
RefSeq protein	RefSeq protein sequences (NP + XP)
Non-RefSeq RNA	GenBank transcript sequence
Non-RefSeq protein	GenBank protein sequences
Build RNA	Proteins generated in the annotation run
Build protein	Proteins generated in the annotation run
Ab initio RNA	Transcripts generated in the annotation run by Gnomon only
Ab initio protein	Proteins generated in the annotation run by Gnomon only
EST	EST sequences by organism
Clone end sequences	Clone end sequences by organism
Traces WGS	Raw sequence reads for genomic assemblies
Traces EST	Raw sequence reads for EST
SNP	Custom database of Single Nucleotide Polymorphism database
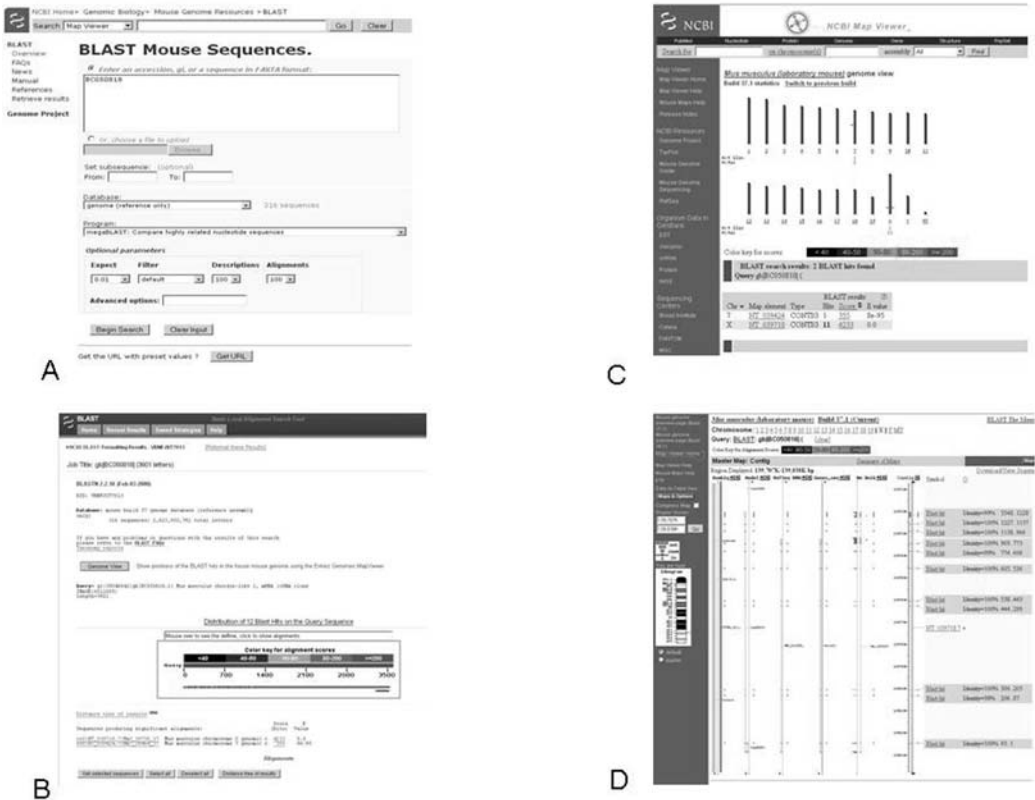
Fig. 2.11. Genomic BLAST: BLAST Mouse Sequences. A Query mouse reference genome with mouse cDNA clone, accession BC050818. B BLAST output page has an additional "Genome View" button that provides an option to show the hits in genome environment via Map Viewer. C Genome overview of BLAST hits. The hits are represented by colored ticks providing the links to zoomed-in view of the chromosome. D Positions of the BLAST hits on the chromosome. The maps shown include Model (NCBI annotation pipeline prediction), RefSeq transcript, and mouse UniGene.Interesting to note that the first exon (hit 3548..1228) is not included in the RefSeq model, although it is supported by UniGene and predicted by NCBI annotation pipeline.

in taxonomic groups or in alphabetical order. A flexible user-friendly interface allows to construct virtual blast databases for the specific searches.

For example, with many closely related microbial genomes sequenced, one might want to exclude the close relatives from consideration in order to reveal more evolutionary interesting remote relationships.

8. FTP Resources for Genome Data

The source genome records can be accessed from the Gen-Bank directory; these are the records that were initially deposited by the original submitters. The reference genomes,

assemblies, and associated genes and proteins can be down-loaded from the Genomes and RefSeq directories. Information on the data content in these FTP directories is located in the README files.

Download the full release database, daily updates, or WGS files:

ftp://ftp.ncbi.nih.gov/genbank/

Download complete genomes/chromosomes, contigs and reference sequence mRNAs and proteins:

ftp://ftp.ncbi.nih.gov/genomes/

Download the curated RefSeq full release or daily updates:

ftp://ftp.ncbi.nih.gov/refseq/

Download curated and non-curated protein clusters from microbial and organelle genomes:

ftp://ftp.ncbi.nih.gov/genomes/Bacteria/CLUSTERS

9. Conclusion

The tremendous increase in genomic data in the last 20 years has greatly expanded our understanding of biology. Genome sequencing projects now span from draft assemblies, complete genomes, large-scale comparative genomic projects, to the new field of metagenomics where genetic material is recovered directly from environmental samples and the entire complement of DNA from a given ecological niche is sequenced. Although these provide an ever greater resource for studying biology, there is still a long way to go from the initial submission of sequence data to the understanding of biological processes. By integrating different types of biological and bibliographical data, NCBI is building a discovery system that enables the researcher to discover more than would be possible from just the original data. By making links between different databases and computing associations within the same database, Entrez is designed to infer relationships between different data that may suggest future experiments or assist in interpretation of the available information. In addition, NCBI is developing the tools that provide users with extra layers of information leading to further discoveries.

Genomics is a very rapidly evolving field. The advance in sequencing technologies has lead to new data types which require different approaches to data management and presentation. NCBI continues to add new databases and develop new tools to address the issue of ever-increasing amounts of information.

Acknowledgments

References

1. Liolios, K., Mavrommatis, K., Tavernarakis, N., Kyrpides, N. C. (2007) The Genomes On Line Database (GOLD) in 2007: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Res.* 36(Database issue), D475–D479.

2. Cochrane, G., Akhtar, R., Aldebert, P., Althorpe, N., Baldwin, A., Bates, K., Bhattacharyya, S., Bonfield, J., Bower, L., Browne, P., Castro, M., Cox, T., Demiralp, F., Eberhardt, R., Faruque, N., Hoad, G., Jang, M., Kulikova, T., Labarga, A., Leinonen, R., Leonard, S., Lin, Q., Lopez, R., Lorenc, D., McWilliam, H., Mukherjee, G., Nardone, F., Plaister, S., Robinson, S., Sobhany, S., Vaughan, R., Wu, D., Zhu, W., Apweiler, R., Hubbard, T., Birney, E. (2008) Priorities for nucleotide trace, sequence and annotation data capture at the Ensembl Trace Archive and the EMBL Nucleotide Sequence Database. *Nucleic Acids Res* 36(Database issue), D5–D12.

3. Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Wheeler, D. L. (2008) GenBank. *Nucleic Acids Res* 36(Database issue), D25–D30.

4. Sugawara, H., Ogasawara, O., Okubo, K., Gojobori, T., Tateno, Y. (2008) DDBJ with new system and face. *Nucleic Acids Res* 36(Database issue), D22–D24.

5. Galperin, M. Y. (2008) The molecular biology database collection: 2008 update. *Nucleic Acids Res* 36(Database issue), D2–D4.

6. Wheeler, D. L., et al. (2008) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res* 36(Database issue), D13–D21.

7. Mailman, M. D., Feolo, M., Jin, Y., Kimura, M., Tryka, K., Bagoutdinov, R., Hao, L., Kiang, A., Paschall, J., Phan, L., Popova, N., Pretel, S., Ziyabari, L., Lee, M., Shao, Y., Wang, Z. Y., Sirotkin, K., Ward, M., Kholodov, M., Zbicz, K., Beck, J., Kimelman, M., Shevelev, S., Preuss, D., Yaschenko, E., Graeff, A., Ostell, J., Sherry, S. T. (2007) The NCBI dbGaP database of genotypes and phenotypes. *Nat Genet* 39(10), 1181–1186.

8. Pruitt, K. D., Tatusova, T., Maglott, D. R. (2007) NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res* 35(Database issue), D61–D65.

9. Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25(17), 3389–3402. Review.

10. Maglott, D. R., Ostell, J., Pruitt, K. D., Tatusova, T. (2007) Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Res* 35(Database issue), D26–D31.

11. Hillary, E. S., Maria, A. S., eds. (2006) *Genomes (Cold Spring Harbor Monograph Series, 46).* Cold Spring Harbor, New York.

12. Salzberg, S. L., Church, D., DiCuccio, M., Yaschenko, E., Ostell, J. (2004) The genome Assembly Archive: a new public resource. *PLoS Biol.* 2(9), E285.

13. Tatusova, T. A., Karsch-Mizrachi, I., Ostell, J. A. (1999) Complete genomes in WWW Entrez: data representation and analysis. *Bioinformatics* 15(7–8), 536–543.

14. Fleischmann, R. D., et al. Whole-genome random sequencing and assembly of Haemophilus influenza Rd. (1995) *Science* 269(5223), 496–512.

15. Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., Koonin, E. V., Krylov, D. M., Mazumder, R., Mekhedov, S. L., Nikolskaya, A. N., Rao, B. S., Smirnov, S., Sverdlov, A. V., Vasudevan, S., Wolf, Y. I., Yin, J. J., Natale, D. A. (2003) The COG database: an updated version includes eukaryotes. *BMC Bioinformatics* 4, 41.

16. Klimke, W., Tatusova, T. (2006) Microbial genomes at NCBI in (Mulder, N., Apweiler, R., eds.) *In Silico Genomics And Proteomics: Functional Annotation of Genomes And Proteins*, Nova Science Publishers; 1st ed., pp. 157–183.

17. Tatusova, T., Smith-White, B., Ostell, J. A. (2006) Collection of plant-specific genomic data and resources at the National Center for Biotechnology Information, in (David, E., ed.), *Plant Bioinformatics: Methods and Protocols (Methods in Molecular Biology)*, *Humana Press*, 1st ed., pp. 61–87.

18. Nakabachi, A., Yamashita, A., Toh, H., Ishikawa, H., Dunbar, H. E., Moran, N. A., Hattori, M. (2006) The 160-kilobase genome of the bacterial endosymbiont. *Carsonella Sci* 314(5797), 267.

19. Schneiker, S., et al. (2007) Complete genome sequence of the myxobacterium Sorangium cellulosum. *Nat Biotechnol* 25(11), 1281–1289.

20. Brügger, K., et al. (2007) The genome of Hyperthermus butylicus: a sulfur-reducing, peptide fermenting, neutrophilic Crenarchaeote growing up to 108 degrees C. *Archaea* 2(2), 127–135.

21. Teeling, H., Lombardot, T., Bauer, M., Ludwig, W., Glockner, F. O. (2004) Evaluation of the phylogenetic position of the planctomycete 'Rhodopirellula baltica' SH 1 by means of concatenated ribosomal protein sequences, DNA-directed RNA polymerase subunit sequences and whole genome trees. *Int J Syst Evol Microbiol* 54, 791–801.

22. Darling, A. C., Mau, B., Blattner, F. R., et al. (2004). Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res* 14(7), 1394–1403.

23. Ahn, S. N., Tanksley, S. D. (1993) Comparative linkage maps of the rice and maize genomes. *Proc Natl Acad Sci USA* 90, 7980–7984.

24. Devos, K. M., Chao, S., Li, Q. Y., Simonetti, M. C., Gale, M. D. (1994) Relationship between chromosome 9 of maize and wheat homeologous group 7 chromosomes. *Genetics* 138, 1287–1292.

25. Kurata, N., Moore, G., Nagamura, Y., Foote, T., Yano, M., Minobe, Y., Gale, M. D. (1994) Conservation of genome structure between rice and wheat. *Biotechnology (NY)* 12, 276–278.

26. van Deynze, A. E., Nelson, J. C., O'Donoghue, L. S., Ahn, S. N., Siripoonwiwat, W., Harrington, S. E., Yglesias, E. S., Braga, D. P., McCouch, S. R., Sorrells, M. E. (1995) Comparative mapping in grasses: oat relationships. *Mol Gen Genet* 249, 349–356.

27. Lederburg, E. M. (1986) Plasmid prefix designations registered by the Plasmid Reference Center 1977–1985. *Plasmid* 1, 57–92.

28. Altschul, S. F., Gish, W., Miller, W., et al. (1990). Basic local alignment search tool. *J Mol Biol* 215(3), 403–410.

29. Cummings, L., Riley, L., Black, L., Souvorov, A., Resenchuk, S., Dondoshansky, I., Tatusova, T. (2002) Genomic BLAST: custom-defined virtual databases for complete and unfinished genomes. *FEMS Microbiol Lett* 216(2), 133–138.

Chapter 3

Protein Sequence Databases

Michael Rebhan

Abstract

Protein sequence databases do not contain just the sequence of the protein itself but also annotation that reflects our knowledge of its function and contributing residues. In this chapter, we will discuss various public protein sequence databases, with a focus on those that are generally applicable. Special attention is paid to issues related to the reliability of both sequence and annotation, as those are fundamental to many questions researchers will ask. Using both well-annotated and scarcely annotated human proteins as examples, it will be shown what information about the targets can be collected from freely available Internet resources and how this information can be used. The results are shown to be summarized in a simple graphical model of the protein's sequence architecture highlighting its structural and functional modules.

Key words: proteins, protein sequence, protein annotation, protein function, databases, knowledgebases, Web resources, expert review, sequence data curation.

1. Introduction

Since Fred Sanger's work paved the way for obtaining the sequences of proteins in the 1950s (1), we have been accumulating information about proteins with an ever-increasing pace. By 1965, 10 years after the publication of the sequence of insulin (1), a few dozen protein sequences were published. This triggered the interest of scientists who started to wonder how this valuable information can best be compared between species and how such comparisons of sequences could help us to elucidate the evolution of molecular mechanisms. Would such distant taxa as mammals, insects, fungi, plants, and bacteria have much in common at the level of sequence? And how would we be able to interpret such similarity? At that time, Margaret Dayhoff (1925–1983), a

scientist with a very interdisciplinary background and a lot of foresight, decided that it is time to put all this information together into a book that has since become the ancestor of all protein collections (2). By providing well-organized information on the 65 proteins sequences that were published at the time, she wanted to make it easier for other scientists to join her in the quest for developing an understanding of their biological meaning through comparison. Soon it turned out that this work, indeed, became one of the foundations of a new field, which is now commonly referred to as "bioinformatics."

In the 1980s, in her last few years, one of Dr. Dayhoff's major efforts was to ensure the continuation of this work, by trying to obtain adequate long-term funding to support the maintenance and further development of the collection. Less than a week before her death, she submitted a proposal to the Division of Research Resources at NIH (3). Her vision was to develop an online system of computer programs and databases which can be accessed by scientists all over the world, for making predictions based on sequences and for browsing the known information. After her death, her colleagues were determined to see her vision realized, by creating the PIR (Protein Information Resource) (4).

Inspired by Dr. Dayhoff's legacy, a Ph.D. student who was busy writing one of the first software packages for sequence analysis in the mid 1980s encountered some problems with the data from PIR. He decided to set up his own collection, to have the freedom to develop it as he pleases. The result of this effort became the Swiss-Prot database (5). Amos Bairoch, its founder, decided to make a bold career move, that is to focus his work on the computational analysis of protein sequences (which, at that time, may not have been an obvious choice to many colleagues). But his foresight was rewarded as well: the EMBL in Heidelberg agreed to distribute it, and as soon as the Internet was mature enough to allow direct access by researchers anywhere in the world, the associated Web site, ExPASy (6), quickly became one of the most fundamental electronic resources to any scientist working with proteins.

With the rise of the Internet in the mid-1990s, thousands of small and large resources related to proteins and genes emerged as well, created by scientists who wanted to share the collections they were developing locally. But this also created increasing confusion, as biologists found it difficult to find the information they were interested in without spending the whole day on the Internet, following an increasingly bewildering forest of hyperlinks between Web sites that often did not last longer than the research project itself due to lack of long-term funding. As a result, resources like GeneCards were developed (7), with the goal of presenting a structured overview of current knowledge on genes and their products, and the ability to drill down into the information to check sources and find additional information if needed. A similar

goal was pursued by scientists at the NCBI, who wanted to create a comprehensive resource for genes of all the main organisms, including expert-reviewed, high-quality, full-length sequences with annotation. This was the beginning of the RefSeq collection of sequences, which is still one of the best sources for reliable nucleotide and protein sequences (8, 9).

With the advent of genome sequencing projects, increasing efforts were made to develop approaches for finding the correct structures of protein-coding genes in those assembled sequences. This led to a new generation of databases that include information on protein sequences with various levels of evidence, such as Ensembl (9) and many organism-specific resources. These genome resources complement the above "high confidence sets" (Swiss-Prot and RefSeq) and their associated large repositories of as yet uncurated sequences that have at least transcript-level experimental evidence – TREMBL (10), Genbank translations (11) – with even more protein sequences. They take advantage of the high reliability of mature genomic sequence and provide candidate protein sequences even for cases where the experimental evidence available at the transcript and protein level may be minimal or absent. As a result, we now have many different protein sequence databases with different strengths and limitations, and protein sequences with many different levels of evidence at the genomic, transcript, and protein levels. Unfortunately, those data are presented in diverse formats and ways of access, which can be a challenge for users. Due to the flood of nucleotide sequences that are likely to be translated into proteins, we now have millions of proteins in the public databases; for example, a search at NCBI Entrez's Protein database (12) at the time of writing returns more than 4.4 million proteins for eukaryotes, and more than 7.7 million proteins for prokaryotes. To guide scientists who are not familiar with the different databases through this complex landscape, we will attempt to discuss some of the key differences in content and use. But, as anything that evolves, this database landscape itself will certainly change, so we will try to emphasize fundamental issues that are likely to persist for some time.

2. The Foundation Is the Sequence

Obviously, the most fundamental piece of information on proteins is the amino acid sequence itself. Before we start to annotate it, we need to be sure that the sequence is reliable enough for our purposes. All kinds of artifacts may complicate further work, either at the computer or in the lab: a small part of the sequence could be wrong, some part may be missing, or the whole sequence may not

even exist in nature as its mRNA is actually not translated. If we are lucky enough that we can rely on the judgment of experts who can assess the reliability of the sequence, such as the curators reviewing sequences for Swiss-Prot and RefSeq, we will of course pick our sequence from their reviewed sequence collections. But in the unfortunate event that our protein of interest is not available in those resources, we have to take what we can get and perform at least some basic checks to make sure that the sequence has sufficient quality. During this analysis, we will first compare our sequence to the reviewed reference sequences and then check for conserved regions.

For example, imagine that we picked up our sequence based on a search for the gene name at NCBI's Entrez query system. By performing a BLAST at ExPASy against the Swiss-Prot database, we can find out if there is clear local similarity to known proteins (rule of thumb: the BLAST score should be at least about 100 using the default search parameters), and do the same using protein BLAST at NCBI against RefSeq (*see* **Fig. 3.1** and **Table 3.1**). Of course, if we can find a match in Swiss-Prot or RefSeq that is almost identical along the entire length (in the right organism), we can use that one as our new reference sequence for further analysis, instead of our original sequence. If we only get clear similarity for a small region in our protein sequence, we can consider this area as more likely to be reliable, but we cannot make a statement about other regions. If we still have insufficient information about the reliability of large regions in our sequence, we can submit the query protein to InterproScan at EBI (13), which will allow us to do a comprehensive search for conserved domains that are covered in one of the many protein family databases. If all those searches fail to result in convincing results for the whole sequence or for the part of the sequence we are interested in, and if other curated
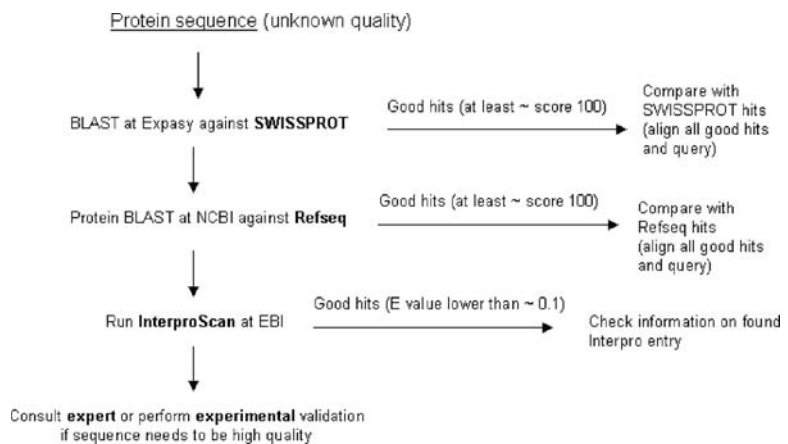
Fig. 3.1. How to estimate the reliability of a protein sequence (see text).

Table 3.1
Overview of key resources for protein sequences and annotations

Name(s)	Relevance	References	URL
Swiss-Prot	Find the (in general) most reliable sequence and annotations	5, 6, 10, 16, 19	http://www.expasy.org/sprot/
RefSeq	Also contains many reliable protein sequences (search the "Protein" database at NCBI, use the "RefSeq" tab, and then check their annotation to find out if they have been reviewed!)	8, 12, 15, 17	http://www.ncbi.nlm.nih.gov/RefSeq/
CCDS (Consensus CDS protein set)	Human and mouse protein sequences that experts in different centers agree on	15	http://www.ncbi.nlm.nih.gov/CCDS/
Mouse Genome Informatics	Curated information on mouse genes and phenotypes	21	http://www.informatics.jax.org
Flybase	Curated information on Drosophila genes	22	http://flybase.bio.indiana.edu
Wormbase	Curated information on nematode genes	23	http://www.wormbase.org
Ensembl	Completed eukaryotic genomes and their genes	9	http://www.ensembl.org
DisProt (database of protein disorder)	Experimentally verified disordered regions in proteins	24	http://www.disprot.org
PhosphoSite	Extensive information on known phosphorylation sites	25	http://www.phosphosite.org

resources such as the organism-specific ones listed in **Table 3.1** do not provide good hits either, it would be advisable to consult an expert (especially if subsequent analysis depends heavily on the quality of the sequence), or even to obtain experimental validation.

In a recent paper that brought the reliability of protein sequences into the forefront again, Michele Clamp and her coworkers performed an analysis of human genes, of which many were so far considered to be protein-coding by default if they did not look similar to known ncRNA genes (14). By carefully assessing the evolutionary conservation of those sequences, they concluded that only about 20,000 of those are showing conservation patterns

typical for validated protein-coding genes, while the others could as well be new types of noncoding RNA genes, in contrast to common practice (to declare them protein-coding by default). In other words, in the absence of evidence to the contrary, we have to take into account the possibility that an ORF predicted in a transcript sequence does not necessarily translate into a real protein in the cell under normal conditions. Fortunately, Swiss-Prot has recently started to include information into their database entries that clearly describes the type of evidence available for the existence of a particular protein.

An additional layer of complexity is provided by alternative splicing, alternative promoter usage, and alternative start codon usage during translation and other phenomena that result in different protein sequences being encoded by the same gene. In fact, experts for deriving protein sequences from gene and transcript data do not always agree on the protein sequence(s) for a particular gene, which can be due to limited transcript data for a gene and a number of experimental artifacts that pollute nucleotide sequence data. An example for efforts that try to address those issues is the CCDS initiative, which defines coding sequences, and therefore encoded protein sequences that different curation teams at NCBI, EBI, Sanger Institute, and UCSC can agree on (15). So, if your protein sequence of interest is part of this collection, this can be interpreted as a good sign (i.e., there is a good chance that this protein exists with exactly this sequence in nature). A gene that exemplifies this kind of complexity is the human form of the microtubule-binding protein tau (approved gene symbol: MAPT), which displays different numbers of protein isoforms, depending on the resource you inspect (*see* **Table 3.1**). So how should such complexity be represented in a protein database? Should every unique sequence get its own entry? Swiss-Prot has since its beginnings decided to try to represent all the isoforms with one reference sequence that is annotated with features that describe which parts of the sequence occur in which isoforms (see the Swiss-Prot entry P10636, which currently contains nine isoforms). Based on this concept, all isoform sequences can then be generated on demand, e.g., for comprehensive sequence similarity searches, while a single reference sequence representing the key properties of a large fraction of the protein molecules is available as well. If an isoform is rare and contains sequence that does not occur in other more common isoforms, it may not be represented in the sequence given in the main entry, but instead in the annotation and when querying specifically for this isoform. Therefore, such areas may be missed in sequence searches that do not search against all isoforms in Swiss-Prot. RefSeq, on the other hand, sometimes provides more than one protein sequence per gene. In any case it may be useful to align all available Swiss-Prot isoform sequences and RefSeq protein sequences (and possibly sequences

from organism-specific databases, *see* **Table 3.1**) for the gene of interest to see where they differ, as there can be substantial differences that require careful examination.

3. What Is Known About the Protein?

Once we can be sure that our protein sequence is reasonably reliable, we can try to find out more about its biological meaning. If it is a protein that has been well characterized, or if it is highly similar to such a protein, we may want to turn to Swiss-Prot to get an overview of the information that has accumulated in the literature. In addition, there can be useful annotation in other resources, like RefSeq, and some organism-specific databases that offer expert-reviewed information as well (*see* **Table 3.1**).

But, what exactly does it mean for a protein to be "well-characterized"? Do we really know all the functions of this protein in all cell types and development stages, and which parts of the sequence play which role in which function? Even in cases where there are hundreds of publications on a particular protein, it is possible that most of the work that has been done has looked at the protein from a particular angle, while much less attention has been paid to additional "moonlighting" functions, functions in other cell types and stages, and some of the transient interactions it engages in. In many cases, we know what particular regions of the protein do, but we cannot make confident statements about the function of the rest of the protein.

On *c-myc*, for example, an impressive amount of information is available. A search in all databases at NCBI reveals more than 10,000 articles in Pubmed, more than 2,000 protein sequences, and 20 macromolecular structures. Few proteins have been investigated so thoroughly. Therefore, one may easily think that almost every interesting aspect of the function of this protein would have been unearthed by now. Let us see what the protein databases provide in such a case. First, we will examine the human entry in Swiss-Prot (16). The human-friendly current "name" (or "ID") is "MYC_HUMAN," but the more stable accession that is independent of gene name changes (which do occur!) is "P01106." In this particular case, a gene name change is not expected, but in any case it is a good habit if we use the stable accession for any type of documentation, to be on the safe side. If you have a careful look at the information in the entry, it shows that the protein was entered into Swiss-Prot a long time ago (in 1986), so actually it may have been one of the first proteins in the database. But since then annotations have been added or modified, the last time quite recently. Note also the "gene name" field, which lists the gene

Fig. 3.2. Architecture of human c-myc, based on annotations in Swiss-Prot, RefSeq, and DisProt. From left to right: "iso" = isoforms generated by alternative translation start sites differ in this region; "T58" = threonine 58 (according to Swiss-Prot numbering), the residue that occurs in phosphorylated and glycolyslated forms; "BM" = basic motif; "DNA" = residues involved in DNA binding; "HLH" = helix-loop-helix motif; "LZ" = leucine zipper. The double arrow marks the area in the N-terminal which is known to be disordered. See the text for details.

name or symbol that can be useful for searches in genome resources. For this protein, there is evidence at the protein level, as you can see in the respective field. Below the references, the "Comments" section provides an overview of the knowledge about its biological function. Information on molecular interactions is given as well, although at this point we do not know which residues mediate which interaction. Further down, a list of keywords is provided that can be very useful for obtaining a quick impression about the protein and for finding other proteins that have been annotated with the same keyword. Then, we can see "features," i.e., annotations that can be localized to particular residues or regions. This includes a helix-loop-helix motif, a potential leucine zipper, and a basic motif (*see* **Fig. 3.2**, rectangles marked HLH, LZ, and BM). There is also some information on the 353 amino acids N-terminal of those C-terminal domains, but they consist mostly of posttranslational modifications (e.g., T58, which can be both phosphorylated and glycosylated), and areas with compositional bias. So, what is actually the function of this large area, and which residues are involved in which aspect of this function? At the end of the features list, we can see the position of secondary structure elements that are experimentally validated, in this case three alpha-helices at the C-terminal end. But this still leaves us wondering what is known about the N-terminal two thirds of the protein. Being aware of the amount of literature that is available, we may wonder if it would make a lot of sense to try to locate this information in Pubmed, as such data are often not obvious from the abstract of a paper.

At RefSeq, we can find the entry "NP_002458.2" (17), which is version 2 of entry "NP_002458." Note the comments on the protein isoform created by the usage of downstream alternative start codons, which seem to have some role in the cell. In the "FEATURES" section, we can again find details on the residues involved in particular functions, in this case DNA binding and dimerization (summarized as ovals in **Fig. 3.2**). But still we did not find much new information on the N-terminal part. To see if this lack of annotation of functional modules is due to a high

degree of structural and functional flexibility, a property of protein sequences that can make their investigation more cumbersome, we check DisProt, a database of protein sequences that offers annotation of regions with experimentally validated intrinsic disorder (see the dedicated chapter, and **Table 3.1**). Indeed, with a keyword search for "c-myc," we can find entry "DP00260," which lists a series of experiments that show the propensity for intrinsic disorder in the N-terminal part of the protein that includes the area around T58, the residue we found to be sometimes phosphorylated and sometimes glycosylated (see the double arrow in **Fig. 3.2** for the disordered area). As disordered regions often carry sites of posttranslational modifications that modulate molecular interactions, and as phosphorylation is usually the best understood, we consult PhosphoSite, a resource specialized on organizing information on posttranslational modifications of proteins (**Table 3.1**). This resource can provide us with a nice overview of the role of different phosphorylation sites, the experimental evidence for the modification, and even the cell types it was found in. For example, click on T58 to find out the various functions this intriguing site has been associated with. Now, we are starting to understand a bit better what this region is doing, although a detailed mapping of residues involved in interactions seems nontrivial.

It can be useful as well to study the Swiss-Prot cross-references, e.g., those linking to family and domain databases, as they often contain information about the location of conserved modules that are for some reason not listed in the features section. For example, go back to "MYC_HUMAN" in Swiss-Prot and see the links to entries in Interpro, especially the "Graphical view," which displays their location in the sequence. In this particular case, the Interpro results confirm only what we saw above.

So much for (relatively) well-characterized proteins, like c-myc. But how is the situation for proteins for which only a few or no papers on their biological function are available? For example, in a search at NCBI you picked up the protein sequence "NP_444283" (can you find it?). Looking at the full entry, you can see that it is supposed to belong to the superfamily of thioesterases, that there is some functional connection to the famous Akt/PKB kinase, and also anti-apoptosis. In the section Comments, you can see that the staff at NCBI have already reviewed this protein, so we can find a nice summary of its known functions right there, which at this time is based on only a handful of papers. In the Features section, the C-terminal half of the protein is annotated with "PaaI_thioesterase" (the location of the conserved thioesterase domain), and a cross-reference to more information about this domain is given. Even some binding sites in this area are localized. Note that a link to CCDS is provided, which increases our confidence in the sequence itself. But what is known about the first 150 amino acids again? To find out if Swiss-Prot can tell us

more about this N-terminal half, we can try to find the correspond- ing entry in Swiss-Prot (should be easy, right?). Just use the RefSeq sequence as a query on ExPASy BLAST against Swiss-Prot; follow- ing the links at the NCBI page to Swiss-Prot is possible but can be more complicated. Looking at their BLAST scores, you can see how the hits at the top have very good scores of at least several hundred. But in this case, we want an identical or almost identical sequence from humans, to compare annotations, so click on the score of the first hit and investigate the alignment between both sequences. It should be 100% identical, or at least very close to that. Therefore, the ID of this protein at Swiss-Prot is THEM4_ HUMAN, at least at the moment (for a relatively new gene, the chance of name change is considerable). If we look at the entry of this protein in our BLAST results, it provides a link to Uniprot for the accession Q5T1C6, which is the more stable identifier for the protein (see above). This brings us to Swiss-Prot, where we can search for additional annotations for this protein. There is some additional information in Comments, but the feature table unfor- tunately does not give us more information on the N-terminal half. Also, the link to the graphical view at Interpro only features the conserved C-terminal domain again. Even DisProt and Phospho- Site, at this time, do not contain annotations for this area. There- fore, by analyzing the information in protein sequence databases, we can at least make a statement about the likely function of the conserved C-terminal domain, but would not be able to say much about the N-terminal half. Of course, we could then apply a variety of prediction methods, such as searches for conserved small linear motifs or of course homology with proteins that have known interactions, to come up with testable hypotheses on the possible functions of particular areas (see other chapters).

For some proteins, there may not even be a conserved domain with some functional information available in the resources discussed above. An analysis with the resources described in **Fig. 3.1** may then help to find out if there are clear similarities to better-characterized proteins in Swiss-Prot, RefSeq, or curated organism databases (*see* **Table 3.1**). If we would like to find out more about such a protein, we may have to leave the realm of the protein sequence databases and look up information in gene expression databases like GEO Profiles at NCBI (18) and other resources (see other chapters).

4. Lower Quality Sequence Databases

So far we have focused on scenarios in which the most reliable sequence is the object of analysis and annotation. It can happen, though, that a comprehensive assessment of all potentially relevant

protein sequences is required, including those that do not have much experimental evidence. Many methods that use protein sequences as input, such as evolutionary analysis and function prediction approaches, require as many protein sequences as possible, at least initially. Often, the basis of further analysis is a high-quality, curated alignment that allows the distinction of conserved regions, residues, and local properties. But how to obtain such a high-quality alignment of protein sequences? One of the best ways of generating such an alignment is by using the myHits server at SIB (19). In addition to Swiss-Prot and RefSeq, many other databases of protein sequences can be searched there, while redundancy clustering can help to use the most representative sequences for a cluster of highly similar sequences (using the rules described above). This includes databases of protein sequences that have some evidence at least at the transcript level (such as TREMBL and most sequences in Ensembl), but also databases of lower quality (20) that contain potential protein sequences that require careful assessment through alignment visualization. In cases where sequences with a certain evolutionary distance are needed to improve the usefulness of an alignment, and where higher quality sequences are not available to obtain informative conservation patterns for the regions of interest, such databases can become useful. Also, they can help to identify homologs of a known protein in an additional organism. They basically provide candidate protein sequences created by methods that use EST evidence, and even gene predictions solely based on genomic DNA. Usually, careful inspection of the alignment by an experienced user can help to decide if they could make useful contributions to the question at hand or not. They should be used with caution due to the considerable amount of artifacts in the sequences.

5. Navigating the Labyrinth of Resources

Although the resources discussed above may be sufficient to answer many common questions about protein sequences and their annotations, the challenge remains to navigate the labyrinth of additional resources, of which many are extremely useful and well-maintained. In this last paragraph, we will therefore provide some tips on how to deal with this challenge, which are based on personal experience and may therefore be highly biased.

If you have about an hour or two to spend on your protein of interest, and if you would like to assemble as much information about its functions and where they are localized in the sequence, we would recommend to spend some time carefully analyzing all sections of both the Swiss-Prot and the RefSeq entry, if available.

This should include the inspection of most of the links available there. To avoid getting lost on the Internet, make sure to note useful information and its source, and if it can be assigned to a particular location in the sequence (some may prefer to keep this in a sequence analysis software, others may record it as text or graphics, as in **Fig. 3.2**). If organism-specific resources are available for highly similar sequences, check those as well (*see* **Table 3.1**). In cases where such expert-reviewed entries are not available, focus on similarity search with BLAST at ExPASy and RefSeq, and on InterproScan, and take some time to carefully examine the hits. Simply by studying the available information thoroughly, you will already have an edge over your less ambitious academic competitors who tend to lose their patience too quickly on such tasks. But not you, of course.

References

1. Stretton, A. O. W. (2002) The first sequence: Fred Sanger and insulin. *Genetics* 162, 527–532.

2. Dayhoff, M. O., Eck, R. V., Chang, M. A., Sochard, M. R. (1965) *Atlas of Protein Sequence and Structure*. Silver Spring, Maryland: National Biomedical Research Foundation.

3. Hunt, L. (1984) Margaret Oakley Dayhoff, 1925–1983. *Bull Math Biol* 46, 467–472.

4. George, D. G., Barker, W. C., Hunt, L. T. (1986) The protein identification resource (PIR). *Nucl Acids Res* 14, 11–15.

5. Bairoch, A., Boeckmann, B. (1991) The SWISS-PROT protein sequence data bank. *Nucl Acids Res* 19, 2247–2249.

6. Appel, R. D., Bairoch, A., Hochstrasser, D. F. (1994) A new generation of information retrieval tools for biologists: the example of the ExPASy WWW server. *Trends Biochem Sci* 19, 258–260.

7. Rebhan, M., Chalifa-Caspi, V., Prilusky, J., Lancet, D. (1998) GeneCards: a novel functional genomics compendium with automated data mining and query reformulation support. *Bioinformatics* 14, 656–664.

8. Maglott, D. R., Katz, K. S., Sicotte, H., Pruitt, K. D. (2000) NCBI's LocusLink and RefSeq. *Nucl Acids Res* 28, 126–128.

9. (2004) *Genome Res* 14(Special issue on Ensembl), 925–995.

10. Bairoch, A., Apweiler, R. (1996) The SWISS-PROT protein sequence data bank and its new supplement TREMBL. *Nucl Acids Res* 24, 21–25.

11. Claverie, J. M., Sauvaget, I., Bouqueleret, L. (1985) Computer generation and statistical analysis of a data bank of protein sequences translated from Genbank. *Biochimie* 67, 437–443.

12. Schuler, G. D., Epstein, J. A., Ohkawa, H., Kans, J. A. (1996) Entrez: molecular biology database and retrieval system. *Methods Enzymol* 266, 141–162.

13. Mulder, N., Apweiler, R. (2007) InterPro and InterProScan: tools for protein sequence classification and comparison. *Methods Mol Biol* 396, 59–70.

14. Clamp, M., Fry, B., Kamal, M., Xie, X., Cuff, J., Lin, M. F., Kellis, M., Lindblad-Toh, K., Lander, E. S. (2007) Distinguishing protein-coding and noncoding genes in the human genome. *Proc Natl Acad Sci USA* 104, 19428–19433.

15. Pruitt, K. D., Tatusova, T., Maglott, D. R. (2007) NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucl Acids Res* 35, D61–D65.

16. http://www.expasy.org/uniprot/P01106

17. http://www.ncbi.nlm.nih.gov/entrez/viewer.fcgi?db=protein&id=71774083

18. Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Rudnev, D., Evangelista, C., Kim, I. F., Soboleva, A., Tomashevsky, M., Edgar, R. (2007) NCBI GEO: mining tens of millions of expression profiles – database and tools update. *Nucl Acids Res* 35, D760–D765.

19. Pagni, M., Ioannidis, V., Cerutti, L., Zahn-Zabal, M., Jongeneel, C. V., Falquet, L. (2004) MyHits: a new interactive resource for protein annotation and domain identification. *Nucl Acids Res* 32, W332–W335.

20. Sperisen, P., Iseli, C., Pagni, M., Stevenson, B. J., Bucher, P., Jongeneel, C. V. (2004) trome, trEST and trGEN: databases of predicted protein sequences. *Nucl Acids Res* 32, D509–D511.

21. Bult, C. J., Eppig, J. T., Kadin, J. A., Richardson, J. E., Blake, J. A. (2008) The Mouse Genome Database (MGD): mouse biology and model systems. *Nucl Acids Res* 36, D724–D728.

22. Drysdale, R. A., Crosby, M. A., FlyBase Consortium (2005) FlyBase: Genes and gene models. *Nucl Acid Res* 33, D390–D395.

23. Stein, L. D., Sternberg, P., Durbin, R., Thierry-Mieg, J., Spieth, J. (2001) WormBase: network access to the genome and biology of *Caenorhabditis elegans. Nucl Acids Res* 29, 82–86.

24. Sickmeier, M., Hamilton, J. A., LeGall, T., Vacic, V., Cortese, M. S., Tantos, A., Szabo, B., Tompa, P., Chen, J., Uversky, V. N., Obradovic, Z., Dunker, A. K. (2007) DisProt: the database of disordered proteins. *Nucl Acids Res* 35, D786–D793.

25. Hornbeck, P. V., Chabra, I., Kornhauser, J. M., Skrzypek, E., Zhang, B. (2004) PhosphoSite: a bioinformatics resource dedicated to physiological protein phosphorylation. *Proteomics* 4, 1551–1561.

Chapter 4

Protein Structure Databases

Roman A. Laskowski

Abstract

Web-based protein structure databases come in a wide variety of types and levels of information content. Those having the most general interest are the various atlases that describe each experimentally determined protein structure and provide useful links, analyses, and schematic diagrams relating to its 3D structure and biological function. Also of great interest are the databases that classify 3D structures by their folds as these can reveal evolutionary relationships which may be hard to detect from sequence comparison alone. Related to these are the numerous servers that compare folds – particularly useful for newly solved structures, and especially those of unknown function. Beyond these there are a vast number of databases for the more specialized user, dealing with specific families, diseases, structural features, and so on.

Key words: protein structure, Protein Data Bank (PDB), wwPDB, RCSB, JenaLib, OCA, PDBe, PDBsum, ESD, CATH, SCOP, secondary structure, fold classification, protein–ligand interactions.

1. Introduction

Looking back to 1971, when the Protein Data Bank (PDB) was founded (1), one cannot help feeling that the study of protein structure must have been a lot simpler then. There were only seven experimentally determined protein structures at the time, and the data for each, including the proteins' atomic coordinates, were stored in simple, fixed-format text files. Admittedly, accessing and displaying this information was more tricky and computers with graphics capabilities tended to be bulky and expensive. These days, access and display of the data over the Web are vastly easier, but with this comes the problem, not so much of the huge increase in the amount of information, but in the multiplicity of sources from which it can be obtained. New servers and services continually appear, while existing ones are modified and improved. Conversely,

other servers are abandoned, switched off or neglected, becoming more and more out of date with time. Thus it has become really difficult to know where to go to get relevant answers most easily. Various lists are available on the Web (for example the Nucleic Acids Research (NAR) list at http://www3.oup.co.uk/nar/database/c). However, this chapter aims to highlight some of the more useful, and up-to-date (at time of writing), sources of information on protein structure that are currently available.

2. Structures and Structural Data

2.1. Terminology

First, it is important to define what is meant by the term "protein structure." It is a term that tends to be somewhat loosely used. A preferable term is "model," as the 3D structures of large molecules such as proteins are models of the atom types, atomic x-, y-, z-coordinates and other parameters that best fit the experimental data. The reason that the term "structure" is so commonly used for experimentally determined models is to distinguish these from "theoretical," or "homology-built," models. Nevertheless, it is important to remember both are models of reality and that only the former type is actually based on experimental evidence.

Another loosely used term is "database." Technically, the databases mentioned here are not databases at all, but rather "data resources" – many of which rely on a database for storing and serving up the data. However, the term "database" is becoming common usage for the types of resources described here (e.g., the NAR Database issues), so it is the meaning we will adopt here.

2.2. The PDB and the wwPDB

The primary repository of 3D structural data on proteins (and other biological macromolecules, including RNA, fragments of DNA, carbohydrates, and different complexes of these molecules) is the PDB. As mentioned above, this was founded in 1971 and located at Brookhaven National Laboratories. In October 1998, the management of the archive was taken over by the Research Collaboratory for Structural Bioinformatics (RCSB), a consortium consisting of Rutgers University, the National Institute of Standards and Technology (NIST), and the San Diego Supercomputer Center (2). Since 2003 the archive has been managed by an international consortium called the world-wide Protein Data Bank (wwPDB) whose partners comprise the RSCB, the Macromolecular Structure Database (MSD, now known as the PDBe) at the European Bioinformatics Institute (EBI), the Protein Data Bank Japan (PDBj) at Osaka University and, more recently, the BioMagResBank (BMRB) at the University of Wisconsin-Madison (3). Access to the primary data is via the wwPDB's web site: http://www.wwpdb.org. The

data come in three different formats: old-style PDB-format files, macro-molecular Crystallographic Information File (mmCIF) format (4), and a XML-style format called PDBML/XML (5). For many of the structures, the wwPDB also makes the original experimental data available. Thus, for structural models solved by X-ray crystallography, one can often download the structure factors from which the model was derived, while for structures solved by nuclear magnetic resonance (NMR) spectroscopy, the original distance and angle restraints can be obtained. As of May 2008, the wwPDB contained nearly 50,000 structural models, each identified by a unique four-character reference code, or PDB identifier.

A key task that the wwPDB has performed is the remediation of the legacy PDB archive to fix and make consistent the entire PDB data, in particular relating to ligands and literature references. Another key task, performed by the MSD in association with the UniProt group at the EBI, has been the mapping of the sequences in the PDB entries onto the appropriate sequences in UniProt.

2.3. Structural Data and Analyses

Rather than download the raw data from the wwPDB for each protein of interest, it is usually more convenient to obtain the information of interest directly from one of the myriad protein structure databases on the Web. These come in many shapes and sizes, catering for a variety of needs and interests.

At the simplest level are the sites that provide "atlas" pages – one for every PDB entry – each containing general information obtained from the relevant PDB file. There are usually graphical representations of the structural model together with links that provide interactive 3D visualizations using Java-based, or other, viewers. Each of the founding members of the wwPDB has their own atlas pages: the RCSB, the PDBe, and PDBj. In addition, there are several other sites that have much to recommend them, and some of these will be mentioned below.

Beyond the atlases, there are a host of other types of sites and servers. These include those that provide information on specific structural motifs, focus on selected protein families, classify protein folds, compare protein structures, provide homology-built models for proteins for which no structure has been determined, and so on. This chapter will cherry–pick a few of the more interesting and useful sites to visit.

3. Atlases

Table 4.1 lists the seven best-known and useful of the atlas sites. All have been developed independently and, not unexpectedly, all have much in common as the information comes from the same

Table 4.1
Protein structure atlases

Server	Location	URL	Ref
JenaLib	Fritz Lipmann Institute, Jena, Germany	http://www.fli-leibniz.de/IMAGE.html	(28)
MMDB	NCBI, USA	http://www.ncbi.nlm.nih.gov/Structure/MMDB/mmdb.shtml	(52)
PDBe	EBI, Cambridge, UK	http://www.ebi.ac.uk/pdbe	(23)
OCA	Weizmann Institute, Israel	http://bip.weizmann.ac.il/oca-bin/ocamain	
PDBj	Osaka University, Japan	http://www.pdbj.org	
PDBsum	EBI, Cambridge, UK	http://www.ebi.ac.uk/pdbsum	(29)
RCSB	Rutgers and San Diego, USA	http://www.rcsb.org/pdb	(2)

source: the PDB entry. So the protein name, authors, key reference, experimental methods, atomic coordinates, and so on are obviously all identical. Also common to most of them are certain derived data, including quality assessment of each structural model, and information about the protein's likely "biological unit."

The first of these, quality assessment, is a crucial issue as not all experimentally determined protein models are equally reliable. Much has been written on this topic over the years (6–10). The main problem is that the results of any experiment contain errors, but with protein structure models it is difficult to estimate the severity of those errors. Thus it is not obvious which models are more reliable than others. For X-ray models, the resolution at which the structure was solved and its R-factor can be a rough guide, while for NMR models there is usually even less information to go on. So it is important to have some sort of guide as to whether a given structural model is a reliable one or not and most atlases provide a rough guide.

The second important issue is the one of knowing what a given protein's biological unit is. This is not always obvious from the PDB entry itself. The problem is that the deposited coordinates from an X-ray crystal structure determination correspond to the molecule(s) in the asymmetric unit. This may give a false impression of how the protein operates *in vivo*. So, for example, what may look like a monomer from the PDB entry is, in real life, a dimer, or a trimer, etc. Conversely, the PDB entry might give the coordinates of a dimer, yet the biological unit happens to be a monomer. For any structural analysis, it is crucial to know what the true biological unit is. For some proteins the biological unit has been determined experimentally, and so is known with great confidence. In others

it has to be deduced computationally by analysis of the packing of the individual chains in the crystal. Some interfaces are more substantial than others and hence likely to represent genuine biological interactions rather than happenstance crystal contacts. Most of the atlases provide information on the known, or predicted, biological unit. The most commonly used prediction methods are protein quaternary structure (PQS) (11), and the method that has now superseded it: protein interfaces, surfaces, and assemblies (PISA) (12).

Beyond these general similarities, the atlases differ in sufficient respects to make them complement one another; they differ in what additional information they pull in, the links they make to external resources, and the analyses of the 3D structure that they provide. Consequently, the atlas of choice can be either a matter of personal preference or depend on the type of information one is after.

A recent review compared these atlases, or "comprehensive information resources" as it called them, and identified their similarities and differences (13). Here we include only those that have aspects that make them unique, useful, or interesting, and we focus on those features. We start with the atlases provided by the founding members of the wwPDB, and then discuss some of the others.

3.1. The RCSB PDB   The RCSB's Web site is a very rich source of information about each PDB entry and can be a little overwhelming for novices. Hence a "Quick Tips" icon periodically offers a handy hint about where to go for specific data. There are also various tutorials, including a narrated one using Flash, to help users get started.

3.1.1. Summary Page   The design of each entry's summary information page (**Fig. 4.1**) aims to make clear which is the "primary" information, coming from the experiment (and entered in the header records of the original PDB file) and which information (shown in red) is secondary – i.e., derived from the primary – such as the SCOP (14) and CATH (15) fold classifications, constituent Pfam domains (16), and Gene Ontology (GO) functional annotation (17). Text in bold blue initiates a search for all other PDB entries having that text in common (e.g., other entries with the same author name, or species, or protein classification, etc).

The thumbnail image of the structure has two modes, and you can click between the two: one mode shows the asymmetric unit and the other shows the biological unit, as described above (although in many cases they are identical). For more recent structures, the RCSB site uses the depositors' information on the likely biological unit. For older structures, where there is no depositor information, the biological unit is as predicted by the PQS algorithm.

Fig. 4.1. RCSB atlas page for PDB entry 1ayy, a glycosylasparaginase showing the summary information for this structural model determined by X-ray crystallography at 2.32 Å resolution.

Below the thumbnail are links to no fewer than six Java-based 3D viewers, which allow one to view the molecule interactively, rotating and moving it about on screen. The viewers are KiNG, Jmol, WebMol, Molecular Biology Toolkit (MBT) SimpleViewer, MBT Protein Workshop, and QuickPDB. Which of these you use soon becomes a matter of personal preference (and patience when download times are long).

3.1.2. Other Information

Besides the summary information, further structural details are presented on additional pages titled Biology & Chemistry, Materials & Methods, Sequence Details, and Geometry. Of most interest on these pages is the schematic diagram of the protein's secondary structure (showing α- and π-helices, β-sheets, turns, and disulphide bonds) on the Sequence Details page (*see* **Fig. 4.2**). Indicated on the diagram are any SCOP structural domains.

For ligands, there is the 3D Java-based Ligand Explorer, which allows you to select and view different types of protein–ligand interactions.

Sequence Details ⓘ   **1AYY**

Chain A (polymer 1) ⓘ 🔲 📄

Description	GLYCOSYLASPARAGINASE
Identical chains [show all chains]	C
Chain Type	polypeptide(L)
UniProt reference	Q47898 [show this sequence below]
Length	151 residues
SCOP domain assignment [hide] [reference]	d1ayy.1 Glycosylasparaginase (aspartylglucosaminidase, AGA): 138 residues 📝
DSSP secondary structure [hide] [reference]	39% helical (5 helices; 59 residues) 13% beta sheet (4 strands; 21 residues)

More annotations
Select ▾

SCOP ▸ ——————————————— d1ayy.1 ———————————————

DSSP

PDB TTNKPIVLSTWNFGLHANVEAWKVLSKGGKALDAVEKGVRLVEDDPTERSVGYGGRPDRD
PDB 2   10   20   30   40   50   60

SCOP ▸ ——————————————— d1ayy.1 ———————————————

DSSP

PDB GRVTLDACIMDENYNIGSVACMEHIKNPISVARAVMEKTPHVMLVGDGALEFALSQGFKK
PDB 61   70   80   90   100   110   120

SCOP ▸ —— d1ayy.1 ——

DSSP

PDB ENLLTAESEKEWKEWLKTSQYKPIVNIENHD
PDB 121   130   139

Fig. 4.2. The sequence details for chain A of entry 1ayy in the RCSB server, showing a schematic diagram of the secondary structure assignments for this chain together with the SCOP structural domain.

The advanced search option allows for quite complex queries, and subqueries, on the data, telling you how many hits each set of conditions returns as you refine your search.

3.1.3. Quality Assessment

For assessing the quality of each protein model, the RCSB provides a Ramachandran plot of the protein's main chain φ and ψ torsion angles as generated by the MolProbity structure validation program (18). The tightness of the clustering of points in the "core" regions of the plot can indicate that the structure is of good quality; conversely, a large number of points lying outside these regions can suggest that part, or all, of the structure may be of dubious reliability. Another quality measure is provided by the "fold deviation score" (FDS), given for each chain on the Geometry page.

3.1.4. Molecule of the Month

One particularly eye-catching feature of the RCSB site is the "Molecule of the Month" written by David S. Goodsell of The Scripps Research Institute and illustrated with his beautiful plots. Each month the structure and function of a different protein or protein family are described, with specific references to the PDB entries that have contributed to the understanding of how the proteins achieve their biological roles. The collection of short articles, which are suitable for specialists and nonspecialists alike, dates back to the year 2000 and now numbers over 100 entries, providing a nice reference and educational resource. Particularly stunning is the poster available from the Molecule of the Month home page. The RCSB also provides other educational material and documents suitable for higher level schooling, which is a good place to start when looking for teaching material.

3.1.5. Structural Genomics Portal

There have been a number of worldwide initiatives to solve protein structures in a high-throughput manner, targeting those whose structures are unknown and which may be of particular interest, whether because they represent new families with no structural representatives, or proteins expected to have a novel fold, or proteins of relevance to disease (19). These Structural Genomics projects now contribute nearly a fifth of all newly released structural models each week, and the RCSB's PDB site has a special section devoted to the analysis of these at http://www.sg.pdb.org. This section has its own "Structure of the Month," which features a recently solved structure of special interest.

3.2. The MSD

The MSD (20–23) is a relational database providing an extensive set of search and analysis tools that allows one to explore and mine the structural data in the PDB. The atlas pages for each entry show the usual summary information describing the structure and the experimental details used to obtain it. As well as this summary

page, additional pages provide information on the Assembly (i.e., biological unit, based on PISA), Sequence, Citation, Similarity, and Visualization.

3.2.1. Search Tools

The MSD provides a bewilderingly large number of search tools, both on the text data associated with the PDB entries and on structural data. The full list can be found on the PDBe home page (http://www.ebi.ac.uk/pdbe), but only a few will be mentioned here.

The simplest is PDBeLite, which is an easy-to-use Web form for searching on text or sequence data. The form allows the user to select additional data to be included in the results, such as various identifiers (e.g., UniProt id).

MSDfold uses the secondary structure similarity matching program SSM (24) to find structurally similar proteins for an uploaded PDB file. This is a fold-matching server; others will be mentioned later.

PDBeMotif, which now incorporates MSDsite (25), allows searches for sequence and structural motifs as well as for ligands and specific protein–ligand interactions. Structural motifs can be defined in terms of patterns of secondary structure, φ/ψ and χ angles, and C^α and side-chain positions. Searches are entered either via a simple Web form or using a graphical query generator. The hits from a search can be viewed in three dimensions, aligned by ligand, PROSITE pattern, active site residues, or by environment. One can generate various statistics on protein–ligand interactions (e.g., to compare the different distributions of residues binding to ATP and GTP). Of particular use is an option to upload a PDB file and scan its ligands and active sites against the MSD data.

MSDpro is a graphical query builder, run as a Java applet, which allows the construction of highly specific and fiendishly complex search queries. Each search term is represented by a box on the screen; the manner in which the boxes are laid out within larger "operator" boxes defines the logic of the query.

PDBeAnalysis allows you to quickly analyze various structural parameters in the data across the PDB. It shows the distribution of the selected parameter as a pie chart or histogram (either 1D or 2D), which can then be explored by using click-and-drag to select bins, or ranges of bins, and obtain, say, the list of PDB codes from which the data come. It can be used to perform geometric validation of a given structure, select data based on various filters, or perform statistical analyses of the data in the PDBe. One can even submit one's own SQL queries direct to the database.

3.2.2. The AstexViewer™

The PDBe's primary visualization tool, which has been partly developed by the MSD, is the AstexViewer™@MSD-EBI (AV-MSD) (26, 27). This is a powerful search, comparison, and display

tool in its own right. It runs as a Java applet and aims to provide a graphical interface for the data in the MSD. As some database searches can return hits to multiple structures, so the viewer can present these hits, structurally superposed, along with the corresponding sequence alignments. Structural analyses are presented in a variety of graphs such as histograms, pie charts, dendrograms, and so on (*see* **Fig. 4.3**). These are dynamically linked to one another as well as to the 3D structure and sequence views. Thus, selection of data in a graph or view – either by a mouse-click or by click-and-drag "data brushing" to select a region of data points – is reflected by appropriate highlighting in all other graphs and views. A nice feature, also found in some other structure viewers, is the way the viewer "flies" between different views of the structure, say to centre on a different residue selected from the sequence display by the user. If the new residue is off-screen, the viewer first zooms out to the whole molecule view before zooming in on the new residue of interest (much like the flight in Google Earth). Other nice features are the "hyperbolic" display of the protein sequence, wherein the current region of interest is magnified relative to the
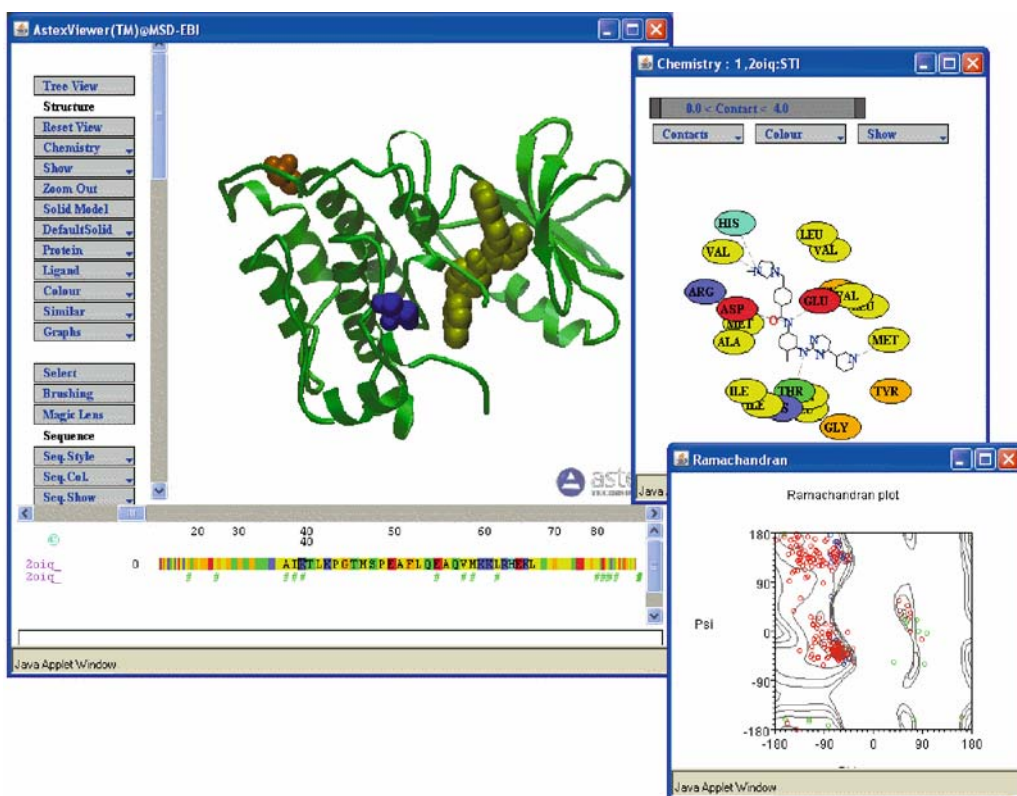
Fig. 4.3. The MSD's Astex viewer showing PDB entry 2oig, tyrosine kinase c-Src, with the bound drug molecule imatinib. Two pop-up windows are shown on the right, one giving the interactions between the ligand and protein residues and the other a Ramachandran plot of the protein's $\psi - \varphi$ torsion angles.

rest of the sequence, and a "magic lens" which, when passed over the 3D structure with the mouse, displays additional structural annotation of the macromolecule within the lens region.

3.3. JenaLib

The Jena Library of Biological Macromolecules, JenaLib (28), was one of the earliest sites offering atlas pages for each PDB entry, and it specializes in hand-curated images of the structures showing functionally informative views. Rather than split information across several pages, JenaLib shows all the information on a single page but has a collapse/expand mechanism for controlling what is shown and what is hidden. In addition to several of the standard 3D viewers, the site features its own: the JenLib Jmol viewer. This viewer is an extension of Jmol, which has a number of options not found in other viewers, such as highlighting of PROSITE motifs, single amino acid polymorphisms, and CATH or SCOP domain structures.

JenaLib has more links to external databases than the other atlas sites and is particularly strong on its many visualizations of each entry – both in terms of its interactive viewing options and preprepared still images.

A particularly useful feature is a form for generating lists of PDB entries according to a number of criteria. Additionally, there are a number of precomputed lists of structures; for example, all nucleic acid structures without protein, all carbohydrate structures, and so on.

3.4. OCA

OCA's main difference from the other atlases is its linkage between proteins and the diseases associated with them. It differs also in that its home page is a search form, much like that of PDBeLite, but with a few additional search options. These include gene name, function, disease, and membrane orientation (for membrane-spanning proteins).

3.5. PDBsum

The last of the atlases described here is PDBsum (29). This aims to be more pictorial than the other sites, illustrating many of its structural analyses by schematic diagrams rather than as tables of numbers. Also, it allows users to upload their own PDB files and get a set of password-protected PDBsum pages generated for them.

3.5.1. Pfam Domain Diagrams

Each entry's summary page has a few useful features not found in the other atlas sites. One of these is a clickable schematic diagram showing how much of the full-length protein sequence is actually represented by the 3D structural model (**Fig. 4.4a**). Often, for example, the 3D structure is only of a single domain. The diagram shows the protein's secondary structure and annotates it with any Pfam sequence domains and CATH structural domains. Occasionally crystallographers assign two separate chain identifiers to different parts of a single protein sequence (perhaps because of a sequence break), and these diagrams can show this immediately (see for example PDB entry 1ayy). The orange "+" icon above the diagram

a.
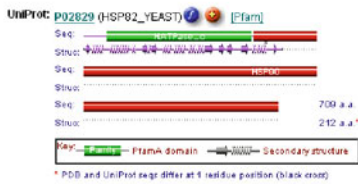
b.

Fig. 4.4. Relationships in PDBsum between 3D structural models and their corresponding UniProt sequence. (a) A schematic Pfam diagram taken from the PDBsum atlas page for PDB entry 2cgf, the N-terminal domain of the yeast HSP90 chaperone. The extent of the 3D structural model is shown beneath the Pfam domains and shows that, indeed, the structural model corresponds to only the N-terminal domain. Clicking on the "+" icon returns all other PDB entries for the given UniProt sequence (HSP82_YEAST). (b) The top 10 PDB entries for this sequence, showing that the structures tend to be of either the N- or C-terminal domains. However the top structure, PDB entry 2cg9, provides the most complete structural model of this protein, albeit at very low resolution and, at the very least, can provide information on how the two domains pack together in 3D.

	identifies other PDB entries containing the same protein sequence. From the list one can see if there any better or more complete structural models of the given protein, as shown in **Fig. 4.4b**.
3.5.2. Quality Assessment	The summary page also provides an at-a-glance guide to the protein's likely reliability by way of a thumbnail Ramachandran plot. Hovering the mouse over the thumbnail pops up a full-size version. As before, a more reliable model will have more points in the core regions (here colored red). Residues in the yellow, disallowed regions are labeled, so if a model has many labeled residues, it might be an idea to look for an alternative. Clicking on the plot goes to a page showing the summary results from the PROCHECK quality assessment program (30) and from this page you can generate a full PROCHECK report.
3.5.3. Enzyme Reactions	For enzymes, the relevant reaction catalyzed by the enzyme is shown by a reaction diagram where possible. If any of the ligands bound to the protein correspond to any of the reactants, cofactors, or products, the corresponding molecule in the diagram is boxed in red. If a ligand is merely similar to one of these, a blue box surrounds the molecule instead and a percentage similarity is quoted.
3.5.4. Figures from Key References	The majority of experimentally determined protein structures are reported in the scientific literature, often in high-profile journals, and each PDB file cites the "key" reference – i.e., the one describing

the structure determination, analysis, and biological significance of the protein. Like the other atlas sites, PDBsum cites this reference, shows its abstract, and provides links to both the PubMed entry and the online version of the article. Where PDBsum differs is that for many of these references, it also gives one or two figures (plus figure legends) taken directly from the key reference itself (31). This is done with permission from the relevant publishers and is useful for two reasons. First, a carefully selected figure can speak volumes about an important aspect of the protein's structure or function. And second, each chapter's lead author is requested to review which figures have been selected by the automated process and, if need be, suggest better choices. About one in six authors take the trouble to do this. And some even add an additional comment to appear on the entry's summary page (e.g., PDB entry 1hz0).

3.5.5. Secondary Structure and Topology Diagrams

From the summary page are various additional pages giving schematic diagrams of different aspects of the 3D structure. The "Protein" page shows a diagram of the chain's secondary structure elements, much like the RCSB's diagram shown in **Fig. 4.2**. Additional features include the annotation of residues that are catalytic – as defined in the Catalytic Site Atlas (CSA) (32) – or are included in the SITE records of the PDB file, or interact with a ligand, DNA/RNA, or metal, or belong to a PROSITE pattern (33). CATH structural domains are marked on the sequence, in contrast to the RCSB's diagram which uses SCOP. Where there is information on the conservation of each residue in the sequence – obtained from the ConSurf-HSSP site (34) – the secondary structure plot can be redisplayed with the residues colored by their conservation.

Next to the secondary structure plot is a topology diagram either of the whole chain or, where it has been divided into its constituent CATH domains, of each domain (**Fig. 4.5**). The diagram shows the connectivity of the secondary structure elements with the constituent β-strands of each β-sheet laid side-by-side, parallel or antiparallel, to show how each sheet in the chain/domain is formed, and where any helices are found relative to the sheets.

3.5.6. Intermolecular Interactions

Some of the other pages for each PDB entry are devoted to schematic representations of intermolecular interactions. Thus for each ligand molecule or metal ion in the structure, there is a schematic LIGPLOT diagram (35) of the hydrogen bonds and nonbonded interactions between it and the residues of the protein to which it is bound (*see* **Fig. 4.6**). Similarly, any DNA–protein interactions are schematically depicted by a NUCPLOT diagram (36). Protein–protein interactions at the interface between two or more chains are shown by two plots: the first shows an overview of which chains interact with which (**Fig. 4.7b**), while the second shows which residues actually interact across the interface (**Fig. 4.7c**).
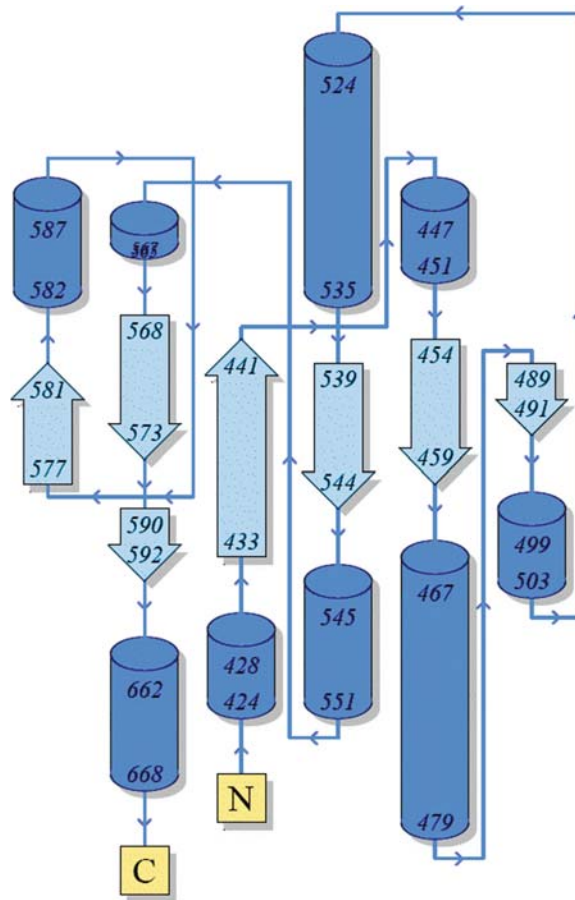
Fig. 4.5. A topology diagram taken from PDBsum for the second domain of chain A in PDB entry 2b6d: a bovine lactoferrin. The diagram illustrates how the β-strands, represented by the block arrows, join up, side-by-side, to form the domain's central β-sheet. The diagram also shows the relative locations of the α-helices, here represented by cylinders. The small arrow indicates the directionality of the protein chain, from the N- to the C-terminus. The numbers within the secondary structural elements correspond to the residue numbering given in the PDB file.

4. Homology Models and Obsolete Entries

4.1. Homology Modeling Servers

As mentioned above, there were nearly 50,000 structural models in the wwPDB as of May 2008. However, some of these were not of proteins and many were duplicates: that is the same protein solved under different conditions, or with different ligands bound, or with one or more point mutations. In terms of unique protein sequences, as defined by the UniProt identifier, this 50,000 corresponded to only about 17,000 unique sequences. [Compare this number with the 105 million sequences in EMBL-Bank (37).] Moreover, for many of these, the 3D structure represents only a part of the full sequence, say merely a fragment or a single domain.
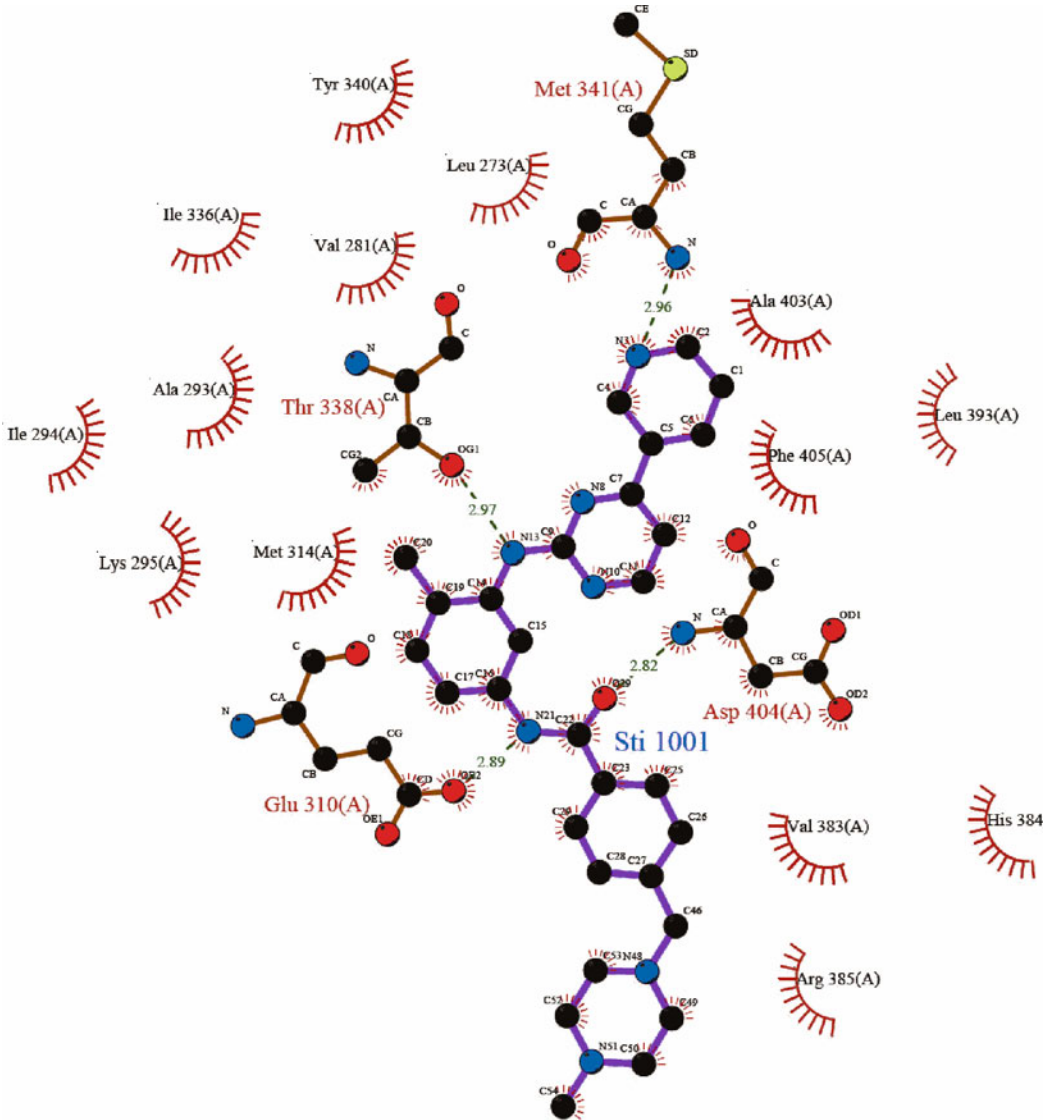
Fig. 4.6. LIGPLOT for PDB entry 2oig, tyrosine kinase c-Src, as given in PDBsum showing the interactions between the bound molecule imatinib (a drug, brand name gleevec) with the residues of the protein. Hydrogen bonds are represented by dashed lines. Residues that interact with the ligand via nonbonded contacts only are represented by the eyelashes.

Thus, if one is interested in a protein for which there are no 3D coordinates available or the coordinates are only of part of the protein, it is common to build a homology model based on the 3D structural model of a closely related protein (if there is one). The PDB used to accept homology-built models together with the experimentally determined ones but, as of 1 July 2002, moved its holding of theoretical models out of the standard PDB archive to a separate ftp site and then, as of October 15, 2006, stopped accepting any new ones. As of May 2008, there were only 1,358 models on the ftp site so, with such a small number, it is unlikely that one's protein of interest will be among them.
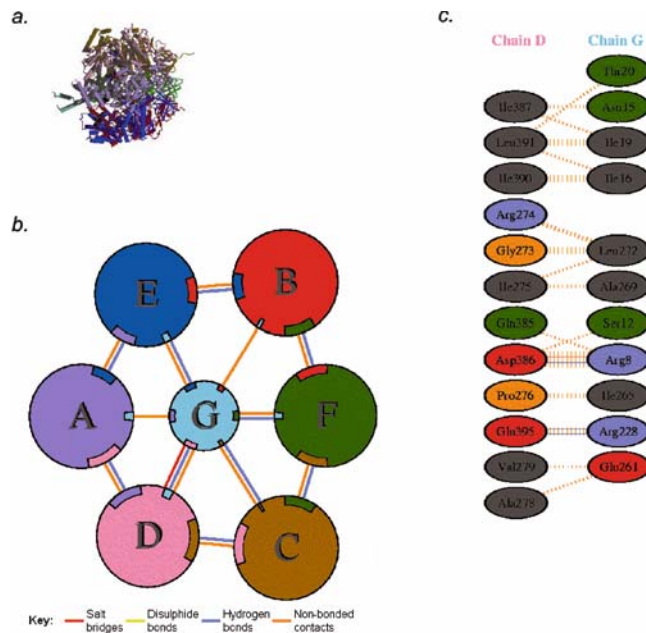
Fig. 4.7. Extracts from the protein–protein interaction diagrams in PDBsum for PDB entry 1cow, bovine mitochondrial F1-ATPase. (a) Thumbnail image of the 3D structural model which contains seven protein chains: three of ATPA1_BOVIN (chains A, B and C), three of ATPB_BOVIN (chains D, E and F) and a fragment of ATPG_BOVIN (chain G). (b) Schematic diagram showing the interactions between the chains. The area of each circle is proportional to the surface area of the corresponding protein chain. The extent of the interface region on each chain is represented by a coloured wedge whose colour corresponds to the colour of the other chain and whose size signifies the interface surface area. (c) A schematic diagram showing the residue–residue interactions across one of the interfaces, namely that between chains D and G. Hydrogen bonds and salt bridges are shown as solid lines while non-bonded contacts are represented by dashed lines.

The alternative is to build a homology model oneself, and there are various servers that will perform the process largely, or completely, automatically. The best known is SWISS-MODEL (38). This accepts a protein sequence and will return a 3D model if it is able to build one. More advanced users can submit multiple sequence alignments and manually refine the final model. It is important to remember that any homology-built model will, at best, be imperfect and at worst totally misleading – particularly if one or more of the structural models that act as a template for the model contain errors. So a key part of SWISS-MODEL is the various validation checks applied to each model to provide the user with an idea of its likely quality.

Table 4.2i shows a list of automated homology modeling Web servers. These are regularly tested by the EVA server (39), which produces statistics on accuracy and ranks the servers by various criteria (**Table 4.2ii**).

Table 4.2
Homology model servers

Server	Location	URL	Ref
(i) Automatic homology modeling			
3D-JIGSAW	Imperial Cancer Research Fund, UK	http://www.bmm.icnet.uk/servers/ 3djigsaw	(53)
CPHmodels	Technical University of Denmark	http://www.cbs.dtu.dk/services/ CPHmodels	(54)
ESyPred3D	University of Namur, Belgium	http://www.fundp.ac.be/urbm/ bioinfo/esypred	(55)
SWISS-MODEL	Biozentrum Basel, Switzerland	http://swissmodel.expasy.org	(38)
(ii) Evaluation of modeling servers			
EVA	Columbia University, USA	http://cubic.bioc.columbia.edu/eva	(39)
(iii) Precomputed homology models			
SWISS-MODEL Repository	Biozentrum Basel, Switzerland	http://swissmodel.expasy.org/ repository	(40)
ModBase	University of California San Francisco, USA	http://modbase.compbio.ucsf.edu	(56)
PDB archive	RCSB, USA	ftp://ftp.wwpdb.org/pub/pdb/ data/structures/models	

Aside from building a model yourself, it may be possible to download a ready-built, off-the-shelf one. The SWISS-MODEL Repository (40) contained over 1.3 million models in May 2008, each accessible by its UniProt accession number or identifier. Similarly, ModBase (41) contains a large number of precomputed models for sequences in the SwissProt and TrEMBL databases – 4.3 million models for 1.3 million proteins in May 2008. **Table 4.2iii** gives the URLs and references for these servers.

4.2. Threading Servers   What about cases where there is no sufficiently similar protein of known structure and thus no possibility of building a homology model? In these cases, it is sometimes necessary to resort to desperate measures such as secondary structure prediction and fold recognition, or "threading." The results from these methods need to be treated with extreme care. Occasionally, these methods approximate the right answer – usually for small, single-domain proteins where they may produce topologically near correct models (42) – but generally, they are wildly wrong and so should be used only as a last resort. A full list of these servers can be found on the LiveBench Web site (http://www.bioinfo.pl/LiveBench), which regularly evaluates these servers (43).

Table 4.3
Fold classification and comparison servers

Server	Location	URL	Ref
(i) Fold classification			
CATH	University College London, UK	http://www.cathdb.info	(57)
SCOP	University of Cambridge, UK	http://scop.mrc-lmb.cam.ac.uk/scop	(14)
(ii) Fold comparison			
CE	University of California San Diego, USA	http://cl.sdsc.edu/ce.html	(58)
Dali	University of Helsinki, Finland	http://ekhidna.biocenter.helsinki.fi/dali_server	(59)
DBAli	University of California San Francisco, USA	http://www.salilab.org/DBAli	(60)
FATCAT	Burnham Institute, USA	http://fatcat.burnham.org	(61)
MATRAS	Nara Institute of Science and Technology, Japan	http://biunit.aist-nara.ac.jp/matras	(62)
SSM	European Bioinformatics Institute, UK	http://www.ebi.ac.uk/msd-srv/ssm	(24)
TOPSCAN	University College London, UK	http://www.bioinf.org.uk/topscan	(63)
VAST	NCBI, USA	http://www.ncbi.nlm.nih.gov/Structure/VAST/vastsearch.html	(64)

4.3. Obsolete Entries   As experimental methods improve, better data sets are collected or earlier errors are detected, so some structural models in the PDB become obsolete. Many are replaced by improved structural models, whereas others are simply quietly withdrawn. None of these obsolete entries disappear entirely, though. Some of the atlases mentioned above include the obsolete entries together with the current ones, and there used to be a special database devoted to their memory: the Archive of Obsolete PDB Entries. Sadly, this, too, is now obsolete. However, the original PDB files can still be found on the wwPDB's ftp servers.

5. Fold Databases

5.1. Classification Schemes   There are currently around 900 known fold groups (44). Many proteins comprise more than one structural domain, with each domain being described by its own fold and often able to fold up

independently of the rest of the protein. There have been a number of efforts to classify protein domains in a hierarchical manner. The two current market leaders in this field are the SCOP and CATH hierarchical classification systems (*see* **Table 4.3i**). In CATH, protein structures are classified using a combination of automated and manual procedures, with four major levels in the hierarchy: Class, Architecture, Topology (fold family), and Homologous superfamily (45). In SCOP, the classification is more manual, although some automated methods are employed. Comparisons between the two classification schemes have shown there to be much in common, although there are differences, primarily in how the structures are chopped into domains (46).

Recently, it has become apparent that protein folds are not the discrete units that these classification schemes might imply, but rather that protein structure space is a continuum (47). However, the two databases are very valuable resources because they group domains by their evolutionary relationships even where this is not apparent from any similarities in the sequences.

5.2. Fold Comparison

Often a given structural domain is associated with a specific biological function. However, the so-called superfolds, which are more common than other folds, tend to be responsible for a wide range of functions (48). There are a large number of Web servers that can identify all proteins sharing a given protein's fold. The main problem is knowing which server to use. Each uses a different algorithm or has a different way of assessing the significance of a match. **Table 4.3ii** lists a selection of the more popular servers. A fuller list, together with brief descriptions of the algorithms and a comparison between them can be found in various comparisons that have been made between them (49, 50).

6. Miscellaneous Databases

6.1. Selection of Data Sets

For any bioinformatics analysis involving 3D structural models, it is important to get a valid and representative dataset of models of as high a quality as possible. To help in this process there are various servers that allow you to obtain such lists based on various selection criteria. **Table 4.4** lists several such servers.

6.2. Uppsala Electron Density Server (EDS)

As has been mentioned a couple of times already, a key aspect of any structural model is how reliably it represents the protein in question. A poor-quality model limits what structural or functional conclusions can be drawn from it. For X-ray models, in addition to the geometrical checks mentioned above, the most useful guide to reliability is how well the model agrees with

Table 4.4
Selection of data sets

Server	Location	URL	Ref
ASTRAL	University of Berkeley, USA	http://astral.berkeley.edu	(65)
JenaLib (Entry Lists)	Fritz Lipmann Institute, Jena, Germany	http://www.fli-leibniz.de/ImgLibPDB/ pages/entry_list-customize.html	
PDBeSelect	European Bioinformatics Institute, UK	http://www.ebi.ac.uk/pdbe-as/ pdbeselect	
PDBselect	University of Applied Sciences, Giessen, Germany	http://bioinfo.tg.fh-giessen.de/pdbselect	(66)
PISCES	Fox Chase Cancer Center, Philadelphia, USA	http://dunbrack.fccc.edu/PISCES.php	(67)

Table 4.5
Miscellaneous servers

Server	Location	URL	Ref
3D Complex	MRC, Cambridge, UK	http://www.supfam.org/elevy/ 3dcomplex/Home.cgi	
Database of macromolecular movements	Yale, USA	http://molmovdb.org	(68)
Electron density server (EDS)	Uppsala, Sweden	http://eds.bmc.uu.se/eds	(51)
Orientations of proteins in membranes (OPM)	University of Michigan, USA	http://opm.phar.umich.edu	(69)
pKnot server	National Chiao Tung University, Taiwan	http://pknot.life.nctu.edu.tw	(70)
Protein knots	Massachusetts Institute of Technology, USA	http://knots.mit.edu	(71)

the experimental data on which it was based. The Uppsala EDS (51), displays the electron density maps for PDB entries for which the experimental structure factors are available. The server also provides various useful statistics about the models. For example, the plots of the real-space R-factor (RSR) indicate how well each residue fits its electron density; any tall red spikes are regions to be wary of. Other useful plots include the occupancy-weighted average temperature factor and a Z-score associated with the residue's RSR for the given resolution.

6.3. Curiosities   Finally, there are various sites which deal with slightly more offbeat aspects of protein structure. Some are included in **Table 4.5**. A couple detects knots in protein folds: Protein Knots and the pKnot Web server. The former lists 44 PDB entries containing knotted proteins, classified according to type of knot. Another interesting site, which can while away part of an afternoon, is the Database of Macromolecular Movement which holds many movies showing proteins in motion. Also included is a "Morph Server" which will produce 2D and 3D animations by interpolating between two submitted protein conformations – very useful for producing animations for presentations or Web sites.

7. Summary

This chapter has described some of the more generally useful protein structure databases. There are many, many more that have not been mentioned. Some are very small and specialized, such as the so-called hobby databases, created by a single researcher and lovingly crafted and conscientiously updated – until, that is, the funding runs out, or the researcher moves on to another post and the database is abandoned and neglected. The larger and more widely used databases have better resources to keep them ticking over, but tend to suffer from a great deal of duplication and overlap. This can be seen in the large numbers of PDB atlases and fold comparison servers. Perhaps one day, a single server of each type will emerge combining the finer aspects of all others to make life a lot easier for the end users of the data.

Acknowledgment

References

1. Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F. Jr., Brice, M. D., Rodgers, J. R., et al. (1977) The Protein Data Bank: a computer-based archival file of macromolecular structures. *J Mol Biol* 112, 535–542.

2. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., et al. (2000) The Protein Data Bank. *Nucleic Acids Res* 28, 235–242.

3. Berman, H. M., Henrick, K., Nakamura, H. (2003) Announcing the worldwide Protein Data Bank. *Nat Struct Biol* 10, 980.

4. Westbrook, J., Fitzgerald, P. M. (2003) The PDB format, mmCIF, and other data formats. *Methods Biochem Anal* 44, 161–179.

5. Westbrook, J., Ito, N., Nakamura, H., Henrick, K., Berman, H. M. (2005) PDBML: the representation of archival macromolecular structure data in XML. *Bioinformatics* 21, 988–992.

6. Brändén, C.-I., Jones, T. A. (1990) Between objectivity and subjectivity. *Nature* 343, 687–689.

7. Hooft, R. W. W., Vriend, G., Sander, C., Abola, E. E. (1996) Errors in protein structures. *Nature* 381, 272.

8. Kleywegt, G. J. (2000) Validation of protein crystal structures. *Acta Crystallogr* D56, 249–265.

9. Laskowski, R. A. (2009) Structural quality assurance, in (Gu, J., Bourne, P. E., eds.) *Structural Bioinformatics*, 2nd ed., John Wiley, New Jersey, pp. 341–375.

10. Brown, E. N., Ramaswamy, S. (2007) Quality of protein crystal structures. *Acta Crystallogr* D63, 941–950.

11. Henrick, K., Thornton, J. M. (1998) PQS: a protein quaternary structure file server. *Trends Biochem Sci* 23, 358–361.

12. Krissinel, E., Henrick, K. (2007) Inference of macromolecular assemblies from crystalline state. *J Mol Biol* 372, 774–797.

13. Hühne, R., Koch, F. T., Sühnel, J. (2007) A comparative view at comprehensive information resources on three-dimensional structures of biological macro-molecules. *Brief Funct Genomic Proteomic* 6, 220–239.

14. Murzin, A. G., Brenner, S. E., Hubbard, T., Chothia, C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol* 247, 536–540.

15. Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., Thornton, J. M. (1997) CATH: a hierarchic classification of protein domain structures. *Structure* 5, 1093–1108.

16. Finn, R. D., Mistry, J., Schuster-Böckler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T. et al. (2006) Pfam: clans, web tools and services. *Nucleic Acids Res* 34, D247–D251.

17. Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., Maslen, J. et al. (2004) The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Res* 32, D262–D266.

18. Lovell, S. C., Davis, I. W., Arendall III, W. B., de Bakker, P. I. W., Word, J. M., Prisant, M. G. et al. (2003) Structure validation by C-alpha geometry: phi, psi, and C-beta deviation. *Proteins Struct Funct Genet* 50, 437–450.

19. Brenner, S. E. (2001) A tour of structural genomics. *Nat Rev Genet* 2, 801–809.

20. Boutselakis, H., Dimitropoulos, D., Fillon, J., Golovin, A., Henrick, K., Hussain, A. et al. (2003) E-MSD: the European Bioinformatics Institute Macromolecular Structure Database. *Nucleic Acids Res* 31, 458–462.

21. Golovin, A., Oldfield, T. J., Tate, J. G., Velankar, S., Barton, G. J., Boutselakis, H. et al. (2004) E-MSD: an integrated data resource for bioinformatics. *Nucleic Acids Res* 32, D211–D216.

22. Velankar, S., McNeil, P., Mittard-Runte, V., Suarez, A., Barrell, D., Apweiler, R. et al. (2005) E-MSD: an integrated data resource for bioinformatics. *Nucleic Acids Res* 33, D262–D265.

23. Tagari, M., Tate, J., Swaminathan, G. J., Newman, R., Naim, A., Vranken, W., et al. (2006) E-MSD: improving data deposition and structure quality. *Nucleic Acids Res.* 34, D287–D290.

24. Krissinel, E., Henrick K. (2004) Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr* D60, 2256–2268.

25. Golovin, A., Dimitropoulos, D., Oldfield, T., Rachedi, A., Henrick, K. (2005) MSDsite: a database search and retrieval system for the analysis and viewing of bound ligands and active sites. *Proteins* 58, 190–199.

26. Hartshorn, M. J. (2002) AstexViewer: a visualisation aid for structure-based drug design. *J Comput-Aided Mol Design* 16, 871–881.

27. Oldfield, T. J. (2004) A Java applet for multiple linked visualization of protein structure and sequence. *J Comput-Aided Mol Design* 18, 225–234.

28. Reichert, J., Sühnel, J. (2002) The IMB Jena Image Library of Biological Macromolecules: 2002 update. *Nucleic Acids Res* 30, 253–254.

29. Laskowski, R. A., Chistyakov, V. V., Thornton, J. M. (2005) PDBsum more: new summaries and analyses of the known 3D structures of proteins and nucleic acids. *Nucleic Acids Res* 33, D266–D268.

30. Laskowski, R. A., MacArthur, M. W., Moss, D. S., Thornton, J. M. (1993) PRO-CHECK – a program to check the stereochemical quality of protein structures. *J Appl Crystallogr* 26, 283–291.

31. Laskowski, R. A. (2007) Enhancing the functional annotation of PDB structures in PDBsum using key figures extracted from the literature. *Bioinformatics* 23, 1824–1827.

32. Porter, C. T., Bartlett, G. J., Thornton, J. M. (2004) The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Res* 32, D129–D133.

33. Sigrist, C. J. A., Cerutti, L., Hulo, N., Gattiker, A., Falquet, L., Pagni, M. et al. (2002) PROSITE: a documented database using patterns and profiles as motif descriptors. *Brief Bioinform* 3, 265–274.

34. Glaser, F., Rosenberg, Y., Kessel, A., Pupko, T., Ben Tal, N. (2004) The ConSurf-HSSP database: the mapping of evolutionary conservation among homologs onto PDB structures. *Proteins* 58, 610–617.

35. Wallace, A. C., Laskowski, R. A., Thornton, J. M. (1995) LIGPLOT: A program to generate schematic diagrams of protein-ligand interactions. *Prot Eng* 8, 127–134.

36. Luscombe, N. M., Laskowski, R. A., Thornton, J. M. (1997) NUCPLOT: a program to generate schematic diagrams of protein-nucleic acid interactions. *Nucleic Acids Res* 25, 4940–4945.

37. Kulikova, T., Akhtar, R., Aldebert, P., Althorpe, N., Andersson, M., Baldwin, A. et al. (2007) EMBL Nucleotide Sequence Database in 2006. *Nucleic Acids Res* 35, D16–D20.

38. Schwede, T., Kopp, J., Guex, N., Peitsch, M. C. (2003) SWISS-MODEL: an automated protein-homology server. *Nucleic Acids Res* 31, 3381–3385.

39. Eyrich, V. A., Marti-Renom, M. A., Przybylski, D., Madhusudhan, M. S., Fiser, A., Pazos, F. et al. (2001) EVA: continuous automatic evaluation of protein structure prediction servers. *Bioinformatics* 17, 1242–1243.

40. Kopp, J., Schwede, T. (2004) The SWISS-MODEL Repository of annotated three-dimensional protein structure homology models. *Nucleic Acids Res* 32, D230–D234.

41. Pieper, U., Eswar, N., Braberg, H., Madhusudhan, M. S., Davis, F. P., Stuart, A. C., et al. (2004) MODBASE: a database of annotated comparative protein structure models, and associated resources. *Nucleic Acids Res* 32, D217–D222.

42. Moult, J. (2005) A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Curr Opin Struct Biol* 15, 285–289.

43. Bujnicki, J. M., Elofsson, A., Fischer, D., Rychlewski, L. (2001) Livebench-1: continuous benchmarking of protein structure prediction servers. *Protein Sci* 10, 352–361.

44. Marsden, R. L., Ranea, J. A. G., Sillero, A., Redfern, O., Yeats, C., Maibaum, M. et al. (2006) Exploiting protein structure data to explore the evolution of protein function and biological complexity. *Phil Trans R Soc B-Biol Sci* 361, 425–440.

45. Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., Thornton, J. M. (1997) CATH: a hierarchic classification of protein domain structures. *Structure* 5, 1093–1108.

46. Jefferson, E. R., Walsh, T. P., Barton, G. J. (2008) A comparison of SCOP and CATH with respect to domain-domain interactions. *Proteins* 70, 54–62.

47. Kolodny, R., Petrey, D., Honig, B. (2006) Protein structure comparison: implications for the nature of 'fold space', and structure and function prediction. *Curr Opin Struct Biol* 16, 393–398.

48. Orengo, C. A., Jones, D. T., Thornton, J. M. (1994) Protein superfamilies and domain superfolds. *Nature* 372, 631–634.

49. Novotny, M., Madsen, D., Kleywegt, G. J. (2004) Evaluation of protein fold comparison servers. *Proteins* 54, 260–270.

50. Carugo, O. (2006) Rapid methods for comparing protein structures and scanning structure databases. *Curr Bioinform* 1, 75–83.

51. Kleywegt, G. J., Harris, M. R., Zou, J.-y., Taylor, T. C., Wählby, Jones T. A. (2004) The Uppsala Electron-Density Server. *Acta Crystallogr* D60, 2240–2249.

52. Chen, J., Anderson, J. B., DeWeese-Scott, C., Fedorova, N. D., Geer, L. Y., He, S. et al. (2003) MMDB: Entrez's 3D-structure database. *Nucleic Acids Res* 31, 474–477.

53. Bates, P. A., Kelley, L. A., MacCallum, R. M., Sternberg, M. J. E. (2001) Enhancement of protein modelling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. *Proteins* 5, 39–46.

54. Lund, O., Frimand, K., Gorodkin, J., Bohr, H., Bohr, J., Hansen, J., Brunak, S. (1997) Protein distance constraints predicted by neural networks and probability density functions. *Protein Eng* 10, 1241–1248.

55. Lambert, C., Leonard, N., De Bolle, X., Depiereux, E. (2002) ESyPred3D: Prediction of proteins 3D structures. *Bioinformatics* 18, 1250–1256.

56. Pieper, U., Eswar, N., Davis, F. P., Braberg, H., Madhusudhan, M. S., Rossi, A. et al. (2006) MODBASE: a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Res* 34, D291–D295.

57. Pearl, F., Todd, A., Sillitoe, I., Dibley, M., Redfern, O., Lewis, T. et al. (2005) The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. *Nucleic Acids Res* 33, D247–D251.

58. Shindyalov, I. N., Bourne, P. E. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng* 11, 739–747.

59. Holm, L., Sander, C. (1996) Mapping the protein universe. *Science* 273, 595–603.

60. Marti-Renom, M. A., Pieper, U., Madhusudhan, M. S., Rossi, A., Eswar, N., Davis, F. P. et al. (2007) DBAli tools: mining the protein structure space. *Nucleic Acids Res* 35, W393–W397.

61. Ye, Y., Godzik, A. (2003) Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics* 19, ii246–ii255.

62. Kawabata, T. (2003) MATRAS: a program for protein 3D structure comparison. *Nucleic Acids Res* 31, 3367–3369.

63. Martin, A. C. R. (2000) The ups and downs of protein topology; rapid comparison of protein structure. *Protein Eng* 13, 829–837.

64. Gibrat, J. F., Madej, T., Bryant, S. H. (1996) Surprising similarities in structure comparison. *Curr Opin Struct Biol* 6, 377–385.

65. Chandonia, J. M., Hon, G., Walker, N. S., Lo Conte, L., Koehl, P., Levitt, M., Brenner, S. E. (2004) The ASTRAL compendium in 2004. *Nucleic Acids Res* 32, D189–D192.

66. Hobohm, U., Scharf, M., Schneider, R., Sander, C. (1992) Selection of representative protein data sets. *Protein Sci* 1, 409–417.

67. Wang, G., Dunbrack, R. L. Jr. (2003) PISCES: a protein sequence culling server. *Bioinformatics* 19, 1589–1591.

68. Gerstein, M., Krebs, W. (1998) A database of macromolecular motions. *Nucleic Acids Res* 26, 4280–4290.

69. Lomize, M. A., Lomize, A. L., Pogozheva, I. D. and Mosberg, H. I. (2006) OPM: Orientations of Proteins in Membranes database. *Bioinformatics* 22, 623–625.

70. Lai, Y. L., Yen, S. C., Yu, S. H., Hwang, J. K. (2007) pKNOT: the protein KNOT web server. *Nucleic Acids Res* 35, W420–W424.

71. Kolesov, G., Virnau, P., Kardar, M., Mirny, L. A. (2007) Protein knot server: detection of knots in protein structures. *Nucleic Acids Res* 35, W425–W428.

Chapter 5

Protein Domain Architectures

Nicola J. Mulder

Abstract

Proteins are composed of functional units, or domains, that can be found alone or in combination with other domains. Analysis of protein domain architectures and the movement of protein domains within and across different genomes provide clues about the evolution of protein function. The classification of proteins into families and domains is provided through publicly available tools and databases that use known protein domains to predict other members in new proteins sequences. Currently at least 80% of the main protein sequence databases can be classified using these tools, thus providing a large data set to work from for analyzing protein domain architectures. Each of the protein domain databases provide intuitive web interfaces for viewing and analyzing their domain classifications and provide their data freely for downloading. Some of the main protein family and domain databases are described here, along with their Web-based tools for analyzing domain architectures.

Key words: protein domain, protein family, InterPro, Gene Ontology, domain architecture.

1. Introduction

A protein family is generally described as a group of evolutionarily related protein sequences, which, by inference means they share similarity in their protein sequences. However, the proteins may not necessarily be conserved throughout the full length of the sequence; they may only share conserved regions. These regions, or mobile elements, that can be found alone or in combination with other conserved regions are referred to as domains, and usually form individual functional units. Domains are independent evolutionary elements that tend to have their functions conserved over time, although they are mobile and free to move and pair with other domains. Protein domain arrangements arise from events such as recombination, exon-shuffling, gene fusion, domain loss,

etc. The representation of protein sequences as sets of ordered functional domains provides a useful way of investigating the evolution of protein functions and multidomain proteins (1). One of the problems with domain analysis, however, is the definition and description of domains. In some cases, particularly where 3D structures are available in the Protein Data Bank (2), the domain boundaries can be clearly defined, while in others, the domain boundaries are estimated based on the beginning and ending of conserved regions. There are different databases that try to predict protein families and domains, and in some cases, the distinction between these two is indistinct, so here I have included information on databases providing protein families as well as domains. In addition, domains can be grouped into domain families, thus adding complexity to the use of this data for domain architecture analysis.

In order to keep up with the task of classifying new protein sequences into families and domains, several protein signature methods have emerged. These are based on conserved regions identified from sequence alignments. These conserved areas of a protein family, domain, or functional site form the basis for developing a protein signature, or "description," of the family using several different methods, including regular expressions (for patterns of conserved residues), profiles, and Hidden Markov Models (HMMs). Regular expressions describe a group of amino acids that constitute a usually short, but highly conserved region within a protein sequence that is thought to be important for function. The regular expression uses the alignment to determine and describe the occurrence of amino acids at each position in the motif. Regular expressions are usually quite specific and proficient at identifying highly conserved functional sites, but, as a consequence, also have low flexibility. There is either a match or no match, with no in betweens. Profiles and HMMs overcome this limitation as they are able to cover longer conserved regions and tolerate mismatches. A profile is a table of position-specific amino acid weights and gap costs and describes the probability of finding an amino acid at a given position in the sequence (3). The scores at each position are used to calculate the similarity between a profile and a sequence based on an original alignment. HMMs (4) are statistical models based on probabilities rather than on scores and are represented by a series of states that can match symbols and transitions between these states, and at each state a symbol is matched with a certain probability. Each transition has a probability associated with it that describes how likely it is to move between any two states.

Regular expressions, profiles, and HMMs have been built by different databases for thousands of known protein families and domains. Examples of well-known databases in the public domain that use these methods or protein sequence clustering for

generating signatures include Pfam (5), PRINTS (6), PROSITE (7), ProDom (8), SMART (9), TIGRFAMs (10), PIRSF (11), SUPERFAMILY (12), Gene3D (13), PANTHER (14), CDD (15), and Everest (16). These databases each have their own area of focus and may differ in their coverage of sequence space; however, since most are based on the same protein sequence sets, usually UniProt Knowledgebase (UniProtKB) (17), there is also a lot of overlap in their families and domains. Each database usually provides information on their families and domains and a search algorithm for identifying new members. These enable researchers to search the protein sequence databases and classify proteins into families and predict the presence of domains. While the number of protein sequences entering the databases is increasing at an exponential rate, the number of new protein domains appearing is tailing off (18).

The vast amount of protein sequence classification data provided by protein signature databases enables not only easier functional annotation of sequences, but also provides the potential for performing evolutionary analyses on the sequences. As mentioned previously, protein domains are independent functional units that have been maintained throughout evolution and moved within and across different proteins and genomes. The combination and order of domains on protein sequences can provide information on the origin of proteins and new functions, as well as identify gene fusion events leading to the formation of multifunctional proteins. It is therefore useful to identify tools that enable in-depth investigation of protein domain architectures. In general, such tools are provided by the databases that generate the domain signatures in the first place and thus are biased toward their own combination of characterized domains; however, there are some tools that integrate data from different databases and enable more comprehensive analyses. Some of the main protein family and domain databases are described here, along with some of the resources for performing domain architecture analyses.

2. Protein Family and Domain Databases

2.1. CDD

The Conserved Domain Database (CDD) (15) (http://www.ncbi. nlm.nih.gov/Structure/cdd/cdd.shtml) contains protein signatures from Pfam (5), SMART (9), and Conserved Domains developed by the CDD group. Since many of these databases describe overlapping domains, to remove redundancy, the different HMMs are clustered based on overlaps in their protein members. HMMs within clusters that do not add significantly to the coverage of the cluster are removed to reduce redundancy, and

where families need to be divided into subfamilies to increase depth, new CDD models are developed. In the development of new models, structural alignments are used to increase the accuracy of domain boundary definition. Position-specific scoring matrices are generated from sequence alignments and are then assembled for searching against using reverse position-specific BLAST (RPS-BLAST) (19). Results for all proteins in Entrez are precomputed, but users can submit query sequences that will be matched against all three libraries of HMMs, with additional matches to Clusters of Orthologous groups (COGs) (20). The sequences are not actually run through HMM-searching algorithms, but rather RPS-BLAST, which is a quicker alternative. The data and tools are available via the Entrez system, which also allows for linking to other Entrez databases.

2.2. InterPro

Instead of describing each of the major protein signature databases individually, an amalgamated database, InterPro (21), is described, as it integrates ten of the major protein signature databases, Pfam, PRINTS, PROSITE, ProDom, SMART TIGR-FAMs, PIRSF, SUPERFAMILY, Gene3D, and PANTHER, into a single resource. InterPro (http://www.ebi.ac.uk/interpro) is an integrated documentation resource for protein families, domains, and functional sites. Integration of the protein signatures is done manually to rationalize where signatures from different databases are describing the same protein family or domain. Related signatures are grouped into InterPro entries, which have high-quality annotation and links to a number of different protein function, specialized protein family, interaction, and literature databases. Where possible, InterPro entries are mapped to Gene Ontology (GO) (22) terms to enable large-scale annotation of the corresponding proteins to GO terms. All matches of the signatures in these databases against the UniProtKB database are precomputed and are viewable in different graphical formats. Users can also submit their own sequences for scanning by InterProScan (23), where each sequence is run through all the different algorithms of the member databases at once, with the results provided in a single format.

Through its matches to corresponding protein signatures from its member databases, InterPro classifies protein sequences into families and domains. Currently, the signatures in InterPro cover 80% of UniProtKB. Although not all of these matches are domains (many are families and a few are small functional sites), the matches provide enormous potential for investigating domain architectures. InterPro, as well as some of its member databases, provides facilities for identifying and retrieving proteins sharing the same architectures or for finding proteins sharing a common domain. Some of these methods are described in more detail in Section 3 below.

2.3. Everest

EVolutionary Ensembles of REcurrent SegmenTs (EVEREST) (http://www.everest.cs.huji.ac.il/) (16) is a database of domains, described by automatically generated HMMs. The underlying definition of a domain used by EVEREST is a "continuous sequence of amino acids that recurs in protein space." In their methodology, a database of protein segments (putative domains) is constructed from all-against-all pairwise sequence comparisons, and these segments are then clustered. One or more HMMs are generated for each cluster that obeys certain criteria for what a domain should look like, given a set of known candidate domains. The process goes through a number of iterations to result in a final set of around 1 million domains covering over 80% of UniProtKB (16). The EVEREST Web interface provides annotation given by SCOP (24), CATH (25), and Pfam-A (curated part of Pfam), as well as visualisation tools for viewing the domain organization of proteins within a family.

3. Tools for Analyzing Protein Domain Architectures

Many of the protein family and domain databases provide intuitive Web interfaces that enable analysis of proteins containing their domains. Some of these are described below and their outputs are illustrated for comparison using a single protein example, the mouse Death domain-containing protein CRADD (UniProtKB accession number O88843).

3.1. ProDom Domain Architecture Viewer

ProDom (http://prodom.prabi.fr/) is a protein domain database generated through automatic clustering of protein sequences (8) and is one of the InterPro member databases. Its automated methods enable ProDom to have high coverage of sequence space, although it also calls into question the validity of the domain boundaries. ProDom provides tools on its Web site for viewing domain architectures, including listing of all proteins sharing a common domain, as well as identifying all proteins with identical domain architectures (*see* **Fig. 5.1**).

3.2. SMART

The SMART HMM database (http://smart.embl-heidelberg.de) of domains provides a Web site for the analysis of domain architectures (9). Users can search for all proteins with identical domain architectures or can find the domain composition of a single protein. From the single protein view (**Fig. 5.2**), links are provided to display all proteins with the same domain composition (proteins containing at least one copy of each of domains of the query protein) or domain organization (proteins containing the same domains as the query protein and in the same order). When the
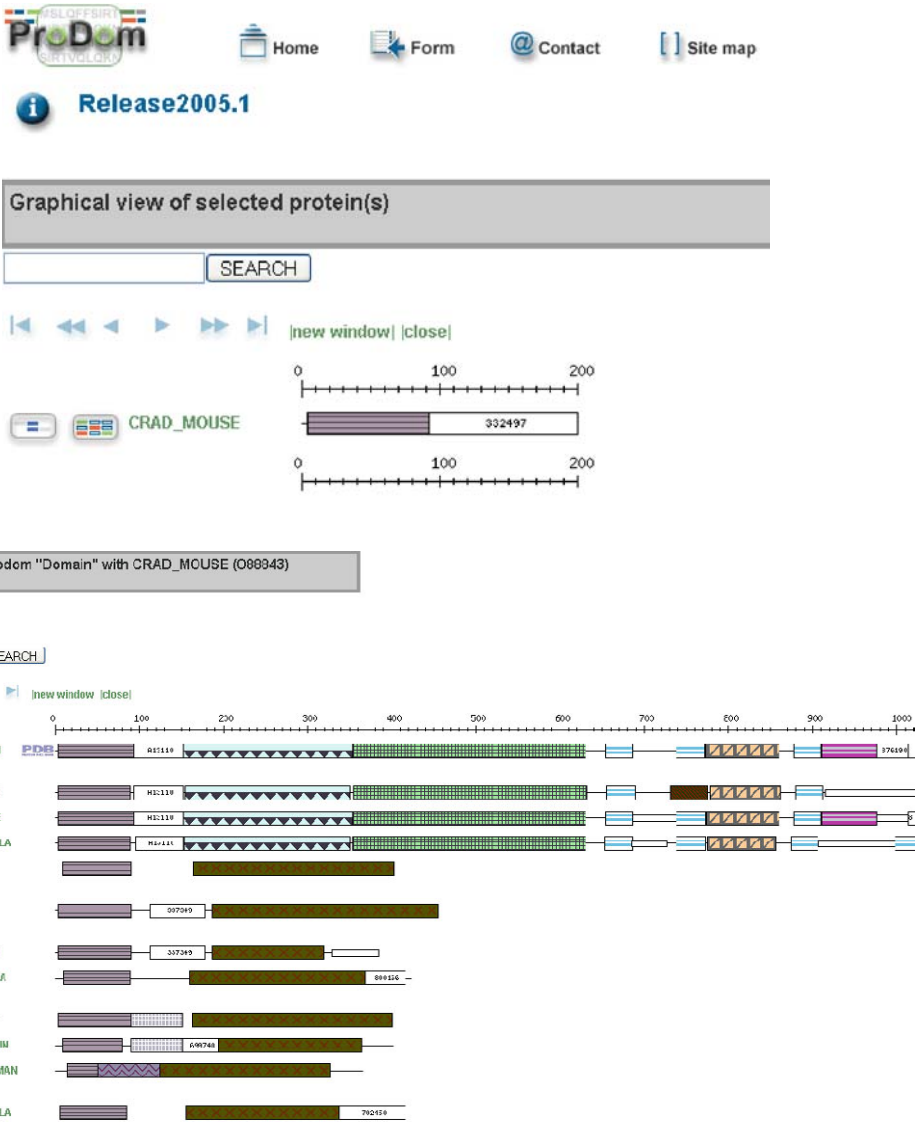
Fig. 5.1. The ProDom Web interface showing a single protein domain architecture (a) and a graphical view of proteins sharing a common domain with this protein (b).

links are followed, the interface provides a taxonomic tree showing how many proteins in the list are from the different taxonomic groups, and the user can select which taxonomic groups to display the domain architecture for. If the query domains are known, from the search page it is possible to enter multiple domains with "AND" or "NOT" to find proteins containing these sets of domains. Again, the search can be limited to specific taxonomic groups.

Domains within *Mus musculus* protein CRADD_MOUSE (O88843)

Death domain-containing protein CRADD
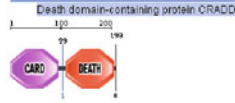
Mouse over domain / undefined region for more info; click on it to go to detailed annotation; right-click to save whole protein as PNG image

Transmembrane segments as predicted by the *TMHMM2* program ( ), coiled coil regions determined by the *Coils2* program ( ), segments of low compositional complexity determined by the *SEG* program ( ). Signal peptides determined by the *SignalP* program ( ). Intron positions are indicated with vertical lines showing the intron phase and exact position in AA.

Protein information	Domain architecture analysis
Display orthology and other data	Display all proteins with similar domain ORGANISATION or COMPOSITION. This domain architecture was probably invented with the emergence of Amniota.

Interaction network

CRADD_MOUSE shown as Cradd

Your selection: **14** Metazoa proteins.

Protein	UPI00001803A9 (source)
Description	PREDICTED: similar to Death domain-containing protein CRADD a death domain)
Species	*Rattus norvegicus*
Domain architecture invented in	Amniota
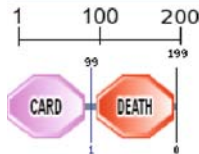
Protein	ENSLAFP00000014571 (source)
Description	no description
Species	*Loxodonta africana*
Domain architecture invented in	Amniota
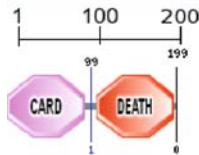
Protein	UPI0000EE009E (source)
Description	PREDICTED: similar to cell death adaptor molecule isoform 1
Species	*Ornithorhynchus anatinus*
Domain architecture invented in	Amniota
Representative of protein cluster	CLUST_UPI0000EE009E
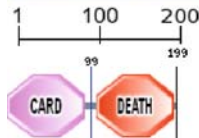
Fig. 5.2. The SMART Web interface view of a single protein (a) and the list of proteins from a specific taxonomic subset sharing the same domain composition (b).

3.3. CDART

CDART (18), provided within the Entrez suite of tools, uses the domains from CDD (see above) and RPS-BLAST (19) to enable users to query the protein database by domains. The query page is available from http://www.ncbi.nlm.nih.gov/Structure/lexington/lexington.cgi, where the user can input a query protein via its accession number or sequence and retrieve its domain architecture, as well as a list of proteins with similar architectures. As with many domain architecture tools, the domain combinations are represented as beads on a string, as this provides an easy way to visualize results. For proteins with similar architectures, these are ranked according to the number of domains in common with the query sequence, and a single representative example is shown where more than one protein shares the same architecture. The list of domains is provided below the results with checkboxes for the user to select a subset of results with specific domains only (**Fig. 5.3**). The results can also be restricted by taxonomy, by choosing the "Subset by Taxonomy" button at the bottom of the page. This takes you to a taxonomic tree with the number of proteins listed next to each organism and a checkbox to tick for inclusion of that taxonomic group. In addition to queries by protein sequences, it is also possible to query the Entrez Domains Database with a domain to find all proteins containing this domain (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=search&DB=cdd). The user can get back to CDART from the results by clicking on the "Proteins" link in the results page (18).

3.4. InterPro Domain Architectures

The InterPro database (http://www.ebi.ac.uk/interpro) provides a domain architecture view of its protein signatures (21). All matches of all the signatures in InterPro against UniProtKB are precomputed and available for viewing in different formats. In the Architectures view, nonoverlapping domain matches are
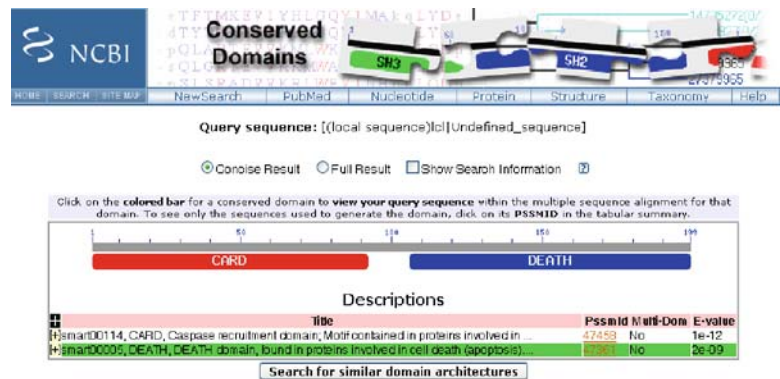
Fig. 5.3. CDART results for a protein search (a), and ranked list of proteins sharing common domains (b). The individual domains can be selected from the bottom of the page to generate a new list.

Fig. 5.3. (continued)

represented as oval shapes on the sequence (**Fig. 5.4**). All proteins
with the same domain architecture are collapsed into a single repre-
sentative example, with the number of proteins containing this
architecture listed and linked to. To search for a single domain

Fig. 5.4. InterPro single protein match view (a), and corresponding Architecture view (b),
showing a representative of each different domain composition.

Showing 1 - 20 of 68 Next »

Count Example Code	Architecture
111 A0GD39 IDA488	Death
48 A2BIB9 IDA2110,906,488	ANK ZU5 Death
42 A5D7R1 IDA1368x4,488	TNFR c6x4 Death
38 A1Y9B2 IDA1368x2,488	TNFR c6x2 Death
38 A2AF18 IDA488,719	Death Prot kinase core
34 A0ELU3 IDA488,157	Death TIR
32 A3KPR8 IDA13098,884x2,906,488	Ig I-set TSP1x2 ZU5 Death
31 A7Z090 IDA906,488	ZU5 Death
29 A3FJ60 IDA11539,2909,2110,488	RHD IPT TIG rcpt ANK Death
18 A3KPR7 IDA1368x3,488	TNFR c6x3 Death
15 A4QMR4 IDA2110,488	ANK Death
13 O88843 IDA1315,488	CARD Death
13 A0N085 IDA1875,488	DED Death

Fig. 5.4. (continued)

and find other proteins containing this domain, the user needs to know the name or accession number of this domain, find the corresponding InterPro entry, and then follow the "Architectures" link. Here, they will find a display of all architectures the domain is found within and they can follow the links to find the proteins.

3.5. PRODOC   The PROtein Domain Organization and Comparison (PRODOC) tools (1) enable the searching and analysis of protein domains in different genomes. PRODOC includes sequence data from the NCBI, certain genome databases (e.g., ENSEMBL, FlyBase, etc.), and UniProtKB (17), and protein domain data from Pfam-A (5). The tools facilitate searching of a specified sequence of domains in various genomes to identify proteins with conserved domain architectures or those containing the same domains without conserving domain order. To achieve this, the user should input a series of domains using the Pfam identifier (a list is available on the Web site). These are then searched against the PRODOC data set to identify proteins containing these domains in the same or a different order.

The PRODOC data set includes a classification of the domains into superfamilies using protein 3D structure information, so that the search can detect not only identical domains, but also those belonging to the same superfamily. This facility enables the identification of domains that may have conserved order, but where one or more has diverged beyond immediate recognition (1).

PRODOC also facilitates the identification of domain fusion events by searching for domains that appear in separate proteins in some genomes and in the same protein in others. Additional genome comparison tools allow the user to compare domain architectures between two genomes, providing an output listing pairs of proteins with at least one domain in common. PRODOC is available via the Web at http://hodgkin.mbu.iisc.ernet.in/ ~prodoc/ (1).

4. Discussion

The analysis of protein domains has the potential to provide information on the evolution of proteins and their functions. All those domains that have been characterized experimentally serve as a valuable source of data for the prediction of functions in uncharacterized proteins, and tools such as protein signatures and their corresponding search algorithms enable this. Through these tools, we are able to classify 80% of the UniProtKB proteins into families and domains, thus providing a large data set for the analysis of protein domain architectures. This data is provided in some form or another and in varying degrees of completeness by a variety of different Web interfaces, as described here. Some, like InterPro, include more databases in their data sets, but have more limited querying tools for domain composition analysis, while others, such as CDD and the individual databases of SMART and ProDom, have fewer domain descriptors, but more advanced tools for filtering, viewing, and ranking proteins sharing domains or domain compositions. The recommendation of which resources to use depends on the user's requirements, research question, and preference for output display.

Most of these resources have a common way of viewing the data, i.e., the beads on a string view, and enable the analysis of a single or small set of domains at a time. If users want to do complex domain architecture analysis across multiple unrelated domains simultaneously, they will be required to download the data sets with the protein sequence and domain boundary information and do the data mining using their own scripts. However, as many of these public databases move toward the provision of programmatic access to their data via Web services, this will enable researchers to do more complex querying and mining of these public data

without the need for large data downloads. This also ensures that the data being used is up to date. As long as the protein domain databases continue to keep up with the deluge of new sequence data, they will continue to provide valuable resources to facilitate new discoveries in protein function and evolution.

References

1. Krishnadev, O., Rekha, N., Pandit, S. B., Abhiman, S., Mohanty, S., Swapna, L. S., Gore, S., Srinivasan, N. (2005) PRODOC: a resource for the comparison of tethered protein domain architectures with in-built information on remotely related domain families. *Nucleic Acids Res* 33, W126–W129.

2. Berman, H., Henrick, K., Nakamura, H., Markley, J. L. (2007) The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. *Nucleic Acids Res* 35, D301–D303.

3. Gribskov, M., Luthy, R., Eisenberg, D. (1990) Profile analysis. *Methods Enzymol* 183, 146–159.

4. Krogh, A., Brown, M., Mian, I. S., Sjolander, K., Haussler, D. (1994) Hidden Markov models in computational biology. Applications to protein modeling. *J Mol Biol* 235(5), 1501–1531.

5. Finn, R. D., Mistry, J., Schuster-Bockler, B., Griffiths-Jones, S., Hollich, V., Lassmann, T., Moxon, S., Marshall, M., Khanna, A., Durbin, R., Eddy, S. R., Sonnhammer, E. L., Bateman, A. (2006) Pfam: clans, web tools and services. *Nucleic Acids Res* 34, D247–D251.

6. Attwood, T. K., Bradley, P., Flower, D. R., Gaulton, A., Maudling, N., Mitchell, A. L., Moulton, G., Nordle, A., Paine, K., Taylor, P., Uddin, A., Zygouri, C. (2003) PRINTS and its automatic supplement, prePRINTS. *Nucleic Acids Res* 31, 400–402.

7. Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., De Castro, E., Langendijk-Genevaux, P. S., Pagni, M., Sigrist, C. J. (2006) The PROSITE database. *Nucleic Acids Res* 34, D227–D230.

8. Bru, C., Courcelle, E., Carrere, S., Beausse, Y., Dalmar, S., Kahn, D. (2005) The Pro-Dom database of protein domain families: more emphasis on 3D. *Nucleic Acids Res* 33, D212–D215.

9. Letunic, I., Copley, R. R., Pils, B., Pinkert, S., Schultz, J., Bork, P. (2006) SMART 5: domains in the context of genomes and networks. *Nucleic Acids Res* 34, D257–D260.

10. Selengut, J. D., Haft, D. H., Davidsen, T., Ganapathy, A., Gwinn-Giglio, M., Nelson, W. C., Richter, A. R., White, O. (2007) TIGRFAMs and Genome Properties: tools for the assignment of molecular function and biological process in prokaryotic genomes. *Nucleic Acids Res* 35, D260–D264.

11. Wu, C. H., Nikolskaya, A., Huang, H., Yeh, L. S., Natale, D. A., Vinayaka, C. R., Hu, Z. Z., Mazumder, R., Kumar, S., Kourtesis, P., Ledley, R. S., Suzek, B. E., Arminski, L., Chen, Y., Zhang, J., Cardenas, J. L., Chung, S., Castro-Alvear, J., Dinkov, G., Barker, W. C. (2004) PIRSF: family classification system at the Protein Information Resource. *Nucleic Acids Res* 32, D112–D114.

12. Wilson, D., Madera, M., Vogel, C., Chothia, C., Gough, J. (2007) The SUPERFAMILY database in 2007: families and functions. *Nucleic Acids Res* 35, D308–D313.

13. Yeats, C., Maibaum, M., Marsden, R., Dibley, M., Lee, D., Addou, S., Orengo, C. A.. (2006) Gene3D: modelling protein structure, function and evolution. *Nucleic Acids Res* 34, D281–D284.

14. Mi, H., Guo, N., Kejariwal, A., Thomas, P. D. (2007) PANTHER version 6: protein sequence and function evolution data with expanded representation of biological pathways. *Nucleic Acids Res* 35, D247–D252.

15. Marchler-Bauer, A., Anderson, J. B., Cherukuri, P. F., DeWeese-Scott, C., Geer, L. Y., Gwadz, M., He, S., Hurwitz, D. I., Jackson, J. D., Ke, Z., Lanczycki, C. J., Liebert, C. A., Liu, C., Lu, F., Marchler, G. H., Mullokandov, M., Shoemaker, B. A., Simonyan, V., Song, J. S., Thiessen, P. A., Yamashita, R. A., Yin, J. J., Zhang, D., Bryant, S. H. (2005) CDD: a Conserved Domain Database for protein classification. *Nucleic Acids Res* 33, D192–D196.

16. Portugaly, E., Linial, N., Linial, M. (2006) EVEREST: a collection of evolutionary conserved protein domains. *Nucleic Acids Res* 34, D1–D6.

17. UniProt Consortium (2007) The Universal Protein Resource (UniProt). *Nucleic Acids Res* 35, D193–D197.

18. Geer, L. Y., Domrachev, M., Lipman, D. J., Bryant, S. H. (2002) CDART: protein homology by domain architecture. *Genome Res* 12(10), 1619–1623.

19. Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25(17), 3389–3402.

20. Tatusov, R. L., Natale, D. A., Garkavtsev, I. V., Tatusova, T. A., Shankavaram, U. T., Rao, B. S., Kiryutin, B., Galperin, M. Y., Fedorova, N. D., Koonin, E. V. (2001) The COG database: new developments in phylogenetic classification of proteins from complete genomes. *Nucleic Acids Res* 29(1), 22–28.

21. Mulder, N. J., Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Buillard, V., Cerutti, L., Copley, R., Courcelle, E., Das, U., Daugherty, L., Dibley, M., Finn, R., Fleischmann, W., Gough, J., Haft, D., Hulo, N., Hunter, S., Kahn, D., Kanapin, A., Kejariwal, A., Labarga, A., Langendijk-Genevaux, P. S., Lonsdale, D., Lopez, R., Letunic, I., Madera, M., Maslen, J., McAnulla, C., McDowall, J., Mistry, J., Mitchell, A., Nikolskaya, A. N., Orchard, S., Orengo, C., Petryszak, R., Selengut, J. D., Sigrist, C. J., Thomas, P. D., Valentin, F., Wilson, D., Wu, C. H., Yeats, C. (2007) New developments in the InterPro database. *Nucleic Acids Res* 35, D224–D228.

22. Gene Ontology Consortium (2006) The Gene Ontology (GO) project in 2006. *Nucleic Acids Res* 34, D322–D326.

23. 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.

24. Andreeva, A., Howorth, D., Brenner, S. E., Hubbard, T. J., Chothia, C., Murzin, A. G. (2004) SCOP database in 2004: refinements integrate structure and sequence family data. *Nucleic Acids Res* 32, D226–D229.

25. Pearl, F., Todd, A., Sillitoe, I., Dibley, M., Redfern, O., Lewis, T., Bennett, C., Marsden, R., Grant, A., Lee, D., Akpor, A., Maibaum, M., Harrison, A., Dallman, T., Reeves, G., Diboun, I., Addou, S., Lise, S., Johnston, C., Sillero, A., Thornton, J., Orengo, C. (2005) The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. *Nucleic Acids Res* 33, D247–D251.

Chapter 6

Thermodynamic Database for Proteins: Features and Applications

M. Michael Gromiha and Akinori Sarai

Abstract

We have developed a thermodynamic database for proteins and mutants, ProTherm, which is a collection of a large number of thermodynamic data on protein stability along with the sequence and structure information, experimental methods and conditions, and literature information. This is a valuable resource for understanding/predicting the stability of proteins, and it can be accessible at //gibk26.bse.kyutech.ac.jp/jouhou/Protherm/protherm.html. ProTherm has several features including various search, display, and sorting options and visualization tools. We have analyzed the data in ProTherm to examine the relationship among thermodynamics, structure, and function of proteins. We describe the progress on the development of methods for understanding/predicting protein stability, such as (i) relationship between the stability of protein mutants and amino acid properties, (ii) average assignment method, (iii) empirical energy functions, (iv) torsion, distance, and contact potentials, and (v) machine learning techniques. The list of online resources for predicting protein stability has also been provided.

Key words: thermodynamics, database, protein stability, prediction.

1. Introduction

Protein structures are stabilized with various noncovalent interactions such as hydrophobic, electrostatic, van der Waals, and hydrogen-bonded interactions (1–4). The importance of such interactions for protein stability has been revealed by site-directed mutagenesis experiments (5–8). We have collected the stability data reported in the literature and developed the thermodynamic database for proteins and mutants, ProTherm, which has more than 20,000 experimental data on protein stability along with sequence and structure information of the protein, experimental conditions, and literature information (9, 10).

Currently, ProTherm is serving as a unique source for understanding and predicting protein stability. It has been used for developing several methods, such as relationship between amino acid properties and protein stability (11–14), empirical energy functions (15, 16), stability scale (17), contact potentials (18), neural networks (19), support vector machines (20, 21), relative importance of secondary structure and solvent accessibility (22), average assignment (23), Bayesian networks (24), distance and torsion potentials (25), decision trees (26), physical force field with atomic modeling (27), etc., for understanding and predicting the stability of proteins upon mutations.

This review is broadly divided into two parts: the first part focuses on the development of thermodynamic database for proteins and mutants and the second part is devoted to analysis and prediction of protein stability upon mutation.

2. Thermodynamic Database for Proteins and Mutants, ProTherm

2.1. Contents of ProTherm

ProTherm is a large collection of thermodynamic data on protein stability, which has the following information (9, 10):

Sequence and Structure Information: Name, source, length, and molecular weight of the protein, codes for Protein Information Resource (PIR) (28), Swiss-Prot (29) and Protein Data Bank (PDB) (30), enzyme commission number (31), mutation details (wild and mutant residue names, residue number and location of the mutant based on secondary structure and solvent accessibility), and number of transition states. The secondary structure of each mutant was assigned using the program, DSSP (32). The solvent accessible surface area (ASA) of all the atoms and the residues were computed using the program ASC (33).

Thermodynamic Data Obtained from Denaturant Denaturation Experiments: Unfolding Gibbs free energy change (ΔG^{H_2O}), difference in unfolding Gibbs free energy change for the mutants [$\Delta\Delta G^{H_2O} = \Delta G^{H_2O}(\text{mutant}) - \Delta G^{H_2O}(\text{wild type})$], midpoint of denaturant concentration (C_m), slope of denaturation curve (m), and reversibility of denaturation.

Thermodynamic Data Obtained from Thermal Denaturation Experiments: Unfolding Gibbs free energy change (ΔG), difference in unfolding Gibbs free energy change for the mutants ($\Delta\Delta G$), transition temperature (T_m), transition temperature change for the mutant (ΔT_m), calorimetric and van't Hoff enthalpy change (ΔH), heat capacity change (ΔC_p), and reversibility of denaturation.

Experimental Methods and Conditions: pH, temperature (T), buffer and ions, and their concentrations, protein concentration, measurement and method.

Functional Information: Enzyme activity, binding constants, etc.

Literature Information: Keywords, reference, authors, and remarks.

2.2. Search and Display Options in ProTherm

We have implemented several search and display options for the convenience to the users.

(i) Retrieving data for a specific protein and source. The search options are also available with PDB code.

(ii) Specifying the type of mutation as single, double, multiple, or wild type, and any mutant/mutated residue.

(iii) Specifying secondary structures having mutations in helix (H), strand (S), turn (T), and coil (C) regions.

(iv) Searching data based on solvent accessibility/solvent ASA (in % or $\mathring{A}^2$) of relevant residue. The mutations are classified into buried (ASA < 20%), partially buried (20% ≤ ASA ≤ 50%) and exposed (ASA > 50%).

(v) Extracting data for a particular measurement (CD, DSC, Fl, etc.) and a specific method (Thermal, GdnHCl, Urea, etc.).

(vi) Limiting data for a particular range of T, T_m, ΔT_m, ΔG, $\Delta\Delta G$, ΔG^{H_2O}, $\Delta\Delta G^{H_2O}$, ΔH, ΔC_p, and pH.

(vii) Obtaining the data reported with two- or three-state transitions and reversible/irreversible denaturation.

(viii) Extracting data with authors, publication year, and key words.

(ix) Specifying output format by selecting various output items and sorting with publication year, wild type residue, mutant residue, residue number, secondary structure, solvent accessibility, pH, T, T_m, ΔT_m, ΔG, $\Delta\Delta G$, ΔG^{H_2O}, $\Delta\Delta G^{H_2O}$, ΔH, ΔC_p, and pH.

Detailed tutorials describing the usage of ProTherm are available at the home page. As an example, the necessary items to be filled or selected to search data for mutations in buried regions by denaturant denaturation and CD measurement at temperatures between 15 and 25°C are shown in **Fig. 6.1a**. In **Fig. 6.1b**, we show the items to be selected for the output and sorting options. In the sorting procedure, the first item has the topmost priority. In this figure, entry, protein, PDB wild, mutation, secondary structure, ASA, $\Delta\Delta G^{H_2O}$, T, pH, and reference are selected for the output. The selected outputs are sorted with temperature as the first priority and residue number as the second priority. The final results obtained from the search conditions (**Fig. 6.1a**) and sorting options of necessary items (**Fig. 6.1b**) are shown in **Fig. 6.1c**.
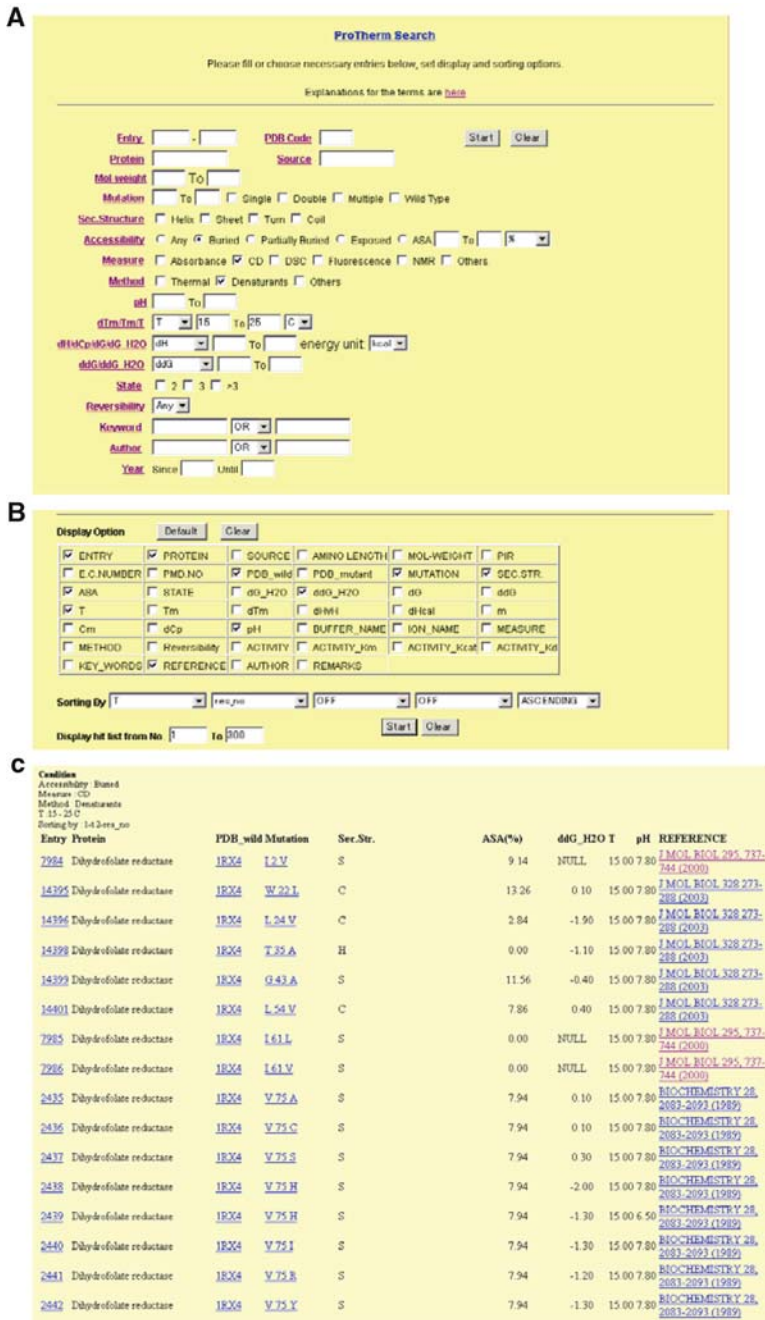
Fig. 6.1. An example of searching conditions, display and sorting options, and results of ProTherm. (a) Main menu for the search options of ProTherm. In this example, items, buried (accessibility), CD (measure), denaturant (method) are selected from the menu and $T(\Delta T_m/T_m/T)$ is specified by filling the boxes for the values from 15 to 25°C. (b) Display and sorting options of ProTherm. In this example, entry, protein, PDB wild, mutation, secondary structure, ASA, $\Delta\Delta G^{H_2O}$, T, pH, and reference are selected for the output. T and residue number are chosen for sorting the results in the order of priority. (c) Part of the results obtained from ProTherm.

2.3. ProTherm Statistics

We update ProTherm quite frequently and at present it has about 22,000 data, which is more than six-fold compared with the first release. The data are obtained from 603 different proteins with 10,948 single and 1,549 double mutations. In terms of secondary structures, 4,765 mutations are in helical segments, 3,682 in strand, 1,828 in turn, and 2,838 in coil regions. According to solvent accessibility, 5,583 mutations are at buried, 3,738 mutations are at partially buried, and 3,357 are at exposed regions. The frequency of mutations in different proteins is presented in **Table 6.1**. Most of the mutational experiments have been carried out with hydrophobic substitutions (replacement of one hydrophobic residue with another, e.g., Val to Ala) and the mutations from any residue into Ala. Further, the aromatic mutations (Tyr to Phe) and few polar mutations (Thr to Ser, Asp to Asn, Glu to Gln, etc.) are dominant in ProTherm.

3. Factors Influencing the Stability of Proteins and Mutants

We have systematically analyzed the influence of specific amino acid properties on the stability of proteins upon amino acid substitutions. We considered a set of 49 diverse amino acid properties, which have been used in protein folding and stability studies (11, 34, 35). The amino acid properties were normalized between 0 and 1 using the expression, $P_{norm}(i) = [P(i) - P_{min}]/[P_{max} - P_{min}]$, where $P(i)$, $P_{norm}(i)$ are, respectively, the original and normalized values of amino acid i for a particular property, and P_{min} and P_{max} are, respectively, the minimum and maximum values. The original and normalized values for the selected 49 physico-chemical, energetic, and conformational properties of the 20 amino acid residues and their brief explanations are available at //www.cbrc.jp/~gromiha/fold_rate/property.html.

3.1. Relationship Between Amino Acid Properties and Protein Stability upon Mutations

We computed the mutation-induced changes in property values, $\Delta P(i)$, using the Eq. (11): $\Delta P(i) = P_{mut}(i) - P_{wild}(i)$, where $P_{mut}(i)$ and $P_{wild}(i)$ are, respectively, the normalized property value of the ith mutant and wild-type residue; i varies from 1 to N and N is the total number of mutants. The computed differences in property values (ΔP) were related to the changes in experimental stability values (ΔT_m, $\Delta\Delta G$ or $\Delta\Delta G^{H_2O}$) using correlation coefficient, $r = [N\Sigma XY - (\Sigma X\Sigma Y)]/\{[N\Sigma X^2 - (\Sigma X)^2] \ [N\Sigma Y^2 - (\Sigma Y)^2]\}^{1/2}$, where N, X and Y are, respectively, the number of data, property and experimental stability values, respectively.

In buried mutations, the properties reflecting hydrophobicity showed a strong correlation with stability indicating the direct relationship between hydrophobicity and stability (11, 12). In

Table 6.1
Frequency of amino acid residues at mutant sites

To

From	Gly	Ala	Val	Leu	Ile	Cys	Met	Phe	Tyr	Trp	Pro	Ser	Thr	Asn	Gln	Asp	Glu	Lys	Arg	His
Gly	–	230	57	5	0	12	0	10	2	6	14	55	2	5	26	28	24	6	19	17
Ala	113	–	131	61	20	29	24	21	5	8	85	74	44	10	18	14	13	41	5	11
Val	73	474	–	258	288	47	64	99	51	24	6	46	89	28	0	26	14	13	23	12
Leu	34	350	114	–	88	55	40	86	2	4	18	16	14	4	4	13	32	12	23	5
Ile	41	244	430	236	–	24	62	93	12	8	7	22	58	3	1	9	20	10	9	12
Cys	4	137	17	26	1	–	2	2	2	1	1	105	86	0	0	0	4	0	0	0
Met	18	65	50	124	45	1	–	17	2	0	0	0	6	1	0	6	6	18	21	0
Phe	5	149	24	117	18	3	10	–	95	63	0	24	5	21	2	4	4	7	0	7
Tyr	31	99	4	25	2	20	1	185	–	72	4	11	4	6	8	17	1	4	2	9
Trp	0	31	1	18	0	3	1	133	69	–	0	2	0	1	5	3	3	0	3	13
Pro	66	187	7	17	2	9	0	4	2	4	–	68	10	4	4	6	5	5	8	3
Ser	28	212	19	18	10	29	2	13	6	2	8	–	28	20	4	66	6	20	37	25
Thr	37	204	126	32	103	38	11	24	28	2	3	118	–	20	27	18	92	7	40	21
Asn	34	155	6	6	51	6	12	4	0	1	1	33	10	–	5	97	22	13	11	41
Gln	42	74	3	23	5	12	3	3	4	0	9	5	1	14	–	10	34	33	14	6
Asp	58	199	8	9	9	33	4	14	7	8	12	20	10	158	11	–	64	67	16	54
Glu	50	299	67	48	6	10	16	25	29	17	8	29	12	13	103	32	–	134	30	13
Lys	85	227	23	12	35	18	46	46	18	27	23	13	18	22	41	13	99	–	79	37
Arg	40	161	14	13	0	19	18	3	0	1	1	8	2	0	20	1	71	26	–	67
His	38	112	10	47	0	7	0	10	69	2	24	6	21	47	51	21	17	6	12	–

partially buried and exposed mutations, the whole set of data did not show significant correlation with any of the properties. However, the classification of data based on the secondary structures improved the correlation between amino acid properties and protein stability significantly (11, 13, 14). In partially buried helical mutations, the stability is attributed mainly with thermodynamic properties. The β-strand tendency (P_β) is the major factor for the stability of mutations in β-strand segments. In exposed helical mutations, the strongest correlation was observed for ΔASA (solvent accessible surface area change for unfolding). In turn segments, P_t (turn tendency), $P_{\Phi-\Psi}$ (backbone dihedral probability), B_l (bulkiness), and M_w (molecular weight) showed significant correlation with protein stability.

3.2. Influence of Neighboring and Surrounding Residues to Protein Mutant Stability

We have also analyzed the influence of neighboring residues of the mutant residue in the amino acid sequence and surrounding residues that are close in the protein 3D structure. The local sequence effect has been included using the Eq. (11):

$$P_{\text{seq}}(i) = \left[\sum_{j=i-k}^{j=i+k} P_j(i) \right] - P_{\text{mut}}(i),$$

where $P_{\text{mut}}(i)$ is the property value of the ith mutant residue and $\Sigma P_j(i)$ is the total property value of a segment of $(2k+1)$ residues, ranging from $i-k$ to $i+k$ about the ith wild-type residue. We used windows of 3 and 9 ($k = 1, 4$) residues to include the influence of short- and medium-range interactions (36).

The structural information, $P_{\text{str}}(i)$, was included using Eq. (11):

$$P_{\text{str}}(i) = \left[\sum_j n_{ij} P_j \right] - P_{\text{mut}}(i),$$

where n_{ij} is the total number of type j residues surrounding the ith residue of the protein within the sphere of radius 8 Å (36), and P_j is the property value of the type j residue.

In buried mutations, the inclusion of neighboring and surrounding residues did not show any significant improvement in the correlation between amino acid properties and protein stability (11, 12). This might be due to the hydrophobic environment of the mutant site, which is surrounded mainly by hydrophobic residues and nonspecific interactions dominate in the interior of proteins. In partially buried and exposed mutations, the inclusion of neighboring and surrounding residues remarkably improved the correlation in all the subgroups of mutations. This result indicates that the information from nearby polar/charged residues and/or the residues those are close in space are important for the stability of partially buried and exposed mutations. Detailed

analysis showed that more than 50% of the neighboring/surrounding residues are polar and charged, and hence the stability of partially buried/exposed mutations is influenced by hydrophobic, hydrogen bonding, and other polar interactions (11, 13, 14).

3.3. Stabilizing Residues in Protein Structures

We have proposed a consensus approach for detecting the stabilizing residues in protein structures based on long-range interactions, hydrophobicity, and conservation of residues (37).

The surrounding hydrophobicity (H_p) of a given residue is defined as the sum of experimental hydrophobic indices (38, 39) of various residues, which appear within the sphere of radius 8 Å radius limit from it (40):

$$H_p(i) = \sum_{j=1}^{20} n_{ij} h_j,$$

where n_{ij} is the total number of surrounding residues of type j around ith residue of the protein and h_j is the experimental hydrophobic index of residue type j in kcal/mol (38, 39).

The long-range order (LRO) for a protein has been computed from the knowledge of long-range contacts (contacts between two residues that are close in space and far in the sequence) in protein structure (41). LRO for a specific residue is calculated using the number of long-range contacts for that residue:

$$\text{LRO}_i = \sum_{j=1}^{N} n_{ij}/N; \quad n_{ij} = 1 \text{ if } |i-j| > 12; \quad n_{ij} = 0 \text{ otherwise,}$$

where i and j are two residues in which the C_α distance between them is ≤ 8 Å and N is the total number of residues in a protein.

Stabilization centers (SCs) in a protein are clusters of residues involved in long-range interactions (42). Two residues are considered to be in long-range interaction if they are separated by at least ten residues in the sequence and at least one of their heavy-atom contact distance is less than the sum of their van der Waals radii of the two atoms, plus 1 Å. Two residues are part of SCs if (i) they are involved in long-range interactions and (ii) two supporting residues can be selected from both of their flanking tetrapeptides, which together with the central residues form at least seven out of the possible nine contacts.

The conservation of residues in each protein has been computed with the aid of the Consurf server (43) (//consurf.tau.ac.il/). This server compares the sequence of a PDB chain with the proteins deposited in Swiss-Prot (29) and finds the ones that are homologous to the PDB sequence.

Based on these four parameters, we have proposed the following conditions to predict the stabilizing residues: (i) $H_p \geq 20$ kcal/mol; (ii) LRO ≥ 0.02; (iii) SC ≥ 1, and (iv) Conservation score ≥ 6. We have compared the stabilizing residues identified by our
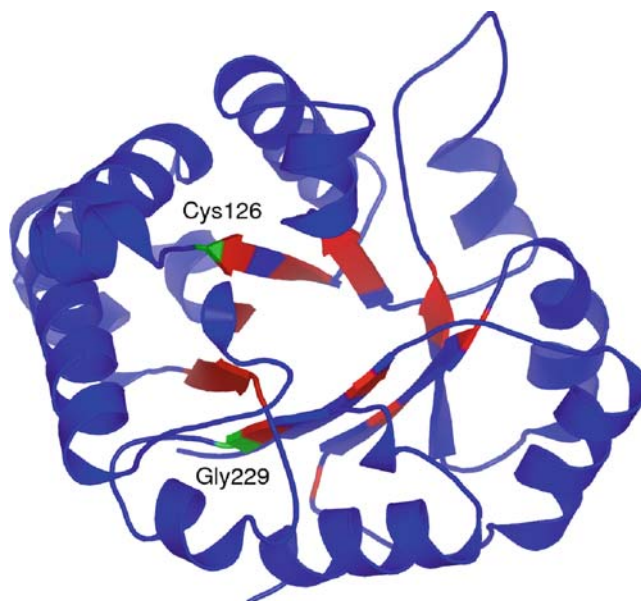
Fig. 6.2. Stabilizing residues in a typical TIM barrel protein, 1 btm. The α-helices are shown as spiral ribbons and the β-strands are drawn as arrows from the amino end to the carboxyl end of the β-strand. The stabilizing residues Gly229 and Cys126 identified by our method and observed by thermodynamic and kinetic experiments are indicated. The picture was generated using PyMOL program (50).

approach with experimental data deposited in ProTherm (10). In triose phosphate isomerase, we have identified the residues Gly229 and Cys126 as stabilizing ones and are shown in **Fig. 6.2**. Interestingly, these residues play important roles to protein stability as observed from thermodynamic and kinetic experiments (10, 44, 45). Further, we found that the replacement of residues in the stabilizing segments (Phe22, Tyr175, Leu209, and Ile232) of tryptophan synthase destabilized the protein with the free-energy change of 1–5 kcal/mol, which showed a good agreement with experiments (10). We set up a Web server for identifying the stabilizing residues in protein structures and it is freely available at //sride.enzim.hu/ (46).

4. Prediction of Protein Mutant Stability

Several methods have been proposed for predicting the stability of proteins upon single amino acid substitutions. The online severs available for predicting protein mutant stability are listed in **Table 6.2**. The performance of these methods has been generally tested with such measures as accuracy, correlation, and

Table 6.2
Online resources for protein stability

Thermodynamic database for proteins and mutants	
ProTherm	//gibk26.bse.kyutech.ac.jp/jouhou/protherm/ protherm.html
Stabilizing residues in protein structures	
SRide	//sride.enzim.hu
Prediction of protein-mutant stability	
FOLD-X	//fold-x.embl-heidelberg.de
CUPSAT	//cupsat.tu-bs.de/
I-Mutant2.0	//gpcr.biocomp.unibo.it/cgi/predictors/ I-Mutant2.0/I-Mutant2.0.cgi.
MUpro	//www.igb.uci.edu/servers/servers.html
iPTREE-STAB	//bioinformatics.myweb.hinet.net/iptree.htm
Eris	//eris.dokhlab.org

mean-absolute error (MAE). The accuracy of distinguishing the stability of mutants (stabilizing/destabilizing) has been determined by using the following expression: Accuracy (%) = $p \times 100.0/N$, where p is the total number of correctly discriminated residues and N is the total number of data used for discrimination. The MAE is defined as the absolute difference between predicted and experimental stability values:

$$\text{MAE} = \frac{1}{N} \sum_i |X_i - Y_i|,$$

where X_i and Y_i are the experimental and predicted stability values, respectively, and i varies from 1 to N, N being the total number of mutants.

4.1. Average Assignment Method

Saraboji et al. (23) developed a method based on average assignment for predicting the stability of protein mutants. In this method, the data set has been classified into 380 possible amino acid substitutions (20 amino acids and 19 substitutions for each of the 20 amino acids). Considering the mutant Ala to Cys, we calculated the average stability change of all Ala→Cys mutants and assigned the same for this specific mutant. This calculation was repeated for all the 380 pairs and the stability change values were assigned. The assigned ΔT_m ($\Delta\Delta G$ and $\Delta\Delta G^{H_2O}$) values are compared with experimental data to obtain the accuracy, correlation, and MAE.

We observed that our method could distinguish the stabilizing and destabilizing mutants to an accuracy of 70–80% at different measures of stability (ΔT_m, $\Delta\Delta G$, or $\Delta\Delta G^{H_2O}$). Further, the classification of mutants based on helix, strand, and coil distinguished the stabilizing/destabilizing mutants at an average accuracy of 82% and the correlation was 0.56; information about the location of residues at the interior, partially buried, or surface of a protein correctly identified the stabilizing/destabilizing residues at an average accuracy of 81% and the correlation was 0.59. The nine subclassifications based on three secondary structures and solvent accessibilities improved the accuracy of assigning stabilizing/destabilizing mutants to an accuracy of 84–89% for the three data sets.

4.2. Empirical Energy Functions

Guerois et al. (15) developed a computer algorithm, FOLDEF, for estimating the important interactions contributing to the stability of proteins. The free energy of unfolding (ΔG) of a target protein is calculated using the equation

$$\Delta G = W_{vdw}\Delta G_{vdw} + W_{solvH}\Delta G_{solvH} + W_{solvP}\Delta G_{solvP} + \Delta G_{wb}$$
$$+ \Delta G_{hbond} + \Delta G_{el} + W_{mc}T\Delta S_{mc} + W_{sc}T\Delta S_{sc},$$

where ΔG_{vdw} is the sum of the van der Waals contributions of all atoms. ΔG_{solvH} and ΔG_{solvP} are the difference in solvation energy for apolar and polar groups, respectively, when going from the unfolded to the folded state. ΔG_{hbond} is the free energy difference for the formation of an intramolecular hydrogen bond with respect to intermolecular hydrogen-bond formation (with solvent). ΔG_{wb} is the extra stabilizing free energy provided by a water molecule forming more than one hydrogen bond to the protein (water bridges) that cannot be taken into account with nonexplicit solvent approximations. ΔG_{el} is the electrostatic contribution of charged groups interactions. ΔS_{mc} is the entropy cost for fixing the backbone in the folded state. This term is dependent on the intrinsic tendency of a particular amino acid to adopt certain dihedral angles. ΔS_{sc} is the entropic cost of fixing a side chain in a particular conformation. The terms W_{vdw}, W_{solvH}, W_{solvP}, W_{mc}, and W_{sc} correspond to the weighting factors applied to the raw energy terms and these weights were obtained from an initial fitting procedure over a database consisting of 339 single-point mutants.

The predictive power of the method was tested using a dataset of 667 mutants. They reported that FOLDEF could predict the stability of protein mutants within the average error of 0.81 kcal/mol for the coverage of 95% of the mutants. The correlation between the experimental and predicted protein free-energy changes was 0.83. A Web server has been developed for predicting the stability of protein mutants and it is available at //fold-x.embl-heidelberg.de.

Bordner and Abagyan (16) proposed an empirical energy function, which included terms representing the energy contributions of the folded and denatured proteins for predicting the stability of protein mutants. They trained the method using a half of the diverse set of 1,816 experimental stability values for single-point mutations in 81 different proteins. They reported that after removing 22 (∼2%) outliers, this method could predict the stability of protein mutants within the standard deviation of 1.08 kcal/mol with a correlation coefficient of 0.82. Further, the prediction method was tested on the remaining half of the experimental data, which could predict the protein mutant stability within the error of 1.10 kcal/mol.

4.3. Torsion, Distance and Contact Potentials

Gilis and Rooman (47, 48) developed torsion and distance potentials for predicting the stability of protein mutants. The torsion potential is mainly based on the neighboring residues in a sequence and hence at the protein surface, the local interactions along the chain are more dominant than hydrophobic interactions. The distance potentials are dominated by hydrophobic interactions, which represent best the main interactions stabilizing the protein core. They have analyzed a set of 106 surface mutations and reported that the correlation coefficient between experimental and computed free-energy change using backbone torsion potentials is 0.67 for all the mutations and it rose up to 0.87 for a subset of 96 mutations (47). Further, the buried and partially buried mutations have been systematically analyzed with distance potentials (48). They reported that for a set of completely buried mutations, the combination of distance potential and torsion potential weighted by a factor of 0.4 yielded the correlation coefficient of 0.80 between the computed and measured changes in folding free energy. For mutations of partially buried residues, the best potential is a combination of torsion potential and a distance potential weighted by a factor of 0.7 and the correlation coefficient is 0.82.

Khatun et al. (18) developed a methodology to determine the contact potentials in proteins, which defines the effective free energy of a protein conformation by a set of amino acid contacts formed in this conformation, from experimental measurements of changes in thermodynamic stability ($\Delta\Delta G$) of proteins upon mutations. They obtained a correlation of 0.66 and 0.46, respectively, for training and split sample validation in a data set of 1,356 mutations. They suggested that the use of an atomistic form of potentials may improve the prediction accuracy of protein mutant stability.

Parthiban et al. (25) analyzed protein stability upon point mutations using distance-dependant pair potential representing mainly through-space interactions and torsion angle potential representing mainly neighboring effects. They have developed the potentials at various ranges of solvent accessibility and at

different secondary structures. This method was trained and tested with 1,538 mutations and contain 101 proteins that share a wide range of sequence identity, which showed the maximum correlation of 0.87 with a standard error of 0.71 kcal/mol between predicted and measured $\Delta\Delta G$ values and a prediction accuracy of 85.3% for discriminating the stabilizing and destabilizing protein mutants. For $\Delta\Delta G^{H_2O}$, they obtained a correlation of 0.78 (standard error 0.96 kcal/mol) with a prediction efficiency of 84.65%. A Web server, CUPSAT, has been developed for predicting the stability of protein mutants and it is available at //cupsat. tu-bs.de/ (49).

4.4. Machine Learning Techniques

Capriotti et al. (19) proposed a neural network-based method to predict if a given mutation increases or decreases the protein thermodynamic stability with respect to the native structure. Using a data set of 1,615 mutations, this method correctly classified >80% of the mutations in the database. Further, when this method was coupled with energy-based methods, the joint prediction accuracy increased up to 90%, suggesting that it can be used to increase the performance of pre-existing methods, and generally protein design strategies. They have also developed a method based on support vector machines for predicting the stability of protein mutants (20). In this method, the stability of protein mutants has been correctly assigned to an accuracy of 80% and the correlation between experimental and computed stabilities is 0.71. A Web server, I-Mutant2.0, has been developed for predicting protein mutant stability and is available at //gpcr.bio comp.unibo.it/cgi/ predictors/I-Mutant2.0/I-Mutant2.0.cgi/.

Cheng et al. (21) used support vector machines to predict protein stability changes for single amino acid mutations from both sequence and structural information. This method could discriminate the stabilizing and destabilizing protein mutants with an accuracy of 84%. They developed a Web server for predicting protein stability changes upon mutations and it is available at //www.igb.uci.edu/servers/servers. html.

Huang et al. (26) developed a method based on interpretable decision tree coupled with adaptive boosting algorithm and a classification and regression tool for predicting protein stability upon amino acid substitutions. This method could correctly discriminate the stabilizing and destabilizing protein mutants at an accuracy of 82% for a data set of 1,859 single mutants. Further, a correlation of 0.70 was obtained between the predicted and experimental stabilities. A Web server, iPTREE-STAB, has been set up for predicting the stability of proteins and it is available at //bioinformatics.myweb.hinet. net/iptree.htm (26).

5. Conclusions

We have developed a thermodynamic database for proteins and mutants, which has several thermodynamic data along with sequence and structure information, experimental methods and conditions, and literature information. The analysis on protein mutant stability revealed that the stability of buried mutations is dominated by hydrophobic interactions whereas the partially buried and exposed mutations are influenced with hydrophobic, hydrogen bonds, and other polar interactions. The information about hydrophobicity, long-range interactions, and conservation of amino acid residues has been used to identify the stabilizing residues in protein structures. The classification of mutants based on secondary structures and solvent accessibility could predict the stability of protein mutants with high accuracy. Different methods have been proposed for predicting protein stability upon amino acid substitution using structural information, mutated and mutant residues, and from amino acid sequence. Further, Web servers have been set up for discriminating the stabilizing and destabilizing mutants as well as predicting protein mutant stability, which can be used for discriminating/predicting the stability of new mutants.

Acknowledgments

References

1. Dill, K. A. (1990) Dominant forces in protein folding. *Biochemistry* 29, 7133–7155.

2. Rose, G. D., Wolfenden, R. (1993) Hydrogen bonding, hydrophobicity, packing, and protein folding. *Annu Rev Biophys Biomol Str* 22, 381–415.

3. Ponnuswamy, P. K., Gromiha, M. M. (1994) On the conformational stability of folded proteins. *J Theor Biol* 166, 63–74.

4. Pace, C. N., Shirely, B. A., McNutt, M., Gajiwala, K. (1996) Forces contributing to the conformational stability of proteins. *FASEB J* 10, 75–83.

5. Yutani, K., Ogasahara, K., Tsujita, T., Sugino, Y. (1987) Dependence of conformational stability on hydrophobicity of the amino acid residue in a series of variant proteins substituted at a unique position of tryptophan synthase alpha subunit. *Proc Natl Acad Sci USA* 84, 4441–4444.

6. Shortle, D., Stites, W. E., Meeker, A. K. (1990) Contributions of the large hydrophobic amino acids to the stability of staphylococcal nuclease. *Biochemistry* 29, 8033–8041.

7. Matthews, B. W. (1995) Studies on protein stability with T4 lysozyme. *Adv Protein Chem* 46, 249–278.

8. Itzhaki, L. S., Otzen, D. E., Fersht, A. R. (1995) The structure of the transition state for folding of chymotrypsin inhibitor 2 analysed by protein engineering methods: evidence for a nucleation-condensation mechanism for protein folding. *J Mol Biol* 254, 260–288.

9. Gromiha, M. M., An, J., Kono, H., Oobatake, M., Uedaira, H., Sarai, A. (1999) ProTherm: thermodynamic database for proteins and mutants. *Nucleic Acids Res.* 27, 286–288.

10. Bava, K. A., Gromiha, M. M., Uedaira, H., Kitajima, K., Sarai, A. (2004) ProTherm, version 4.0: Thermodynamic Database for Proteins and Mutants. *Nucleic Acids Res* 32, D120–D121, Database issue.

11. Gromiha, M. M., Oobatake, M., Kono, H., Uedaira, H., Sarai, A. (1999) Role of structural and sequence information in the prediction of protein stability changes: comparison between buried and partially buried mutations. *Protein Eng* 12, 549–555.

12. Gromiha, M. M., Oobatake, M., Kono, H., Uedaira, H., Sarai, A. (1999) Relationship between amino acid properties and protein stability: buried mutations. *J Protein Chem* 18, 565–578.

13. Gromiha, M. M., Oobatake, M., Kono, H., Uedaira, H., Sarai, A. (2000) Importance of surrounding residues for predicting protein stability of partially buried mutations. *J Biomol Str Dyn* 18, 281–295.

14. Gromiha, M. M., Oobatake, M., Kono, H., Uedaira, H., Sarai, A. (2002) Importance of mutant position in Ramachandran plot for predicting protein stability of surface mutations. *Biopolymers* 64, 210–220.

15. Guerois, R., Nielsen, J. E., Serrano, L. (2002) Predicting changes in the stability of proteins and protein complexes: a study of more than 1000 mutations. *J Mol Biol* 320, 369–387.

16. Bordner, A. J., Abagyan, R. A. (2004) Large-scale prediction of protein geometry and stability changes for arbitrary single point mutations. *Proteins* 57, 400–413.

17. Zhou, H., Zhou, Y. (2002) Stability scale and atomic solvation parameters extracted from 1023 mutation experiments. *Proteins* 49, 483–492.

18. Khatun, J., Khare, S. D., Nikolay, V., Dokholyan. (2004) Can contact potentials reliably predict stability of proteins? *J Mol Biol* 336, 1223–1238.

19. Capriotti, E., Fariselli, P., Casadio, R. (2004) A neural-network-based method for predicting protein stability changes upon single point mutations. *Bioinformatics* 20, I63–I68.

20. Capriotti, E., Fariselli, P., Calabrese, R., Casadio, R. (2005) Predicting protein stability changes from sequences using support vector machines. *Bioinformatics* 21, ii54–ii58.

21. Cheng, J., Randall, A., Baldi, P. (2005) Prediction of protein stability changes for single-site mutations using support vector machines. *Proteins* 62, 1125–1132.

22. Saraboji, K., Gromiha, M. M., Ponnuswamy, M. N. (2005) Relative importance of secondary structure and solvent accessibility to the stability of protein mutants: a case study with amino acid properties and energetics on T4 and human lysozymes. *Comp Biol Chem* 29, 25–35.

23. Saraboji, K., Gromiha, M. M., Ponnuswamy, M. N. (2006) Average assignment method for predicting the stability of protein mutants. *Biopolymers* 82, 80–92.

24. Caballero, J., Fernandez, L., Abreu, J. I., Fernandez, M. (2006) Amino Acid Sequence Autocorrelation vectors and ensembles of Bayesian-Regularized Genetic Neural Networks for prediction of conformational stability of human lysozyme mutants. *J Chem Inf Model* 46, 1255-1268.

25. Parthiban, V., Gromiha, M. M., Hoppe, C., Schomburg, D. (2007) Structural analysis and prediction of protein mutant stability using distance and torsion potentials: role of secondary structure and solvent accessibility. *Proteins* 66, 41–52.

26. Huang, L. T., Gromiha, M. M., Ho, S. Y. (2007) iPTREE-STAB: interpretable decision tree based method for predicting protein stability changes upon mutations. *Bioinformatics* 23, 1292–1293.

27. Yin, S., Ding, F., Dokholyan, N. V. (2007) Eris: an automated estimator of protein stability. *Nat Methods* 4, 466–467.

28. Barker, W. C., Garavelli, J. S., Huang, H., McGarvey, P. B., Orcutt, B. C., Srinivasarao, G. Y., Xiao, C., Yeh, L. S., Ledley, R. S., Janda, J. F. et al. (2000) The protein information resource (PIR). *Nucleic Acids Res* 28, 41–44.

29. Bairoch, A., Apweiler, R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res* 28, 45–48.

30. Berman, H. M., Westbrook, J., Feng, Z, Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., Bourne, P. E. (2000) The Protein Data Bank. *Nucleic Acids Res* 28, 235–242.

31. Schomburg, I., Chang, A., Hofmann, O., Ebeling, C., Ehrentreich, F., Schomburg, D. (2002) BRENDA: a resource for enzyme data and metabolic information. *Trends Biochem Sci* 27, 54–56.

32. Kabsch, W., Sander, C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22, 2577–2637.

33. Eisenhaber, F., Argos, P. (1993) Improved strategy in analytical surface calculation for molecular system- handling of singularities and computational efficiency. *J Comp Chem* 14, 1272–1280.

34. Gromiha, M. M. (2005) A statistical model for predicting protein folding rates from amino acid sequence with structural class information. *J Chem Inf Model* 45, 494–501.

35. Gromiha, M. M., Thangakani, A. M., Selvaraj, S. (2006) FOLD-RATE: prediction of protein folding rates from amino acid sequence. *Nucleic Acids Res* 34, W70–W74.

36. Gromiha, M. M., Selvaraj, S. (2004) Inter-residue interactions in protein folding and stability. *Prog Biophys Mol Biol* 86, 235–277.

37. Gromiha, M. M., Pujadas, G., Magyar, C., Selvaraj, S., Simon, I. (2004) Locating the stabilizing residues in (alpha/beta)8 barrel proteins based on hydrophobicity, long-range interactions, and sequence conservation. *Proteins* 55, 316–329.

38. Nozaki, Y., Tanford, C. (1971) The solubility of amino acids and two glycine peptides in aqueous ethanol and dioxane solutions. Establishment of a hydrophobicity scale. *J Biol Chem* 246, 2211–2217.

39. Jones, D. D. (1975) Amino acid properties and side-chain orientation in proteins: a cross correlation approach. *J Theor Biol* 50, 167–183.

40. Ponnuswamy, P. K. (1993) Hydrophobic characteristics of folded proteins. *Prog Biophys Mol Biol* 59, 57–103.

41. Gromiha, M. M., Selvaraj, S. (2001) Comparison between long-range interactions and contact order in determining the folding rates of two-state proteins: application of long-range order to folding rate prediction. *J Mol Biol* 310, 27–32.

42. Dosztányi, Z., Fiser, A., Simon, I. (1997) Stabilization centers in proteins: identification, characterization and predictions. *J Mol Biol* 272, 597–612.

43. Glaser, F., Pupko, T., Paz, I., Bell, R. E., Bechor, D., Martz, E., Ben-Tal, N. (2003) ConSurf: identification of functional regions in proteins by surface-mapping of phylogenetic information. *Bioinformatics* 19, 163–164.

44. Kursula, I., Partanen, S., Lambeir, A. M., Wierenga, R. K. (2002) The importance of the conserved Arg191-Asp227 salt bridge of triosephosphate isomerase for folding, stability, and catalysis. *FEBS Lett* 518, 39–42.

45. González-Mondragón, E., Zubillaga, R. A., Saavedra, E., Chánez-Cárdenas, M. E., Pérez-Montfort, R., Hernández-Arana, A. (2004) Conserved cysteine 126 in triosephosphate isomerase is required not for enzymatic activity but for proper folding and stability. *Biochemistry* 43, 3255–3263.

46. Magyar, C., Gromiha, M. M., Pujadas, G., Tusnády, G. E., Simon, I. (2005) SRide: a server for identifying stabilizing residues in proteins. *Nucleic Acids Res* 33, W303–W305.

47. Gilis, D., Rooman, M. (1996) Stability changes upon mutation of solvent-accessible residues in proteins evaluated by database-derived potentials. *J Mol Biol* 257, 1112–1126.

48. Gilis, D., Rooman, M. (1997) Predicting protein stability changes upon mutation using database-derived potentials: solvent accessibility determines the importance of local versus non-local interactions along the sequence. *J Mol Biol* 272, 276–290.

49. Parthiban, V., Gromiha, M. M., Schomburg, D. (2006) CUPSAT: prediction of protein stability upon point mutations. *Nucleic Acids Res* 34, W239–W242.

50. DeLano, W. L. (2002) The PyMOL Molecular Graphics System. DeLano Scientific, San Carlos, CA, USA. //www.pymol.org.

Chapter 7

Enzyme Databases

Dietmar Schomburg and Ida Schomburg

Abstract

Enzymes are catalysts for the chemical reactions in the metabolism of all organisms and play a key role in the regulation of metabolic steps within the cells, as drug targets, and in a wide range of biotechnological applications. With respect to reaction type, they are grouped into six classes, namely oxidoreductases, transferases, hydrolases, lyases, and ligases. EC-Numbers are assigned by the IUBMB. Enzyme functional databases cover a wide range of properties and functions, such as occurrence, kinetics of enzyme-catalyzed reactions, structure, or metabolic function. BRENDA stores a large variety of different data for all classified enzymes whereas KEGG, MEROPS, MetaCyc, REBASE, CAzy, ESTHER, PeroxiBase, and KinBase specialize in either certain aspects of enzyme function or specific enzyme classes, organisms, or metabolic pathways. Databases covering enzyme nomenclature are ExplorEnz, SIB-ENZYME, and IntEnz.

Key words: enzyme, database, metabolism, enzyme kinetics, pathway, catalysis.

1. Introduction

Enzymes represent the largest and most diverse group of all proteins, catalyzing all chemical reactions in the metabolism of all organisms. They play a key role in the regulation of metabolic steps within the cell. With respect to the rapid development and progress in the fields of structural and functional genomics, metabolomics, and systems biology, the systematic collection, accessibility, and processing of enzyme data become even more important in order to analyze and understand the complex networks of biological processes.

2. Classification and Nomenclature

Although enzymes are the only protein family where names are recommended by the IUBMB/IUPAC enzyme commission, these recommendations are unfortunately often ignored in the literature. In many cases, the same enzyme became known by several different names, while conversely the same name was sometimes given to different enzymes. Many of the names conveyed little or no idea of the nature of the reactions catalyzed and similar names were sometimes given to enzymes of quite different types. The International Commission on Enzymes was founded in 1956 by the International Union of Biochemistry. Since then the system of enzyme code numbers (EC numbers) with systematic and recommended names has been established (1). Currently, there are 4,072 active EC numbers plus 795 inactive numbers for entries that have been deleted or transferred to another class. The old numbers have not been allotted to new enzymes; instead the place has been left vacant with comments concerning the fate of the enzyme (deletion or transfer). In the EC number system, an enzyme is defined by the reaction it catalyses. In some cases where this is not sufficient, additional criteria are employed such as cofactor specificity of the reaction. The 4,072 active EC numbers currently account for ∼43,000 synonyms (taken from the BRENDA enzyme resource). The number of classified enzymes is increasing by about 100 each year.

The enzyme code numbers, prefixed by EC, which are now widely in use, contain four elements separated by points, with the following meaning:

1. the first number shows to which of the six main divisions (classes) the enzyme belongs,

2. the second figure indicates the subclass,

3. the third figure gives the sub-subclass,

4. the fourth figure is the serial number of the enzyme in its sub-subclass.

5. **Table 7.1** gives an overview on the enzyme classes.

The *accepted name* is the most frequently used and recommended, although it may not be the most reasonable name. The *systematic name* consists of two parts. The first contains the name of the substrate or, in the case of a bimolecular reaction, of the two substrates separated by a colon. The second part, ending in *-ase*, indicates the nature of the reaction. Thus, the enzyme class EC 2.1.1.12 with the systematic name *S*-adenosyl-L-methionine:L-methionine *S*-methyltransferase transfers a methyl group from *S*-adenosyl-L-methionine to L-methionine producing *S*-adenosyl-L-homocysteine and *S*-methyl-L-methionine.

Table 7.1
Overview on enzyme classes defined by the NC-IUBMB

EC Class	Enzyme type	# of sub-sub-classes
1	Oxidoreductases	1,115
2	Transferases	1,178
3	Hydrolases	1,124
4	Lyses	369
5	Isomerases	163
6	Ligases	141

All proteins found to catalyze the same reaction are summarized under one EC number. The catalyzed reaction is written according to the rules for a chemical reaction including the stoichiometry. In the case of reversible reactions, the direction is the same for all the enzymes in a given class, even if this direction has not been demonstrated for all. Thus, the reaction may be written in a defined direction, even though only the reverse of this has been actually demonstrated experimentally. For some enzyme subclasses, namely proteases which hydrolyze peptide bonds in proteins or glycolases which hydrolyze glycosidic linkages it is not possible to draft an equation, therefore the reaction is replaced by a sentence, describing the specificity of the enzyme.

Where available each entry is equipped with a link to a graphical representation of the reaction frequently also displaying the enzyme in its metabolic context.

3. Enzyme Information Resources

The currently available enzyme databases can be grouped into global databases which cover all hitherto classified enzymes with or without their functional properties and databases for special enzymes classes or special enzyme-catalyzed reactions.

3.1. General Enzyme Databases

3.1.1. Enzyme Nomenclature Web Sites

The classification of enzymes according to the rules of enzyme nomenclature is the responsibility of the Enzyme Commission of the International Union for Biochemistry and Molecular Biology (IUBMB). The outcome of the decisions made by the commission is deposited in the enzyme list, which is made accessible by several Web sites (IUBMB website, ExplorEnz, SIB-ENZYME, IntEnz).

They provide forms for searching the enzyme's accepted name, the systematic name, some synonyms, the reaction, cofactors, and literature references.

3.1.1.1. IUBMB Nomenclature and ExplorEnz

The Enzyme Commission is the curator of the ExplorEnz database (http://www.enzyme-database.org/) (**Fig. 7.1**).

Main topics are classification and nomenclature. In a concise way it contains the basic data for all classified enzymes. Changes to the enzyme list, e.g., corrections in names, references, or reactions are displayed on a separate Web site (**Fig. 7.2**).

ExplorEnz also offers an input form for researchers to report on enzymes which are currently not classified in the enzyme list and also for requesting changes to existing entries.

Fig. 7.1. ExplorEnz, example.

Fig. 7.2. ChangeLog in ExplorEnz.

The compilation of new enzyme classes issued by the NC-IUBMB is followed by a period of public review. Enzymes undergoing this process are displayed on the ExplorEnz Web sites, where scientists can add their comments or request changes.

The contents of the ExplorEnz database are also displayed, often together with reaction diagrams, on the Enzyme Nomenclature pages of the IUBMB (http://www.chem.qmul.ac.uk/iubmb/enzyme/) In addition, this site gives detailed information on the rules for naming enzymes and on the nomenclature of biochemical molecules.

3.1.1.2. SIB-ENZYME Nomenclature Database

SIB-ENZYME (http://www.expasy.ch/enzyme/) connects the nomenclature of enzymes (2) with sequence information as stored in UNIPROT. A report form for an error or an update of existing entries can be used to draw the attention of the editor to enzymes and other catalytic entities missing from this list.

A special feature is the links for the protein sequences, which are deposited in UniProt enabling a direct access to individual enzyme proteins (*see* **Fig. 7.3**).

3.1.1.3. IntEnz

IntEnz (http://www.ebi.ac.uk/intenz/) (3) also contains enzyme data that are curated and approved by the Nomenclature Committee. Enzyme names and reactions are taken from the enzyme list of the NC-IUBMB (see above).

Fig. 7.3. Enzyme database at the SIB.

Some enzyme data are connected to the ChEBI (4) database, which provides a definitive dictionary of compounds to improve the quality of the IntEnz vocabulary. ChEBI stands for dictionary of Chemical Compounds of Biological Interest. The ChEBI database is also hosted at the European Bioinformatics Institute (EBI). IntEnz entries also provide links to the protein sequences stored in the UniProt database (**Fig. 7.4**).

3.1.2. Enzyme-Functional Databases

Unlike the above-mentioned databases, BRENDA (http://www.brenda-enzymes.org/) covers the full range of enzyme properties such as

3.1.2.1. BRENDA

- Classification and nomenclature
- Reaction and specificity
- Functional parameters
- Organism-related information
- Enzyme structure
- Isolation and preparation

Fig. 7.4. IntEnz, the enzyme database at the European Informatics Institute.

- Literature references
- Application and engineering
- Enzyme–disease relationships

The section on *Classification and Nomenclature* is based on the enzyme names as defined by the NC-IUBMB and is supplemented with all synonyms found in the ~79,000 literature references, which have been manually annotated so far. In BRENDA, all literature references are manually annotated and the data are quality controlled by scientists ensuring a high standard. *Reaction and Specificity* covers the complete range of natural and artificial substrates accepted by a particular enzyme. Many enzymes may have a wider substrate specificity and accept different substrates. Additional sections provide lists of inhibitors, cofactors, metal ions, and activating compounds. Since in biological sciences very often trivial names are used instead of International Union of Pure and Applied Chemistry (IUPAC) nomenclature, many compounds are known with a variety of names. Thus even simple molecules may have a dozen or more names. Brenda is equipped with a thesaurus for ligand names based on the IUPAC International Chemical Identifier (INChI) codes for 66,000 different compound names amounting to ~46,000 different chemical entities.

Enzyme-catalyzed reactions and compounds interacting with the enzyme protein (cofactors, inhibitors, activating compounds, etc.) can be viewed as graphical representations. A tool for substructure searches can be used for drawing a molecule and searching this or its more complex derivatives in the database. The molecular structures are also stored as molfiles enabling a wide range of bioinformatic and cheminformatic usages.

The enzyme information system BRENDA was founded in 1987 at the German National Research Center for Biotechnology (GBF) then was continued at the Cologne University Bioinformatics Centre and is now curated since 2007 at the Technical University (5). First, BRENDA was published as a series of books. The second edition was started in 2001. About 39 volumes are published so far, each containing about 500–600 pages encompassing 50–150 EC classes (6).

All data are stored in a relational database system. The user can choose from nine search modes:

- Quick search can be used for a direct search in one of the 54 data fields providing a fast and direct access, e.g., via enzyme names or metabolites (**Fig. 7.5**).
- Fulltext search performs a search in all sections of the database, including commentaries.
- Advanced search allows a combinatorial search for text or numerical data fields.

Fig. 7.5. BRENDA quick search.

- Substructure search is a tool for drawing a molecule which then is searched in the database. The results are exact matches or any molecule containing the plotted structure (*see* **Fig. 7.6**).

- TaxTree explorer allows to search for enzymes or organisms in the taxonomic tree.

Fig. 7.6. BRENDA substructure search.

- EC explorer can be used to browse or search the hierarchical tree of enzymes.
- Sequence search is useful for enzymes with a known protein sequence.
- Genome explorer connects enzymes to genome sequences. The location of classified enzymes is displayed in their genomic context.
- Ontology explorer allows to simultaneously search in all biochemically relevant ontologies, among them BrendaTissueOntology (BTO).

About 1.4 million functional and property data describing enzymes are stored in the database covering ~50 datafields. A summary of the amount of data is displayed in **Table 7.2**. All data in BRENDA are linked to the original paper reference.

Functional data are often context-dependent. Since every laboratory carries out their experiments on enzyme characterizations under individually defined conditions, and since they depend on the given experimental know-how, methods, and technical equipment available, raw data for the same enzyme are not comparable. In order to account for these differences, BRENDA very often includes the experimental conditions together with the data.

Table 7.2
Overview on enzyme data in BRENDA

	Data entries	Literature links
Enzyme with functional data	49,972	116,012
Nomenclature and classification	60,445	107,033
Substrates/products	557,794	820,970
Natural substrate/products	119,390	199,205
Inhibitors	118,371	135,544
Cofactor/activating substances	57,850	84,557
Kinetic data	170,723	180,577
Organism/localization/tissue	79,682	113,915
Enzyme structure	73,278	96,218
pH and temperature optima	50,198	57,633
Purification and cloning	46,370	77,244
Application and Engineering	33,666	38,534

Because until now there is no standardization for documenting these, the experimental and other details are given as a commentary directly linked to the functional data. Each entry is linked to a literature reference, allowing the researcher to go back to the original literature for further details.

Example: for aminobutyraldehyde dehydrogenase (EC-Number 1.2.1.19) from rat, two different K_M values, measured at different conditions, are reported:

0.018 mM (aminobutyraldehyde) 250 mM phosphate buffer, 1 mM NAD^+

0.081 mM (aminobutyraldehyde) 400 mM phosphate buffer, 1 mM NAD^+

Kinetic data can be submitted directly to the database (http://www.brenda-enzymes.org/strenda/).

All data in BRENDA are connected to the biological source of the enzyme, that is, the organism, the tissue, the subcellular localization, and the protein sequence (if available); consequently data for different isoenzymes can be identified. For the organisms in BRENDA, the taxonomy-lineage is given if the respective organism can be found in the NCBI taxonomy database (National Center for Biotechnology Information, USA). Using the TaxTree search mode, the user can search for enzymes along the taxonomic tree and move to higher or lower branches to get either an overview or restrict the search.

Different isoenzymes in different tissues may be found. Sometimes enzymes restricted to a single tissue or any organ may express a specific isoenzyme. The BRENDA tissues grouped into a hierarchical tissue ontology (Brenda Tissue Ontology, BTO), which was developed by the BRENDA team, is available from OBO and meanwhile used by a large number of different groups.

3.1.2.2. AMENDA/FRENDA

AMENDA (Automatic Mining of ENzyme DAta) and FRENDA (Full Reference ENzyme DAta) are supplements to BRENDA. AMENDA contains a large amount of enzyme data which are automatically extracted from ~18 million PubMed abstracts (US National Library of Medicine) using modern optimized text-mining procedures. FRENDA aims at providing an exhaustive collection of literature references containing organism-specific enzyme information. The use of these databases is restricted to the academic community. As the development of AMENDA and FRENDA could not financed by public money, the data are available for the academic community free of charge but commercial users have to obtain a license http://www.biobase-international.com/

3.1.2.3. KEGG

In the KEGG database (Kyoto Encyclopedia of Genes and Genomes, **Fig. 7.7**), enzyme information is stored as a part of the LIGAND database (http://www.genome.jp/ligand/) (7). This is a composite database currently consisting of

KEGG LIGAND Database
Molecular building blocks of life in the chemical space

| KEGG2 | ATLAS | PATHWAY | BRITE | GENES | SSDB | LIGAND | DBGET |

Chemical Substances and Reactions

KEGG LIGAND contains our knowledge on the universe of chemical substances and reactions that are relevant to life. It is a composite database currently consisting of COMPOUND, DRUG, GLYCAN, REACTION, RPAIR, and ENZYME databases. ENZYME is derived from the Enzyme Nomenclature, but the others are internally developed and maintained.

Database		Identifier	Content	Specialized entry point
LIGAND	COMPOUND	C number	Chemical compound structures	KEGG COMPOUND
	DRUG	D number	Drug structures	KEGG DRUG
	GLYCAN	G number	Glycan structures	KEGG GLYCAN
	REACTION	R number	Biochemical reactions	
	RPAIR	A number	Reactant pair alignments	KEGG REACTION
	ENZYME	EC number	Enzyme nomenclature	
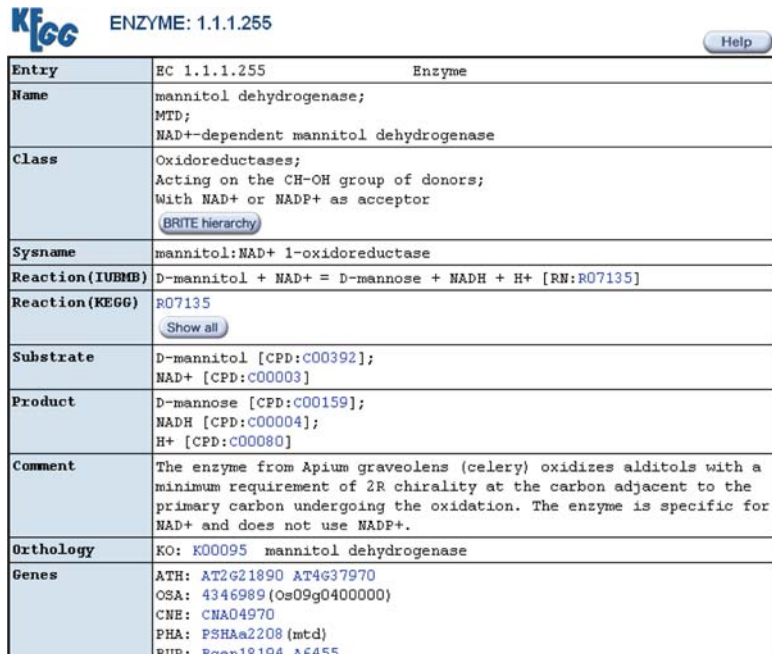
Search LIGAND ⌄ for _____ [Go] [Clear]
⊙ bfind mode ○ bget mode

Fig. 7.7. KEGG ligand database.

- Compound
- Drug
- Glycan
- Reaction
- Repair
- Enzyme

KEGG-ENZYME is also derived from the IUBMB Enzyme Nomenclature, but the other datasets like compound, drug, glycan, reaction, repair are developed and maintained by the Kanehisa Laboratories in the Bioinformatics Center of Kyoto University and the Human Genome Center of the University of Tokyo. In addition to the nomenclature enzyme data comprise substrates, products, reactions, gene names, and links to chemical structures of metabolites, reaction diagrams, and metabolic pathways.

Enzyme-catalyzed reactions are stored in the KEGG REACTION database containing all reactions from KEGG ENZYME and additional reactions from the KEGG metabolic pathways, the latter without an EC classification. Each reaction is identified by the R number, such as R06466 for the isomerization of (*S*)-2,3-epoxysqualene to lupeol. Reactions are linked to ortholog groups of enzymes as defined by the KEGG ORTHOLOGY database, enabling integrated analysis of genomic (enzyme genes) and chemical (compound pairs) information. **Figure 7.8** shows the entry for mannitol dehydrogenase as an example.

Fig. 7.8. KEGG, entry of EC 1.1.1.255.

3.2. Special Enzyme Databases

Whereas the above-described databases cover all enzymes which have been classified by the NC-IUBMB, there are some databases which are specialized on certain enzyme classes.

3.2.1. MEROPS

The MEROPS database is a manually curated information resource for peptidases (also known as proteases, proteinases, or proteolytic enzymes), their inhibitors, and substrates (8). The database has been in existence since 1996 and can be found at http://merops.sanger.ac.uk/. Releases are made quarterly. Peptidases and protein inhibitors are arranged in the database according to a hierarchical classification. The classification is based on sequence comparisons of the domains known to be important for activity (known as the peptidase or inhibitor unit). A protein that has been sequenced and characterized biochemically is chosen as a representative ("holotype"). All sequences that represent species variants of the holotype are grouped into a "protein species." The sequences of statistically significant related protein species are grouped into a "family." Families that are believed to have had a common ancestor, either because the tertiary structures of the proteins are similar or (in the case of peptidases) active site residues are in the same order in the sequence, are grouped into a "clan."

The substrate specificity is described in two ways:

1. For any peptidase with more than ten known cleavages, a display is presented that gives an indication of the amino acids preferred at its substrate binding sites. This display

Fig. 7.9. MEROPS, entry for caspase-1.

uses the WebLogo software (9). Details of the amino acid sequences around the cleavage sites are displayed in the "Specificity Matrix."

1. In addition to the logo, a text string describing the specificity is also shown.

2. Artificial or model substrates are summarized in text-sheets, including literature references (*see* **Fig. 7.9** with the entry for caspase).

3.2.2. MetaCyc

MetaCyc (http://metacyc.org/) is a nonredundant reference database of small-molecule metabolism that contains experimentally verified metabolic pathway and enzyme information obtained from the scientific literature (10). The metabolic pathways and enzymes in MetaCyc are from a wide variety of organisms with an emphasis on microbial and plant metabolism, although a significant number of animal pathways are also included. Enzymes can be searched via the IN-IUBMB EC number or via their names. They are displayed within the various pathways or with a graphic reaction diagram and links to the connected pathways (**Fig. 7.10**).

3.2.3. REBASE

REBASE is a comprehensive database of information about restriction enzymes, DNA methyltransferases, and related proteins involved in the biological process of restriction-modification (11). It contains fully referenced information about recognition and cleavage sites, isoschizomers, neoschizomers, commercial availability, methylation sensitivity, crystal and sequence data (**Fig. 7.11**). Experimentally characterized homing endonucleases

Fig. 7.10. MetaCyc display of EC 1.1.1.13, arabitol dehydrogenase.

Fig. 7.11. REBASE entry for Eco1051.

are also included. All newly sequenced genomes are analyzed for the presence of putative restriction systems and these data are included within the REBASE. The contents or REBASE may be browsed from the Web (http://rebase.neb.com/rebase/rebase. ftp.html) and selected compilations can be downloaded by ftp (ftp.neb.com).

3.2.4. Carbohydrate-Active Enzymes (CAzy)

The CAzy database (http://www.cazy.org/index.html) describes the families of structurally related catalytic and carbohydrate-binding modules (or functional domains) of enzymes that degrade, modify, or create glycosidic bonds (12). The NC-IUBMB Enzyme nomenclature of glycoside hydrolases is based on their substrate specificity and occasionally their molecular mechanism. Such a classification does not reflect (and was not intended to) the structural features of these enzymes. A

Fig. 7.12. CAzy, entry for gylcoside hydrolase family 1.

classification of glycoside hydrolases in families based on amino acid sequence similarities (example for glycoside hydrolase family is shows in **Fig. 7.12**) has been proposed a few years ago.

The biosynthesis of disaccharides, oligosaccharides, and polysaccharides involves the action of hundreds of different glycosyltransferases (EC 2.4.x.y), enzymes which catalyze the transfer of sugar moieties from activated donor molecules to specific acceptor molecules, forming glycosidic bonds. In similar manners, classifications for polysaccharide lyases and carbohydrate esterases are presented.

Because there is a direct relationship between sequence and folding similarities, these classifications

- reflect the structural features of these enzymes better than their sole substrate specificity
- help to reveal the evolutionary relationships between these enzymes
- provide a convenient tool to derive mechanistic information

3.2.5. Databases Based on Sequence Homologies

Numerous enzyme databases on the Web are specialized in the analysis of protein and gene sequences for enzyme groups. Examples are

The **ESTHER Database** is dedicated to the analysis of protein and nucleic acid sequences belonging to the superfamily of alpha/ beta hydrolases homologous to cholinesterases (http://bioweb. ensam.inra.fr/ESTHER/definition) (13).

PeroxiBase is curated in collaboration with the Swiss Institute of Bioinformatics (SIB). The goal of this peroxidase database is to centralize most of the peroxidase superfamilies encoding sequences, to follow the evolution of peroxidase among living

organism and compile the information concerning putative functions and transcription regulation (http://peroxibase.isb-sib.ch/index.php) (14).

KinBase holds information on over 3,000 protein kinase genes found in the genomes of human and many other sequenced genomes. It explores the functions, evolution, and diversity of protein kinases, the key controllers of cell behavior with a focus on the kinome, the full complement of protein kinases in any sequenced genome. This includes the extensive KinBase (http://kinase.com/) database (15).

References

1. *Enzyme Nomenclature* (1992 and Supplements) Academic Press.

2. Bairoch, A. (2000) The ENZYME database in 2000. *Nucleic Acids Res* 28, 304–305.

3. Fleischmann, A., Darsow, M., Degtyarenko, K., Fleischmann, W., Boyce, S., Axelsen, K. B., Bairoch, A., Schomburg, D., Tipton K. F., Apweiler, R. (2004) IntEnz, the integrated relational enzyme database. *Nucleic Acids Res* 32, D434–D437.

4. Degtyarenko, K., Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Darsow, M., Guedj, M., Ashburner, M. (2008) ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res* 36 (Database issue), D344–D350.

5. Barthelmes, J., Ebeling, C., Chang, A., Schomburg, I., Schomburg, D. (2007) BRENDA, AMENDA and FRENDA: the enzyme information system in 2007. *Nucleic Acids Res* 35, D511–D514.

6. Schomburg, D., Schomburg, I. (2001) *Springer Handbook of Enzymes*, 2nd ed. Springer, Heidelberg.

7. Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., Yamanishi Y. (2008) KEGG for linking genomes to life and the environment. *Nucleic Acids Res* 36, D480–D484.

8. Rawlings, N. D., Morton, F. R., Kok, C. Y., Kong, J., Barrett, A. J. (2008) MEROPS: the peptidase database. *Nucleic Acids Res* 36, D320–D325.

9. Crooks, G. E., Hon, G., Chandonia, J. M., Brenner, S. E. (2004) WebLogo: a sequence logo generator. *Genome Res* 14(6), 1188–1190.

10. Caspi, R., Foerster, H., Fulcher, C. A., Kaipa, P., Krummenacker M., Latendresse, M., Paley, S., Rhee, S. Y., Shearer, A. G., Tissier, C., Walk, T. C., Zhang, P., Karp, P. D. (2008) The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Res* 36 (Database issue), D623–D631.

11. Roberts, R. J., Vincze, T., Posfai, J., Macelis, D. (2007) REBASE-enzymes and genes for DNA restriction and modification. *Nucleic Acids Res* 35 (Database issue), D269–D270.

12. Coutinho, P. M., Henrissat, B. (1999) Carbohydrate-active enzymes: an integrated database approach, in (Gilbert, H. J., Davies, G., Henrissat, B., Svensson, B., eds.) *Recent Advances in Carbohydrate Bioengineering*. The Royal Society of Chemistry, Cambridge, pp. 3–12.

13. Hotelier, T., Renault, L., Cousin, X., Negre, V., Marchot, P., Chatonnet, A. (2004) ESTHER, the database of the alpha/beta-hydrolase fold superfamily of proteins. *Nucleic Acids Res* 32 (Database issue), D145–D147.

14. Passardi, F., Theiler, G., Zamocky, M., Cosio, C., Rouhier, N., Teixera, F., Margis-Pinheiro, M., Ioannidis, V., Penel, C., Falquet, L., Dunand C. (2007) PeroxiBase: The peroxidase database. *Phytochemistry* 68(12), 1605–1611.

15. Manning, G., Whyte, D. B., Martinez, R., Hunter, T., Sudarsanam S. (2002) The protein kinase complement of the human genome. *Science* 298, 1912–1934.

Chapter 8

Biomolecular Pathway Databases

Hong Sain Ooi, Georg Schneider, Teng-Ting Lim, Ying-Leong Chan, Birgit Eisenhaber, and Frank Eisenhaber

Abstract

From the database point of view, biomolecular pathways are sets of proteins and other biomacromolecules that represent spatio-temporally organized cascades of interactions with the involvement of low-molecular compounds and are responsible for achieving specific phenotypic biological outcomes. A pathway is usually associated with certain subcellular compartments. In this chapter, we analyze the major public biomolecular pathway databases. Special attention is paid to database scope, completeness, issues of annotation reliability, and pathway classification. In addition, systems for information retrieval, tools for mapping user-defined gene sets onto the information in pathway databases, and their typical research applications are reviewed. Whereas today, pathway databases contain almost exclusively qualitative information, the desired trend is toward quantitative description of interactions and reactions in pathways, which will gradually enable predictive modeling and transform the pathway databases into analytical workbenches.

Key words: biomolecular pathway database, pathway, KEGG.

1. Introduction

Recent years have shown a rapidly growing interest in biological pathway analysis as can be deduced from the number of biological pathway resources listed in Pathguide (1). Currently, Pathguide contains almost 300 resources; this is an about 50% increment compared to the situation 2 years ago (190 resources). On average, one new resource was introduced every 2 weeks during the past 2 years. The explosive growth of interest in pathway analysis was triggered by the availability of high-throughput methods involving complete sequencing of several model organisms, large-scale expression studies, etc. It is the first time that one can study a biological problem reasonably at the system level, at which various types of biological information such as the functions of genes and

proteins, molecular interaction networks, and biological pathways need to be put together to form a computational working model of a living cell (2). Such models have the ultimate potential to become useful for better understanding the cellular behavior in response to outside stimuli including drug treatment. There is hope that, at the end, system-level studies might lead to more successful rational design of effective therapeutic agents that are free of undesired side effects, although we have to admit that there is a long way to go. At present, pathway libraries mainly fulfill the function of knowledge depositories. Sophisticated pathway modeling and simulation are in their infancy and rarely accompanied by an outcome of results with biological relevance due to the crudeness of assumptions and the incompleteness of the networks known so far.

It must be emphasized that the category of a pathway is an idealized human construct for dissecting the whole, very complex network of interacting genes and proteins into more comprehensible smaller parts that can be associated with a certain cellular or physiological function. It is assumed that the coordinated action of this subgroup of genes and proteins is especially important for the given function (or biological process) and that their interaction with other parts of the network is relevant to a lesser degree than within the subgroup. The three major types of biological pathways are responsible for (i) gene regulation, (ii) metabolic processes, and (iii) signal transduction. In gene regulation pathways, transcription factors play a crucial role in activating or inhibiting the expression of a set of genes, which in turn, may trigger other pathways. Metabolic pathways consist of a series of biochemical reactions governed by sets of enzymes that convert low-molecular (typically organic) compounds into one another. For example, in the glycolysis pathway, glucose is processed to pyruvate with concomitant generation of free energy equivalents for use in other cellular processes. Signal transduction cascades, the cell's communication mechanisms, convey messages from one part of the cell to another through a series of binding events, transmitter redistributions, and protein modifications.

In the existing pathway databases, pathways are described independently and the respective information is typically represented in isolation. However, in reality, all these pathways are part of one complete interaction network. Over the past few years, while many pathway databases have been created (*see* **Table 8.1** for a list of important public domain pathway databases), some databases have focused on a particular type of pathways. Hence, collecting the information about genes or pathways of interest can be a daunting task for a researcher who is not familiar with the scope and type of the existing databases.

Table 8.1
Public pathway databases

Database	Description	Type of pathways	Organisms	Data collection	URL
BioCarta	A curated collection of pathways represented as interactive diagrams and its contents can be submitted or updated by anyone	Metabolic, signaling	Human, mouse	Manually curated	http://www.biocarta.com
BioCyc	BioCyc contains pathways from two manually curated databases, MetaCyc and EcoCyc, as well as pathways predicted using Pathway Tools. Some of the predicted pathways have been manually reviewed as well	Metabolic	Over 900 organisms	Manually curated and predicted using Pathway Tools	http://www.biocyc.org
GenMAPP	GenMAPP provides a compilation of pathways from several sources, such as KEGG and SGD, in addition to their own manually curated pathways. GenMAPP also contains tools for annotating and visualizing results obtained from expression arrays onto the pathways	Metabolic, signaling	Several organisms (such as human, zebrafish, and cow)	Manually curated and inferred	http://www. genmapp.org
Kyoto Encyclopedia of Genes and Genomes (KEGG)	A pathway resource that shows a common network map for all organisms and highlights known orthologues on the map when a particular species is selected	Metabolic, signaling, disease pathways	Over 650 organisms	Manually curated and inferred	http://www.genome. jp/kegg/pathway.html
MIPS CYGD (Comprehensive Yeast Genome Database)	This database is a yeast genome database and contains several pathways and protein annotations for yeast	Metabolic, signaling	Yeast	Manually curated	http://mips.gsf.de/ genre/proj/yeast

(continued)

Table 8.1 (continued)

Database	Description	Type of pathways	Organisms	Data collection	URL
PANTHER	PANTHER is a tool that classifies genes by their function, using experimental evidence and prediction algorithms and provides annotations such as gene ontologies and pathways	Signaling	Human	Manually curated	http://www.pantherdb.org
Protein Lounge	A collection of pathways, which are shown as interactive diagrams. For each pathway, there is a summary with citations to publications	Metabolic, signaling	Several organisms (mammals, plants, bacteria, fungi)	Manually curated	http://www.proteinlounge.com/Pathway/Default.aspx
Reactome	Reactome contains reaction networks and pathways in a form useful for systems modeling. The pathways curated in the database are mainly from human and inferred to other organisms through their respective orthologues, either putative or known	Metabolic, signaling	Human	Manually curated and inferred	http://www.reactome.org
Science Database of Cell Signaling	A resource that collects the various pathways from publications in Science STKE. It is possible to search and browse the pathways, summaries and references are provided	Signaling	Several organisms (vertebrates, plants, bacteria)	Manually curated	http://stke.sciencemag.org/cm
Saccharomyces Genome Database (SGD)	A collection of pathways generated by the PathoLogic module from Pathway Tools, which are manually curated and corrected	Metabolic	Yeast	Predicted using Pathway Tools and manually curated	http://www.yeastgenome.org/biocyc

From the technical point of view, pathway studies based on database collections depend heavily on software tools and computational algorithms, database technologies, and the quality of information provided by experimental methods. To date, there is a plethora of pathway analysis tools, ranging from a simple visualization device to a full-fledged commercial solution dedicated to pathway analysis. For example, mapping of the gene expression data to the collection of pathways allows the assessment of biological processes involved in the experiment and expedites the understanding of the interaction between genes, proteins, and metabolites. Thus, pathway knowledge and understanding is a crucial element of current attempts of biological data interpretation and many groups have dedicated enormous efforts to constructing and maintaining pathway databases. While a number of problems still remain to be solved, pathway analysis has advanced our understanding in several areas of biological study. For example, studies in cancer analysis have suggested that pathways rather than individual genes control the nature of tumorigenesis (3, 4).

It must be emphasized that high-throughput data sets represent the smaller part of knowledge about pathways and most of the information is buried in the scientific literature that is not easy to digest for producing electronically readable pathway information. In the end, this requires reading and abstracting of thousands of articles in a formalized manner by specialized annotators. The long breath (both in a time perspective and from the viewpoint of funding) necessary for such an effort is frequently out of range for academic teams and, thus, commercial efforts see an opportunity to step in. Not surprisingly, companies like GeneGo (www.genego.com), BIOBASE (www.biobase-international.com), Molecular Connections (www.molecularconnections.com), or Ingenuity (www.ingenuity.com) hold the largest and best annotated pathway databases (*see* **Table 8.2** for a list of important commercial pathway database systems); yet, they are not immediately accessible for the public domain. In this chapter, we review some of the widely used publicly available pathways databases, focusing on their scope, annotation quality, and tools.

2. Current Development in Pathway Databases

The diverseness of the pathway resources available today presents a great challenge to researchers. Searching for the necessary information in numerous databases and understanding the different content structuring approaches are laborious. Furthermore, the lack of uniform data models and data access methods renders data

Table 8.2
Commercially supported pathway databases

Database	Description	Type of pathways	Organisms	Data collection	URL
Ingenuity Pathway Analysis (IPA)	A Web-based commercial pathway database coupled with analysis tools from Ingenuity Systems. IPA also includes metabolic signaling pathway information obtained from KEGG	Metabolic, signaling	Human, mouse, rat, dog	Manually curated and obtained from KEGG	http://www.ingenuity.com/products/pathways_analysis.html
Linnea Pathways	Maintained by Invitrogen, Linnea Pathways is a publicly available curated resource for displaying pathways in interactive maps and highlighting products from Invitrogen such as primers and antibodies	Metabolic, signaling	Human	Manually curated	http://escience.invitrogen.com/ipath
MetaCore	Released by GeneGo, MetaCore is a discovery platform for studying pathways using data obtained from high-throughput experimental techniques such as microarrays, SAGE gene expression profiles, proteomics, and metabolomics	Metabolic, signaling	Human, mouse, dog, fruit-fly	Manually curated	http://www.genego.com/metacore.php
PathArt	A commercial database by Jubilant Biosys comprising three modules: pathways, interaction maps, and drug molecules. Also allows mapping of gene sets onto pathways	Metabolic, signaling, disease pathways	Various organisms such as human, mouse, and rat	Manually curated	http://www.jubilantbio.com/pathart.html
ResNet		Signaling	Human, mouse, rat, plants, yeast, worm, fruit-fly	Manually curated and predicted from other organisms	http://www.ariadnegenomics.com/products/databases/ariadne-resnet

integration a Herculean task. While several efforts have been developed to assist the exchange of pathway information, it is still in a primitive state. These data standards are relatively recent compared with traditions followed for several pathway databases that have existed for more than a decade such as KEGG (5) and EcoCyc (6). In some cases, these databases employ data representations that cannot be exactly represented in the exchange formats and their curators may even decide not to adopt the exchange formats at all. Currently, there is no quick solution to all these issues. The worst of all possibilities is the representation of pathways just in graphics form that are almost impossible to parse electronically, although this type of representation appears useful for the human eye (e.g., the Science Database of Cell Signaling at http://stke.sciencemag.org/cm.) Other attempts result in encyclopedia-type compilations of pathway descriptions (*see* **Table 8.3**).

In the following section, we discuss several of the most widely used pathway databases that could serve as a starting point to find genes or pathways of interest. Readers are advised to visit Pathguide (http://www.pathguide.org) for a complete list of available pathway databases. In **Table 8.4**, we list several important pathway analysis tools.

2.1. Kyoto Encyclopedia of Genes and Genomes (KEGG)

KEGG (5, 7) is a suite of databases, which can be grouped into four main classes of information, genes and proteins (KEGG GENES), pathway information (KEGG PATHWAY), relationship between various biological objects (KEGG BRITE), and chemical compounds (KEGG LIGAND). KEGG was first introduced in 1995 and many significant improvements have been made since then. KEGG PATHWAY currently contains more than 90,000 organism-specific pathways generated from 335 reference pathways. KEGG maintains pathway information concerning metabolism, genetic information processing, environmental information processing, cellular processes, human diseases, and drug responses. The richness of pathway information stored makes KEGG one of the most widely used pathway databases. The KEGG data are available for academic users and may be downloaded from the KEGG FTP site at ftp://ftp.genome.jp/pub/kegg.

In KEGG, each subdatabase can be browsed or searched. The KEGG PATHWAY page lists all manually drawn pathway maps. A particular pathway map or entry is retrievable by the corresponding map number. The pathway is displayed as a reference pathway or an organism-specific pathway. Users can switch between different organisms using the dropdown menu. The pathway map is clickable and navigation from one pathway to another is easily possible. The pathway entry page provides a more detailed description of the pathway and lists the corresponding literature references.

Table 8.3
Public pathway repositories

Repository	Description	Type of pathways	Organisms	Data collection	URL
Biological Biochemical Image Database (BBID)	BBID is a collection of biological pathway images obtained from scientific publications with indices using keywords, genes, and proteins names	Signaling	Human	Manually curated	http://bbid.grc.nia.nih.gov
ExPASy Biochemical Pathways	This repository is a collection of digitized wall charts showing all biochemical pathways using the material of the famous Boehringer Mannheim wall charts	Metabolic	Human	Manually curated	http://www.expasy.org/cgi-bin/search-biochem-index
WikiPathways	WikiPathways is an open concept and displays a collection of pathways as static pictures	Metabolic, signaling	Human, mouse, rat, fruit-fly, worm, yeast	Manually curated and obtained from GenMAPP	http://www.wikipathways.org/index.php/WikiPathways

Table 8.4
Some important pathway tools

Repository	Description	Nature of tool	URL
Database for Annotation, Visualization and Integrated Discovery (DAVID)	This tool uses a list of genes using various identifiers such as their affymetrix identifiers as input and provides annotations, which include gene ontology (GO) terms, protein–protein interactions and pathways. The pathways are consolidated from BBID, BioCarta, and KEGG	Annotations for a list of genes	http://david.abcc.ncifcrf.gov
PathwayExplorer	A Java-based Web client that maps and clusters gene lists with their expression profiles onto pathways obtained from BioCarta, GenMAPP, and KEGG	Mapping of genes onto pathways	https://pathwayexplorer.genome.tugraz.at
Pathway Studio	Pathway Studio, a product from Ariadne Genomics, aids researchers in analyzing a list of genes or probe sets by providing functionalities such as heat maps, clustering, and pathway mapping. The pathways can be extracted from various databases such as ResNet, KEGG, STKE, and Prolexys HyNet	Mapping of genes onto pathways	http://www.ariadnegenomics.com/products/pathway-studio
Dynamic signaling maps	A tool that allows one to input a pathway in the DSM language and automatically generate pathway images	Creation of pathway maps	http://www.hippron.com/hippron/index.html
Graphviz	Graphviz is a tool to automatically generate network graphs (such as interaction networks or pathways) for display using a set of layout algorithms	Creation of pathway maps	http://www.graphviz.org
GenePath	GenePath can aid researchers in inferring genetic networks through a set of relationship patterns	Prediction of genetic networks	http://genepath.org
Pathway Tools	A suite of algorithms to predict the metabolic pathways of an organism as well as to provide annotations of "omics" data	Prediction of pathways	http://bioinformatics.ai.sri.com/ptools
Cell Illustrator	Cell Illustrator allows users to draw, model, and simulate biological processes and networks for testing hypotheses	Network modeling	http://www.cellillustrator.com

The latest addition to KEGG is KEGG Atlas (8), which consists of a global metabolism map and a viewer. The global map is built by manually combining around 120 KEGG metabolic pathways. The KEGG Atlas allows users to align a set of genes or compounds to the global map and user-specified entries will be highlighted as colored lines or circles. This feature is especially useful in determining which pathways are activated or depressed in a particular gene expression experiment. Instead of downloading a local copy of KEGG data, users can also assess the data through KEGG-specific application program interfaces (APIs).

2.2. Reactome

Reactome is a knowledgebase of biological processes where the information is represented in a reaction-centered form (9). A reaction is defined as a path of conversion of inputs into outputs. Both inputs and outputs can be physical entities such as small molecules, proteins, nucleotides, or complexes. This data representation is flexible enough to encompass most biological interactions. The reactions are then linked together to form a full pathway. Reactome mainly concentrates on the curation of human reactions, except when there are gaps in human data (10). At the time of writing, 926 human pathways were available. Pathways for other organisms can also be retrieved via orthology relationships from the respective human pathways.

The database Reactome comes with an intuitive browsing interface that allows users to easily locate the necessary information. The reaction map displays all available pathways of a particular organism and an individual component can be selected for further investigation. A list of available biological processes is also provided under the reaction map (this is known as the Pathway Topic List). The human reaction map is used by default; however, alternative organisms can be selected from the dropdown menu. The information is organized in a tree-like structure where a particular biological process represents the root of the tree. Each branch points to a more specific component of the biological processes and eventually leads to a reaction. At each level, additional data such as equivalent events in other organisms, participating molecules, and descriptions are provided. The information can be viewed or downloaded in several formats such as BioPAX (http://www.biopax.org), SBML (11), or Cytospace (12). Basically, one can search for information in Reactome by reaction, gene, and protein names as well as several other identifiers. But an advanced search interface with a variety of options is also available which can limit the search to particular fields. PathFinder is a useful tool for finding pathways connecting a compound or reaction to one or more compounds or reactions. If several outputs are specified, the shortest path is returned. Given a set of user-defined genes, Skypainter can be used to identify the reactions or pathways that are statistically overrepresented. Skypainter recognizes a

variety of gene and protein identifiers including Affymetrix probe sets. Reactome provides an extensive help guide and users are encouraged to refer to it.

2.3. BioCyc

BioCyc, a meta-database, is a collection of pathway/genome databases (PGDB) (13). The databases in BioCyc can be classified into three tiers, based on their annotation quality level. Tier 1 components undergo intensive curation and contain EcoCyc (14) and MetaCyc (13). EcoCyc is an organism-specific database and contains extensive information about the metabolic and regulatory network, genes and gene products of *Escherichia coli* K-12 collected from the literature (from more than 17,000 publications) in a systematic manner. On the other hand, MetaCyc stores more than 1,100 metabolic pathways obtained from more than 1,500 different organisms through extensive literature curation. Tier 2 contains 20 network databases for organisms with fully sequenced genome, but insufficient available experimental information about gene function. These networks are inferred from MetaCyc with moderate manual curation through the use of the PathoLogic program, a part of the Pathway Tools software (15). The remaining tier contains 354 databases predicted with PathoLogic without any further curation. The data files can be obtained in two popular formats, BioPAX and SBML.

BioCyc provides two search interfaces, a standard form and an advanced query page. In the former, several types of queries are listed together. The user first selects the database of interest and then chooses a particular type of query to be performed. Only one query can be executed per request. It is possible to select an entry in the "Genome Browser" for visualization of a whole chromosome, display statistics and updates of the history of the database, perform a text search using the "Query" field, browse through ontologies, execute BLAST searches using protein or nucleotide sequences or display the list of all pathways, proteins, genes, or compounds. The advanced query page allows formulating requests using a language known as BioVelo. There is no need to know this syntax of BioVelo since the input form will translate the request into BioVelo and execute it. The query language allows users to create requests such as, for example, to find all human proteins with the Gene Ontology (16) term "localized to cytoplasm." For a more detailed description of the query language, users are advised to consult the online documentation. For users already familiar with the query language BioVelo, a free form advanced query page is also available.

2.4. Pathway Interaction Database (PID)

PID is a collaboration between the National Cancer Institute (NCI) and the Nature Publishing Group to provide the cancer research community with high-quality pathway information pertaining to human molecular signaling and regulatory events (17).

The data are curated from peer-reviewed literature and constantly updated. PID also contains data directly imported from Reactome and BioCarta. Currently, there are 88 human pathways (4,944 interactions) curated by NCI–Nature and 318 human pathways (6,538 interactions) imported from BioCarta and Reactome. The number of pathways listed excludes subnetworks. The data are freely available in PID XML and BioPAX formats, without any restriction on use.

A user can select a pathway of interest from the list of all available pathways. Each pathway is displayed as a clickable static image. Several popular file formats are supported, for example, GIF, SVG, PID XML, and BioPAX. The "Molecule list" shows all participating molecules and the complexes they form. The list of references for the selected pathway is also available. PID provides a simple search box to query the database. The search field supports Entrez Gene identifiers, UniProt identifiers, HUGO gene symbols, and Gene Ontology biological process terms or identifiers. A more advanced search option is also available. For NCI–Nature curated data, searches can be limited by evidence codes. The "Connected molecules" query allows users to construct a single network containing all molecules with a minimal amount of interaction links.

3. Problems Associated with Pathway Analysis Based on Public Databases

A pathway, by itself, is a human construction that represents the current understanding of a particular biological process, which is partially complete and subject to changes when new knowledge emerges. Pathway analysis depends heavily on the completeness of the pathway information and, obviously, incomplete information may lead to misinterpretation of sets of target genes obtained from an experimental screen. In a particular case, it has been shown that the existing pathway databases do not even contain all pathway information presented in the public literature (18). In the study, the authors argue that many parts of the fatty acid metabolism pathway are missing or incomplete in four widely used pathway databases (KEGG, Reactome, GenMAPP (19), and BioCarta (http://www.biocarta.com)). Furthermore, some records have not been updated for years, even for KEGG. This has raised concerns about pathway analysis, which highly depends on the quality of the pathway information.

Therefore, there is a clear need to consolidate pathway information by integrating the data from different pathway sources. This approach provides several advantages such as filling in gaps in

pathways and enabling the cross-validation of existing pathway components. It might sound relatively easy to integrate all information of a particular pathway from various sources to create one complete pathway. However, in reality, researchers are often discouraged to do so. One of the main issues is that each pathway database provider has special preferences in choosing the data model and conventions in representing the pathway knowledge and it is laborious to squeeze the available data of a given database into a more general data model. Various data exchange standards have been developed to overcome this issue.

Currently, the two most widely used exchange formats for pathways are BioPAX (http://www.biopax.org) and SBML (11). SBML is mainly used for mathematical modeling and simulations of pathways. On the other hand, BioPAX is more flexible and the data model can represent molecular interactions and is capable of accommodating additional information such as chemical structures and mathematical models. While it is generally possible to convert the pathway data from one format to another, the level of information may differ or the same information cannot be fully presented at all. The problem is further complicated if the database providers employ data models that are not fully compatible with the exchange format. In this case, while the syntax for the pathway data in a particular exchange format is correctly followed, the semantics of the data might not be the same. For example, in Reactome, an EntitySet, a set of physical entities with their functions interchangeable in a given situation, is encoded as a "generalized" protein with several external references to different proteins in the BioPAX file format. It should be emphasized that, following the syntax of BioPAX, a protein entry has to link to a single sequence, possibly with several external reference points to entries in other databases, instead of to different proteins.

We performed a study of the proteins in human pathways in a variety of pathway databases. **Tables 8.5** and **8.6** show the contents of pathways based on the BioPAX Level 2 specification (http://www.biopax.org/release/biopax-level2.owl). **Table 8.5** lists the number of pathways together with the biomolecules in them. Following **Table 8.6**, only Reactome contains so-called physical entities. While the physical entity entry can be used to represent an entity with physical structure, instances of a physical entity should never be created based on the BioPAX specification. These entries were removed in the Reactome data set in PID (PID Reactome), which suggests that some curation was performed during the import stage, although there is some ambiguity as a result. We observed that the number of proteins in Reactome is lower than the corresponding number in PID Reactome.

Table 8.5
The number of human pathways and biomolecules available in PID, Reactome, and BioCyc

	PID	PID Reactome	PID BioCarta	Reactome	BioCyc
Pathways	141	823	254	926	327
Physical entities	0	0	0	519	0
Proteins	2,527	2,795	2,392	2,410	2,239
Complexes	2,038	1,701	880	2361	24
Small molecules	132	624	205	737	1,262
DNA	0	0	0	0	0
RNA	6	23	14	31	0

For PID, the data imported from Reactome and BioCara are also provided. Only Reactome contains the data type "physical entity," which is a super class of biological molecules (proteins, complexes, etc). An example of such a physical entity is the Fatty Acid anion "head-in" in the human fatty acid cycling model. There is no physical entity in PID Reactome, which may suggest that the entities were removed or refined to other biological molecules.

Table 8.6
The number of interactions available in the same databases

	PID	PID Reactome	PID BioCarta	Reactome	BioCyc
Interaction	931	839	412	0	0
Physical interaction	0	0	0	0	0
Control	4,383	1,766	2,642	0	0
Conversion	0	0	0	0	0
Catalysis	0	0	0	1,716	3,155
Modulation	0	0	0	30	54
Complex assembly	1,447	909	646	0	0
Biochemical reaction	2,077	1,401	1,699	3,034	1,325
Transport	196	267	245	0	1
Transport with biochemical reaction	111	9	1	0	7

The type of interactions is based on the BioPax Level 2 specification. The data show that the level of information provided by various databases differs drastically. For example, PID has the tendency to represent the regulation and modulation events as control events. On the other hand, Reactome and BioCyc further refine the processes into catalysis or modulation events. PID also uses the most general class "Interaction" to represent molecular interactions; yet, such instances should not be created based on the BioPax specification.

Table 8.6 shows the diversity of interaction types used to describe pathways in different databases. For example, PID encodes all regulation or modification events as a control. However, Reactome and BioCyc further classify a control event into catalysis or a modulation event.

4. Conclusions and Outlook

While there are many pathway databases, even an idealized unified version of them is still far from being comprehensive. Most of the database providers are focused on a particular type of biological processes, reflecting the research interest and expertise of a specific group. The databases vary greatly in their content, quality, and completeness. Furthermore, the lack of resources limits the ability of most database providers to offer up-to-date pathway knowledge since the scientific literature to digest is very large and constantly accumulating. Currently, the information stored in pathway databases still falls behind the knowledge presented in scientific articles. An integrative approach seems to be a natural solution to the problems; yet, it is hindered by issues such as heterogeneous data models and lack of standardized data access methods. Various data exchange standards have been developed to assist the storage, organization, and exchange of pathway information. However, they are still in an early developmental stage. To overcome the issues mentioned above, a unified data model and data access method must be used to minimize the issues in data exchange. Furthermore, an automatic workflow for inferring and updating the pathway information from experimental data such as molecular interactions or gene expression data would greatly advance pathway analysis.

At the end, pathway databases will ultimately evolve into workbenches for modeling and predicting the influence of metabolic pathways, signaling and gene activity regulatory cascades on cellular

Table 8.7
List of platforms for cell reconstruction and modeling

Repository	Web URL
Virtual cell	http://www.nrcam.uchc.edu
E-cell	http://www.e-cell.org
Silicon cell	http://homepages.cwi.nl/~gollum/SiC
CyberCell	http://redpoll.pharmacy.ualberta.ca/CCDB
WebCell	http://webcell.org

reactions. As ambitious as this goal might sound and as little biological insight has come out of so-called *in silico* cell projects (*see* **Table 8.7** for examples) so far, this is, nevertheless, the final goal of biomolecular mechanism-focused life science research.

Referencess

1. Bader, G.. D., Cary, M.P., Sander, C. (2006) Pathguide: a pathway resource list. *Nucleic Acids Res* 34, D504–D506.

2. Ideker, T., Galitski, T., Hood, L. (2001) A new approach to decoding life: systems biology. *Annu Rev Genomics Hum Genet* 2, 343–372.

3. Wood, L. D., Parsons, D. W., Jones, S., Lin, J., Sjoblom, T., Leary, R. J., Shen, D., Boca, S. M., Barber, T., Ptak, J., et al. (2007) The genomic landscapes of human breast and colorectal cancers. *Science* 318, 1108–1113.

4. Vogelstein, B., Kinzler, K. W. (2004) Cancer genes and the pathways they control. *Nat Med* 10, 789–799.

5. 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.

6. Karp, P. D., Riley, M., Paley, S. M., Pelligrini-Toole, A. (1996) EcoCyc: an encyclopedia of *Escherichia coli* genes and metabolism. *Nucleic Acids Res* 24, 32–39.

7. Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T., et al. (2008) KEGG for linking genomes to life and the environment. *Nucleic Acids Res* 36, D480–D484.

8. Okuda, S., Yamada, T., Hamajima, M., Itoh, M., Katayama, T., Bork, P., Goto, S., Kanehisa, M. (2008) KEGG Atlas mapping for global analysis of metabolic pathways. *Nucleic Acids Res* 36, W423–W426.

9. Matthews, L., Gopinath, G., Gillespie, M., Caudy, M., Croft, D., de Bono, B., Garapati, P., Hemish, J., Hermjakob, H., Jassal, B., et al. (2009) Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Res* 37, D619–D622.

10. Joshi-Tope, G.., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de, B. B., Jassal, B., Gopinath, G.. R., Wu, G.. R., Matthews, L., et al. (2005) Reactome: a knowledgebase of biological pathways. *Nucleic Acids Res* 33, D428–D432.

11. Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., et al. (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 19, 524–531.

12. Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res* 13, 2498–2504.

13. Caspi, R., Foerster, H., Fulcher, C. A., Kaipa, P., Krummenacker, M., Latendresse, M., Paley, S., Rhee, S. Y., Shearer, A. G., Tissier, C., et al. (2008) The MetaCyc Database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. *Nucleic Acids Res* 36, D623–D631.

14. Keseler, I. M., Bonavides-Martinez, C., Collado-Vides, J., Gama-Castro, S., Gunsalus, R. P., Johnson, D. A., Krummenacker, M., Nolan, L. M., Paley, S., Paulsen, I. T., et al. (2009) EcoCyc: a comprehensive view of Escherichia coli biology. *Nucleic Acids Res* 37, D464–D470.

15. Karp, P. D., Paley, S., Romero, P. (2002) The Pathway Tools software. *Bioinformatics* 18 Suppl 1, S225–S232.

16. Lomax, J. (2005) Get ready to GO! A biologist's guide to the Gene Ontology. *Brief Bioinform* 6, 298–304.

17. Schaefer, C. F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., Buetow, K. H. (2009) PID: the Pathway Interaction Database. *Nucleic Acids Res* 37, D674–D679.

18. Adriaens, M. E., Jaillard, M., Waagmeester, A., Coort, S. L., Pico, A. R., Evelo, C. T. (2008) The public road to high-quality curated biological pathways. *Drug Discov Today* 13, 856–862.

19. Dahlquist, K. D., Salomonis, N., Vranizan, K., Lawlor, S. C., Conklin, B. R. (2002) GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. *Nat Genet* 31, 19–20.

Chapter 9

Databases of Protein–Protein Interactions and Complexes

Hong Sain Ooi, Georg Schneider, Ying-Leong Chan, Teng-Ting Lim, Birgit Eisenhaber, and Frank Eisenhaber

Abstract

In the current understanding, translation of genomic sequences into proteins is the most important path for realization of genome information. In exercising their intended function, proteins work together through various forms of direct (physical) or indirect interaction mechanisms. For a variety of basic functions, many proteins form a large complex representing a molecular machine or a macromolecular super-structural building block. After several high-throughput techniques for detection of protein–protein interactions had matured, protein interaction data became available in a large scale and curated databases for protein–protein interactions (PPIs) are a new necessity for efficient research. Here, their scope, annotation quality, and retrieval tools are reviewed. In addition, attention is paid to portals that provide unified access to a variety of such databases with added annotation value.

Key words: protein–protein interaction, protein-complex database, PPI database.

1. Introduction

Protein–protein interactions (PPIs) are a critical attribute of most cellular processes. Protein interactions can either be direct (physical) via the formation of an interaction complex (with varying affinity of interaction and duration of complex formation) or they can be indirect (just functional) via a variety of genetic dependencies, transcriptional regulation mechanisms, or biochemical pathways. Traditionally, instances of PPI have been studied by genetic, biophysical, and biochemical techniques. Until less than a decade ago, their experimental detection was cumbersome; the cost of such a laborious effort restricted the number of known complexes and the main information source about PPI was scientific journal articles that, typically, described one or a handful of interactions only.

The first high-throughput PPI detection technology was provided with the yeast two-hybrid technology (1) followed by several others, among which tag-based mass-spectrometric techniques (2) have recently become the state of the art. Other major sources are correlated expression profiles (3, 4) and genetic interaction data (5) (e.g., on synthetic lethality) but theoretical, in silico computed approaches based on interaction predictions from gene context studies (gene fusion events (6–9), gene neighborhood (10–13), and gene co-occurrences/absences, also called the method of phylogenetic profiles (14–18)) increasingly contribute to our understanding of protein networks.

It is evident that, at present, we know only a fraction of the interaction network in cellular systems (and, of course, only in a qualitative manner). Nevertheless, the sheer size of the available data about interactions requires their collection in electronically readable databases. Currently, there are a number of competing database projects that vary in their scope, annotation quality, and availability to the public. Some of these databases are ambitious projects that try to collect all possible known interactions between proteins of every organism. The Biomolecular Interaction Network Database (BIND) (19, 20) (it was recently renamed Biomolecular Object Network Databank – BOND – and commercialized) is one of the most comprehensive databases of protein–protein interactions and complexes. Among its many features, it not only has an interactive Web portal for searching and browsing through the records, but also provides standardized application interfaces (APIs) for various computer languages like Perl, Java, C, and C++ to allow another avenue to access its data. Other databases on the other hand can be specific for certain diseases or organisms only. For example, NCBI's HIV-1, Human Protein Interaction Database (http://www.ncbi.nlm.nih.gov/projects/RefSeq/HIVInteractions) attempts to collect all known interacting proteins between the various HIV-1 viral proteins and human proteins. Such databases are very specific and therefore, usually contain less data and have less functionality than the general interaction databases.

Protein interaction databases, in turn, will become useful only with respective retrieval tools and, most importantly, with their integration into annotation pipelines that enables them to become means for discovery of new biomolecular mechanisms. For example, there is a recent publication that describes the use of information about protein complexes in yeast to predict the phenotypic effect of gene mutation (21) and that this approach can possibly be extended to predicting and investigating the genes of Mendelian or complex diseases. We need to admit that, at this front, there are still many open issues and the qualitative change in biological theory aimed at more system biological understanding is still a matter of the future.

2. Recent Status of Protein–Protein Interaction and Complex Databases

Entry items in PPI databases are interactions or complexes. A protein–protein interaction usually refers to a binary relationship between one protein and another. On the other hand, protein complexes consist of several subunits and, thus, refer to a set of proteins. Each protein pair in this set forms an interaction and some pairs even interact physically in a direct manner. More generally, a protein complex can be viewed as a special case of a set of proteins with a common functional description. Other examples are the set of proteins in a pathway or the set of coexpressed targets under specific biological conditions. The amount of interactions measured with a specific method depends on the degree of interaction (e.g., its affinity) and the duration of this interaction. The duration of the interaction may be long term and with high affinity (so that the complex can survive the harsh purification procedures); it may also be rather transient as in enzyme substrate complexes.

In the following section, we mention the most important sources of PPIs currently available. We classify the protein–protein interaction databases into three main categories, based on the methods used to collect or generate the data. A majority of these databases are repositories of experimental data, which were collected either through manual curation, computational extraction, or direct deposit by the authors, such as DIP (22), MINT (23), and IntAct (24). The second type of databases stores predicted protein–protein interactions. Examples of these are PIPs (25), OPHID (26), and HomoMINT (27). Finally, the last category is a portal that provides unified access to a variety of protein interaction databases. The most advanced example of this category is STRING (28, 29). A comparison of primary databases for PPIs is provided by Mathivanan et al. (30). There are also databases for PPI in bacteria (31, 32). For a more complete list of protein–protein interaction databases, readers can refer to Pathguide (33), which contains information about 290 biological pathway and interaction resources.

2.1. Database of Interacting Proteins (DIP)

The main aim of DIP (22) is to provide the scientific community with a single, user-friendly online database by integrating the existing experimentally determined protein–protein interactions from various sources. It mainly records binary protein–protein interactions that were manually curated by experts. In recent years, DIP has been extended to include interactions between protein ligands and protein receptors (DLRP) (34). The database is consistently updated and the interaction data together with the protein sequences can be downloaded in several formats including tab-delimited and PSI-MI (35–37).

Access to DIP requires registration and is free for academic users. An extensive help page and a search guide are provided. A search for proteins can be performed in a number of ways such as by node identifiers (a node is a protein in DIP), descriptions, keywords, BLAST query of a protein sequence, sequence motifs, or literature articles. The search returns a list of proteins that matched the search criteria. The "Links" field lists all the interactions of a particular protein. The link under the "Interaction" field provides the experimental evidence and the corresponding publication support for the interaction. The detailed description of a protein is also given and can be viewed by selecting the "Interactor(s)" field. The "graph" link opens the interaction map for the current protein. To provide a reasonable visualization, only nodes up to two edges from the root node are displayed. The width of the edges reflects the number of independent experiments supporting this interaction and is useful to identify highly confident interactions. The interaction maps generated have links to all nodes and this allows navigation from one protein to another.

2.2. Molecular INTeraction database (MINT)

The Molecular INTeraction database (MINT) (23) is an endeavor to document experimentally verified protein–protein interactions, which are mined from the scientific literature by expert curators. While the main focus of the team is on protein–protein interactions, other interaction data such as enzymatic modifications of the interacting partners are also recorded. Although most of the interactions come from high-throughput experiments, the main value of MINT resides in the high number of curated articles. The data can be freely downloaded and are available in several formats.

The search interface presents several query options. The user can retrieve the list of proteins from an article based on PUBMED ID or authors. The query might also be based on protein or gene names, protein accession numbers, keywords and limited to specific data sets (all taxa, mammalian, yeast, worm, fly, or viruses). Finally, a BLAST search can be performed to find proteins which are homologous to the query protein. The search returns a list of proteins with information such as a brief description of protein function, Uniprot AC, taxonomy, and domains. The detailed page of the protein shows a summary of the protein features in the left panel while the set of the interacting partners of the query protein is given in the right panel. The type of evidence support from the literature is also specified together with their respective scores. The interaction network is visualized with the MINT Viewer. The viewer provides advanced features such as filtering the network based on scores as well as expanding and collapsing network sections. The result can be exported in several formats, for example, flat file, Osprey (38), and PSI-MI (36).

2.3. IntAct

IntAct (24), by itself, is an open source database and software framework. The system provides a flexible data model which can accommodate a high level of experimental details. It also contains a suite of tools that can be used to visualize and analyze the interaction data. The interaction data are manually extracted from public literature and annotated to a high level of detail through the extensive use of controlled vocabulary. Most of the interaction data come from protein–protein interactions, but IntAct also captures nonprotein molecular interactors such as DNA, RNA, and small molecules. IntAct is updated weekly and can be downloaded in the PSI-MI format (36). Both the IntAct software Rintact (39) and the data are freely available to all users.

A simple, yet flexible search engine is provided. Users can search for a broad range of identifiers, accession numbers, names, and aliases. The search results may also be filtered with criteria such as publication ID, first author, experiment type, and interaction type. The search result is displayed in tabular form for easy browsing and can be downloaded in the PSI-MI format. To visualize the interactions for a particular protein, the link with IntAct accession instead of that of the Uniprot accession has to be selected. The new page displays basic information about the selected protein and a number of interactions involving the current protein. Then, one can select the protein and click on the "Graph" link. The interactive viewer provides a number of unique features such as highlighting the node based on the molecule type, Gene Ontology (40), InterPro (41) annotation, experimental and biological role or species. Similar to the MINT viewer, the interaction network can also be expanded or refocused to a new protein. The result of the navigation can be immediately exported in PSI-MI format (36).

2.4. BioGRID

The Biological General Repository for Interaction Data sets (Bio-GRID) (42, 43) is an effort developed to collect both protein and genetic interactions from major model organisms. BioGRID provides the most up-to-date and virtually complete set of interaction data reported in the published literatures for both the budding yeast *Saccharomyces cerevisiae* and the fission yeast *Schizosaccharomyces pombe* (42). The database contains data from both high-throughput and conventional studies. It is updated monthly. The data can be downloaded freely in several formats such as PSI-MI (36), tab-delimited, and Osprey. The data can also be downloaded ordered by gene, publication, organism, or experimental system.

A search can be performed with a wide variety of identifiers, for example, cDNA accession and GI numbers as well as with Ensembl, Entrez gene and Uniprot accessions (see their Help page for full descriptions). The result page contains a list of matched items with and without associations. The description page of a selected protein shows the standard annotations, links to external databases, Gene

Ontology, the number of both protein and genetic interactions. Subsequently, a list of interacting partners is displayed and so are the experimental support and the corresponding publications. The interaction type can be recognized via the color code of the experiments. It is possible to download the data for each interaction or publication in the supported formats. Currently, no visualization is available; however, Osprey can be used (38).

2.5. Human Protein Reference Database (HPRD)

The main purpose of HPRD (44) is to build a complete catalogue of human proteins pertaining to health and disease. While HPRD is not a protein–protein interaction database, it contains an extensive list of interaction data of human proteins. All data in HPRD are manually extracted from public literature and curated by a team of trained biologists. The data are freely available for academic users and can be downloaded in either tab-delimited or XML formats. Users can download the whole database or only protein–protein interaction data without annotations in a tab-delimited or PSI-MI format (36).

The database can be searched by keywords or by sequences. The "Query" page provides various keyword fields; these include protein names, accession numbers, gene symbols, chromosome locations, molecular classes, domains or motifs, and posttranslational modifications. The "Browse" page organizes the list of proteins into different categories for easy browsing. It is a unique feature of HPRD that the annotations for a particular protein are organized in tabs. The "Interactions" tab provides the list of protein interactors together with the experiment type. Nonprotein interactors are also listed on the same page. No interaction visualizer is provided. The "Pathways" tab leads to the corresponding protein entry in NetPath (www.nethpath.org), which contains a number of immune- and cancer-signaling pathways. From Net-Path, users can download the corresponding pathway in popular file formats.

2.6. MPact

MPact (45) is an organism-specific database focusing on manually curated protein–protein interactions and complexes from *S. cerevisiae* and acts as an access point to PPI resources available in CYGD (46). As the database is part of CYGD, the rich set of information in CYGD is directly accessible from MPact. Due to its quality, the data set has been used in numerous studies and is widely considered as a gold standard for yeast protein–protein interactions (47–49). The latest version of data is available for download in PSI-MI format (36).

A "Quick Search" box is provided for quick access to the interaction data by protein ID and gene name. More specific queries can be performed by using the "Query by Protein" search page. Here, protein attributes such as names or aliases, functional categories, cellular localization, and EC numbers can be specified.

Additional criteria such as evidence and interaction type, publication ID, and an option to exclude high-throughput experiments are available. This feature is useful to select interaction data based on the strength of the detection methods. The results can be viewed in two formats. In the short format, only protein ID, gene name, a simple description, and the link to CYGD are listed. The long format provides additional information such as the type of experimental evidence, the publication ID, the full function description, and the type of the interaction. The search result can be downloaded in PSI-MI format (36). A simple visualizer is available for illustrating the interaction network. The nodes are colored based on functional categories and the color of the edges reflects the level of supporting evidence for the corresponding interactions. The network can also be downloaded in PDF format for offline use.

2.7. STRING

STRING was first introduced in the year 2000 (50) and evolved from a Web server of predicted functional association between proteins into a comprehensive Web portal of protein–protein interactions (28, 51). It integrates data from numerous sources, not only from experimental repositories, but also includes computational prediction methods, and automated text mining of public text collections such as PUBMED. To facilitate the integration of multiple data sets, the interactions are mapped onto a consistent set of proteins and identifiers. During the integration, isoforms are reduced to a single representative protein sequence. While this approach enables unique comparison and efficient storage, the interaction information may lead to misinterpretation of the result in later stages as some interactions only occur for a particular isoform of the protein. While STRING data can be freely downloaded mostly in flat file or as a database dump, the complete data set is only available under a license agreement, which is free for academic users.

The interaction networks can be searched by protein names and accessions and a variety of accession types is supported. The search returns a list of proteins that match the term and the user can select the best candidate. A similar search can also be performed using the protein sequence with the best-matched protein selected automatically. STRING provides a powerful network visualizer together with a rich set of annotations. Several visualization tools are available for analysis and facilitate navigation within the interaction network. Users are encouraged to refer to the online help page for more information. STRING also provides a search interface for querying the interaction network with a protein list that tries to connect all or most of them via interactions in the STRING database.

2.8. Unified Human Interactome (UniHI)

Unified Human Interactome (UniHI) (52) provides unified access to human protein interaction data from various sources including both computational and experimental repositories. The aim of UniHI is to be the most comprehensive platform to

study the human interactome. Currently, it contains interaction data extracted from six public experimental repositories, and large-scale Y2H screenings, and computational extraction through text-mining and orthologue transfer. The integration of proteins was performed using information from Ensmart (PMID: 14707178) and HGNC (PMID: 11810281).

Users can query UniHI using the UniHI search tool. A variety of protein identifiers are supported and users can submit a set of proteins to obtain their functional information and interacting partners. The search returns a list of matched proteins together with the original source database names. UniHI provides an interactive viewer to visualize the interaction networks. This software offers several options to refine the network. In addition, UniHI provides two powerful tools to analyze the human interactome. The first one is UniHI Express, which can be used to identify tissue-specific interaction networks. Users can refine the interaction networks based on gene expression in selected tissues to construct a tissue-specific network. While an interaction network is of great help, pathway information provides more detail about the information flow in the biological process. Thus, it is useful to compare an interaction network with a known pathway. This can be done with the help of UniHI Scanner, the second tool. UniHI Scanner compares the extracted networks with the pathways from KEGG (53) and, thus, enables the detection of new components in existing pathways. It also allows identifying proteins that are involved in multiple pathways, which might be useful for disease-related studies. UniHI provides detailed help pages about the available tools. Users are strongly recommended to read the documentation before starting their search.

3. Comparison of Protein–Protein Interaction and Complex Databases

Storage of protein–protein interaction and complex data including information about experimental and theoretical evidence for the interaction and functional annotations of the proteins involved requires a complex data structure. Standardization is a necessary requirement to allow electronic access to various data sources by programs and exchange of data sets among research teams. There are currently two major data formats available to represent protein–protein interaction and protein complex data. The first format known as the Proteomics Standard Initiative-Molecular Interactions (PSI-MI) (36) is clearly advocated and has been adopted by many major existing databases. This XML-based format allows the user to further analyze the data with existing tools [e.g., Cytoscape (54)] as these tools are usually PSI-MI compatible. BioPAX, an acronym for Biological Pathways Exchange, is an

alternative, concurrently used format. It is defined in terms of OWL-XML (http://www.w3.org/TR/owl-ref/) and the current version (55) includes definitions for molecular interactions. The relative merits of various standards have been reviewed (37, 56). It should be noted that, although databases might use the same format, the level of information provided (especially of the annotation) might differ considerably.

The amount and completeness of the data in PPI databases are of major concern. As can be seen in **Tables 9.1** and **9.2**, there is a considerable overlap between the major general interaction databases with regard to the representation of interacting proteins as well as with regard to interactions described; yet, any applied effort for network creation in context with a biological research task needs to draw information from essentially all primary databases if one wishes not to omit essential interactions. Not surprisingly, some fusion of primary databases is the first benefit that is provided by portals such as STRING (28) or UNIHI (52) for access to interactions. There is no overlap between MPact (45), a yeast database, with the resources for human protein interactions, GNP (http://genome network.nig.ac.jp/public/sys/gnppub/portal.do) and HPRD (44).

Our in-house effort to integrate the interaction data from multiple databases using the PSI-MI format (36) showed several further issues exist. First, not all proteins can be associated with unique UniProt entries (*see* **Table 9.2**), mostly in context with obsolete entries and identifiers that can be mapped to multiple entries. While other identifiers can be introduced, the rich set of annotation provided by the UniProt Knowledgebase cannot be used at a later stage. Second, not all information is presented in the same level of detail. For example, some experiments are annotated using the term "experimental interaction detection" instead of describing the real experiment method. This not only hinders the analysis based on experiment type, it also produces wrong statistics. A more severe issue occurs when the same laxness is applied to the taxonomy. For example, proteins can be assigned to mammals instead of to human.

Finally, the quality of the data with regard to the significance of the interaction in the biological context is problematic. PPI databases contain lots of interactions of proteins with chaperones, ribosomal proteins, and other similarly sticky proteins, interactions that are not informative about biological functions. It can be expected that the experimental conditions for interaction registration have created a substantial number of interactions that will not occur under physiological conditions. It is difficult to assess this fraction of data. The quality of high-throughput experiments has been long criticized with high false-positive rates of 50–70% (57). While the rate is high, there is a belief that the experiments might produce the correct physical interaction data, even though the interactions might not be biological meaningful. Von Mering et al. (29, 48) and others (58–60) have also provided some criteria for the

Table 9.1
Overlap among proteins described as interactors in protein interaction databases

	IntAct	MINT	BioGRID	DIP	HPRD	MPact	GNP	MPPI
IntAct	38,085							
MINT	21,669	27,109						
BioGRID	16,578	15,323	26,015					
DIP	15,188	15,283	13,335	18,808				
HPRD	5,526	4,759	3,773	1,057	9,496			
MPact	4,629	4,567	4,328	4,528	0	4,787		
GNP	663	591	366	151	771	0	1,014	
MPPI	564	542	219	256	432	0	81	863
IntAct	38,085							
MINT	79.93%	27,109						
BioGRID	63.72%	58.90%	26,015					
DIP	80.75%	81.26%	70.90%	18,808				
HPRD	58.19%	50.12%	39.73%	11.13%	9,496			
MPact	96.70%	95.40%	90.41%	94.59%	0.00%	4,787		
GNP	65.38%	58.28%	36.09%	14.89%	76.04%	0.00%	1,014	
MPPI	65.35%	62.80%	25.38%	29.66%	50.06%	0.00%	9.39%	863
Total	40,010	28,322	26,668	19,795	9,550	4,835	1,071	926
With Uniprot ID	38,085	27,109	26,015	18,808	9,496	4,787	1,014	863
With Uniprot ID (%)	95.19	95.72	97.55	95.01	99.43	99.01	94.68	93.20
With sequence	39,705	27,993	26,015	18,863	9,550	4,787	1,014	863
With sequence (%)	99.24	98.84	97.55	95.29	100.00	99.01	94.68	93.20

This table provides information about the sets of proteins described as interactors in interactions (as of January 2009) enlisted in the respective databases IntAct (24), MINT (23), BioGRID (42,43), DIP (22), HPRD (44), MPact (45), GNP (http://genomenetwork.nig.ac.jp/public/sys/gnppub/portal.do), and MPPI (69). In the upper part, the absolute overlap among proteins in the database is listed (the diagonal shows the number of proteins with Uniprot ID in each database). The middle section shows the overlap as a percentage of the total protein number of the database in each column. The bottom part of the table provides information about the total number of protein entries, the absolute and relative (in %) numbers of proteins with Uniprot IDs and with explicit sequence information.

assessment of this part of the data and they arrive at similar fractions of most primary interaction data sets that are apparently spurious. The lower accuracy is associated with mRNA synexpression and the better value is given to mass-spectrometric methods of tag-purified

Table 9.2
Overlap among binary interactions described in protein interaction databases

	INTACT	MINT	BIOGRID	DIP	HPRD	MPACT	GNP	MPPI
INTACT	82,712							
MINT	36,796	68,071						
BIOGRID	22,652	32,312	138,383					
DIP	23,111	30,336	30,161	49,730				
HPRD	8,078	7,102	5,615	794	36,899			
MPACT	5,095	5,621	6,983	6,027	0	12,207		
GNP	13	20	18	7	67	0	1,292	
MPPI	82	106	64	30	299	0	0	833
INTACT	82,712							
MINT	54.06%	68,071						
BIOGRID	16.37%	23.35%	138,383					
DIP	46.47%	61.00%	60.65%	49,730				
HPRD	21.89%	19.25%	15.22%	2.15%	36,899			
MPACT	41.74%	46.05%	57.20%	49.37%	0.00%	12,207		
GNP	1.01%	1.55%	1.39%	0.54%	5.19%	0.00%	1,292	
MPPI	9.84%	12.73%	7.68%	3.60%	35.89%	0.00%	0.00%	833

This table provides information about the sets of binary interactions (as of January 2009) enlisted in the respective databases IntAct (24), MINT (23), BioGRID (42, 43), DIP (22), HPRD (44), MPact (45), GNP (http://genomenetwork.nig.ac.jp/public/sys/gnppub/portal.do), and MPPI (69). In the upper part, the absolute overlap among proteins in the database is listed (the diagonal shows the number of proteins in each database). The bottom section shows the overlap as a percentage of the total protein number of the database in each column.

complexes (61). The accuracy can be enhanced by combining methods; yet, the coverage does go down hand in hand with the improvement of reliability. On the other hand, it remains unclear which segments (domains) of the two proteins really interact with each other.

4. Conclusions and Future Developments

The analysis of interaction data from the biological viewpoint is ongoing and appears to be most advanced in yeast and human. For

example, Schwikowski et al. (62) did a global analysis of the interactions in yeast and registered 2,358 interactions among 1,548 proteins. They found that 63% of the interactions occurred in proteins with common functionality and 76% in proteins with common subcellular localization. Fraser and Plotkin (21) managed to use functional genomic data from the yeast *S. cerevisia* to show that it is possible to best predict the phenotype of a protein knockout by using the phenotype of the knockout of other proteins that form complexes with it. This result can be explained by considering protein complexes as a form of functional interaction that is especially tightly knit. Jensen et al. (63, 64) showed that by comparing large-scale datasets of protein complexes, the periodicity of expression of the regulated subunits of each protein complex differs greatly between organisms (63, 64). The data in protein–protein interaction databases can be used to build a macromolecular biological network by combining the known interactions with pathway information. These networks can then be used to predict and study possible novel signaling or metabolic pathways within the organism (60). These networks can also be systematically analyzed to aid the interpretation of high-throughput experimental data for the purpose of identifying, validating, and prioritizing potential drug targets (65–67).

Despite all efforts, the availability of protein interaction data has, so far, not had a great impact on biological theory and did not create the desired system-wide understanding. Possibly, we just know too small a part of the total network and it might also be required that a higher resolution is necessary with regard to the quality of the interactions (activation/inactivation, etc.). Noort et al. (68) proposed a method to combine all sources of evidence for adding quality labels to protein–protein interactions for *S. cerevisiae* and they use this information to predict the nature (metabolic or physical interactions) of newly discovered protein–protein interactions.

References

1. Uetz, P., Giot, L., Cagney, G., Mansfield, T. A., Judson, R. S., Knight, J. R., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P., et al. (2000) A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature* 403, 623–627.

2. Yu, H., Braun, P., Yildirim, M. A., Lemmens, I., Venkatesan, K., Sahalie, J., Hirozane-Kishikawa, T., Gebreab, F., Li, N., Simonis, N., et al. (2008) High-quality binary protein interaction map of the yeast interactome network. *Science* 322, 104–110.

3. Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., Bennett, H. A., Coffey, E., Dai, H., He, Y. D., et al. (2000) Functional discovery via a compendium of expression profiles. *Cell* 102, 109–126.

4. Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., et al. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell* 2, 65–73.

5. Tong, A. H., Evangelista, M., Parsons, A. B., Xu, H., Bader, G. D., Page, N., Robinson, M., Raghibizadeh, S., Hogue, C. W., Bussey, H., et al. (2001) Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science* 294, 2364–2368.

6. Marcotte, E. M., Pellegrini, M., Ng, H. L., Rice, D. W., Yeates, T. O., Eisenberg, D. (1999) Detecting protein function and protein-protein interactions from genome sequences. *Science* 285, 751–753.

7. Date, S. V., Marcotte, E. M. (2003) Discovery of uncharacterized cellular systems by genome-wide analysis of functional linkages. *Nat Biotechnol* 21, 1055–1062.

8. Enright, A. J., Iliopoulos, I., Kyrpides, N. C., Ouzounis, C. A. (1999) Protein interaction maps for complete genomes based on gene fusion events. *Nature* 402, 86–90.

9. Kamburov, A., Goldovsky, L., Freilich, S., Kapazoglou, A., Kunin, V., Enright, A. J., Tsaftaris, A., Ouzounis, C. A. (2007) Denoising inferred functional association networks obtained by gene fusion analysis. *BMC Genomics* 8, 460.

10. Dandekar, T., Snel, B., Huynen, M., Bork, P. (1998) Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem Sci* 23, 324–328.

11. Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G. D., Maltsev, N. (1999) The use of gene clusters to infer functional coupling. *Proc Natl Acad Sci USA* 96, 2896–2901.

12. Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G. D., Maltsev, N. (1999) Use of contiguity on the chromosome to predict functional coupling. *In Silico Biol* 1, 93–108.

13. Korbel, J. O., Jensen, L. J., von, M. C., Bork, P. (2004) Analysis of genomic context: prediction of functional associations from conserved bidirectionally transcribed gene pairs. *Nat Biotechnol* 22, 911–917.

14. Makarova, K. S., Koonin, E. V. (2003) Filling a gap in the central metabolism of archaea: prediction of a novel aconitase by comparative-genomic analysis. *FEMS Microbiol Lett* 227, 17–23.

15. Pellegrini, M., Marcotte, E. M., Thompson, M. J., Eisenberg, D. Yeates, T. O. (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci USA* 96, 4285–4288.

16. Sato, T., Yamanishi, Y., Kanehisa, M., Toh, H. (2005) The inference of protein-protein interactions by co-evolutionary analysis is improved by excluding the information about the phylogenetic relationships. *Bioinformatics.* 21, 3482–3489.

17. Sato, T., Yamanishi, Y., Horimoto, K., Kanehisa, M., Toh, H. (2006) Partial correlation coefficient between distance matrices as a new indicator of protein-protein interactions. *Bioinformatics* 22, 2488–2492.

18. Morett, E., Korbel, J. O., Rajan, E., Saab-Rincon, G., Olvera, L., Olvera, M., Schmidt, S., Snel, B., Bork, P. (2003) Systematic discovery of analogous enzymes in thiamin biosynthesis. *Nat Biotechnol* 21, 790–795.

19. Bader, G. D., Betel, D., Hogue, C. W. (2003) BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Res* 31, 248–250.

20. Bader, G. D. and Hogue, C. W. (2000) BIND – a data specification for storing and describing biomolecular interactions, molecular complexes and pathways. *Bioinformatics* 16, 465–477.

21. Fraser, H. B., Plotkin, J. B. (2007) Using protein complexes to predict phenotypic effects of gene mutation. *Genome Biol* 8, R252.

22. Xenarios, I., Salwinski, L., Duan, X. J., Higney, P., Kim, S. M., 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.

23. Zanzoni, A., Montecchi-Palazzi, L., Quondam, M., Ausiello, G., Helmer-Citterich, M., Cesareni, G. (2002) MINT: a Molecular INTeraction database. *FEBS Lett* 513, 135–140.

24. Kerrien, S., am-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., Dimmer, E., Feuermann, M., Friedrichsen, A., Huntley, R., et al. (2007) IntAct – open source resource for molecular interaction data. *Nucleic Acids Res* 35, D561–D565.

25. McDowall, M. D., Scott, M. S., Barton, G. J. (2009) PIPs: human protein-protein interaction prediction database. *Nucleic Acids Res* 37, D651–D656.

26. Brown, K. R., Jurisica, I. (2005) Online predicted human interaction database. *Bioinformatics* 21, 2076–2082.

27. Persico, M., Ceol, A., Gavrila, C., Hoffmann, R., Florio, A., Cesareni, G. (2005) HomoMINT: an inferred human network based on orthology mapping of protein interactions discovered in model organisms. *BMC Bioinformatics* 6(Suppl 4), S21.

28. Jensen, L. J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., et al. (2009) STRING 8 – a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res* 37, D412–D416.

29. von Mering, C., Huynen, M., Jaeggi, D., Schmidt, S., Bork, P., Snel, B. (2003) STRING: a database of predicted functional associations between proteins. *Nucleic Acids Res* 31, 258–261.

30. Mathivanan, S., Periaswamy, B., Gandhi, T. K., Kandasamy, K., Suresh, S., Mohmood, R., Ramachandra, Y. L., Pandey, A. (2006) An evaluation of human protein-protein interaction data in the public domain. *BMC Bioinformatics* 7(Suppl 5), S19.

31. Noirot, P., Noirot-Gros, M. F. (2004) Protein interaction networks in bacteria. *Curr Opin Microbiol* 7, 505–512.

32. Su, C., Peregrin-Alvarez, J. M., Butland, G., Phanse, S., Fong, V., Emili, A., Parkinson, J. (2008) Bacteriome.org – an integrated protein interaction database for *E. coli. Nucleic Acids Res* 36, D632–D636.

33. Bader, G. D., Cary, M. P., Sander, C. (2006) Pathguide: a pathway resource list. *Nucleic Acids Res* 34, D504–D506.

34. Graeber, T. G., Eisenberg, D. (2001) Bioinformatic identification of potential autocrine signaling loops in cancers from gene expression profiles. *Nat Genet* 29, 295–300.

35. Hermjakob, H., Montecchi-Palazzi, L., Bader, G., Wojcik, J., Salwinski, L., Ceol, A., Moore, S., Orchard, S., Sarkans, U., von Mering, C., et al. (2004) The HUPO PSI's molecular interaction format – a community standard for the representation of protein interaction data. *Nat Biotechnol* 22, 177–183.

36. Kerrien, S., Orchard, S., Montecchi-Palazzi, L., Aranda, B., Quinn, A. F., Vinod, N., Bader, G. D., Xenarios, I., Wojcik, J., Sherman, D., et al. (2007) Broadening the horizon – level 2.5 of the HUPO-PSI format for molecular interactions. *BMC Biol* 5, 44.

37. Stromback, L., Lambrix, P. (2005) Representations of molecular pathways: an evaluation of SBML, PSI MI and BioPAX. *Bioinformatics* 21, 4401–4407.

38. Breitkreutz, B. J., Stark, C., Tyers, M. (2003) Osprey: a network visualization system. *Genome Biol* 4, R22.

39. Chiang, T., Li, N., Orchard, S., Kerrien, S., Hermjakob, H., Gentleman, R., Huber, W. (2008) Rintact: enabling computational analysis of molecular interaction data from the IntAct repository. *Bioinformatics* 24, 1100–1101.

40. Lomax, J. (2005) Get ready to GO! A biologist's guide to the Gene Ontology. *Brief Bioinformatics* 6, 298–304.

41. Hunter, S., Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Das, U., Daugherty, L., Duquenne, L., et al. (2009) InterPro: the integrative protein signature database. *Nucleic Acids Res* 37, D211–D215.

42. Breitkreutz, B. J., Stark, C., Reguly, T., Boucher, L., Breitkreutz, A., Livstone, M., Oughtred, R., Lackner, D. H., Bahler, J., Wood, V., et al. (2008) The BioGRID Interaction Database: 2008 update. *Nucleic Acids Res* 36, D637–D640.

43. Stark, C., Breitkreutz, B. J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M. (2006) BioGRID: a general repository for interaction datasets. *Nucleic Acids Res* 34, D535–D539.

44. Keshava Prasad, T. S., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., Raju, R., Shafreen, B., Venugopal, A., et al. (2009) Human Protein Reference Database – 2009 update. *Nucleic Acids Res* 37, D767–D772.

45. Guldener, U., Munsterkotter, M., Oesterheld, M., Pagel, P., Ruepp, A., Mewes, H. W. and Stumpflen, V. (2006) MPact: the MIPS protein interaction resource on yeast. *Nucleic Acids Res* 34, D436–D441.

46. Guldener, U., Munsterkotter, M., Kastenmuller, G., Strack, N., van Helden, J., Lemer, C., Richelles, J., Wodak, S. J., Garcia-Martenez, J., Perez-Ortin, J. E., et al. (2005) CYGD: the Comprehensive Yeast Genome Database. *Nucleic Acids Res* 33, D364–D368.

47. Wuchty, S. (2004) Evolution and topology in the yeast protein interaction network. *Genome Res* 14, 1310–1314.

48. von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., Bork, P. (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature* 417, 399–403.

49. Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N. J., Chung, S., Emili, A., Snyder, M., Greenblatt, J. F., Gerstein, M. (2003) A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science* 302, 449–453.

50. Snel, B., Lehmann, G., Bork, P., Huynen, M. A. (2000) STRING: a web-server to retrieve and display the repeatedly occurring

neighbourhood of a gene. *Nucleic Acids Res* 28, 3442–3444.

51. von Mering, C., Jensen, L. J., Kuhn, M., Chaffron, S., Doerks, T., Kruger, B., Snel, B., Bork, P. (2007) STRING 7 – recent developments in the integration and prediction of protein interactions. *Nucleic Acids Res* 35, D358–D362.

52. Chaurasia, G., Malhotra, S., Russ, J., Schnoegl, S., Hanig, C., Wanker, E. E., Futschik, M. E. (2009) UniHI 4: new tools for query, analysis and visualization of the human protein-protein interactome. *Nucleic Acids Res* 37, D657–D660.

53. Okuda, S., Yamada, T., Hamajima, M., Itoh, M., Katayama, T., Bork, P., Goto, S., Kanehisa, M. (2008) KEGG Atlas mapping for global analysis of metabolic pathways. *Nucleic Acids Res* 36, W423–W426, PMID: 18077471.

54. Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T. (2003) Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res* 13, 2498–2504.

55. Jiang, K., Nash, C. (2006) Application of XML database technology to biological pathway datasets. *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society IEEE Engineering in Medicine and Biology Society Conference* 1, 4217–4220.

56. Cerami, E. G., Bader, G. D., Gross, B. E., Sander, C. (2006) cPath: open source software for collecting, storing, and querying biological pathways. *BMC Bioinformatics* 7, 497.

57. Hart, G. T., Ramani, A. K., Marcotte, E. M. (2006) How complete are current yeast and human protein-interaction networks? *Genome Biol* 7, 120.

58. Chiang, T., Scholtens, D., Sarkar, D., Gentleman, R., Huber, W. (2007) Coverage and error models of protein-protein interaction data by directed graph analysis. *Genome Biol* 8, R186.

59. Gentleman, R., Huber, W. (2007) Making the most of high-throughput protein-interaction data. *Genome Biol* 8, 112.

60. Thorne, T., Stumpf, M. P. (2007) Generating confidence intervals on biological networks. *BMC Bioinformatics* 8, 467.

61. Gavin, A. C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., Rau, C., Jensen, L. J., Bastuck, S., Dumpelfeld, B., et al. (2006) Proteome survey reveals modularity of the yeast cell machinery. *Nature* 440, 631–636.

62. Schwikowski, B., Uetz, P., Fields, S. (2000) A network of protein-protein interactions in yeast. *Nat Biotechnol* 18, 1257–1261.

63. Jensen, L. J., Jensen, T. S., de, L. U., Brunak, S., Bork, P. (2006) Co-evolution of transcriptional and post-translational cell-cycle regulation. *Nature* 443, 594–597.

64. Jensen, L. J., de, L. U., Jensen, T. S., Brunak, S., Bork, P. (2008) Circular reasoning rather than cyclic expression. *Genome Biol* 9, 403.

65. Nikolsky, Y., Ekins, S., Nikolskaya, T., Bugrim, A. (2005) A novel method for generation of signature networks as biomarkers from complex high throughput data. *Toxicol Lett* 158, 20–29.

66. Nikolsky, Y., Nikolskaya, T., Bugrim, A. (2005) Biological networks and analysis of experimental data in drug discovery. *Drug Discov Today* 10, 653–662.

67. Nikolsky, Y., Sviridov, E., Yao, J., Dosymbekov, D., Ustyansky, V., Kaznacheev, V., Dezso, Z., Mulvey, L., Macconaill, L. E., Winckler, W., et al. (2008) Genome-wide functional synergy between amplified and mutated genes in human breast cancer. *Cancer Res* 68, 9532–9540.

68. van Noort, V., Snel, B., Huynen, M. A. (2007) Exploration of the omics evidence landscape: adding qualitative labels to predicted protein-protein interactions. *Genome Biol* 8, R197, PMID: 17880677.

69. Pagel, P., Kovac, S., Oesterheld, M., Brauner, B., Dunger-Kaltenbach, I., Frishman, G., Montrone, C., Mark, P., Stumpflen, V., Mewes, H. W., et al. (2005) The MIPS mammalian protein-protein interaction database. *Bioinformatics* 21, 832–834.

Section II

Data Mining Techniques

Chapter 10

Proximity Measures for Cluster Analysis

Oliviero Carugo

Abstract

The present chapter provides the basic information about the measures of proximity between two subjects or groups of subjects. It is obvious that these concepts must be clear in order to apply them to any pattern recognition analysis, both supervised and unsupervised.

Key words: cluster analysis, distance, proximity, similarity.

1. Introduction

The cluster analysis is probably the most widely used technique of unsupervised pattern recognition. Its fundamental objective is to look for clusters in a given population of subjects, each characterized by a certain number of variables. In other words, the cluster analysis is performed in order to see if the subjects can be classified in different groups. The applications of cluster analyses are very numerous in very different scientific fields. A typical example, in biology, is the study of the taxonomy of the species or the delineation of evolutionary trees on the basis of protein sequence alignments. The present chapter will provide some basic information about the measures of proximity (distance or similarity) between subjects that must be classified with cluster analysis. Clustering techniques will be described in the next chapter.

The cluster analysis, like many other statistical tools, may give different results depending on how it is used. For example, if we take a frog, a cat, a salmon, and an eagle, we can classify them in different ways as a function of the classification criterion we adopt. If we decide to group the subjects on the basis of the place where they live, we can get three groups, or clusters, one containing the

salmon and the chicken, which live outside water, one containing the salmon, which lives in water, and the last containing the frog, which can live both inside and outside the water. On the contrary, we get only two clusters if we focus the attention on the ability to fly, since only the eagle is a subject able to fly, while the other subjects, the frog, the dog, and the salmon are unable to fly. This trivial example clearly shows how the results of the unsupervised pattern recognition methods can be fragile, since they strongly depend on the variables that are associated with the statistical units and on the criteria with which the statistical units are grouped into discrete clusters. Nevertheless, these statistical techniques are very precious in the real life of data miners. When little is known about the structure of the data, the cluster analysis can provide a starting point for further investigations.

The definition of cluster is, per se, a rather ambiguous exercise. Certainly, the classification of entities is a very ancient human ability, deeply inserted into the human nature. Anybody possesses the ability to recognize a dog and to put into the dog group a newly observed associated with the basic features a dog must possess. Nevertheless, the theoretical definition of cluster, the cluster of the dogs, is very ambiguous. Several definitions have been proposed and the most close to the human perception is that of natural clusters. They are defined as a continuous region in the space, containing a high density of subjects, and separated from other clusters by regions containing a low density of subjects. The exact separation between two clusters is therefore defined in a rather arbitrary way. Natural clusters are also termed hard, or crisp, since each single subject may belong to one and only one cluster. Alternatively, it is possible to use the concept of fuzzy cluster and allow the subjects to belong to more than a single cluster, proportionally to their degree of similarity with each cluster. Although this second approach may be extremely useful in various disciplines, from neurobiology to economics, we prefer to concentrate here on the concept of natural clusters. From an operational point of view, this means that similar statistical units must be grouped together while dissimilar units must be put in different clusters.

Given its intrinsic ambiguities, it is necessary to examine accurately all steps of cluster analysis.

First at all, the statistical variables must be carefully selected. On the one hand, the inclusion of too many variables may have two major drawbacks: the overall analysis lacks elegance and the computations can become very expensive. On the other hand, some redundancy may be tolerated if this ensures better results. The selection of the right set of variables depends on the data mining objective and, consequently, it must be performed or, at least, checked by experts in the field in which the data miner operates.

Second, it is necessary to define the proximity measure between subject pairs. The proximity may be evaluated by distance measures or by similarity measures. Although these are conceptually alternatives, there is little difference in practice. Many different possibilities have been explored and proposed.

Third, the clustering criterion must be defined. In other words, it is necessary to decide under which conditions two statistical units must be grouped together and also if two clusters, each containing more than one unit, must be fused into a single group. Different clustering criteria may produce different results because of the structure of the data. For example, it is obvious that compact clusters (**Fig. 10.1a**) should be compared in a different way than elongated clusters (**Fig. 10.1b**). It is nevertheless nearly impossible to select a priori the optimal clustering criterion in pattern recognition, especially when each statistical unit is associated with a high number of variables. In this case, in fact, each unit corresponds to a point in a high-dimensional space and the data structure can hardly be perceived by the common methods of human perception.

Fourth, a very large variety of clustering algorithms is available. Also at this point, the results of a cluster analysis markedly depend on the choice of an algorithm over another. We will nevertheless concentrate the attention on a particular type of algorithms, the hierarchical ones, which are mostly used in molecular biology. It must, however, be remembered that often the results may change considerably by changing the strategy with which the clustering is carrier out.

Eventually, it is necessary to validate and interpret the results of the cluster analysis. The latter point, like the selection of the variables, strictly depends on the reason why the cluster analysis is performed and consequently relies on the scientific experience of

(a)   (b)
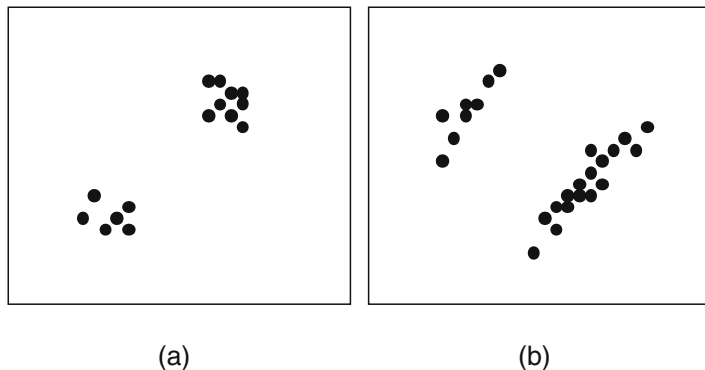
Fig. 10.1. Example of data that show a different clustering tendency. In (**a**) the points tend to cluster in a compact manner while in (**b**) the clusters are rather elongated.

the data miner. On the contrary, the result validation is an objective procedure intended to verify the correctness of the cluster analysis output and it is usually performed through appropriate tests.

In conclusion, it appears that cluster analysis starts and ends with two steps that need the advice of people experienced in the field that is investigated (the selection of the variables and the interpretation of the results). In between the initial and the final steps, it is necessary to define and apply four computational steps that may be common to analyses performed in different fields, like sociology, economics, or biology (definition of proximity measures, clustering criteria, clustering algorithms, and result validation).

2. Proximity Measures

The proximity between statistical units can be measured by a distance or by a similarity. This is nevertheless a trivial problem, though the difference between distance and similarity must be kept in mind, especially during computations. On the contrary, it is not trivial to consider the fundamental properties of the proximity measures. They can be divided into two classes: those that are metric and those that are not. For both types of measures, given the statistical units $\mathbf{X}$ and $\mathbf{Y}$, it must be true that

$$d(\mathbf{X}, \mathbf{X}) = d_{\min} \tag{1}$$

where $d_{\min}$ is the minimal, possible distance, which can be encountered when the statistical unit $\mathbf{X}$ is compared to itself. It must be, moreover, always true that

$$-\infty < d_{\min} \leq d(\mathbf{X}, \mathbf{Y}) < +\infty \tag{2}$$

which means that the statistical units $\mathbf{X}$ and $\mathbf{Y}$ may be identical, if their distance is equal to $d_{\min}$, or different if their distance is higher than $d_{\min}$. Any type of distance is moreover always commutative, since

$$d(\mathbf{X}, \mathbf{Y}) = d(\mathbf{Y}, \mathbf{X}) \tag{3}$$

Exactly the same properties hold also if the proximity is evaluated by means of similarity measures. In this case it is always true that

$$s(\mathbf{X}, \mathbf{X}) = s_{\max} \tag{4}$$
$$-\infty > s(\mathbf{X}, \mathbf{Y}) \leq s_{\max} \leq +\infty \tag{5}$$
$$s(\mathbf{X}, \mathbf{Y}) = s(\mathbf{Y}, \mathbf{X}) \tag{6}$$

Distances and similarities are metric, in the mathematical sense, only if the triangular inequality holds. This implies that given the statistical units **X**, **Y**, and **Z**

$$d(\mathbf{X}, \mathbf{Z}) \leq d(\mathbf{X}, \mathbf{Y}) + d(\mathbf{Y}, \mathbf{Z}) \tag{7}$$

$$s(\mathbf{X}, \mathbf{Z}) \geq [s(\mathbf{X}, \mathbf{Y})s(\mathbf{Y}, \mathbf{Z})]/[s(\mathbf{X}, \mathbf{Y}) + s(\mathbf{Y}, \mathbf{Z})] \tag{8}$$

This inequality is of fundamental importance in data mining, when the data must be examined by means of unsupervised pattern recognition methods. Nevertheless, we must be aware of the fact that many measures of distance or similarity, which are used in molecular biology, are not metric. For example, the proximity between protein three-dimensional structures is very often estimated by means of the root-mean-square distance between equivalent and optimally superposed pairs of atoms. Well, this very popular proximity measure is metric only for very similar subjects (e.g., apo and holo metallo-proteins or different single point mutants), but it is not a metric when the data include proteins with very different shapes and sizes.

Beside these theoretical considerations, there are three types of proximities that one might handle: the proximity between individual units, that between a single unit and a group of units, and that between two clusters, each containing more than one statistical unit.

2.1. Proximity Between Two Statistical Units

The most commonly used distance measure is the Minkowski metric. Given two units, $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ and $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}$, it is defined as

$$d_{Mp} = \left(\sum_{i=1}^{n} w_i |x_i - y_i|^p \right)^{1/p} \tag{9}$$

where the weights $0 \leq w_i \leq 1$ can be equal to 1 in the case of unweighted distances, or not in the case of weighted distances. The parameter p can assume any positive, integer value. If $p = 1$, the distance is also known as the Manhattan norm:

$$d_{MN} = \sum_{i=1}^{n} w_i |x_i - y_i| \tag{10}$$

and if $p = 2$, the distance is also known as the Euclidean distance:

$$d_{E} = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2} \tag{11}$$

Several other distance measures have been used in various applications. For example, the d_{max} norm is defined as

$$d_{max} = \max_{1 \leq i \leq n}(w_i |x_i - y_i|) \tag{12}$$

The d_G distance includes some information about all the statistical units that are examined since it is defined as

$$d_G = -\log_{10}\left(1 - \frac{1}{n}\sum_{i=1}^{n}\frac{|x_i - y_i|}{M_i - m_i}\right) \qquad (13)$$

where M_i and m_i are the maximal and minimal values of the ith statistical variable within the ensemble of all the statistical units that are examined. Consequently, the distance d_G between the units $\mathbf{X}$ and $\mathbf{Y}$ may vary if it is computed when $\mathbf{X}$ and $\mathbf{Y}$ are part of a certain ensemble of units or part of another ensemble of units. An alternative is the d_Q distance, defined as

$$d_Q = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - y_i}{x_i + y_i}\right)^2} \qquad (14)$$

All the above distances can be applied to real-type variables. In the case of qualitative variables, nominal or ordinal, the distance between two statistical units must be defined in different ways. The most popular of them is certainly the Hamming distance, defined as the number of places where two vectors differ. From a formal point of view, this can be expressed by means of the contingency table. If the variables of the units $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ and $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}$ can assume m states, the contingency table is a square $m \times m$ matrix $\mathbf{A}$, the elements a_{ij} of which are the number of times the ith possible value present in $\mathbf{X}$ has been substituted by the jth possible value in $\mathbf{Y}$. For example, if $m = 3$ and the possible values, or states, of the variables are 1, 2, and 3, the contingency table that compares the unit $\mathbf{X} = \{2, 1, 2, 3, 1, 2\}$ with $\mathbf{Y} = \{2, 2, 3, 1, 2, 3\}$ is

$$\begin{pmatrix} 0 & 2 & 0 \\ 0 & 1 & 2 \\ 1 & 0 & 0 \end{pmatrix} \qquad (15)$$

As an example, the element $a_{1,2}$ is equal to 2 since it happens twice that a variable $x = 1$ is associated with a variable $y = 2$ ($x_2 = 1$ and $y_2 = 2$; $x_5 = 1$ and $y_5 = 2$).

The Hamming distance d_H can be therefore defined as

$$d_H = \sum_{i=1}^{m}\sum_{j=1, i\neq j}^{m} a_{ij} \qquad (16)$$

given that the elements a_{ij}, with $i \neq j$, of the contingency table indicate the number of times $x_i \neq y_i$. In the case of $\mathbf{X} = \{2, 1, 2, 3, 1, 2\}$ and $\mathbf{Y} = \{2, 2, 3, 1, 2, 3\}$, therefore, $d_H = 5$ because only the first variables of $\mathbf{X}$ and $\mathbf{Y}$ have the same status, $x_1 = y_1 = 2$, while for

all the other 5 values of i, $x_i \neq y_i$. The computation of the Hamming distance by means of the contingency table is certainly not necessary in simple cases, like that presented above, in which both the number of possible statuses of the variables ($m = 3$) and the number of variables associated with each statistical unit ($n = 6$) are small. In other instances, where both m and n can be very large, the use of the contingency table makes the computations of the Hamming distance much easier.

A different distance definition for discrete variables is

$$d_{\mathrm{D}} = \sum_{i=1}^{n} |x_i - y_i| \qquad (17)$$

which is obviously equivalent to the Hamming distance when the data are binary, that is, when each variable may assume only two values, for example, 1 and 2.

If the statistical units are characterized by different types of variables, some of which may assume real number values and some others may assume only discrete values, the distance between two units cannot be measured with the methods described above and other definitions of distance must be used. Various solutions of this problem have been proposed, the most widely used of which is based on the discretization of the real variables. If, for example, the ith variable assumes real values in the closed interval (a, b), which means that a and b are the minimal and maximal values that the ith variable assumes within the statistical units under exam or in an absolute scale, where the ith variable cannot be minor than a or major than b, the x_i values can be described by a histogram. The interval (a, b) is divided into m subintervals and if the variable x_i falls into the jth subinterval, it is transformed into $j - 1$.

As mentioned in the introductory paragraph of this section, the degree of proximity between two statistical units can be measured not only with distance measures, like those described above, but also with similarity measures.

Two similarity measures are used very often. One is the correlation coefficient. Given two vectors $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ and $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}$, it is defined as

$$s_{\mathrm{cc}} = \frac{\displaystyle\sum_{i=1}^{n} (x_i - x_{\mathrm{av}})(y_i - y_{\mathrm{av}})}{\sqrt{\displaystyle\sum_{i=1}^{n} (x_i - x_{\mathrm{av}})^2 \sum_{i=1}^{n} (y_i - y_{\mathrm{av}})^2}} \qquad (18)$$

and it ranges between -1 and $+1$, being 0 if the two statistical units $\mathbf{X}$ and $\mathbf{Y}$ are totally independent from each other. The maximal value of $+1$ is encountered if $\mathbf{X}$ and $\mathbf{Y}$ are identical, i.e. perfectly correlated, and the minimal value of -1 indicates that $\mathbf{X}$ and $\mathbf{Y}$ are

perfectly anticorrelated, i.e. $\mathbf{X} = -\mathbf{Y}$. The second measure of similarity that is used very often is the inner product. Given the statistical units described above, it is defined as

$$s_{\text{in}} = \mathbf{X}^T \mathbf{Y} = \sum_{i=1}^{n} x_i y_i \tag{19}$$

Generally, the inner product is computed after normalization of the vectors $\mathbf{X}$ and $\mathbf{Y}$, so that both have unit length, by means of

$$x_i' = \frac{x_i}{\sqrt{\sum_{i=1}^{n} x_i^2}} \tag{20}$$

In this way, the inner product lies in the interval $(-1, +1)$ and depends on the angle between $\mathbf{X}$ and $\mathbf{Y}$. Identical statistical units are associated with the inner product equal to +1 while a value equal to -1 indicates that the units are opposite, i.e., $\mathbf{X} = -\mathbf{Y}$.

Other measures of similarity can also be used to compare statistical units characterized by variables that can assume real values. A widely used similarity measure is, for example, the Tanimoto similarity s_T, defined as

$$s_T = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} (x_i - y_i)^2} \tag{21}$$

and, if the vectors $\mathbf{X}$ and $\mathbf{Y}$ have been normalized to unit length, can be rewritten as

$$s_T = \frac{\sum_{i=1}^{n} x_i y_i}{2 - \sum_{i=1}^{n} x_i y_i} \tag{22}$$

and may range between -0.33 and +1 for opposite or identical vectors, respectively.

If the variables do not assume real values but can be associated with discrete values or statuses, the Tanimoto measure of similarity between the vectors $\mathbf{X}$ and $\mathbf{Y}$ is defined as the ratio between the number of elements they have in common and the number of elements they do not have in common. By using the contingency table described above, the Tanimoto measure can be defined as

$$s_T = \frac{\sum_{i=1}^{m} a_{ii}}{\sum_{i=1}^{m} \sum_{j=1, j \neq i}^{m} a_{ij}} \tag{23}$$

Alternatively, it is possible to define a similarity that is based on the ratio between the number of elements that $\mathbf{X}$ and $\mathbf{Y}$ have in common and the number of variables that characterize each unit. Such a similarity measure can be computed as

$$s_A = \frac{\sum_{i=1}^{m} a_{ii}}{n} \tag{24}$$

Like for the distance measures, when the variables are of different types, the definitions of similarity described above cannot be used. In these cases, it is necessary to measure the degree of similarity by other means. Like for the distance measures, a possible solution of this problem is based on the discretization of the real variables by means of histograms. The similarity between the statistical units $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ and $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}$ can be measured as the sum of the similarity between each pair of variables x_i and y_i

$$s_Q = \sum_{i=1}^{n} s_i \tag{25}$$

The s_i is the similarity between the ith pair of variables and it can be differently computed depending on the type of variable. If the latter is a real number, s_i may be defined as

$$s_i = 1 - \frac{|x_i - y_i|}{r_i} \tag{26}$$

where r_i is the interval of values that is possible or is observed within the ith variable. Thus, if $x_i = y_i$, s_i reaches its maximal value equal to 1, while if the absolute difference between x_i and y_i is equal to r_i, s_i assumes its minimal value equal to 0. On the contrary, if the ith variable is not a real variable, s_i is equal to 1 if $x_i = y_i$ and it is equal to 0 if $x_i \neq y_i$. Independently of the type of variable, each individual similarity s_i may have values ranging from 0 and 1, and the s_Q measure of proximity has a minimal value of 0 and a maximal value equal to n, the number of variables associated with each statistical unit.

2.2. Proximity Between a Single Unit and a Group of Units

The proximity between a single statistical unit and a group of several (two or more than two) units must be measured with techniques that are different from those that are used to measure the proximity between two statistical units. It is necessary to compute the proximity between a unit and a group of units in several circumstances, in both supervised and unsupervised pattern recognition methods.

There are two types of proximity measures between a single statistical unit and a group of various units: the single subject can be compared to all the members of the group or it can be compared to a profile of the group, which summarizes the most

important features of the group. In the first case, the problem can be handled with the definitions of proximity between pairs of units, though it is necessary to solve the problem of how to handle the n proximities between the single subject and the n units belonging to a group. In the second case, the problem is the definition of the profile that summarizes all the elements of the group, and the proximity between a single unit and a group of units is measured by the proximity between the single unit and the summarizing profile.

If the proximity between a single unit and a group of units is measured as a function of the individual proximities between the single unit and each of the elements of the group, three extreme possibilities exist. The distance can be defined as the maximal distance between the single subject and the group members, as the minimal distance between the individual unit and the elements of the group, or as the average distance between the single unit and the members of the group. Analogously, the similarity between a single unit and a group of units can be defined as the maximal, minimal, or average similarity between the single subject and the elements of the group. From a formal point of view, this can be described as follows. Given a single unit $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ and the group $\mathbf{Y}$ of m units $\mathbf{Y}_1 = \{y_{11}, y_{12}, \ldots, y_{1n}\}$, $\mathbf{Y}_2 = \{y_{21}, y_{22}, \ldots, y_{2n}\}$, $\ldots$, and $\mathbf{Y}_m = \{y_{m1}, y_{m2}, \ldots, y_{mn}\}$, the proximity $P(\mathbf{X},\mathbf{Y})$ between $\mathbf{X}$ and $\mathbf{Y}$ can be measured as

$$P(\mathbf{X}, \mathbf{Y}) = \max_{1 \leq i \leq m}(P(\mathbf{X}, \mathbf{Y}_i)) \tag{27}$$

$$P(\mathbf{X}, \mathbf{Y}) = \min_{1 \leq i \leq m}(P(\mathbf{X}, \mathbf{Y}_i)) \tag{28}$$

$$P(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=1}^{m} P(\mathbf{X}, \mathbf{Y}_i)}{m} \tag{29}$$

where $P(\mathbf{X},\mathbf{Y}_i)$ is the proximity between the single unit $\mathbf{X}$ and the ith elements $\mathbf{Y}_i$ of the group $\mathbf{Y}$.

A number of intermediate definitions are also possible. For example, the proximity between the single unit and a group of units may be estimated as the average proximity between it and the n members of the cluster that are more similar to it or most dissimilar to it.

A completely different approach to estimate the proximity between a single subject and a cluster of units is based on the definition of a profile of the group. The simplest way to do it is to define the average or centroid $\mathbf{Y}_{av}$ of the group, the elements of which are

$$y_{av,i} = \frac{\sum_{i=1}^{m} y_i}{m} \tag{30}$$

The proximity between the single unit $\mathbf{X}$ and the group $\mathbf{Y}$ is then measured by the proximity between $\mathbf{X}$ and $\mathbf{Y}_{av}$. Alternatively, it is possible to select one of the members of the group, measure the proximity between the individual subject and such a representative element, and assume that such a proximity measures the proximity between the single unit and the group. There are various strategies to select an element that can be considered as a representative of the group of elements. Given a group $\mathbf{Y}$ of elements $\mathbf{Y}_i$, with $\mathbf{Y}_1 = \{y_{11}, y_{12}, \ldots, y_{1n}\}$, $\mathbf{Y}_2 = \{y_{21}, y_{22}, \ldots, y_{2n}\}$, etc., the most common procedure consists in summing for each $\mathbf{Y}_i$, the distances between $\mathbf{Y}_i$ and all the other elements $\mathbf{Y}_j\,(i \neq j)$. The minimum value of these sums is associated with the element that is the most representative of the group. This is in fact the element that is, on average, most close to the other elements of the group. Alternatively, the distances between each element $\mathbf{Y}_i$ and all the other elements $\mathbf{Y}_j\,(i \neq j)$ are computed and their median value M_i is stored. The minimal value of all the M_i values, $M = \min(M_i)$, is than searched for and it is associated with the element most representative of the group.

2.3. Proximity Between Two Groups

The proximity between two groups of statistical units can be evaluated by measuring the proximity between pairs of units or by comparing the profiles of the two clusters. In the first case, the proximity P between two groups $\mathbf{X}$ and $\mathbf{Y}$, with m_X and m_Y elements, respectively, can be defined as

$$P = \min_{1 \leq i \leq m_X, 1 \leq j \leq m_Y}(p_{ij}) \tag{31}$$

where p_{ij} is the proximity between the elements $\mathbf{X}_i$ and $\mathbf{Y}_j$. This is known as the nearest neighbour criterion of proximity. Alternatively, P can be measured as

$$P = \max_{1 \leq i \leq m_X, 1 \leq j \leq m_Y}(p_{ij}) \tag{32}$$

or as

$$P = \frac{\sum_{i=1}^{m_X}\sum_{j=1}^{m_Y} p_{ij}}{m_X m_Y} \tag{33}$$

Of course, it is also possible to tune the above definitions, for example, by defining P as the average value of the n highest or lowest p_{ij}, with $n \geq 2$. All these definitions are based on the comparison of all the m_X members of the group $\mathbf{X}$ with all the m_Y elements of the group $\mathbf{Y}$. If on the contrary, one prefers to build a profile for both $\mathbf{X}$ and $\mathbf{Y}$, the proximity P becomes the proximity $P(\mathbf{R}_X, \mathbf{R}_Y)$ between two points $\mathbf{R}_X$ and $\mathbf{R}_Y$, which are representative of $\mathbf{X}$ and $\mathbf{Y}$, respectively. Two formulations are widely used. One is simply

$$P = P(R_X, R_Y) \tag{34}$$

and the other considers explicitly the possibility that the number of elements in $\mathbf{X}$ may by different from the number of units in Υ and is thus

$$P = \sqrt{\frac{m_X m_\Upsilon}{m_X + m_\Upsilon}} P(\mathbf{R}_X, \mathbf{R}_\Upsilon) \qquad (35)$$

In both cases, the selection of the representative points can be performed in various ways, like those described in the section dealing with the measure of proximity between a single unit and a cluster. It is thus possible to select the centroid of the group or one member, which is the most representative.

An interesting property of the proximities P between two sets of statistical units is that they are only seldom metrics, in the mathematical sense. For example, if p_{ij} is a distance, $P = \max(p_{ij})$ cannot have the minimal, possible value when a cluster is compared to itself. If, on the contrary, p_{ij} is a similarity, P is not a metric, in the mathematical sense, because the triangular inequality is not satisfied. It appears, therefore, that different choices of proximity estimators may produce completely different results and that only an expert in the field of application can decide, on a rational basis, which choice is preferable.

Further Reading

Theodoris, S., Koutroumbas, K. (2003) *Pattern Recognition*. Academic Press, Amsterdam.

Corrigan, M. S. (2007) *Pattern Recognition in Biology*. Nova Science, Lancaster.

Kaufman, L., Rousseeuw, P. J. (2005) *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, New York.

Romesburg, H. C. (2004) *Cluster Analysis for Researchers*. Lulu Press, North Carolina.

Everitt, N. S., Landau, S., Leese, M. (2001) *Cluster Analysis*. Hodder Arnold Publications, Oxford.

Chapter 11

Clustering Criteria and Algorithms

Oliviero Carugo

Abstract

Cluster analysis is an unsupervised pattern recognition frequently used in biology, where large amounts of data must often be classified. Hierarchical agglomerative approaches, the most commonly used techniques in biology, are described in this chapter. Particular attention is put on techniques for validating the optimal cluster number and the clustering quality.

Key words: cluster analysis, agglomerative hierarchical clustering, clustering tendency.

1. Introduction

A number of clustering criteria and clustering algorithms have been proposed and used in various scientific fields. A common and somehow mysterious question is: how many clusters of statistical units do exist within a certain population of subjects? Several answers can be given to this question. Several techniques can be applied to solve this problem. In molecular biology, nevertheless, a family of techniques found a wide popularity. This family of techniques is known as the family of the agglomerative, hierarchical methodologies. They are, for example, the basis of any taxonomical classification, when phenotypes of molecular features are considered. Any phylogenetic tree, from the Darwinist approaches to the socio-biological applications, is basically built through an agglomerative, hierarchical cluster analysis.

2. Hierarchical Agglomerative Clustering

The agglomerative cluster analysis of a set of m statistical units $\mathbf{X}_i$, with $1 \leq i \leq m$, starts with a clustering C_0, in which there are m clusters, each containing a single unit $\mathbf{X}_i$. The first step is to merge into a unique cluster the two units $\mathbf{X}_i$ and $\mathbf{X}_j$, which are most similar. This produces a new clustering C_1, in which there are $m-1$ clusters. All of them are occupied by a single element of the examined set with the exception of the cluster that contains $\mathbf{X}_i$ and $\mathbf{X}_j$. The second step produces a new clustering C_2, in which there are $m-2$ clusters. Two possibilities exist for C_2. In the one hand, one of the $m-2$ clusters contains three elements ($\mathbf{X}_i$, $\mathbf{X}_j$, and $\mathbf{X}_k$ if $\mathbf{X}_k$ is merged in the cluster of C_1 that contains $\mathbf{X}_i$ and $\mathbf{X}_j$). On the other hand, C_2 could be formed by one cluster containing $\mathbf{X}_i$ and $\mathbf{X}_j$, the elements of which were already clustered together in C_1, one cluster containing $\mathbf{X}_k$ and $\mathbf{X}_l$, and $m-4$ clusters containing only one statistical unit. The second possibility occurs if the elements $\mathbf{X}_k$ and $\mathbf{X}_l$ are the most similar clusters found at the C_1 level. The first possibility, on the contrary, occurs if the best proximity is found by comparing $\mathbf{X}_k$ with the cluster of C_1 that contains $\mathbf{X}_i$ and $\mathbf{X}_j$.

An agglomerative clustering algorithm implies therefore $m-1$ sequential steps. At the beginning, level $= 0$, there are m singly occupied clusters, and at the end, level $= m-1$, there is only one cluster, which contains all the m statistical units of the ensemble that is examined. At each intermediate step, level $= L$, the two most similar clusters formed in the previous step, level $= L-1$, are grouped together. At each step L, a new clustering C_L is produced. C_L contains one cluster less than the clustering C_{L-1}. From a formal point of view, the clustering C_{L-1} is nested into the clustering C_L.

There are two important considerations that must be made about agglomerative clustering algorithms: the overall clustering process can be summarized by a dendrogram and the algorithm does not provide a unique answer. A dendrogram is depicted, for example, in **Fig. 11.1**. The set of units that are examined includes five subjects. A dendrogram is a simple, pictorial representation of the agglomerative procedure. The scale provides a visual perception of the proximity between the units and between the clusters. Consequently, the dendrogram shown in the figure indicates that the subjects $\mathbf{X}_1$ and $\mathbf{X}_2$ cluster together first, because they are very similar to each other. The initial clustering C_0, which contains five groups, each with a single unit, evolves into the clustering C_1 that contains four clusters. One of them is formed by two clusters of C_0, the subjects $\mathbf{X}_1$ and $\mathbf{X}_2$. The other three are occupied by single units. The subsequent clustering C_2 results in three clusters. One
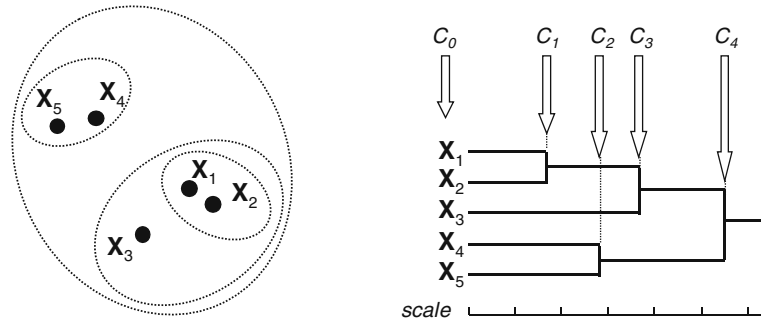
Fig. 11.1. Example of dendrogram that summarizes the clustering of five statistical units.

of them, containing $\mathbf{X}_1$ and $\mathbf{X}_2$, was already present in C_1. Also another cluster of C_2, containing the single unit $\mathbf{X}_3$, was already present in C_1. On the contrary, the cluster of C_2 containing the units $\mathbf{X}_4$ and $\mathbf{X}_5$ was not present on C_1, where these elements were segregated in different groups. While there are five clusters in C_0, four in C_1, and three in C_2, there are only two groups in the clustering C_3. At this clustering level, the element $\mathbf{X}_3$ merges into the cluster of $\mathbf{X}_1$ and $\mathbf{X}_2$. Consequently, at the C_3 level, one cluster with three members ($\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$) and one cluster with two elements ($\mathbf{X}_4$ and $\mathbf{X}_5$) are present. Further on, the clustering C_4 includes all the five statistical units into a unique cluster, the final one, which contains all the elements that are examined. It is important to observe that the lengths of the branches of the dendrogram, i.e., the horizontal lines in the figure, are proportional to the proximity between the clusters that are progressively merged. For example, the proximity between $\mathbf{X}_1$ and $\mathbf{X}_2$ is higher than the proximity between $\mathbf{X}_4$ and $\mathbf{X}_5$.

The second-important consideration about the agglomerative clustering algorithms is that they do not provide a unique clustering. In the general case in which little or nothing is known about the distribution of the data, it is impossible to predict a priori if one or more clusters are present within the elements that are examined and it is consequently impossible to know how many clusters exist. The results of an agglomerative cluster analysis can therefore be ambiguous. In the example described above, for instance, it is possible to hypothesize that the data are structured into three clusters if the clustering C_2 is considered although five clusters are observed at the C_0 level or one cluster is obtained at the C_4 level. Sometimes, a visual inspection of the dendrograms may help in taking a decision on the optimal number of clusters in which a certain data set can be divided. For example, the dendrogram of **Fig. 11.2a** could suggest the presence of two clusters, one containing $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$, and $\mathbf{X}_4$, and the other with $\mathbf{X}_5$ and $\mathbf{X}_6$. On the contrary three clusters, one with $\mathbf{X}_1$ and $\mathbf{X}_2$, the second with $\mathbf{X}_3$ and $\mathbf{X}_4$, and the third with $\mathbf{X}_5$ and $\mathbf{X}_6$, can be easily identified by visual inspection of the dendrogram of
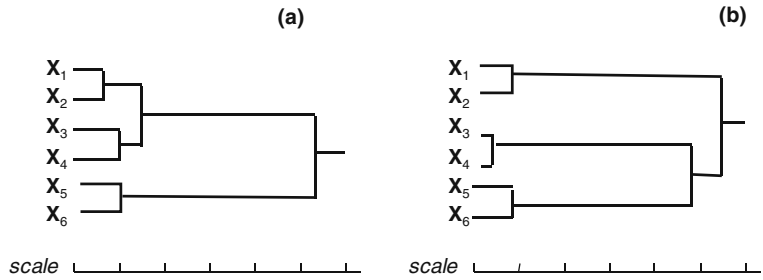
Fig. 11.2. Examples of dendrograms that indicate two very different clusterings of the statistical units.

Fig. 11.2b. Nevertheless, in general, the identification of the number of clusters in which the data can be optimally divided is not obvious. Various techniques have been developed to determine the optimal number of groups in which the data can be partitioned. All of them are rather arbitrary. It must be remembered, therefore, that a careful analysis of the dendrograms, especially if the help of expert scientists is available, is essential.

The general agglomerative clustering algorithm can be summarized by means of the following scheme.

(a) At the level 0, the clustering C_0 consists of m clusters, each of which contains a single subject.

(b) Go to the next level L, by searching amongst the clusters of C_{L-1} the two clusters that have the maximal proximity.

(c) This results in the new clustering C_L that consists of a number of clusters equal to the number of clusters of C_{L-1} minus one.

(d) Go to point (b), unless all the elements that are examined are clustered together.

Of course, if two clusters of C_{L-1} merge into a single cluster in C_L, they continue to be merged for all the subsequent clustering C_{L+k}, with $k \geq 1$.

In practice, most of the agglomerative algorithms use the matrix approach. This means that an ensemble of m subjects, each characterized by n variables, is represented by a $m \times n$ data matrix **D**, each row of which is associated with a statistical unit and each column of which is associated with a statistical variable. The data matrix **D** is translated into an $m \times m$ pattern (or proximity) matrix **P**. Each element p_{ij} of **P** indicates the proximity between the ith and the jth element. Such a square matrix **P** is symmetrical with respect to the main diagonal. Any type of proximity measure can be used to go from **D** to **P**. This means that more than a matrix **P** can be obtained from a single matrix **D** and that it is impossible to know the matrix **D** by knowing the matrix **P**. Nevertheless, at the clustering level 0, the **P** matrix is uniquely determined by the measure of proximity between the elements of the set that must be

analyzed. As an example, the following data matrix **D** has 12 rows and 2 columns. It represents a data set of 12 statistical units, each characterized by 2 variables (*see* **Table 11.1**).

These 12 subjects are plotted on a bi-dimensional space in **Fig. 11.3**. It clearly appears that the 12 subjects tend to cluster into two distinct groups, one at low values of both the first and the

Table 11.1
List of 12 statistical units, each characterized by two variables

Subject	Variable 1	Variable 2
1	1.0	1.0
2	2.0	1.0
3	1.5	1.5
4	2.5	1.5
5	2.0	2.0
6	1.5	2.5
7	2.0	3.0
8	5.0	4.0
9	5.5	3.5
10	5.5	4.5
11	6.0	4.0
12	6.5	4.5
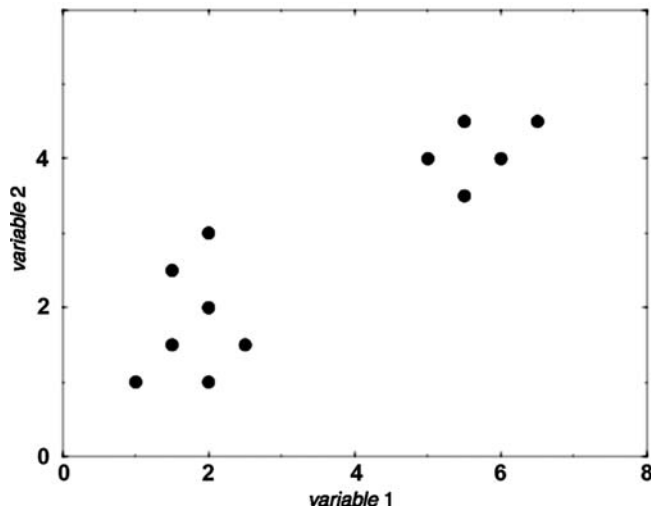
Fig. 11.3. Scatter plot of the 12 statistical units defined in **Table 1.**

second variable, and the other at higher values of both the first and the second variable. The proximity matrix **P**, associated with the data matrix reported above, can be obtained by using the Euclidean distance. It is reported in **Table 11.2**. A different proximity matrix **P** is obtained by measuring the proximity though the City Block distances and is reported in **Table 11.3**.

In going from the clustering level 0 to the level 1, it is necessary to find the pair of units $\mathbf{X}_i$ and $\mathbf{X}_i$, the proximity of which is maximal. The subjects $\mathbf{X}_i$ and $\mathbf{X}_j$ are then clustered together and $m-1$ clusters are present in C_1. While m clusters are observed at the level 0, $m-1$ clusters are present at the level 1. The $m \times m$ matrix **P** of the C_0 level is therefore substituted by an $m-1 \times m-1$ matrix, by deleting the ith and the jth rows and columns, and by substituting them with a single row and a single column. The latter contains the measures of proximity between the newly formed cluster and all the other statistical units. At each clustering step, therefore, the proximity matrix leaves a row and a column. Consequently, the number of operations that must be performed is

$$\frac{(m-1)m(m+1)}{6} \tag{1}$$

given a set of m statistical units. To appreciate the computational complexity of an agglomerative clustering procedure, it is necessary to think that if $m = 10$, that is, if there are only 10 subjects, it is

Table 11.2
Euclidean distances between the 12 statistical units reported in Table 1

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	1.0	0.7	1.6	1.4	1.6	2.2	5.0	5.1	5.7	5.8	6.5
2	1.0	0.0	0.7	0.7	1.0	1.6	2.0	4.2	4.3	4.9	5.0	5.7
3	0.7	0.7	0.0	1.0	0.7	1.0	1.6	4.3	4.5	5.0	5.1	5.8
4	1.6	0.7	1.0	0.0	0.7	1.4	1.6	3.5	3.6	4.2	4.3	5.0
5	1.4	1.0	0.7	0.7	0.0	0.7	1.0	3.6	3.8	4.3	4.5	5.1
6	1.6	1.6	1.0	1.4	0.7	0.0	0.7	3.8	4.1	4.5	4.7	5.4
7	2.2	2.0	1.6	1.6	1.0	0.7	0.0	3.2	3.5	3.8	4.1	4.7
8	5.0	4.2	4.3	3.5	3.6	3.8	3.2	0.0	0.7	0.7	1.0	1.6
9	5.1	4.3	4.5	3.6	3.8	4.1	3.5	0.7	0.0	1.0	0.7	1.4
10	5.7	4.9	5.0	4.2	4.3	4.5	3.8	0.7	1.0	0.0	0.7	1.0
11	5.8	5.0	5.1	4.3	4.5	4.7	4.1	1.0	0.7	0.7	0.0	0.7
12	6.5	5.7	5.8	5.0	5.1	5.4	4.7	1.6	1.4	1.0	0.7	0.0

Table 11.3
City Block distances between the 12 statistical units reported in Table 1

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	1.0	1.0	2.0	2.0	2.0	3.0	7.0	7.0	8.0	8.0	9.0
2	1.0	0.0	1.0	1.0	1.0	2.0	2.0	6.0	6.0	7.0	7.0	8.0
3	1.0	1.0	0.0	1.0	1.0	1.0	2.0	6.0	6.0	7.0	7.0	8.0
4	2.0	1.0	1.0	0.0	1.0	2.0	2.0	5.0	5.0	6.0	6.0	7.0
5	2.0	1.0	1.0	1.0	0.0	1.0	1.0	5.0	5.0	6.0	6.0	7.0
6	2.0	2.0	1.0	2.0	1.0	0.0	1.0	5.0	5.0	6.0	6.0	7.0
7	3.0	2.0	2.0	2.0	1.0	1.0	0.0	4.0	4.0	5.0	5.0	6.0
8	7.0	6.0	6.0	5.0	5.0	5.0	4.0	0.0	1.0	1.0	1.0	2.0
9	7.0	6.0	6.0	5.0	5.0	5.0	4.0	1.0	0.0	1.0	1.0	2.0
10	8.0	7.0	7.0	6.0	6.0	6.0	5.0	1.0	1.0	0.0	1.0	1.0
11	8.0	7.0	7.0	6.0	6.0	6.0	5.0	1.0	1.0	1.0	0.0	1.0
12	9.0	8.0	8.0	7.0	7.0	7.0	6.0	2.0	2.0	1.0	1.0	0.0

necessary to perform 165 operations. If the number of units is considerably larger, for example, if $m = 500$, it is necessary to perform more than 20 millions of operations. **Figure 11.4** shows the increase of complexity of an agglomerative clustering algorithm as a function of the number of statistical units that must be analyzed. Apparently, the cluster analyses are rather expensive.
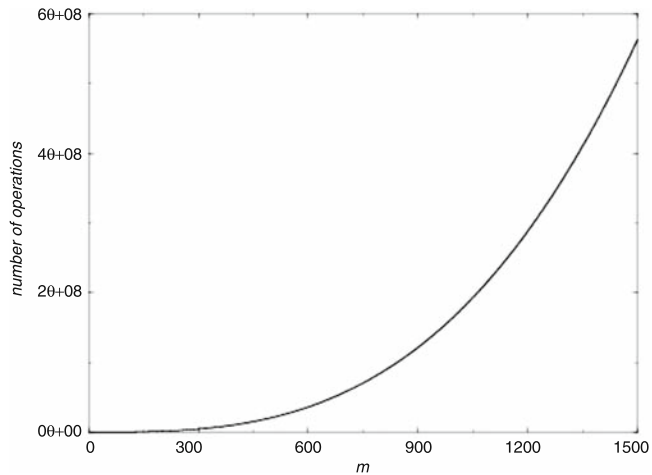
Fig. 11.4. Increase of the complexity of an agglomerative-clustering algorithm as a function of the number of statistical units m that must be clustered.

There is a simple equation that allows one to define each type of hierarchical, agglomerative clustering procedure. The central question is the definition of the proximity between two clusters. Such a definition is of fundamental importance in order to build, at each clustering step, the new proximity matrix **P**. This simple equation that defines the proximity between two clusters C_1 and C_2 depends on the clusters C_i and C_j that have been merged, in the last clustering step, into the new cluster C_1 (**Fig. 11.5**).

clustering direction
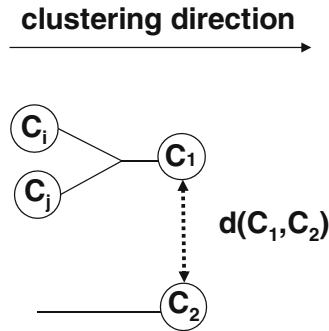
Fig. 11.5. The cluster C_1 has just been formed by merging the clusters C_i and C_j. The distance $d(C_1,C_2)$ must be measured in order to decide if the newly formed cluster C_1 has to be merged with the cluster C_2 in the next clustering step.

Thereinafter, the proximity is assumed to be a distance, although similar considerations can be done if the proximity is estimated by means of a similarity measure. The distance between the newly formed cluster C_1 and an existing cluster C_2 can be measured as

$$d(C_1, C_2) = a_1 d(C_i, C_2) + a_2 d(C_j, C_2) + a_3 d(C_i, C_j) + a_4 |d(C_i, C_2) - d(C_j, C_2)| \qquad (2)$$

With such an equation it is possible to define all the possible clustering strategies, as a function of the values of the parameters a_i, with $1 \leq i \leq 4$. For example, if $a_1 = a_2 = 1/2$, $a_3 = 0$, and $a_4 = -1/2$, $d(C_1, C_2)$ assumes the minimal value of the distances between C_i and C_2 and between C_j and C_2. From a formal point of view, this can be written as

$$d(C_1, C_2) = \min\left[d(C_i, C_2), d(C_j, C_2)\right] \qquad (3)$$

and it is usually referred to as the single-link clustering criterion. Alternatively, if $a_1 = a_2 = a_4 = 1/2$ and $a_3 = 0$, the distance $d(C_1, C_2)$ assumes the minimal value of the distances between C_i and C_j and C_2. This is usually known as the complete-link clustering criterion and, from a formal point of view, it can be written as

$$d(C_1, C_2) = \max\left[d(C_i, C_2), d(C_j, C_2)\right] \qquad (4)$$

Obviously, if the proximity between the clusters is estimated by means of a similarity measure, instead of a distance measure, the operators min and max must be permuted.

The single-link and the complete-link clustering criteria adopt two extreme definitions of distance between the clusters C_1 and C_2. An infinite number of intermediate criteria are possible. Here only the most commonly used are described.

An intermediate clustering criterion, which is very popular in molecular biology, is the unweighted pair group method average (UPMGA) algorithm. It is defined by $a_1 = m_i/(m_i + m_j)$, $a_2 = m_j/(m_i + m_j)$, and $a_3 = a_4 = 0$, where m_i and m_j are the number of statistical units contained in the clusters C_i and C_j. In this case, the distance between the clusters C_1 and C_2 is defined as

$$d(C_1, C_2) = \frac{m_i}{m_i + m_j} d(C_i, C_2) + \frac{m_j}{m_i + m_j} d(C_j, C_2) \quad (5)$$

Alternatively, the unweighted pair group method centroid (UPGMC) criterion can be used. In this case, $a_1 = m_i/(m_i + m_j)$, $a_2 = m_j/(m_i + m_j)$, $a_3 = -m_i m_j/(m_i + m_j)^2$, and $a_4 = 0$. Therefore,

$$d(C_1, C_2) = \frac{m_i}{m_i + m_j} d(C_i, C_2) + \frac{m_j}{m_i + m_j} d(C_j, C_2) - \frac{m_i m_j}{(m_i + m_j)^2} d(C_i, C_j) \quad (6)$$

The weighted pair group method average (WPGMA) clustering criterion is a further possibility, where $a_1 = a_2 = 1/2$ and $a_3 = a_4 = 0$. Consequently,

$$d(C_1, C_2) = \frac{d(C_i, C_2) + d(C_j, C_2)}{2} \quad (7)$$

The weighted pair group method centroid (WPGMC) criterion is defined by $a_1 = a_2 = 1/2$ and $a_3 = -1/4$, and $a_4 = 0$ and therefore

$$d(C_1, C_2) = \frac{d(C_i, C_2) + d(C_j, C_2)}{2} - \frac{d(C_i, C_j)}{4} \quad (8)$$

Another clustering criterion that is often employed is the Ward (or minimum variance) algorithm. This estimates the proximity between two clusters C_A and C_B by means of the Euclidean distance between their centroids. Such a distance is then weighted as

$$d(C_A, C_B)' = \frac{m_A m_B}{m_A + m_B} d(C_A, C_B) \quad (9)$$

where m_A and m_B are the number of subjects contained in cluster C_A and cluster C_B, respectively. The weighted distance between the clusters C_1 and C_2 is thus computed as

$$d(C_1, C_2)' = \frac{m_i+m_2}{m_i+m_j+m_2} d(C_i, C_2)' +$$
$$\frac{m_j+m_2}{m_i+m_j+m_2} d(C_j, C_2)' - \qquad (10)$$
$$\frac{m_2}{m_i+m_j+m_2} d(C_i, C_j)'$$

Several clustering criteria are therefore possible within a hierarchical, agglomerative clustering procedure. Different results can be obtained by changing the clustering criteria. This is not, however, the only problem. Another problem is that, as anticipated above, a hierarchical clustering algorithm does not provide the optimal clustering. It gives only a series of possible clusterings, where the clustering C_{L-1}, at the step $L-1$, is nested in the clustering C_L, at the step L. A series of possible clusterings is thus provided and there is not a unique and definitive answer to the question of how many clusters exist in a certain population of statistical units. It is, nevertheless, very important, in general, to know how many clusters can be found within an ensemble of subjects. A simple answer to such a question cannot be given by the agglomerative clustering algorithms. Several heuristic approaches have thus been developed in order to decide, on an objective way, the optimal number of partitions in which an ensemble of statistical units can be subdivided. These can be classified into two types, the extrinsic and the intrinsic approaches. The intrinsic approaches require the analysis of the specific structure of the data. On the contrary, the extrinsic approaches imply the comparison between the clustering and some specific, a priori information.

The simplest example of the intrinsic method is summarized here. The best clustering C_l is that in which the proximity between the members of a cluster is higher that the proximity between the subjects that belong to different clusters. From a formal point of view, if the proximity is measured with a distance d, this means that an optimal clustering is characterized by the following relationship

$$d(C_i, C_j) > \max\left[d(C_i, C_i), d(C_j, C_j)\right] \qquad (11)$$

for any i and j belonging to the data set, which is analyzed, and where $d(C_i, C_j)$ is the distance between clusters C_i and C_j, and where $d(C_i, C_i)$ is the distance between the elements belonging to the same cluster C_i. **Figure 11.6** depicts such a condition. The distance $d(C_i, C_i)$ is the maximal distance between any pair of members of cluster C_i. $d(C_j, C_j)$ is the maximal distance between any pair of members of cluster C_j. And $d(C_i, C_j)$ is the minimal distance between a member of cluster C_i and a member of cluster
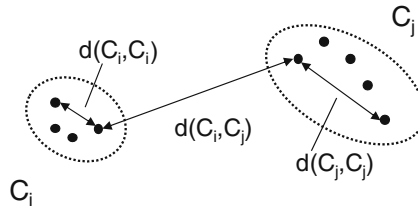
Fig. 11.6. Intrinsic method to evaluate the clustering quality.

C_j. The condition that $d(C_iC_j)$ must be larger than any distance $d(C_i,C_i)$ or $d(C_j,C_j)$ implies that the intracluster similarity is maximized relative to the intercluster similarity.

Contrary to the intrinsic approaches, the extrinsic ones imply some a priori information. The simplest extrinsic approach is described here. If the proximity between the members of cluster C_i is measured with the distance $d(C_i,C_i)$, the clustering level C_L, in which C_i is obtained, can be an optimal clustering level if

$$d(C_i, C_i) \leq \vartheta \tag{12}$$

where θ is an arbitrary threshold. The distance $d(C_i,C_i)$ can be defined in various ways, like it has been described in the previous sections. For example, $d(C_i,C_i)$ can be measured as

$$d(C_i, C_i) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{m} d(\mathbf{X}_i, \mathbf{X}_j)}{2m} \tag{13}$$

where m is the number of statistical units that the cluster C_i contains, and $\mathbf{X}_i$ and $\mathbf{X}_j$ are members of the cluster C_i. In this way, $d(C_i,C_i)$ is simply the average distances between the pairs of members of the cluster. Alternatively, $d(C_i,C_i)$ can be defined as the maximal distance between the elements of C_i

$$d(C_i, C_i) = \max_{1 \leq i \leq m, 1 \leq j \leq m} \left[d(\mathbf{X}_i, \mathbf{X}_j) \right] \tag{14}$$

or as the minimal distances between the elements of C_i

$$d(C_i, C_i) = \min_{1 \prec i \leq m, 1 \leq j \leq m, i \neq j} \left[d(\mathbf{X}_i, \mathbf{X}_j) \right] \tag{15}$$

The condition that $d(C_i,C_i)$ must be smaller than or equal to a threshold value θ means that the self-similarity, between members of each cluster, must not be smaller than a threshold, over which the clusters are no more compact enough. The value of θ is of course ill defined. Any arbitrary threshold can be used. A simple way to reduce the degree of arbitrariness is to select a θ value that depends on the data that are analyzed. For example, it is possible to define θ as

$$\theta = \mu + \lambda\sigma \tag{16}$$

where μ and σ are the mean value and the associated standard deviation of the distribution of the distances between all statistical units that are analyzed. According to this approach, the only a priori parameter, the value of which must be defined, is λ. Usually, λ is set to be 3, though any other value can be selected. The value of λ indicates, anyway, to which extent the intracluster proximity must exceed the intercluster proximity. In other words, the value of λ indicates the extent to which the proximity between the members of a cluster can be widened. Over the threshold θ, the cluster is considered unreliable, because it is too inhomogeneous.

3. Clustering Validation

The final procedure of any cluster analysis is the critical assessment of the results. This does not mean the interpretation of the results, which is possible only for scientists that are expert in the specific field in which the cluster analysis is performed. This means, on the contrary, the evaluation of the clustering reliability, from a strictly statistical point of view.

In general, two types of validations may be needed. On the one hand, it may be necessary to determine how reasonable an individual cluster is. A well-formed cluster must obviously be compact and isolated from other clusters. On the other hand, it may be necessary to determine if a clustering reflects well the structure of the data. A good clustering must of course be a "good" translation of the proximity matrix **P**. Besides these two types of validation, it is also sometimes necessary to compare two clusterings, C_a and C_b, obtained by alternative procedures, and several statistical tests have been developed to solve such a task.

3.1. Validation of an Individual Cluster

A general definition of what a good cluster is depends on the shape of the cluster, that is, on how its members, each characterized by n statistical variables, are distributed in the n-dimensional space. Nevertheless, the main feature that a cluster must possess is the compactness. The m points must be close to each other and well separated from the other subjects, classified in different clusters. Such a feature may be monitored in many different ways, some of which have already been discussed in the sections devoted to the methods that can be used to determine the optimal number of clusters in the hierarchical, agglomerative clustering algorithms. Many other statistical tools have been developed to estimate the quality of an individual cluster. One of them, despite its simplicity, is nearly independent of the cluster shape and deserves particular
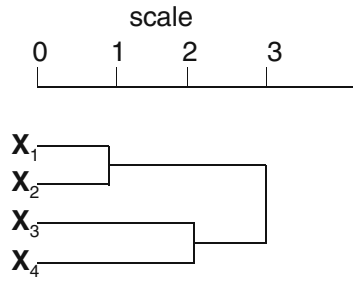
Fig. 11.7. Example of dendrogram used to compute the Lind index.

attention because of its versatility. It is known as the Lind index and it is based on the clustering lifetime, L. Such a quantity is defined as

$$L = e - b \qquad (17)$$

where b is the clustering level at which the cluster is formed and e is the clustering level at which the cluster is absorbed by a larger cluster. In **Fig. 11.7**, for example, the cluster containing the statistical units X_1 and X_2 is formed at the clustering level 1 and it disappears at the clustering level 3, when it is fused with the cluster containing the statistical units X_3 and X_4, which were merged into a cluster at the clustering level 2. The lifetime of the cluster containing the subjects X_1 and X_2 is therefore $L = 3 - 1 = 2$. On the contrary, the cluster of X_3 and X_4 has a lifetime $L = 1$, since it appears at the clustering level 2 and it disappears at the clustering level 3. This means that the cluster formed by the statistical units X_1 and X_2 is better than the cluster formed by X_3 and X_4, since its lifetime is larger ($L = 2 > L = 1$).

The Lind index, nevertheless, requires some standardization. If it is used as it is described above, it can just allow one to rank the quality of various individual clusters. This is certainly important but it does not allow one to decide if the cluster is good or not. The only way to accomplish such a task is to perform a series of cluster analyses on randomized subjects. By using, for instance, the example of **Fig. 11.7**, it is necessary to build up a large number u of ensembles of statistical units X_i ($1 \leq i \leq 4$), randomly generated. A number u of cluster analyses of the randomized statistical units is then performed. The lifetime of the cluster formed by the subjects X_1 and X_2 must be compared with the lifetimes of the randomized X_1 and X_2. If it exceeds a statistical threshold, the cluster formed by X_1 and X_2 is considered good. There are therefore two problems. On the one hand, it is necessary to design a randomization procedure and, on the other hand, it is necessary to design a statistics that is able to compare the real clustering with the randomizations. The first problem can be solved by a random number

generator. The second problem can be solved by using a standard statistical test, like, for example, the t-test. So, once a high number of random data sets have been clustered, a high number of clusterings is obtained and the distribution of these clusterings allows one to estimate the statistical significance of the analysis of the genuine, real data set.

3.2. Validation of a Clustering Level

Beside the validation of an individual cluster, it may be necessary to examine the entire partition, produced by a cluster analysis, in order to verify if it reproduces with sufficient fidelity the real structure of the statistical units. The real structure of m data is efficiently summarized by the $m \times m$ proximity matrix **P**, each element p_{ij} of which is the proximity between the ith and the jth statistical units. A similar matrix, the cophenetic matrix **C**, is able to efficiently summarize a hierarchical, agglomerative clustering process. Its elements indicate the level at which two statistical units are grouped, for the first time, into the same cluster. For example, given the dendrogram of **Fig. 11.8**, it is possible to build the following cophenetic matrix **C**

$$\mathbf{C} = \begin{pmatrix} 0 & 1 & 2 & 5 & 5 \\ 1 & 0 & 2 & 5 & 5 \\ 2 & 2 & 0 & 5 & 5 \\ 5 & 5 & 5 & 0 & 3 \\ 5 & 5 & 5 & 3 & 0 \end{pmatrix} \tag{18}$$
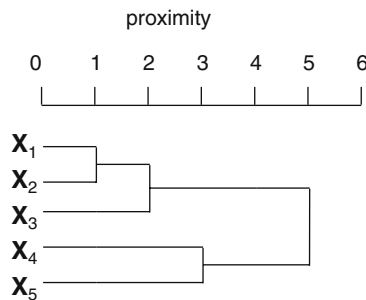
Fig. 11.8. Example of dendrogram used to compute the cophenetic matrix.

Obviously, it is symmetric relative to the main diagonal and all the elements c_{ii} are equal to 0. The element $c_{1,2}$ is equal to 1 because the units $\mathbf{X}_1$ and $\mathbf{X}_2$ meet at a proximity equal to 1. The element $c_{4,5}$ is equal to 3 because the units $\mathbf{X}_4$ and $\mathbf{X}_5$ are merged into the same cluster at a proximity equal to 3. And so on.

The comparison between the real structure of the data and the agglomerative clustering, produced by a cluster analysis, can be performed by comparing the proximity matrix $\mathbf{P}$ and the cophenetic matrix $\mathbf{C}$.

The most popular statistical index used for this task is the cophenetic correlation coefficient. Given that both $\mathbf{P}$ and $\mathbf{C}$ are symmetric and have the elements along the diagonal equal to 0 (it is assumed that the proximity is measured by means of a distance measure), it is sufficient to consider only the upper diagonal elements. If $\mathbf{P}$ and $\mathbf{C}$ contain m rows and m columns, $u = m(m-1)/2 = (mm-m)/2$. The cophenetic correlation coefficient CCC is defined as

$$CCC = \frac{\dfrac{\displaystyle\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\left(p_{ij}c_{ij}-\mu_{P}\mu_{C}\right)}{u}}{\sqrt{\dfrac{\left[\displaystyle\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\left(p_{ij}^{2}-\mu_{P}^{2}\right)\right]\left[\displaystyle\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\left(c_{ij}^{2}-\mu_{C}^{2}\right)\right]}{u^{2}}}} \qquad (19)$$

where p_{ij} and c_{ij} are the elements of $\mathbf{P}$ and $\mathbf{C}$, respectively, and μ_P and μ_C are their mean values, computed as

$$\mu_P = \frac{\displaystyle\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} p_{ij}}{u} \qquad (20)$$

$$\mu_C = \frac{\displaystyle\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} c_{ij}}{u} \qquad (21)$$

The values of the cophenetic correlation coefficient may range between -1 and $+1$. High values indicate a good agreement between the proximity matrix $\mathbf{P}$ and the cophenetic matrix $\mathbf{C}$. This happens if the real structure of the data is well translated into the hierarchical clustering. A major problem of CCC is that its value depends markedly on u, that is, on the number of statistical units that are analyzed. As usual, this kind of problems can be solved by determining, empirically, the distribution of the values that the quantity, in which we are interested in, can assume. In practice, it is necessary to build a large number of random data sets, make cluster analyses of them, and compute the cophenetic correlation coefficients. In this way, it is possible to give a statistical interpretation of the cophenetic correlation coefficient that has been computed on the basis of the ensemble of real statistical units.

A statistical index, different from the cophenetic correlation coefficient, is also widely used. It is usually known as the γ index. Again, since both the proximity matrix $\mathbf{P}$ and the cophenetic matrix $\mathbf{C}$ are symmetrical, it is enough to consider their upper

diagonal elements. Taken two pairs of elements of **P** and **C**, for example, p_{ij} and c_{ij}, on the one hand, and p_{kl} and c_{kl}, on the other; four possibilities can be encountered:

(a) $p_{ij} > c_{ij}$ and $p_{kl} > c_{kl}$,

(b) $p_{ij} < c_{ij}$ and $p_{kl} < c_{k}$,

(c) $p_{ij} > c_{ij}$ and $p_{kl} < c_{k}$,

(d) $p_{ij} < c_{ij}$ and $p_{kl} > c_{k}$.

In the first two cases, the matrices **P** and **C** are said to be concordant. In the last two cases, they are said to be discordant. The γ statistical index is defined as

$$\gamma = \frac{N_p - N_m}{N_p + N_m} \tag{22}$$

where N_p the number of times in which the **P** and **C** matrices are concordant, and N_m is the number of times in which they are discordant. By definition, the values of the γ index may range from -1 to $+1$, and high values indicate a good agreement between the data structure, monitored by the proximity matrix **P**, and the derived hierarchical clustering, monitored by the cophenetic matrix **C**. Unfortunately, like for the cophenetic correlation coefficient, also in the case of the γ index, the statistical significance of an observed γ value cannot be estimated with certainty. Again, it is therefore necessary to simulate a large number of data sets and perform many cluster analyses, in order to get the distribution of the possible γ values and assign an accurate, statistical meaning to the γ index computed in the cluster analysis of the real data.

3.3. Comparison Between Alternative Clusterings

It may be necessary to compare two partitions of the data obtained by two different clustering procedures. In some cases, one of the two partitions can be known a priori. The comparison between the results of a cluster analysis and an a priori known partition may be needed to assess the performance of a certain cluster-analysis method.

The simplest approach to compare two partitions is based on the analysis of all pairs of statistical units. Two subjects can be grouped together in both partitions; they can be grouped together in the first partition and not in the second; they can be grouped together in the second partition and not in the first; or they can be classified in different groups in both partitions. In the example of **Fig. 11.9**, the cluster analysis of an ensemble of four statistical units results in two clusters, one with three subjects and one with a single element, by using a certain clustering method (first partition). By means of a different algorithm, a different partition (second partition) is obtained, where two clusters contain two
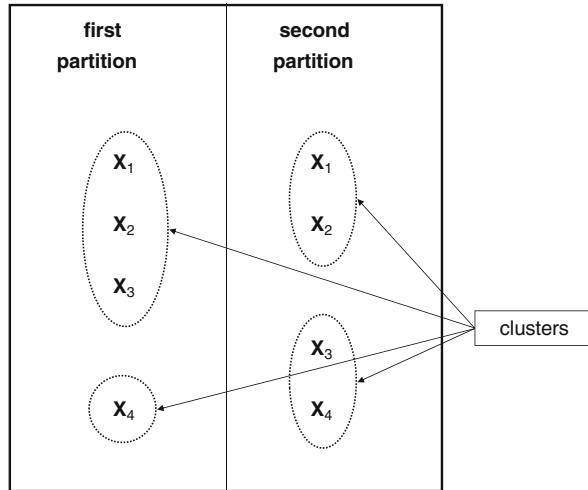
Fig. 11.9. Example of alternative clusterings of four statistical units.

statistical units each. Each pair of statistical units can therefore be classified into one of following four types: tt (together–together), ts (together–separated), st (separated–together), and ss (separated–separated), as a function of their relationships in the two partitions that are examined. In the example of **Fig. 11.9**, for instance, the pair of subjects X_1 and X_2 are classified in the same cluster in both partitions and it is therefore of type tt. An example of pair of type ss is made by X_1 and X_4, which are classified in different clusters in both partitions. The statistical units X_1 and X_3 are of type ts since they are grouped together in the first partition and are separated in the second partition. An example of pair of type st is given by X_3 and X_4, which are separated in the first partition while they are grouped together in the second. A complete classification of the six unique pairs of subjects of **Fig. 11.9** is given in **Table 11.4**.

Table 11.4
Classification of the six unique pairs of Fig. 11.9

Subject	X_1	X_2	X_3	X_4
X_1	—	tt	ts	ss
X_2	—	—	ts	ts
X_3	—	—	—	st
X_4	—	—	—	—

Once all pairs have been classified as tt, ts, st, or ss, it is possible to define different statistical indices. For example, the Fowlkes and Mallows index, FM, is defined as

$$FM = \frac{tt}{\sqrt{(tt + ts)(tt + st)}} \tag{23}$$

Alternatively, it is possible to use the Rand coefficient R, defined as

$$R = \frac{tt + ss}{tt + st + ts + ss} \tag{24}$$

or the Jaccard statistic, J, define as

$$J = \frac{tt}{tt + ts + st} \tag{25}$$

The three coefficients range between 0, if the two partitions that are compared are completely different, and 1, if the two partitions that are compared are statistically identical.

A more complex index is the Hubert statistic, H, defined as

$$H = \frac{M \times tt - (tt + ts)(tt + st)}{\sqrt{(tt + ts)(tt + st)[(M - (tt + ts))][M - (tt + st)]}} \tag{26}$$

where $M = tt + ts + st + ss$. In all these statistics, the exact meaning of what is computed is nevertheless unclear. In other words, only a simulation, consisting of a very large number of cluster analyses of randomized data sets, allows one to evaluate the statistical significance of the observed coefficients.

4. Clustering Tendency

The term clustering tendency refers to the problem of deciding whether a set of subjects has an intrinsic predisposition to cluster into distinct groups. This is also referred to as the spatial randomness problem. In formal terms, there are two extreme hypotheses;

(a) the data are intrinsically aggregated (mutual attraction),

(b) the data are randomly arranged (mutual repulsion).

There are two prominent approaches to test these hypotheses, the Hopkins and the Cox–Lewis statistics. Given m subjects, in a n-dimensional space, $k << m$ random geometrical points and subjects are selected. If u_i is the distance between the ith random point and its closest subject (*see* **Fig. 11.10**) and if w_i is the distance between the ith subject randomly selected and its closest subject (*see* **Fig. 11.10**), it is possible to define
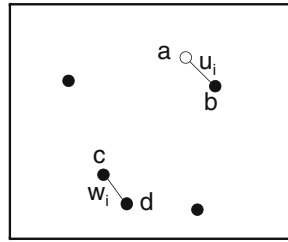
Fig. 11.10. Example of Hopkins statistic. Filled circles represent subjects in a bidimensional space. Open circles represent geometrical points. u_i is the minimal distance between a geometrical point (**a**) and a subject (**b**). w_i is the minimal distance between a subject (**c**) and another subject (**d**). Both a and c are randomly selected.

$$U = \sum_{i=1}^{k} u_i \qquad (27)$$

and

$$W = \sum_{i=1}^{k} w_i \qquad (28)$$

and the Hopkins coefficient H is defined as

$$H = \frac{U}{U + W} \qquad (29)$$

Values of H around 0.5 are expected for randomly distributed data sets (extreme hypothesis b), because $U \approx W$. Values close to 1 are expected for a well-clustered data set (extreme hypothesis a), because W will be close to 0. It is in general accepted that the statistical units can be considered to be naturally clustered if $H \geq 0.75$.

The Cox–Lewis coefficient is defined in a very similar way. Given m statistical units, each characterized by n variables, $k << m$ geometrical points are randomly selected in the n-dimensional space. The lowest distance u_i between the ith point and one of the m units is then recorded together with the minimal distance w_i between such a subject and another unit (*see* **Fig. 11.11**). The ratio R_i

$$R_i = \frac{u_i}{w_i} \qquad (30)$$

is then computed and the average values R of the k R_i values

$$R = \frac{\sum_{i=1}^{k} R_i}{k} \qquad (31)$$

are the Cox–Lewis coefficient. R values close to 1 are expected for uniformly distributed data, because $w_i \approx u_i$ (extreme hypothesis b). Values much larger than 1 are on the contrary expected to arise if the subjects tend to cluster into well-defined groups (extreme hypothesis a).
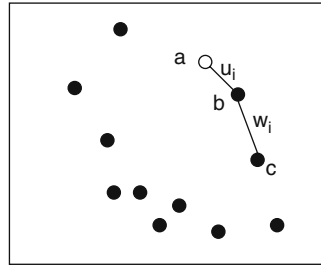
Fig. 11.11. Example of Cox–Lewis statistic. Filled circles represent subjects in a bidimensional space. Open circles represent geometrical points. u_i is the minimal distance between a geometrical point (**a**) and a subject (**b**). w_i is the minimal distance between the same subject (**b**) and another subject (**c**). The geometrical point a is randomly selected.

A completely different approach, useful especially for spherical or nearly-spherical clusters, is followed by the Lacey-Cole statistic. Given an ensemble of m statistical units, the $m(m-1)/2$ distances between each unique pair of subjects are computed and their distribution is analyzed (*see* **Fig. 11.12**). If the statistical units tend to be grouped in clusters, such a distribution is bi-modal and the minimum intercalated between the two picks indicates roughly the cluster dimension.

A more sophisticated and computationally expensive statistic is the Index$_{MST}$, which is based on graph theory. An ensemble of m subjects can be considered a graph, in which each unit is a vertex and each pair of units defines an edge, characterized by the distance between the two subjects. It is thus possible to build the minimum spanning tree, which is a tree where $m-1$ edges connect all the m vertices and where the overall sum of the distances between connected vertices is minimal. A practical approach to build a minimum spanning tree is provided by the Prism's algorithm. An object X_1 is randomly selected amongst the m units. Its closest subject, X_2, is searched for and the units X_1 and X_2 are considered to be connected by an edge of the minimum spanning
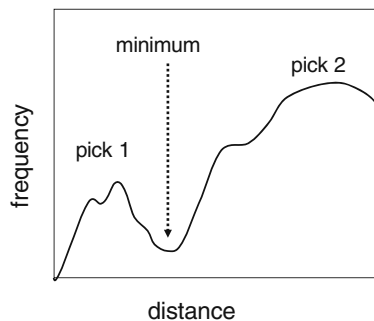
Fig. 11.12. Example of Lacey-Cole statistic. The distribution of the distances between pairs of subjects is bimodal, with an evident minimum of frequency of observations.

tree. Then, the subjects X_3, which is the element closest to X_1, and X_4, which is the subject closest to X_2 are searched amongst the remaining $m-2$ statistical units. If the distance between X_1 and X_3 is lower than the distance between X_2 and X_4, the element X_3 is included into the minimum spanning tree, which is therefore constituted by the three vertices X_1, X_2, and X_3 an by two edges, one between X_1 and X_2 and one between X_1 and X_3. On the contrary, if the distance between X_1 and X_3 is larger that that between X_2 and X_4, the latter subject is included into the minimum spanning tree, which consists of the three vertices X_1, X_2, and X_4 and by two edges, one between X_1 and X_2 and one between X_2 and X_4. The problem is then iterated until all the statistical units have been included into the minimum spanning tree.

The distribution of the lengths of the edges of the minimum spanning tree is then analyzed in order to determine the distance d_{crit}, over which there are only 5% of the edge lengths. The clustering tendency is then estimated with the $Index_{MST}$, defined as

$$Index_{MST} = \sum_{d > d_{crit}} \left(\frac{d}{d_{crit}} - 1 \right) \tag{32}$$

where d are the edge lengths larger than d_{crit}. For well-clustered data sets (extreme hypothesis a), it is very probable that a considerable fraction of the d values will be much higher than d_{crit}. The $Index_{MST}$ will therefore be higher than 1. On the contrary, in uniformly distributed data (extreme hypothesis b), the d values will not be much higher than d_{crit} and consequently, the $Index_{MST}$ will approach 0.

5. Monotonicity and Crossover

The agglomerative, hierarchical clustering algorithms do not provide automatically the optimal partition of an ensemble of statistical units. They rather result in dendrograms that summarize the progressive grouping on the subjects. An insidious phenomenon that sometimes occurs is the crossover. In principle, the series of partitions that are produced, sequentially, by such algorithms must be monotonous. Such a condition, called monotonicity, implies that each cluster is formed at a higher dissimilarity level than any one of its components. For example, if cluster A, formed in a previous step of the procedure by merging of clusters A_1 and A_2, and cluster B, formed previously by fusion of clusters B_1 and B_2, are grouped together into cluster $C = A + B$, the distance between A and B must be higher than the distance between A_1 and A_2, on the one hand, and the distance between B_1 and B_2, on the other hand.
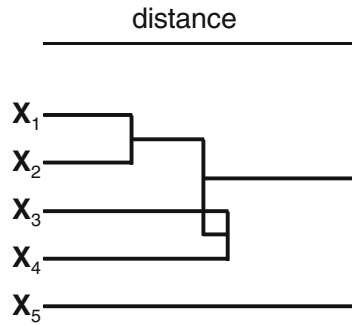
Fig. 11.13. Example of a dendrogram that summarizes a clustering with crossover.

Because of the variability of definitions of proximity between two groups of statistical units, the contrary may happen. In other words, it is possible that $d(A,B) \leq d(A_1,A_2)$ or that $d(A,B) \leq d(B_1,B_2)$, where $d(X,Y)$ is the distance between cluster X and cluster Y. This undesirable phenomenon is called crossover. A dendrogram with a crossover is shown in **Fig. 11.13**, where the cluster formed by X_1 and X_2 is merged into the cluster containing the elements X_3 and X_4 at a distance minor than that at which the units X_3 and X_4 merge together. The algorithms UPGMC and WPGMC are particularly prone to lead to dendrograms with crossovers, though this is absolutely not the rule. It is nevertheless always necessary to pay some attention and verify that the monotonicity rule is not violated.

Further Reading

Theodoris, S., Koutroumbas, K. (2003) *Pattern Recognition*. Academic Press, Amsterdam.

Corrigan, M. S. (2007) *Pattern Recognition in Biology*. Nova Science, Lancaster.

Kaufman, L., Rousseeuw, P. J. (2005) *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, New York.

Romesburg, H. C. (2004) *Cluster Analysis for researchers*. Lulu Press, North Carolina.

Everitt, N. S., Landau, S., Leese, M. (2001) *Cluster Analysis*. Hodder Arnold Publications, Oxford.

Chapter 12

Neural Networks

Zheng Rong Yang

Abstract

Neural networks are a class of intelligent learning machines establishing the relationships between descriptors of real-world objects. As optimisation tools they are also a class of computational algorithms implemented using statistical/numerical techniques for parameter estimate, model selection, and generalisation enhancement. In bioinformatics applications, neural networks have played an important role for classification, function approximation, knowledge discovery, and data visualisation. This chapter will focus on supervised neural networks and discuss their applications to bioinformatics.

Key Words: Neural networks, learning rule, learning algorithm, regression, classification, evaluation, generalization, cross-validation, bioinformatics.

1. Introduction

Neural networks are a class of computational algorithms mimicking human brain with the support of modern fast and sometimes parallel computational facility. In terms of this, neural networks are regarded as a class of information processing systems as well. The interpretation of this is that neural networks can reconstruct an unknown function using the available data without any prior knowledge about function structures and parameters. Information processing has two meanings. The first is that neural networks can help to estimate function structures and parameters without domain experts involved. This is perhaps the most important reason for neural networks being so popular in many areas. The second is that neural networks are a class of intelligent learning machines, which can store knowledge through learning as human brain for pattern recognition, decision making, novelty detection

and prediction. Combining these two important factors, neural networks then become a powerful computational approach for handling data for various learning problems.

Neural network studies and applications have experienced several important stages. In the early days, neural network studies only focused on theoretical subjects, i.e. investigating if a machine can replace human for decision-making and pattern recognition. The pioneer researchers are Warren McCulloch and Walter Pitts (1) where they showed the possibility of constructing a net of neurons which can interact to each other. The net was based on symbolic logic relations. **Table 12.1** shows one of McCulloch and Pitts OR logic, where the output is a logic OR function of two inputs.

Table 12.1
McCulloch and Pitts OR logic

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

This earlier idea of McCulloch and Pitts was not based on rigorous development as indicated by Fitch (2) that "in any case there is no rigorous construction of a logical calculus". However, the study on neural networks was continuing. For instance, Hebb in his book published in 1949 gave the evidence that the McCulloch–Pitts model certainly works (3). He showed how neural pathways can be strengthened whenever it is activated. In his book, he indicated that "when an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolite change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased". In 1954, Marvin Minsky completed his doctorial study on neural networks. His dissertation was titled as "Theory of Neural-Analog Reinforcement Systems and its Application to Brain-Model Problem". Later he published a paper about this work in a book (4). This triggered a wide scale of neural network research. In 1958, Frank Rosenblatt built a computer at Cornell University called the Perceptron (later being called single-layer perceptron), which can learn new skills by trial and error through mimicking human thought process. However, this work was evaluated by Minsky in 1969 (5) showing its incapability in dealing with complicated data. Minsky's book then blocked the further study of neural networks for many years.

In the period of 1970s and 1980s, neural network research was in fact not completely ceased. For instance, the self-organising map (6) and the Hopfiled net were intensively studied (7). In 1974, Paul Werbos conducted his doctorial study at Harvard University and studied the training process called back propagation of errors. The work was published later in his book (8). This important contribution led to the work of David Rumelhart and his colleagues in the 1980s. In 1986, the back propagation algorithm was introduced by Rumelhart and his colleagues with the implementation called the delta rule for supervised learning problems (9). From this, neural networks became very popular for data mining or machine learning in both theoretical studies and practical exercises.

The most important contribution of Rumelhart and his colleagues' work is that a simple training or learning algorithm based on trial-and-error principle has been implemented and has demonstrated its powerfulness in dealing with problems which were declared impossible by Minsky in 1969. In contrast to Rosenblatt's single-layer perceptron (SLP), Rumelhart's model is called multilayer perceptron (MLP) where the most important difference is the introduction of hidden neurons.

Shown in **Fig. 12.1** (a) is a structure of an SLP, where there are three input neurons named as x1, x2, and x3 with a single output neuron named as y. In contrast, a structure of an MLP is shown on the right panel in **Fig. 12.1**, where in addition to three input neurons and an output neuron, three hidden neurons named as z1, z2, and z3 are inserted between the input and output neurons. Generally, x1, x2, and x3 represent observed values for

(a) SLP   (b) MLP
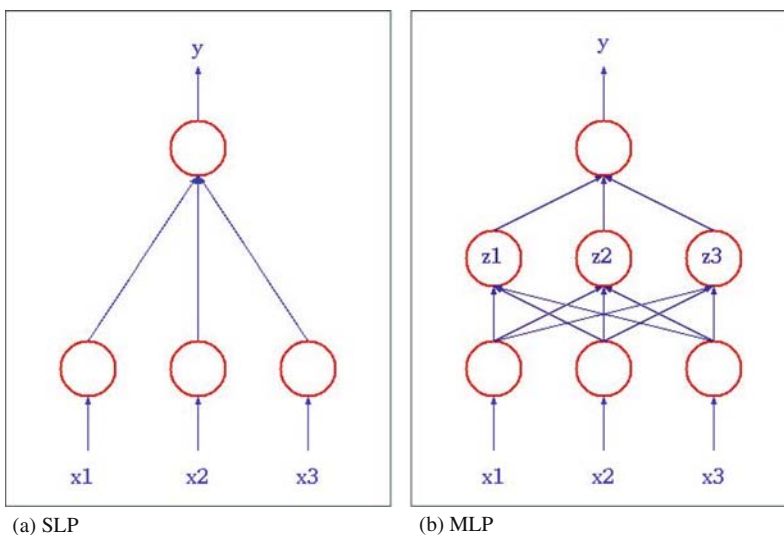
Fig. 12.1. SLP and MLP structure.

three independent variables (or the input variables) while y corresponds to observed values for a dependent variable (or the output variable). The hidden neurons represent variables which are not observed. We will see later in this chapter that the introduction of hidden neurons makes it possible to model nonlinear data.

Practically, there are three related subjects in using neural networks for any applications. They are model construction, model selection, and model evaluation. Before discussing these three related practical issues, we need to know other three theoretical issues. They are parameter estimate, learning rule, and learning algorithms. Parameter estimate is a learning process by which knowledge in data is extracted and expressed quantitatively in a neural network model. The extracted knowledge is ultimately used for making predictions on unseen data. The requirements for this are the accuracy and the robustness of the parameters. In most cases, we have no idea what values should be assigned to model parameters when we have data only. This means that the data obtained are the only source for us to estimate model parameters. Besides we need to determine an optimised model structure to represent true knowledge hidden in data. This involves neural learning. For different supervised learning projects, different learning algorithms are needed. Moreover, there might be many variants in one type of supervised learning. For instance, in classification analysis, we may have simple or complicated data distribution. For these two kinds of applications, different neural learning algorithms may be employed. Even for complicated classification projects, we may have a data set where the number of data points is much larger than the number of variables or we may have a data set where the number of data points is much less than the number of variables. For these two kinds of applications, different treatments are then needed for applying a proper supervised learning algorithm. There are various learning rules for use. Some are based on numerical methods and some are based on statistical approaches. Some are fast for some types of data and some are accurate for some types of data. We will focus on numerical approaches for deriving neural network learning rules.

2. Learning Theory

2.1. Parameterisation of a Neural Network

A neural network without parameters will have no capability of associative memory. In particular, a neural network whenever its structure has been determined must possess the power for prediction in a supervised learning project. In order to make a neural network capable of prediction, it must have parameters which represent processed information. We explain this by a simple

example. Suppose we are interested in studying if a metabolite in a specific pathway is related with its upstream genes. We first denote this metabolite as y. Meanwhile, we denote three upstream genes for the metabolite y as x_1, x_2, and x_3. Suppose we have had some observations for x_1, x_2, x_3, and y. Our objective is to construct a model which can establish the relationship between x_1, x_2, x_3, and y as a predictive function $y = f(x_1, x_2, x_3)$. If $f(x_1, x_2, x_3)$ is properly parameterised, say $y = f(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3)$, where w_0 is a bias term, w_1, w_2 and w_3 are parameters for three upstream genes, we can make a prediction whenever we have new values for x_1, x_2, and x_3, respectively.

It is normally believed that parameters in a neural network model represent the knowledge in data. For instance, if a neural network model is expressed as $y = \sigma(0.1 + 0.03 x_1 + 5.1 x_2 + 0.002 x_3)$, all three input variables have the same magnitude and $\sigma(z)$ is a monotonic linear function of z, i.e. $\sigma(z) \propto z$, we can believe that x_2 plays a key role for y. In other words, x_2 is the dominant factor for y and ignoring the other two input variables will not lead to loss of much precision in prediction whilst making a predictive system less loaded.

2.2. Learning Rules

Before discussing learning rules, we need to establish a proper objective function. There are normally two types of objective functions for supervised learning. They are the square error function and the cross-entropy function. The former is used for regression analysis, which addresses a type of problems of continuous function approximation. The latter is used for classification analysis, which addresses a class of applications of data partitioning.

2.2.1. Regression Analysis

We normally denote a regression function as

$$y_n = f(\mathbf{x}_n, \mathbf{w}) \tag{1}$$

Here $\mathbf{w} \in \Re^H$ is a numerical parameter vector of H dimensions and $\mathbf{x}_n \in \Re^D$ is a numerical input vector of D dimensions describing the nth object in a data set, where $\Re$ is the real number set. Correspondingly, $y_n \in \Re$ is the model output for $\mathbf{x}_n$. H is heavily depending on a model's structure, which will be discussed later in this chapter. For $\mathbf{x}_n$, we normally have its observed phenotypic property called target $t_n \in \Re$. Note that t_n does not represent a true value in most cases. Normally, it is called a corrupted function value. For instance, a true function is a sin function $5\sin(x)$. We may have observed corrupted values from a noise added sin function $5\sin(x) + G(0, 1)$, where $G(0, 1)$ is called a white noise. The existence of noise is normally unavoidable in many experiments. Many factors can result in noise. In order to estimate the parameter vector $\mathbf{w}$, we need to make the distance between y_n and t_n ($t_n - y_n$)

as small as possible during learning. Based on this, we have commonly used the square error function (mean square error function) for regression analysis as below

$$\varepsilon = \frac{1}{\ell} \sum_{n=1}^{\ell} (t_n - y_n)^2 \tag{2}$$

Here ℓ is the number of observed pairs (x_n, t_n). A learning rule must ensure the model parameter vector satisfying

$$\tilde{\mathbf{w}} = \min_{\text{arg}} \left\{ \frac{1}{\ell} \sum_{n=1}^{\ell} (t_n - f(\mathbf{x}_n, \hat{\mathbf{w}}))^2 \right\} \forall \hat{\mathbf{w}} \in \Re^H \tag{3}$$

Here $\hat{\mathbf{w}}$ is a vector (a point) in a H-dimensional space (called a parameter space) and $\tilde{\mathbf{w}}$ is the optimal vector among many (normally infinite) $\hat{\mathbf{w}}$s.

2.2.2. Classification Analysis

In classification, we will use a different objective function if the model output y_n is constrained in the interval $[0, 1]$. We will see later in this chapter that neural networks employing the sigmoid function can easily fulfil this requirement. The cross-entropy function is normally employed for classification analysis suppose there are only two classes, where $t_n \in \{0, 1\}$

$$O = \prod_{n=1}^{\ell} y_n^{t_n} (1 - y_n)^{1-t_n} \tag{4}$$

In most cases, negative logarithm is applied to this objective function leading to

$$O = - \sum_{n=1}^{\ell} t_n \log y_n + (1 - t_n) \log(1 - y_n) \tag{5}$$

A learning process is aiming to minimise this objective function so that

$$\tilde{\mathbf{w}} = \min_{\text{arg}} \left\{ - \sum_{n=1}^{\ell} t_n \log f(\mathbf{x}_n, \hat{\mathbf{w}}) + (1 - t_n) \log(1 - f(\mathbf{x}_n, \hat{\mathbf{w}})) \right\} \tag{6}$$

$$\forall \hat{\mathbf{w}} \in \Re^H$$

It is then obvious that we have to analyse the function $f(\mathbf{x}, \mathbf{w})$ before discussing the learning rule. In neural networks, the sigmoid function is normally used for $f(\mathbf{x}, \mathbf{w})$ because it has two advantages, i.e. being derivable and parallelism. The former makes it possible to apply conventional numerical approximation approaches which heavily depend on derivatives to parameter estimate and the latter makes it possible to use parallel computing

techniques because the calculation of each neuron output is completely independent from the calculations of other neuron's outputs in the same layer. The sigmoid function is defined as below

$$f(z) = \frac{1}{1 + \exp(-z)} \tag{7}$$

It is not very difficult to see that the sigmoid function squashes the value of z into the interval $(0,1)$ as

$$\lim_{z \to -\infty} \frac{1}{1 + \exp(-z)} = 0 \tag{8}$$

and

$$\lim_{z \to +\infty} \frac{1}{1 + \exp(-z)} = 1 \tag{9}$$

In addition, the other advantage of the sigmoid function is that its derivative is easily calculated as the entropy as below

$$\frac{df(z)}{dz} = f(z)(1 - f(z)) \tag{10}$$

We now use regression analysis as an example for the analysis of the learning rule. In most cases, we will have no knowledge of what values should be assigned to model parameters. Like statistical learning, neural learning also starts from a random guess, i.e. assigning random values to model parameters (called initialisation) and based on these random parameters, we start to search the way by which an objective function can be decreased, hence bringing the current model parameters ($\hat{\mathbf{w}}$) more closer to the optimal solution ($\tilde{\mathbf{w}}$). As we know the regression analysis adopts a quadratic-like objective (error) function. In a quadratic function, there will be always some relationship between the derivatives and the optimal solution. **Figure 12.2** shows such a relationship for a case where there is only one model parameter. Two filled dots are the possible random guesses. It can be seen that the optimal model parameter must sit in the bottom of the valley of the
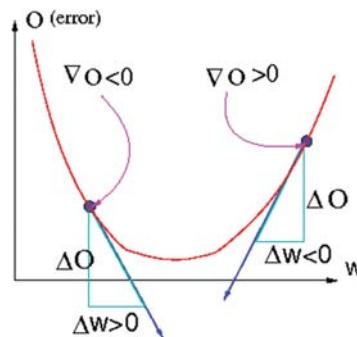
Fig. 12.2. The relationship between a model output using the current model parameter and the direction of the optimal model parameter.

quadratic objective function. The slop (first derivative) denoted by a straight line of the random guess on the left side of the optimal solution shows a negative sign while the slop denoted by another straight line of the random guess on the right side of the optimal solution shows a positive sign. The negative sign means that when the value of w increases, the error O decreases. The positive sign means that when the value of w increases, the error O increases. From this, it can be seen that we must increase the model parameter when the slop of the model output based on the current model parameter shows a negative sign. We must decrease the model parameter when the slop of the model output based on the current model parameter shows a positive sign. The slop of model output is mathematically defined as the first derivative of the objective function with respect to the model parameter as below

$$\nabla O = \frac{\mathrm{d}O}{\mathrm{d}w} \tag{11}$$

Before defining the quantitative learning rule which will be used to update model parameters stochastically, we need to analyse the qualitative relationship between parameter change and the slop. The next thing is to determine the learning rule quantitatively.

If the change (increase or decrease) on w is denoted by Δw, we then have a qualitative relationship from **Fig. 12.3** that if the absolute value of the slop is larger, the current position is more
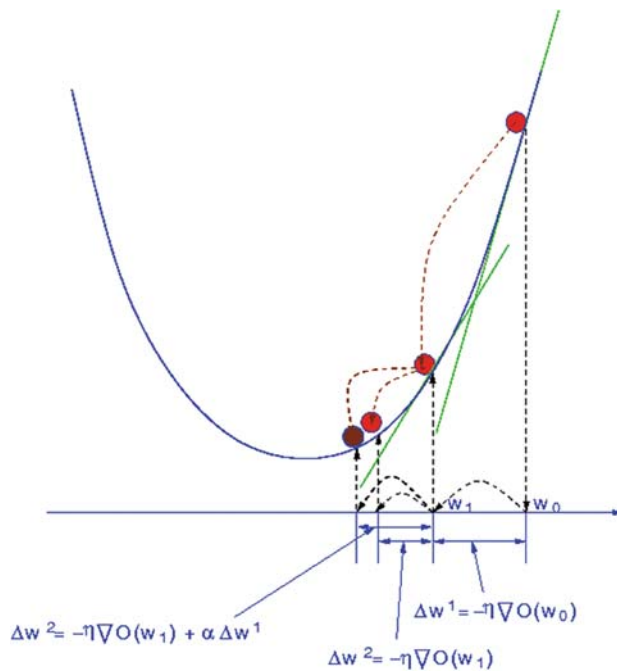
Fig. 12.3. The quantitative relationship between slop and the magnitude of model parameter change.

departed from the optimal solution and if the absolute value of the slop is smaller, the current position is closer to the optimal solution. When w is closer to the optimal solution, we must have a smaller change on w so that we will not miss the optimal solution. When w is more departed from the optimal solution, we can have a larger change of w. From this, we then have a qualitative learning rule defined as below

$$|\Delta w| \propto |\nabla O| \qquad (12)$$

Quantitatively, the learning rule (also called the delta rule) is defined as below

$$\Delta w = -\eta \nabla O \qquad (13)$$

Here $\eta \in (0, 1)$ is called the learning rate.

The delta rule may not be always working properly. It is quite often that a new solution of w may miss the optimal solution. For instance, the new solution w_1 generated using the delta rule from w_0 misses the optimal solution, i.e. the valley of the quadratic curve as seen in **Fig. 12.4**. From w_1 the delta rule will lead to w_2^A which again misses the optimal solution. However, we have noticed that the first derivatives at w_0 and w_1 have different signs meaning that they have a complementary function. If the first derivative at w_{t+1} has a different sign as the one at w_t, it means that w_t and w_{t+1} are sitting on the opposite sites of the optimal solution; *see* **Fig. 12.2**. The move from w_{t+1} to w_{t+2} may miss the optimal solution again. If we can correct the move from w_{t+1} to w_{t+2} using a momentum
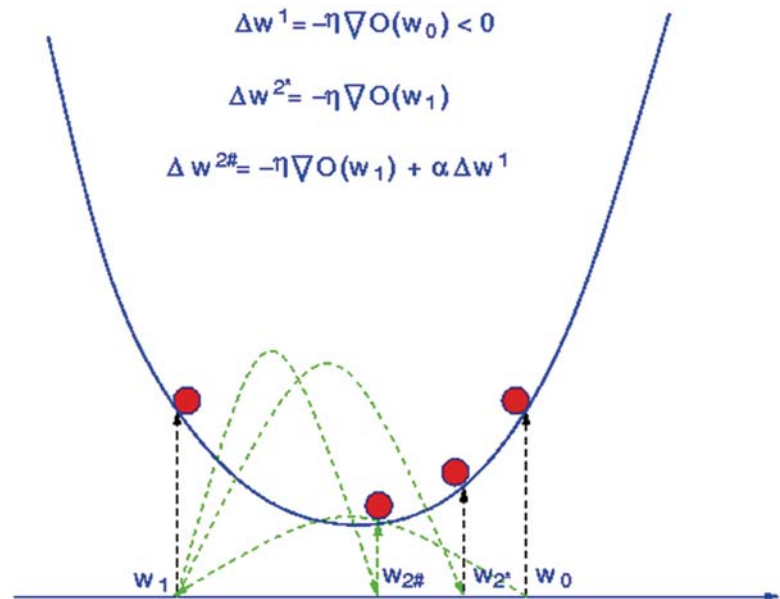
Fig. 12.4. The illustration of the use of the momentum factor for fast learning.

which has a different first derivative sign as the one at w_{t+1}, such risk can be possibly reduced. Remember that the first derivative at w_t is different as the one at w_{t+1}, we can design a revised delta rule for this purpose

$$\Delta w^{t+1} = -\eta \nabla O^t + \alpha \Delta w^t \qquad (14)$$

Here Δw^{t+1} is the update of w at time $t+1$, Δw^t is the update of w at time t, ∇O^t is the first derivative of O with respect to w at time t and $\alpha \in (0, 1)$ is a positive number called the momentum factor. In **Fig. 12.4**, we can see that this revised delta rule can reduce this risk. This time, the move from w_1 is to w_2^B rather than w_2^A.

According to Equation [14], we can see that

$$\Delta w^{t+1} = -\eta \left(\nabla O^t + \alpha \nabla O^{t-1} \right) + \alpha^2 \Delta w^{t-1} \qquad (15)$$

From the above equation, we can conclude two aspects. First, if ∇O^t and ∇O^{t-1} have the same sign, the previous update instruction (∇O^{t-1}) will enhance the new update instruction (∇O^t) otherwise ∇O^{t-1} will reduce the impact of ∇O^t. Second, if Δw^{t+1} and Δw^{t-1} have the same sign, Δw^{t-1} will enhance Δw^{t+1}. Otherwise, Δw^{t-1} will reduce the impact of Δw^{t+1}.

In using the delta rule or the revised delta rule, the user needs to tune the learning rate and the momentum factor to proper values. This is not an easy job. There is another numerical method which uses second derivative information for weight update, where we normally do not need the learning rate and the momentum factor. Shown in **Fig. 12.5** , we can see that the weight update amount for the case in the left panel should be smaller than that for the case in the right panel. If we use the same amount of weight update for both cases, the left panel case may have missed the optimal solution while the right panel may not. This is because the right panel case shows a small curvature while the left case demonstrates a large curvature. A point located in a large curvature area means that it is close to the optimal solution. A point located
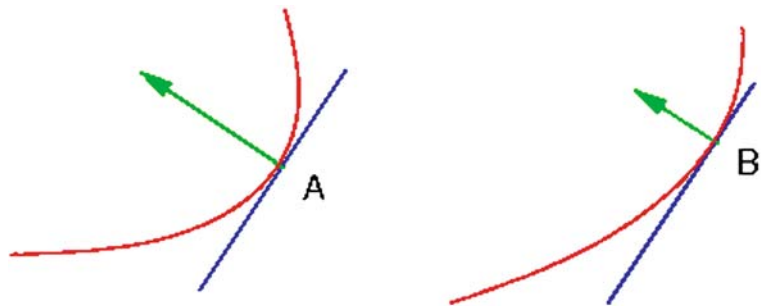
Fig. 12.5. Illustration of using second-derivative information for weight update.

in a small curvature area means that it may be far away from the optimal solution. As we know the second derivative can be used to quantify function curvatures. This means that

$$|\Delta w| \propto |\nabla O| |\Delta w| \propto \frac{1}{|\nabla \nabla O|} \tag{16}$$

Here ∇O and $\nabla \nabla O$ are the first and second derivatives with respect to w, respectively. The update rule using the second derivative information is called the Newton–Raphson method. In application to neural network parameters, it is illustrated as below

$$\Delta w = -\frac{\nabla O}{\nabla \nabla O} \tag{17}$$

or

$$\Delta \mathbf{w} = -\mathbf{H}^{-1} \nabla O \tag{18}$$

Here $\mathbf{w}$ is a weight vector and $\mathbf{H}$ is called a Hessian matrix of second derivatives as below

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 O}{\partial w_1 \partial w_1} & \frac{\partial^2 O}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 O}{\partial w_1 \partial w_m} \\ \frac{\partial^2 O}{\partial w_2 \partial w_1} & \frac{\partial^2 O}{\partial w_2 \partial w_2} & \cdots & \frac{\partial^2 O}{\partial w_2 \partial w_m} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 O}{\partial w_m \partial w_1} & \frac{\partial^2 O}{\partial w_m \partial w_2} & \cdots & \frac{\partial^2 O}{\partial w_m \partial w_m} \end{pmatrix} \tag{19}$$

where $\frac{\partial^2 O}{\partial w_i \partial w_j}$ is the second derivative of O with respect to w_i and w_j.

2.3. Learning Algorithms

In this subsection, we discuss two learning algorithms for regression and classification analyses respectively, where different objective functions are used.

2.3.1. Regression

In regression analysis, the target variable is commonly a numerical variable $t_n \in \Re$ (or $t_n \in [0, 1]$). In this case, the least mean square error function is used as the objective function as seen in equation [2]. Using the revised delta rule (Equation [14]), we then have two update rules as below. First, the update rule for the weights between hidden neurons and an output neuron (for instance, between output neuron y and hidden neurons z_1, z_2, z_3 in **Fig. 12.1**) is

$$\Delta \mathbf{w}_0^{t+1} = \eta \mathbf{Z}^T \mathbf{B} \mathbf{e} + \alpha \Delta \mathbf{w}_0^t \tag{20}$$

Here $\mathbf{w}_0 = (w_{01}, w_{02}, \cdots, w_{0H})^T$ is the hidden weight vector with w_{0h} connecting the hth hidden neuron to the output neuron, $\mathbf{e} = (e_1, e_2, \cdots, e_\ell)^T$ is the error vector with $e_n = t_n - y_n$, $\mathbf{B} = diag\{y_n(1 - y_n)\}$ is the diagonal entropy matrix of outputs

with ℓ rows and ℓ columns, and $\mathbf{Z}$ is the matrix recording the outputs from all the hidden neurons with ℓ rows and H columns (H hidden neurons). Second, the update rule for the weights between input neurons and the hth hidden neuron (for instance, between the hidden neuron z_1 and input neurons in **Fig. 12.1**) is shown as below

$$\Delta \mathbf{w}_h^{t+1} = w_{0\,h}\eta\mathbf{X}^T\mathbf{B}\mathbf{Q}_h\mathbf{e} + \alpha\Delta\mathbf{w}_h^t \qquad (21)$$

Here $\mathbf{w}_h = (w_{h1}, w_{h2}, \cdots, w_{hD})^T$ is the input weight vector with w_{hd} connecting the hth hidden neuron to the dth input neuron, $\mathbf{Q}_h = diag\{z_{nh}(1 - z_{nh})\}$ is the diagonal entropy matrix for the hth hidden neuron with ℓ rows and ℓ columns, and $\mathbf{X}$ is the matrix recording all the input vectors, i.e. having ℓ rows and D columns (ℓ input vectors and D input variables).

2.3.2. Classification

For a classification problem, the target variable is commonly a discrete variable $t_n \in I$ with I_1 meaning integers. We study discrimination problems where $t_n \in \{0, 1\}$ in this chapter. The cross-entropy function is commonly used as the objective function for classification projects as seen in equation [4]. Applying the revised delta rule to equation [5], we will also have two update rules. First, the update rule for the weights between hidden neurons and the output neuron if we have one output neuron (for instance, between the output neuron y and hidden neurons z_1, z_2, z_3 in **Fig. 12.1**) is

$$\Delta \mathbf{w}_0^{t+1} = \eta\mathbf{Z}^T\mathbf{e} + \alpha\Delta\mathbf{w}_0^t \qquad (22)$$

Second, the update rule for the weights between input neurons and the hth hidden neuron (for instance, between hidden neuron z_1 and input neurons in **Fig. 12.1**) is

$$\Delta \mathbf{w}_h^{t+1} = w_{0\,h}\eta\mathbf{X}^T\mathbf{Q}_h\mathbf{e} + \alpha\Delta\mathbf{w}_h^t \qquad (23)$$

2.3.3. Procedure

During learning, the above equations [20, 21, 22, 23] will be used iteratively until some criteria are satisfied. The learning procedure will be

- Step 1, Initialisation: assigning random values to all network parameters;
- Step 2, Estimation: estimate model outputs and errors by feeding input vectors;
- Step 3, Update: update all the model parameters using the above update rules;
- Step 4, Check: check if the desired criteria are satisfied, if so stop, otherwise go to Step 2.

There are commonly three stop criteria for use. They are the maximum learning cycle, the error threshold, and the stability. If the learning cycle has exceeded the maximum learning cycle, a

learning process will be terminated. In some situations, if the training error has already been below the desired error threshold, a learning process will also be halted. For some complicated learning problems, we may not be interested to reach the maximum learning cycle and may not be able to set a proper error threshold. In this case, we can check if the change of weights is small enough. There can be two reasons when there is nearly no change on weights. First, a model has been well trained whilst the desired error threshold is too small and the maximum learning cycle is too long. Second, an inappropriate setting of the learning parameters (the learning rate, the momentum factor, and the number of hidden neurons) leads to bad learning. If this happens, a learning process must be stopped manually for seeking new settings of learning parameters. In most cases, a large learning rate may end up with pre-matured learning process where the change of weights will diminish much early than what it should be.

Due to page limit, the learning algorithms using the second derivatives are omitted in this chapter. Readers can refer to Bishop's book for details (10).

3. Neural Networks in Action

3.1. Model Evaluation

Every model must be as accurate, appropriate, efficient, effective, and fast as possible in application. Much like many engineering projects, where there is always a problem of careful evaluation and selection of the available tools. Without a careful evaluation and selection, a built neural network model may have a limited usage. In addition to model parameter optimisation using the delta rule (or the Newton–Raphson method) as discussed above, we have to consider how to evaluate a model and select the most appropriate model from many candidates. There are two important issues in neural network model construction. The first is how to evaluate a neural network model. Different neural network models (regression or classification projects) need different evaluation standards. During model evaluation, there is also an issue that which part of the model should be evaluated. Do we need to evaluate weights or do we need to evaluate how the predictions are close to the targets or to evaluate how the predictions are well separated? Moreover, what method can be recognised as an un-biased evaluation method? The second is how to select the most appropriate model based on the evaluation. There are always some hyper-parameters which are normally not updatable using a quantitative rule like the delta rule in most machine learning models. For instance, the number of hidden neurons in neural networks cannot

be updated using any quantitative rule like the delta rule. We therefore must employ a proper learning procedure to determine the optimal number of hidden neurons.

Suppose there are two data sets generated from the same application, i.e. they have the same physical background, $D^A = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\ell^A})$ and $D^B = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\ell^B})$, where ℓ^A and ℓ^B are the number of input vectors in two data sets. D^A and D^B are used to train two neural network models (A and B) using two different model structures separately. Both models are trained until the same minimum error is satisfied, i.e. $\varepsilon^A < \varepsilon^o$ and $\varepsilon^B < \varepsilon^o$, where ε^o, ε^A, and ε^B are the error threshold, the error occurred to model A, and the error occurred to model B. If we use D^A and D^B to evaluate model A and model B, respectively, two models may have the same error, i.e. $\varepsilon^A(\text{data A}) = \varepsilon^B(\text{data B})$, where $\varepsilon^X(\text{data Y})$ means the test error for model X using data Y. Because two models are with two different model structures, it is very likely that $\varepsilon^A(\text{data B}) \neq \varepsilon^A(\text{data A})$ and $\varepsilon^B(\text{data A}) \neq \varepsilon^B(\text{data B})$. This means that using the training data to evaluate a model is inappropriate as it does not provide rigorous evidence for model selection. As we will discuss below, a built model can very likely over fit to data noise, hence an independent test using a separate data must be followed as an appropriate methodology for any neural network model evaluation.

In most cases, there is no separate test data available when a user starts to build a new neural network model. We then need to consider how to use the available data for proper model construction, in particular, model evaluation and selection. This means that we need to reserve part of the available data for model evaluation. The basic principle is that the reserved data must not involve model parameter estimate. There are three commonly used methods for this purpose. They are cross-validation, re-sampling, and Jackknife. All these methods are using the same principle in that the evaluation data must not involve any process of model parameter estimation. This means that the available data must be divided into two parts. One is for model parameter estimation, which is commonly referred to as training. The other is for model evaluation and selection. The difference between these three methods is the strategy used for the division of a given data set.

3.2. Evaluation Statistics

3.2.1. Regression Analysis

For regression analysis problems, we normally have two evaluation statistics. They are the normalised error and the prediction correlation. The former measures the error degree compared with data noise while the second measures how predictions fit observations. The normalised error is defined as below:

$$\tilde{\varepsilon} = \frac{\varepsilon}{\sigma} \tag{24}$$

Here ε is defined in equation [2] and

$$\sigma = \sqrt{\frac{1}{\ell} \sum_{n=1}^{\ell} (y_n - \bar{y})^2} \qquad (25)$$

with $\bar{y}$ as the mean of the observations (target values). It should be noted that simply presenting ε may mislead because a sample with a large variance very unlikely has a smaller ε. Using $\tilde{\varepsilon}$, different systems can be comparable. The prediction correlation is defined as below:

$$\rho = \frac{\sum_{n=1}^{\ell} (t_n - \bar{t})(y_n - \bar{y})}{\sqrt{\sum_{n=1}^{\ell} (t_n - \bar{t})^2 \sum_{n=1}^{\ell} (y_n - \bar{y})^2}} \qquad (26)$$

If $\rho \to 1$, the prediction is perfect. If $\rho \to 0$, the predictions have no correlation with the target values and if $\rho \to -1$ the predictions are negatively correlated with the target values.

3.2.2. Classification Analysis

In classification analysis projects, we normally have three evaluation statistics. The first is called the confusion matrix, the second is called the Mathew correlation coefficient, and the last is called the receiver operating characteristics. The confusion matrix is as shown in **Table 12.2**. In **Table 12.2**, the true negatives and the true positives are correctly classified input vectors of the negative and the positive classes, respectively. The false positives are misclassified negative input vectors. The false negatives are misclassified positive input vectors. The specificity is the fraction of correctly predicted negative input vectors over the total negative input vectors meaning the prediction accuracy of negative input vectors. It also indicates the probability for a known negative input vector being correctly classified. The sensitivity is the fraction of correctly predicted positive input vectors over the total positive input vectors meaning the probability that a known positive input vector can be correctly classified. The negative prediction power is the fraction of the true negatives over the sum of the true negatives and the false negatives. It is the probability that a negative prediction is a true negative input vector. The positive prediction power

Table 12.2
Confusion matrix

	Negative	Positive	
Negative	True negatives	False positives	Specificity
Positive	False negatives	True positives	Sensitivity
	Negative prediction power	Positive prediction power	Total accuracy

is the fraction of the true positives over the sum of the true positives and the false positives. It is the probability that a positive prediction is a true positive input vector. Finally, the total accuracy is the fraction of all correctly classified input vectors over the total number of input vectors.

The Matthew correlation coefficient (11) is used to measure the prediction accuracy when data are unbalanced for both positive and negative input vectors. Let TN, TP, FN, FP denote true negatives, true positives, false negatives, and false positives, respectively. The definition of the Matthews correlation coefficient (MCC) is

$$\text{MCC} = \frac{\text{TN} \times \text{TP} \ - \ \text{FP} \times \text{FN}}{\sqrt{(\text{TN} + \text{FN})(\text{TN} + \text{FP})(\text{TP} + \text{FN})(\text{TP} + \text{FP})}} \quad (27)$$

The larger the Matthews correlation coefficient, the better the model fits the target. If the value of the Matthews correlation coefficient is one, it represents a complete correlation. If the value is zero, the prediction is completely random. If the value is negative, the prediction is on the opposite side of the target.

Both the confusion matrix and MCC evaluations are a point estimate approach, i.e. the evaluation statistics only depends on a single pre-defined threshold used for discrimination. In practice, a built model may be used using different thresholds by different users. For instance, a low threshold is commonly used in medical diagnosis because a false negative (cancers are wrongly diagnosed as noncancers) will undertake a much larger cost compared with a false positive (non-cancers are wrongly diagnosed as cancers). The problem is if the model still shows good performance when we change the threshold. This is certainly not a single point estimate problem. It is a problem that if a built model is robust. The receiver operating characteristics (ROC) analysis provides a good way to analyse system robustness when we change the threshold (12). In ROC analysis, we normally use the false positive fraction as the horizontal axis and the true positive fraction as the vertical axis to collectively present these two evaluation statistics for all available thresholds. This means that each point in this two-dimensional space represents these two evaluation statistics for a specific threshold used for discrimination. When we connect all the points in this two-dimensional space, we then form a curve which is called the ROC curve. Such a curve should be as close to the top-left corner as possible to demonstrate good robustness. Any system showing a curve near the diagonal line in this two-dimensional space shows a completely random system. On the left panel of **Fig. 12.6**, we have two probabilistic density functions for two classes of input vectors. There are five thresholds, named by A, B, C, D, and E, respectively. A is supposed to be the threshold generating the best separation between two

classes of input vectors. Based on these five thresholds, the values of the false positive rate and the true positive rate will certainly vary. If the threshold is moving towards left from A, for instance to B or D, both the false positive rate and the true positive rate will be increased. These new false positive and true positive rates will be mapped to two points in the two-dimensional ROC space shown on the right panel of **Fig. 12.6**. When the threshold is moving towards right from A, for instance to C or E, two new points are mapped to the two-dimensional ROC space as seen on the right panel of **Fig. 12.6** (points C and E). Connecting these five points in the two-dimensional ROC space will generate a curve referred to as a ROC curve shown on the right panel of **Fig. 12.6**.
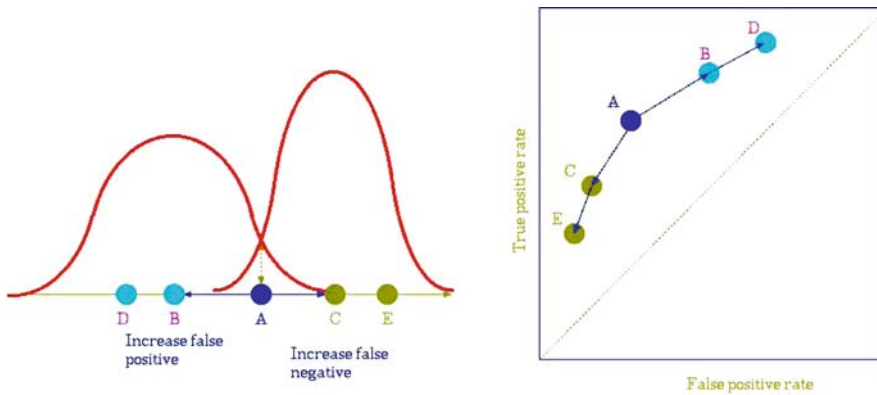
Fig. 12.6. ROC curves.

If two probabilistic density functions are far away from each other, changing threshold will not make a big difference in changing the false positive rate and the true positive rate. In this case, we refer to such a model as a robust one and the ROC curve will be very close to the top-left corner in the two-dimensional ROC space.

If there are a number of models built on different data sets or using different algorithms or different model structures, different ROC curves will be generated. In this situation, we can certainly compare these models by visualising these ROC curves. The closer a ROC curve is to the top-left corner in the two-dimensional ROC space, the more robust the model is. In **Fig.12.7**, there are three ROC curves for three models. It can be seen that model A is the worst one while model C is the best one. A quantitative measure of model robustness can be derived using the area under an ROC curve (AUR). In **Fig. 12.7**, it is obvious that curve C has the largest AUR while curve A has the smallest AUR (the value is 0.5).
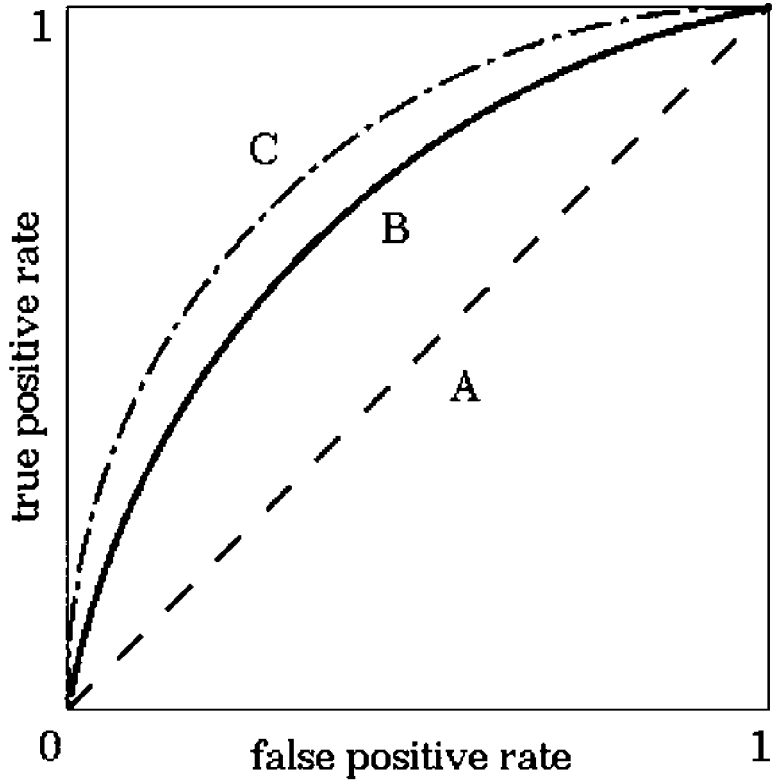
Fig. 12.7. ROC curves for model comparison.

3.3. Generalisation Issue

3.3.1. Over-Training

In the above discussion, we have been familiar with the delta rule. It is not very difficult to find out the relationship between $\Delta\varepsilon$ (or ΔO), $\nabla\varepsilon$ (or ∇O), and Δw from **Fig. 12.2** as shown below

$$\nabla\varepsilon = \frac{\Delta\varepsilon}{\Delta w} \tag{28}$$

Using the delta rule, we can derive the following relationship:

$$\Delta\varepsilon = \Delta w \nabla\varepsilon = -\eta\nabla\varepsilon^2 < 0 \tag{29}$$

This means that the use of the delta rule can always reduce the error until the minimum theoretically. The question is then whether this continuous reduction of the training error is a good thing in model construction. If two data sets are randomly drawn from the same sample, each will contain a different distribution of noise. Suppose the sample noise is a Gaussian $G(\mu_0, \sigma_0^2)$ and noise distributions in two data sets are $G(\mu_1, \sigma_1^2)$ and $G(\mu_2, \sigma_2^2)$. A model (M_1) generated on the first data set may not be able to generalise well on the second data set. If $\mu_1 < \mu_0$, it can be expected that the predictions (or generalisations) on the second data set using M_1 will be generally under-estimated. If $\mu_1 > \mu_0$, an over-estimate of

predictions may happen. We refer to the first data set as the training data set, i.e. all the input vectors in the first data set will be used for optimising model parameters using the delta rule. We refer to the second data set as validation (evaluation) data set, i.e. all the input vectors in this data set will not be used for optimising model parameters using the delta rule. Rather, they will be used for checking if a training process is biased. Such a bias is commonly called over-training or over-fitting. **Figure 12.8** shows such a case, where the training errors are continuously decreasing while the validation errors show a V-shape. After passing a point (called the early stop point), the validation errors have reached the minimum and start to increase continuously. A proper training of neural network then needs to identify this early stop point. The way of finding such an early stop point is based on the principle we have discussed, i.e. randomly dividing a sample into training and validation data sets.
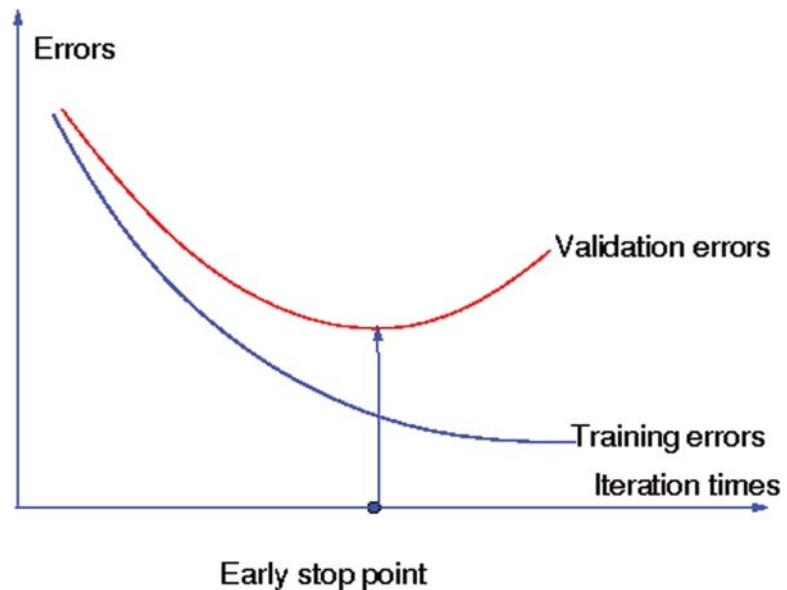
Fig. 12.8. Demonstration of over-training.

3.3.2. Over-Sized

Bad models not only result from over-training, but also from over-size. A model with a fewer parameters is always preferred compared with a model with more parameters when both have a similar training error. The first reason is the consideration of data significance, which is expressed as the ratio of the number of data points over the number of model parameters. The ratio should be as large as possible. In applications, it is not easy to have this condition satisfied. In most bioinformatics applications, it is very difficult to have large data. It is then very important to reduce model parameters to raise the data significance. The larger the data

significance, the more robust the model is. The second reason is that redundant model parameters are very often used for data noise. Removing these redundant model parameters can prevent possible over-fitting, which is closely related with the model generalisation capability.

3.4. Data Organisation for Model Evaluation

We now discuss how to organise data for proper model validation (evaluation). There are three approaches, re-sampling (or re-substitution), cross-validation, and Jackknife.

With the re-sampling method, we randomly sample a certain percentage of data for training and the rest for validation. Such a process is repeated many times. Suppose there are ℓ data points and we repeat the sampling process for m times. There will be m alidation models, each of which has different training and validation data with some possible overlapped data points. Note that all m validation models use the same hyper-parameters, for instance, they all use H hidden neurons. The parameters of the ith model are first estimated using the ith training data set with $k_i < \ell$ data points. The ith model is then validated on the ith validation data set with $\ell - k_i$ data points. Because we use different training data sets at each time, it is then expected that the parameters of the ith model will be different from those of the jth model. The validation performance of the ith model is certainly different from that of the jth model as well. We denote by $\varepsilon_i^{\vartheta}$ ($\vartheta = h$ implying neural networks with different hidden neurons) the validation error for the ith model. The evaluation statistic of the model with designed hyper-parameters ϑ can follow

$$\mu_{\vartheta} = \frac{1}{m} \sum_{i=1}^{m} \varepsilon_i^{\vartheta} \quad \text{and} \quad \sigma_{\vartheta}^2 = \frac{1}{m} \sum_{i=1}^{m} (\varepsilon_i^{\vartheta} - \mu_{\vartheta})^2 \qquad (30)$$

In order to determine the proper values for the hyper-parameters so that we can select a proper model, we can vary the values assigned to the hyper-parameters. If we have g hyper-parameters for selection, the selection will be taken in a g-dimensional space where each grid is a combination of hyper-parameters. Suppose we only need to determine the number of hidden neurons, we then have a series of μ_{ϑ} and σ_{ϑ}^2. The best model can be selected through

$$H = \arg\max\{\mu_{\vartheta}\} \sim \arg\min\{\sigma_{\vartheta}^2\} \qquad (31)$$

It should be noted that for the re-sampling method, some data points may be used for multiple times in training or validation.

In cross-validation, we normally randomly divide a data set into m folds. Each fold contains distinctive data points. If we denote by Ω_i as the set of data points in the ith fold, we will have $\Omega_i \bigcap \Omega_j = \phi$ meaning that two sets have no elements in common. Every time, we select one fold as the validation set and the remaining m-1 folds are used as the training set for model parameters

estimate. Such a process is repeated for m times until each fold has been used for validation once. This means that there are m models. Again, all these models use the same hyper-parameters, for instance, the same number of hidden neurons. The ith model is trained using the folds except for the ith fold and validated on the ith fold. The parameters of the ith model will also be different from those of the jth model and the validation performance of different models will vary. Note that each data point will be validated only once. Equations [30] and [31] can be used for whole system evaluation.

When data size is not too large, one commonly prefers to use the Jackknife (often called leave-one-out cross-validation) method. In using the Jackknife method, we normally pick up one data point for validation and use the remaining data points for training. Such a process is repeated for ℓ times until each data point has been exhausted for validation. This means that there are ℓ models for ℓ data points. Obviously all these models use the same hyper-parameters. The ith model is trained using all the data points except for the ith data point and validated on the ith data point. Equations [30] and [31] can also be used for whole system evaluation, but m is replaced by ℓ.

4. Applications to Bioinformatics

We will discuss some applications of neural networks to bioinformatics projects in this section.

4.1. Bio-chemical Data Analysis

Quantitative structure–activity relationship (QSAR) models are a class of bio-chemical models and normally involved with binary input variables for chemical properties with a very large dimensionality. The use of neural networks is normally for relational study or dimensionality reduction. Each input vector in these applications therefore represents a binary vector, i.e. $\mathbf{x} \in \{0, 1\}^D$. Each input vector is associated with a target value indicating compound property. In order to find the mapping function relating the chemical properties with the compound property, classification analysis approaches can be used. Neural networks can be used in these tasks for nonlinear modelling. For instance, a recent study using neural networks was studying the inhibition function of mutant PfDHFR (13). In microbiological research, Bacillus species identification is not an easy task. The application of neural networks on 1,071 fatty acid profiles is proved as a power tool for the identification (14). The neural networks have also been applied to the study of the relationship between compound chemical structures and human oestrogen receptor (α and β) binding affinity, where the

inputs are the molecular descriptors calculated from docking methods (15). Heparanase inhibitors' activity was also predicted using neural networks based on QSAR data (16).

4.2. Gene Expression Data Analysis

Gene expression data have been widely studied for understanding how genes are responding to external environmental cues. Gene expression data are normally numerical inputs with also a large dimensionality, but a few number of samples. In this case, data significance is a very serious problem in applying neural networks for data analysis. In recent studies, gene expression data have been used for disease diagnosis. In these applications, the expressions of genes are commonly sitting in a high-dimensional space ($\mathbf{x} \in \Re^D$, where D is the number of genes and $\mathbf{x}$ is a vector of the expression values for D genes). Each expression vector ($\mathbf{x}$) has an associated target value, declaring the corresponding sample is disease-free or not. It can be seen that this is then a classification problem. If the relation between expression vector and target is nonlinear, neural network is one of the candidates for model construction and prediction. For instance, neural networks were used for the investigation of the distinguishing power of childhood acute lymphoblastic leukaemia (ALL) diagnostic bone marrow (17), for influenza identification based on microarray data (18). Neural networks have also been used for gene network re-construction (19) and cancer-related regulatory modelling (20).

4.3. Protein Structure Data Analysis

Protein structures are always an important subject for studying how proteins are interacting with each other forming complexes for cellular signalling responding to environmental cues. Wagner et al. applied neural networks to the function prediction of inhibitory activity of serotonin and NF-kappaB (21). It was found that the relationship between structure and activity is essential to cellular signalling for the inhibitory function of serotonin and NF-kappaB. In the study of the detection of drug-induced idiosyncratic liver toxicity using QSAR data, it was reported that a neural network model is able to achieve 84% (22).

4.4. Bio-marker Identification

In bioinformatics research, the identification of bio-markers has a great importance in bio-medical applications. The major purpose in these applications is to identify the most important identities which can be genes, compounds, chemicals, proteins, or metabolites for predictive usages. This means that we need to combine classification analysis approaches with feature selection approaches to identify a minimum subset of input variables which can achieve maximum discrimination capability between disease and disease-free samples. For instance, surface-enhanced laser desorption/ionisation time-of-flight mass spectrometry was used to detect proteomic patterns in the serum of women with endometriosis (23). Neural networks have been used for detecting early stage

epithelial ovarian cancer using multiple serum markers from four institutes (24). Sixty-six Chinese patients with hepatocellular carcinoma was used to generate proteomic profiling study by two-dimensional gel electrophoresis analysis and neural networks are used to analyse the profiling data for delineating significant patterns for discriminating hepatocellular carcinoma from nonmalignant liver tissues (25).

4.5. Sequence Data Analysis

Sequence data analysis is one of the most important subjects in bioinformatics. Its main objective is to predict structures or functions based on sequence compositions. Neural networks have been recently applied to the prediction of transcription start sites (26). In discriminating mesophilic and thermophilic proteins, neural networks have been used to build a classifier achieving 91% accuracy (27), where amino acid frequency was used as feature or input variable. Neural networks have also been used to predict phosphorylation sites (28), disorder proteins (29), and secondary structures (30–32). In recent studies, a new function called bio-basis function was proposed (33) for protein functional site prediction. The neural network built based on the bio-basis function is then called the bio-basis function neural network. Denote by s_i and s_j two sequences with D residues (amino acids). The similarity between these two sequences using a mutation matrix (34–36) is defined as below:

$$\rho(\mathbf{s}_i, \mathbf{s}_j) = \sum_{d=1}^{D} M(s_{id}, s_{jd}) \tag{32}$$

Here $M(s_{id}, s_{jd})$ can be obtained from a mutation matrix through the table-loop-up method. The bio-basis function is then defined as

$$\sigma(\mathbf{s}_i, \mathbf{s}_j) = \exp\left(\frac{\rho(\mathbf{s}_i, \mathbf{s}_j) - \rho(\mathbf{s}_j, \mathbf{s}_j)}{\rho(\mathbf{s}_j, \mathbf{s}_j)}\right) \tag{33}$$

It can be seen that if s_i and s_j are identical, $\sigma(\mathbf{s}_i, \mathbf{s}_j) = 1$ while $\sigma(\mathbf{s}_i, \mathbf{s}_j)$ will be small if s_i and s_j are very different. We treat s_j as a training sequence and s_i as a testing sequence. Suppose we have ℓ training sequences, the model of bio-basis function neural network is defined as a linear combination of the bio-basis functions as below:

$$y_i = \sum_{n=1}^{\ell} w_n \sigma(\mathbf{s}_i, \mathbf{s}_n) \tag{34}$$

This bio-basis function neural network can then be used for regression analysis or classification analysis. The bio-basis function neural network has been successfully applied to the prediction of trypsin cleavage sites (33), HIV protease cleavage sites (33), (37), disorder proteins (38), phosphorylation sites (39–41), O-linkage

sites (42), factor X cleavage sites (43), caspase sites (44), SARS cleavage sites (45), T-cell epitopes (46), and HCV protease cleavage sites (47).

5. Summary

This chapter has discussed the basic principle of neural networks, specifically supervised neural networks, the learning rule (delta rule), the learning algorithms, and their applications to bioinformatics. We have discussed in detail how the delta rule is developed, its property, and the revised delta rule. In discussing the learning algorithms, we have discussed in detail how the delta rule is specialised to regression and classification learning tasks. After this, we have discussed the important subjects in using neural networks for bioinformatics research and applications, namely data organisation for model construction and model selection. We have discussed three approaches, namely Jackknife, cross-validation, and re-substitution. Meanwhile, some model evaluation criteria and their properties are discussed for both regression and classification applications.

It must be noted that like many other machine learning algorithms, neural networks have the difficulty of potential over-fitting or biased learning. We normally do not have any prior knowledge about how parameters should be distributed. Because of this, a model with its parameters estimated at one point in a parameter space as discussed in this chapter may not represent the true knowledge. Taking all the points in the parameter space into the consideration, on the other hand, is not an easy job. This is why a Bayesian chapter has been included in this book, where the Bayesian approach, particularly Bayesian learning and Bayesian inference approaches, are detailed and are fundamental to increase generalisation capability of neural network models.

References

1. McCulloch W and Pitts W (1943) A logical calculus the ideas immanent in nervous activity, *Bulletin Mathematical Biophysics*, **5**, 115–133.

2. Fitch FB, Review: McCulloch WS and Pitts W (1944) A logic calculus of the ideas immanent in nervous activity, *Journal Symbolic Logic*, **9**, 49–50.

3. Hebb DO (1949) *The Organization of Behaviour*, John Wiley and Sons Inc.

4. Minsky M (1954) Steps towards artificial intelligence. In Feigenbaum, EA and Feldman, J, eds *Computers & Thought*, MIT Press, Cambridge, MA, USA.

5. Minksy M (1969) *Perceptron*, MIT Press, Cambridge, MA, USA.

6. Kohonen T (1982) Analysis of a simple self-organizing process, *Biological Cybernetics*, **44**, 135–140.

7. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities, *PNSA*, **79**, 2554–2558.

8. Werbos PJ (1994) *The Roots of Backpropagation: From Ordered Derivatives to Neural*

Networks and Political Forecasting, Willey-Interscience.

9. Rumelhart DE, Hinton GE, and Williams RJ (1986) Learning internal representations by error propagation. In Rumelhart DF and McClelland, JL eds, *Parallel Distributed Processing*, 318–362, The MIT press, Cambridge, MA, USA.

10. Bishop CM (1995) *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford.

11. Matthews BW (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme, *Biochimica et Biophysica Acta*, **405**, 442–451.

12. Metz CE (1986) ROC Methodology in radiologic imaging, *Investigative Radiology*, **21**, 720–733.

13. Hecht D, Cheung M, and Fogel GB (2008) QSAR using evolved neural networks for the inhibition of mutant PfDHFR by pyrimethamine derivatives, *Biosystems*, **92**, 10–15.

14. Slabbinck B, De Baets B, Dawyndt P, and De Vos P (2008) Genus-wide Bacillus species identification through proper artificial neural network experiments on fatty acid profiles, *Antonie Van Leeuwenhoek* (in press).

15. Spreafico M, Boriani E, Benfenati E, and Novic M (2008) Structural features of diverse ligands influencing binding affinities to estrogen alpha and estrogen beta receptors. Part II. Molecular descriptors calculated from conformation of the ligands in the complex resulting from previous docking study, *Molecular Diversity* (in press).

16. Jalali-Heravi M, Asadollahi-Baboli M, and Shahbazikhah P (2008) QSAR study of heparanase inhibitors activity using artificial neural networks and Levenberg-Marquardt algorithm, *European Journal of Medicinal Chemistry*, **43**, 548–556.

17. Catchpoole D, Lail A, Guo D, Chen QR, and Khan J (2007) Gene expression profiles that segregate patients with childhood acute lymphoblastic leukaemia: an independent validation study identifies that endoglin associates with patient outcome, *Leukemia Research*, **31**, 1741–1747.

18. Moore CL, Smagala JA, Smith CB, Dawson ED, Cox NJ, Kuchta RD, and Rowlen KL (2007) Evaluation of MChip with historic subtype H1N1 influenza A viruses, including the 1918 "Spanish Flu" strain, *Journal Clinic Microbiology*, **45**, 3807–3810.

19. Xu R, Venayagamoorthy GK, and Wunsch DC (2007) Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization, *Neural Networks*, **20**, 917–927.

20. Chiang JH, Chao SY (2007) Modeling human cancer-related regulatory modules by GA-RNN hybrid algorithms, *BMC Bioinformatics*, Mar 14;8:91.

21. Wagner S, Arce R, Murillo R, Terfloth L, Gasteiger J, and Merfort I (2008) Neural networks as valuable tools to differentiate between sesquiterpene lactones' inhibitory activity on serotonin release and on NF-kappaB, *Journal of Medicinal Chemistry*, **51**, 1324–1332.

22. Cruz-Monteagudo M, Cordeiro MN, and Borges F (2008) Computational chemistry approach for the early detection of drug-induced idiosyncratic liver toxicity, *Journal Computer Chemistry*, **29**, 533–549.

23. Wang L, Zheng W, Mu L, and Zhang SZ (2008) Identifying biomarkers of endometriosis using serum protein fingerprinting and artificial neural networks, *International Journal Gynaecol Obstet* (in press).

24. Zhang Z, Yu Y, Xu F, Berchuck A, van Haaften-Day C, Havrilesky LJ, de Bruijn HW, van der Zee AG, Woolas RP, Jacobs IJ, Skates S, Chan DW, and Bast RC Jr (2007) Combining multiple serum tumor markers improves detection of stage I epithelial ovarian cancer, *Gynecologic Oncology*, **107**, 526–531.

25. Luk JM, Lam BY, Lee NP, Ho DW, Sham PC, Chen L, Peng J, Leng X, Day PJ, and Fan ST (2007) Artificial neural networks and decision tree model analysis of liver cancer proteomes, *Biochemistry Biophysics Research Communication*, **361**, 68–73.

26. Won HH, Kim MJ, Kim S, and Kim JW (2008) EnsemPro: an ensemble approach to predicting transcription start sites in human genomic DNA sequences, *Genomics*, **91**, 259–266.

27. Gromiha MM and Suresh MX (2008) Discrimination of mesophilic and thermophilic proteins using machine learning algorithms, *Proteins*, **70**, 1274–1279.

28. Blom N, Gammeltoft S, and Brunak S (1999) Sequence and structure based prediction of eukaryotic protein phosphorylation sites, *Journal of Molecular Biology*, **294**, 1351–1362.

29. Dunker AK, Obradovic Z, Romero P, Garner EC, and Brown CJ (2000) Intrinsic protein disorder in complete genomes, *Genome Information*, **11**, 161–171.

30. Qian N and Sejnowski TJ (1988) Predicting the secondary structure of globular proteins

using neural network models. *Journal of Molecular Biology*, **202**, 865–884.

31. Baldi P, Pollastri G, Andersen C, and Brunak S (2000) Matching protein beta-sheet partners by feedforward and recurrent neural networks, *Proceedings of International Conference on Intelligent Systems for Molecular Biology, ISMB*, **8**, 25–36.

32. Edelman J and White SH (1989) Linear optimization of predictors for secondary structure: Application to transbilayer segments of membrane proteins, *Journal of Molecular Biology*, **210**, 195–209.

33. Thomson R, Hodgman TC, Yang ZR, and Doyle AK (2003) Characterising proteolytic cleavage site activity using bio-basis function neural networks, *Bioinformatics*, **19**, 1741–1747.

34. Dayhoff MO, Schwartz RM, and Orcutt BC (1978) A model of evolutionary change in proteins. matrices for detecting distant relationships, In Dayhoff MO ed, *Atlas of protein sequence and structure*, **5**, 345–358.

35. Altschul SF, Gish, W, Miller, W, Myers, E, and Lipman, DJ (1990) Basic local alignment search tool, *Journal of Molecular Biology*, **215**, 403–410.

36. Johnson MS and Overington JP (1993) A structural basis for sequence comparisons-an evaluation of scoring methodologies, *Journal of Molecular Biology*, **233**, 716–738.

37. Yang ZR and Berry E (2004) Reduced bio-basis function neural networks for protease cleavage site prediction, *Journal of Computational Biology and Bioinformatics*, **2**, 511–531.

38. Yang ZR, Thomson R, McNeil P, and Esnouf R (2005) RONN: use of the bio-basis function neural network technique for the detection of natively disordered regions in proteins, *Bioinformatics*, **21**, 3369–3376.

39. Berry E, Dalby A, and Yang ZR (2004) Reduced bio basis function neural network for identification of protein phosphorylation sites: Comparison with pattern recognition algorithms, *Computational Biology and Chemistry*, **28**, 75–85.

40. Senawongse P, Dalby AD, and Yang ZR (2005) Predicting the phosphorylation sites using hidden Markov models and machine learning methods, *Journal of Chemical Information and Computer Science*, **45**, 1147–1152.

41. Sidhu A and Yang ZR (2006) Predict signal peptides using bio-basis function neural networks, *Applied Bioinformatics*, **5**, 13–19.

42. Yang ZR and Chou KC (2004) Bio-basis function neural networks for the prediction of the O-linkage sites in glyco-proteins, *Bioinformatics*, **20**, 903–908.

43. Yang ZR, Dry J, Thomson R, and Hodgman C (2006) A bio-basis function neural network for protein peptide cleavage activity characterisation. *Neural Networks*, **19**, 401–407.

44. Yang ZR (2005) Prediction of caspase cleavage sites using Bayesian bio-basis function neural networks, *Bioinformatics*, **21**, 1831–1837.

45. Yang ZR (2005) Mining SARS-CoV protease cleavage data using decision trees, a novel method for decisive template searching, *Bioinformatics*, **21**, 2644–2650.

46. Yang ZR and Johnathan F (2005) Predict T-cell epitopes using bio-support vector machines, *Journal of Chemical Informatics and Computer Science*, **45**, 1142–1148.

47. Yang ZR (2006) Predicting hepatitis C virus protease cleavage sites using generalised linear indicator regression models, *IEEE Transanctions on Biomedical Engineering*, **53**, 2119–2123.

Chapter 13

A User's Guide to Support Vector Machines

Asa Ben-Hur and Jason Weston

Abstract

The Support Vector Machine (SVM) is a widely used classifier in bioinformatics. Obtaining the best results with SVMs requires an understanding of their workings and the various ways a user can influence their accuracy. We provide the user with a basic understanding of the theory behind SVMs and focus on their use in practice. We describe the effect of the SVM parameters on the resulting classifier, how to select good values for those parameters, data normalization, factors that affect training time, and software for training SVMs.

Key words: Kernel methods, Support Vector Machines (SVM).

1. Introduction

The Support Vector Machine (SVM) is a state-of-the-art classification method introduced in 1992 by Boser, Guyon, and Vapnik (1). The SVM classifier is widely used in bioinformatics due to its high accuracy, ability to deal with high-dimensional data such as gene expression, and flexibility in modeling diverse sources of data (2). See also a recent paper in Nature Biotechnology titled "What is a support vector machine?" (3).

SVMs belong to the general category of *kernel methods* (4, 5). A kernel method is an algorithm that depends on the data only through dot-products. When this is the case, the dot product can be replaced by a *kernel function* which computes a dot product in some possibly high-dimensional feature space. This has two advantages: First, the ability to generate nonlinear decision boundaries using methods designed for linear classifiers. Second, the use of kernel functions allows the user to apply a classifier to data that

have no obvious fixed-dimensional vector space representation. The prime example of such data in bioinformatics are sequence, either DNA or protein, and protein structure.

Using SVMs effectively requires an understanding of how they work. When training an SVM, the practitioner needs to make a number of decisions: how to preprocess the data, what kernel to use, and finally, setting the parameters of the SVM and the kernel. Uninformed choices may result in severely reduced performance (6). In this chapter, we aim to provide the user with an intuitive understanding of these choices and provide general usage guidelines. All the examples shown in this chapter were generated using the PyML machine learning environment, which focuses on kernel methods and SVMs, and is available at http://pyml.sourceforge.net. PyML is just one of several software packages that provide SVM training methods; an incomplete listing of these is provided in **Section 9**. More information is found on the Machine Learning Open Source Software Web site http://mloss.org and a related paper (7).

This chapter is organized as follows: we begin by defining the notion of a linear classifier (**Section 2**); we then introduce kernels as a way of generating nonlinear boundaries while still using the machinery of a linear classifier (**Section 3**); the concept of the margin and SVMs for maximum margin classification are introduced next (**Section 4**). We then discuss the use of SVMs in practice: the effect of the SVM and kernel parameters (**Section 5**), how to select SVM parameters and normalization (**Sections 6 and 8**), and how to use SVMs for unbalanced data (**Section 7**). We close with a discussion of SVM training and software (**Section 9**) and a list of topics for further reading (**Section 10**). For a more complete discussion of SVMs and kernel methods, we refer the reader to recent books on the subject (5, 8).

2. Preliminaries: Linear Classifiers

Support vector machines are an example of a linear two-class classifier. This section explains what that means. The data for a two-class learning problem consist of objects labeled with one of two labels corresponding to the two classes; for convenience we assume the labels are +1 (positive examples) or −1 (negative examples). In what follows, boldface $\mathbf{x}$ denotes a vector with components x_i. The notation $\mathbf{x}_i$ will denote the ith vector in a dataset composed of n labeled examples $(\mathbf{x}_i, y_i)$ where y_i is the label associated with $\mathbf{x}_i$. The objects $\mathbf{x}_i$ are called *patterns* or *inputs*. We assume the inputs belong to some set X. Initially we assume the inputs are vectors, but once we introduce kernels this

assumption will be relaxed, at which point they could be any continuous/discrete object (e.g., a protein/DNA sequence or protein structure).

A key concept required for defining a linear classifier is the *dot product* between two vectors, also referred to as an *inner product* or *scalar product*, defined as $\mathbf{w}^T\mathbf{x} = \sum_i w_i x_i$. A linear classifier is based on a linear *discriminant function* of the form

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b. \qquad [1]$$

The vector $\mathbf{w}$ is known as the *weight vector*, and b is called the bias. Consider the case $b = 0$ first. The set of points $\mathbf{x}$ such that $\mathbf{w}^T\mathbf{x} = 0$ are all points that are perpendicular to $\mathbf{w}$ and go through the origin – a line in two dimensions, a plane in three dimensions, and more generally, a *hyperplane*. The bias b translates the hyperplane away from the origin. The hyperplane divides the space into two according to the sign of the discriminant function $f(\mathbf{x})$ defined in Equation [1] – *see* **Fig. 13.1** for an illustration. The boundary between regions classified as positive and negative is called the *decision boundary* of the classifier. The decision boundary defined by a hyperplane is said to be *linear* because it is linear in the input examples (cf. Equation [1]). A classifier with a linear decision boundary is called a linear classifier. Conversely, when the decision boundary of a classifier depends on the data in a nonlinear way (*see* **Fig. 13.4** for example), the classifier is said to be nonlinear.
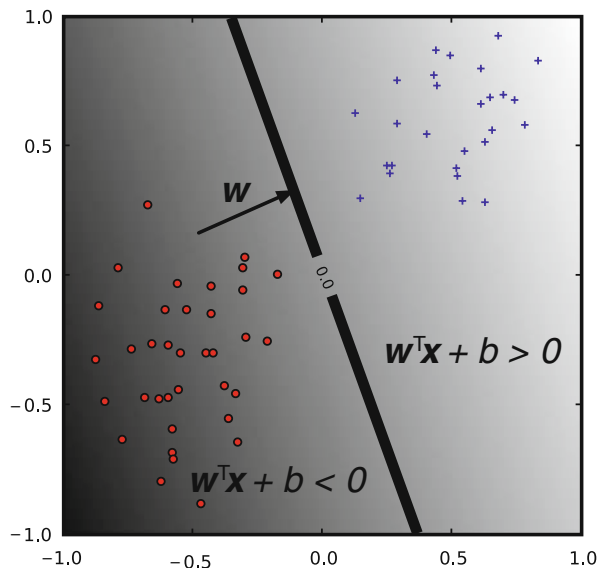
Fig. 13.1. A linear classifier. The hyper-plane (line in 2-d) is the classifier's decision boundary. A point is classified according to which side of the hyper-plane it falls on, which is determined by the sign of the discriminant function.

3. Kernels: from Linear to Nonlinear Classifiers

In many applications a nonlinear classifier provides better accuracy. And yet, linear classifiers have advantages, one of them being that they often have simple training algorithms that scale well with the number of examples (9, 10). This begs the question: can the machinery of linear classifiers be extended to generate nonlinear decision boundaries? Furthermore, can we handle domains such as protein sequences or structures where a representation in a fixed-dimensional vector space is not available?

The naive way of making a nonlinear classifier out of a linear classifier is to map our data from the input space X to a feature space F using a nonlinear function ϕ. In the space F, the discriminant function is

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \qquad [2]$$

Example 1 Consider the case of a two-dimensional input-space with the mapping $\phi(\mathbf{x}) = \left(x_1^2, \sqrt{2}x_1 x_2, x_2^2\right)^T$, which represents a vector in terms of all degree-2 monomials. In this case

$$\mathbf{w}^T \phi(\mathbf{x}) = w_1 x_1^2 + w_2 \sqrt{2}x_1 x_2 + w_3 x_2^2,$$

resulting in a decision boundary for the classifier which is a conic section (e.g., an ellipse or hyperbola). The added flexibility of considering degree-2 monomials is illustrated in **Fig. 13.4** in the context of SVMs.

The approach of explicitly computing nonlinear features does not scale well with the number of input features: when applying a mapping analogous to the one from the above example to inputs which are vectors in a d-dimensional space, the dimensionality of the feature space F is quadratic in d. This results in a quadratic increase in memory usage for storing the features and a quadratic increase in the time required to compute the discriminant function of the classifier. This quadratic complexity is feasible for low-dimensional data; but when handling gene expression data that can have thousands of dimensions, quadratic complexity in the number of dimensions is not acceptable. The situation is even worse when monomials of a higher degree are used. Kernel methods solve this issue by avoiding the step of explicitly mapping the data to a high-dimensional feature space. Suppose the weight vector can be expressed as a linear combination of the training examples, i.e., $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i$. Then

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i^T \mathbf{x} + b$$

In the feature space, F, this expression takes the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b$$

The representation in terms of the variables α_i is known as the *dual* representation of the decision boundary. As indicated above, the feature space F may be high dimensional, making this trick impractical unless the kernel function $k(\mathbf{x}, \mathbf{x}')$ defined as

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$$

can be computed efficiently. In terms of the kernel function, the discriminant function is

$$f(\mathbf{x}) = \sum_{i=1}^{n} k(\mathbf{x}, \mathbf{x_i}) + b. \tag{3}$$

Example 2 Let us go back to the example of the mapping $\phi(\mathbf{x}) = \left(x_1^2, \sqrt{2} x_1 x_2, x_2^2 \right)^T$. An easy calculation shows that the kernel associated with this mapping is given by $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') = \left(\mathbf{x}^T \mathbf{x}' \right)^2$, which shows that the kernel can be computed without explicitly computing the mapping ϕ.

The above example leads us to the definition of the degree-*d* polynomial kernel

$$k(\mathbf{x}, \mathbf{x}') = \left(\mathbf{x}^T \mathbf{x}' + 1 \right)^d. \tag{4}$$

The feature space for this kernel consists of all monomials whose degree is less or equal to d. The kernel with $d = 1$ is the *linear kernel*, and in that case the additive constant in Equation [4] is usually omitted. The increasing flexibility of the classifier as the degree of the polynomial is increased is illustrated in **Fig. 13.4**. The other widely used kernel is the Gaussian kernel defined by

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2 \right), \tag{5}$$

where $\gamma > 0$ is a parameter that controls the width of Gaussian, and $\|\mathbf{x}\|$ is the *norm* of $\mathbf{x}$ and is given by $\sqrt{\mathbf{x}^T \mathbf{x}}$. The parameter γ plays a similar role as the degree of the polynomial kernel in controlling the flexibility of the resulting classifier (*see* **Fig. 13.5**).

We saw that a linear decision boundary can be "kernelized," i.e. its dependence on the data is only through dot products. In order for this to be useful, the training algorithm needs to be kernelizable as well. It turns out that a large number of machine learning algorithms can be expressed using kernels – including ridge regression, the perceptron algorithm, and SVMs (5, 8).

4. Large-Margin Classification

In what follows, we use the term *linearly separable* to denote data for which there exists a linear decision boundary that separates positive from negative examples (*see* **Fig. 13.2**). Initially, we will assume linearly separable data and later show how to handle data that are not linearly separable.
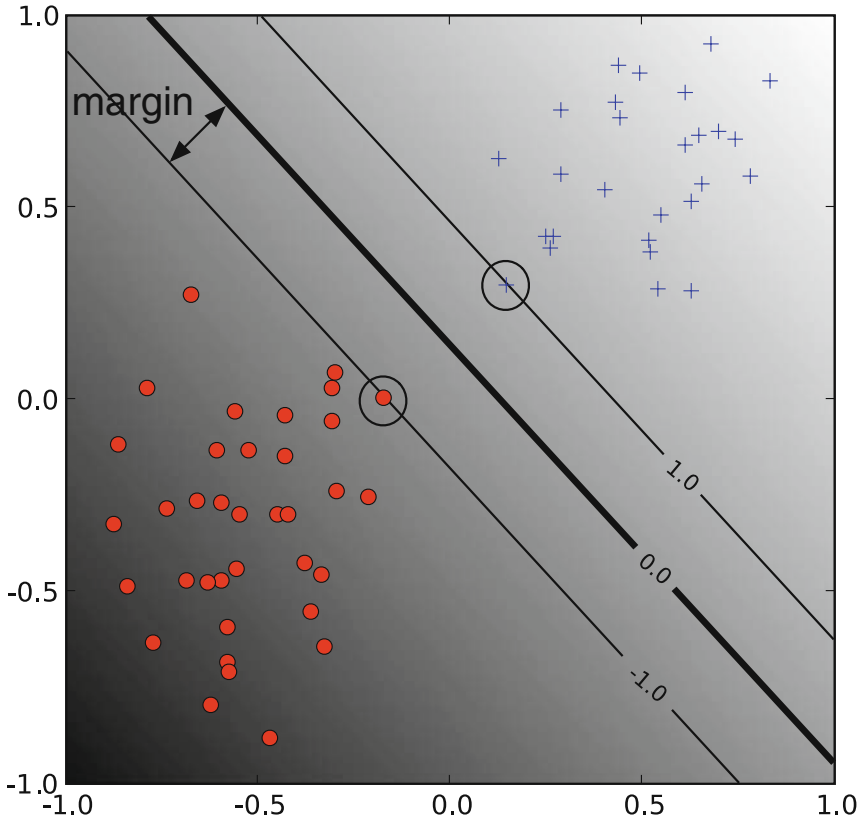
Fig. 13.2. A linear SVM. The circled data points are the *support vectors* – the examples that are closest to the decision boundary. They determine the margin with which the two classes are separated.

4.1. The Geometric Margin

In this section, we define the notion of a margin. For a given hyperplane, we denote by $\mathbf{x}_+(\mathbf{x}_-)$ the closest point to the hyperplane among the positive (negative) examples. From simple geometric considerations, the margin of a hyperplane defined by a weight vector $\mathbf{w}$ with respect to a dataset D can be seen to be

$$m_D(\mathbf{w}) = \frac{1}{2}\hat{\mathbf{w}}^{\mathbf{T}}(\mathbf{x}_+ - \mathbf{x}_-), \qquad [6]$$

where $\hat{\mathbf{w}}$ is a unit vector in the direction of $\mathbf{w}$, and we assume that $\mathbf{x}_+$ and $\mathbf{x}_-$ are equidistant from the decision boundary, i.e.,

$$f(\mathbf{x}_+) = \mathbf{w}^T\mathbf{x}_+ + b = a$$

$$f(\mathbf{x}_+) = \mathbf{w}^T\mathbf{x}_+ + b = -a \qquad [7]$$

for some constant $a > 0$. Note that multiplying the data points by a fixed number will increase the margin by the same amount, whereas in reality, the margin has not really changed – we just

changed the "units" with which it is measured. To make the geometric margin meaningful, we fix the value of the discriminant function at the points closest to the hyperplane, and set $a = 1$ in Equation [7]. Adding the two equations and dividing by $||\mathbf{w}||$, we obtain the following expression for the margin:

$$m_D(\mathbf{w}) = \frac{1}{2}\hat{\mathbf{w}}^{\mathbf{T}}(\mathbf{x}_+ - \mathbf{x}_-) = \frac{1}{||\mathbf{w}||}. \qquad [8]$$

4.2. Support Vector Machines

Now that we have the concept of a margin, we can formulate the maximum margin classifier. We will first define the hard-margin SVM, applicable to a linearly separable dataset, and then modify it to handle nonseparable data.

The maximum-margin classifier is the discriminant function that maximizes the geometric margin $1/||\mathbf{w}||$, which is equivalent to minimizing $||\mathbf{w}||^2$. This leads to the following constrained optimization problem:

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to: } y_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) \geq 1 \ i = 1, \dots, n. \qquad [9]$$

The constraints in this formulation ensure that the maximum-margin classifier classifies each example correctly, which is possible since we assumed that the data are linearly separable. In practice, data are often not linearly separable; and even if they are, a greater margin can be achieved by allowing the classifier to misclassify some points. To allow errors we replace the inequality constraints in Equation [9] with

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i,$$

where ξ_i are *slack variables* that allow an example to be in the margin ($1 \geq \xi_i \geq 0$, also called a margin error) or misclassified ($\xi_i \geq 1$). Since an example is misclassified if the value of its slack variable is greater than 1, the sum of the slack variables is a bound on the number of misclassified examples. Our objective of maximizing the margin, i.e., minimizing $||\mathbf{w}||^2$ will be augmented with a term $C\sum_i \xi_i$ to penalize misclassification and margin errors. The optimization problem now becomes

$$\underset{\mathbf{w},b}{\text{minimize}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i$$

$$\text{subject to: } y_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b) \geq 1 - \xi_i \qquad \xi_i \geq 0. \qquad [10]$$

The constant $C > 0$ sets the relative importance of maximizing the margin and minimizing the amount of slack. This formulation is called the *soft-margin SVM* and was introduced by Cortes and

Vapnik (11). Using the method of Lagrange multipliers, we can obtain the *dual* formulation, which is expressed in terms of variables α_i (11, 5, 8):

$$\underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x_j} \qquad [11]$$

$$\text{subject to} \sum_{i=1}^{n} y_i \alpha_i = 0, \qquad 0 \le \alpha_i \le C.$$

The dual formulation leads to an expansion of the weight vector in terms of the input examples

$$\mathbf{w} = \sum_{i=1}^{n} y_i \alpha_i \mathbf{x}_i. \qquad [12]$$

The examples for which $\alpha_i > 0$ are those points that are on the margin, or within the margin when a soft-margin SVM is used. These are the so-called *support vectors*. The expansion in terms of the support vectors is often sparse, and the level of sparsity (fraction of the data serving as support vectors) is an upper bound on the error rate of the classifier (5).

The dual formulation of the SVM optimization problem depends on the data only through dot products. The dot product can therefore be replaced with a nonlinear kernel function, thereby performing large-margin separation in the feature space of the kernel (*see* **Figs. 13.4** and **13.5**). The SVM optimization problem was traditionally solved in the dual formulation, and only recently it was shown that the primal formulation, Equation [10], can lead to efficient kernel-based learning (12). Details on software for training SVMs is provided in **Section 9**.

5. Understanding the Effects of SVM and Kernel Parameters

Training an SVM finds the large-margin hyperplane, i.e., sets the values of the parameters α_i and b (c.f. Equation [3]). The SVM has another set of parameters called *hyperparameters*: the soft-margin constant, C, and any parameters the kernel function may depend on (width of a Gaussian kernel or degree of a polynomial kernel). In this section, we illustrate the effect of the hyperparameters on the decision boundary of an SVM using two-dimensional examples.

We begin our discussion of hyperparameters with the soft-margin constant, whose role is illustrated in **Fig. 13.3**. For a large value of C, a large penalty is assigned to errors/margin errors. This is seen in the left panel of **Fig. 13.3**, where the two points closest to the hyperplane affect its orientation, resulting in a hyperplane that comes close to several other data points. When C is decreased (right panel of the figure), those points become margin errors; the hyperplane's orientation is changed, providing a much larger margin for the rest of the data.
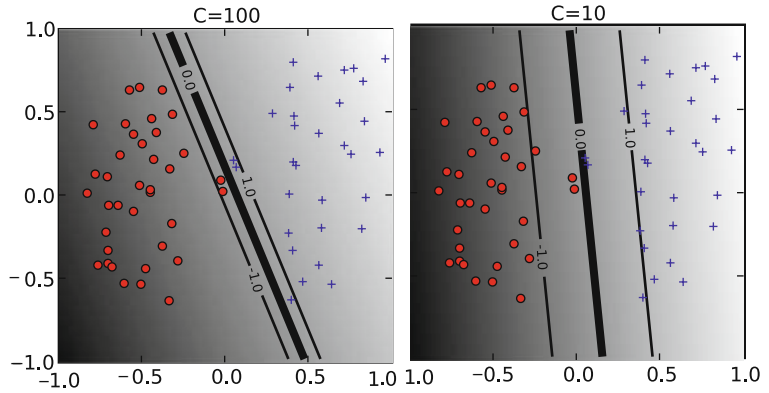
Fig. 13.3. The effect of the soft-margin constant, *C*, on the decision boundary. A smaller value of *C* (*right*) allows to ignore points close to the boundary and increases the margin. The decision boundary between negative examples (*circles*) and positive examples (*crosses*) is shown as a thick line. The lighter lines are on the margin (discriminant value equal to −1 or +1). The grayscale level represents the value of the discriminant function, dark for low values and a light shade for high values.

Kernel parameters also have a significant effect on the decision boundary. The degree of the polynomial kernel and the width parameter of the Gaussian kernel control the flexibility of the resulting classifier (**Figs. 13.4** and **13.5**). The lowest degree polynomial is the linear kernel, which is not sufficient when a nonlinear relationship between features exists. For the data in **Fig. 13.4** a degree-2 polynomial is already flexible enough to discriminate between the two classes with a sizable margin. The degree-5 polynomial yields a similar decision boundary, albeit with greater curvature.

Next we turn our attention to the Gaussian kernel defined as $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$. This expression is essentially zero if the distance between $\mathbf{x}$ and $\mathbf{x}'$ is much larger than $1/\sqrt{\gamma}$; i.e., for a fixed $\mathbf{x}'$ it is localized to a region around $\mathbf{x}'$. The support vector expansion, Equation [3] is thus a sum of Gaussian "bumps" centered around each support vector. When γ is small (top left panel in **Fig. 13.5**) a given data point $\mathbf{x}$ has a nonzero kernel value relative
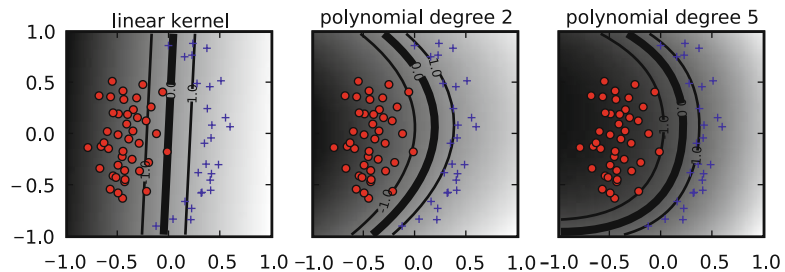
Fig. 13.4. The effect of the degree of a polynomial kernel. Higher degree polynomial kernels allow a more flexible decision boundary. The style follows that of **Fig. 13.3**.
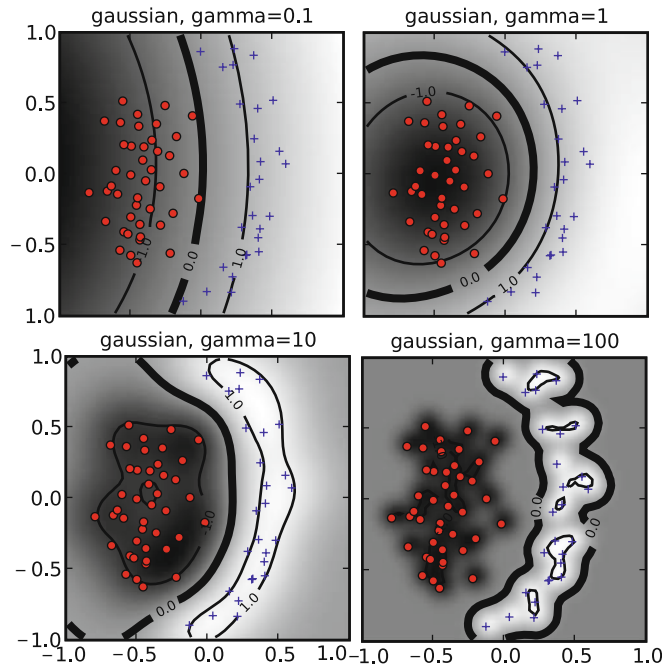
Fig. 13.5. The effect of the inverse-width parameter of the Gaussian kernel (γ) for a fixed value of the soft-margin constant. For small values of γ (*upper left*) the decision boundary is nearly linear. As γ increases the flexibility of the decision boundary increases. Large values of γ lead to overfitting (*bottom*). The figure style follows that of **Fig. 13.3**.

to any example in the set of support vectors. Therefore, the whole set of support vectors affects the value of the discriminant function at **x**, resulting in a smooth decision boundary. As γ is increased, the locality of the support vector expansion increases, leading to greater curvature of the decision boundary. When γ is large, the value of the discriminant function is essentially constant outside the close proximity of the region where the data are concentrated (*see* bottom right panel in **Fig. 13.5**). In this regime of the γ parameter, the classifier is clearly overfitting the data.

As seen from the examples in **Figs. 13.4** and **13.5**, the parameter γ of the Gaussian kernel and the degree of polynomial kernel determine the flexibility of the resulting SVM in fitting the data. If this complexity parameter is too large, overfitting will occur (bottom panels in **Fig. 13.5**).

A question frequently posed by practitioners is "which kernel should I use for my data?" There are several answers to this question. The first is that it is, like most practical questions in machine learning, data dependent, so several kernels should be tried. That being said, we typically follow the following procedure: try a linear kernel first, and then see if you can improve on its performance using a nonlinear kernel. The linear kernel provides a useful baseline, and in many bioinformatics applications provides the best results:

the flexibility of the Gaussian and polynomial kernels often leads to overfitting in high-dimensional datasets with a small number of examples, microarray datasets being a good example. Furthermore, an SVM with a linear kernel is easier to tune since the only parameter that affects performance is the soft-margin constant. Once a result using a linear kernel is available, it can serve as a baseline that you can try to improve upon using a nonlinear kernel. Between the Gaussian and polynomial kernels, our experience shows that the Gaussian kernel usually outperforms the polynomial kernel in both accuracy and convergence time if the data are normalized correctly and a good value of the width parameter is chosen. These issues are discussed in the next sections.

6. Model Selection

The dependence of the SVM decision boundary on the SVM hyperparameters translates into a dependence of classifier accuracy on the hyperparameters. When working with a linear classifier, the only hyperparameter that needs to be tuned is the SVM soft-margin constant. For the polynomial and Gaussian kernels, the search space is two-dimensional. The standard method of exploring this two-dimensional space is via grid-search; the grid points are generally chosen on a logarithmic scale and classifier accuracy is estimated for each point on the grid. This is illustrated in **Fig. 13.6**. A classifier is then trained using the hyperparameters that yield the best accuracy on the grid.
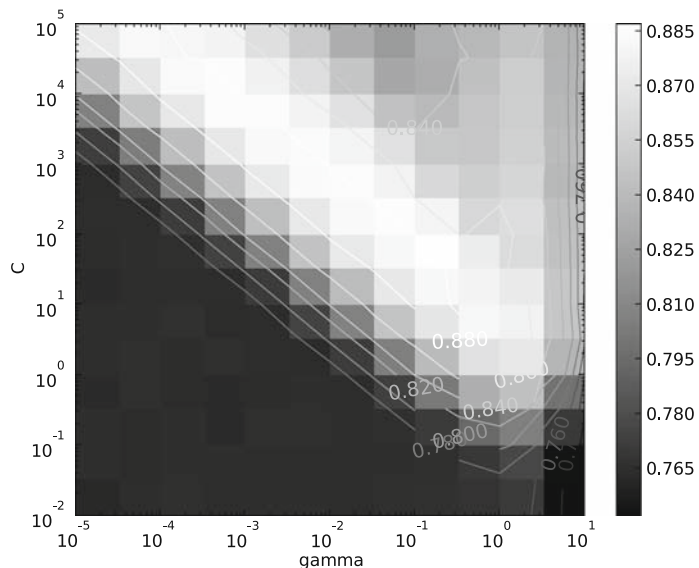
Fig. 13.6. SVM accuracy on a grid of parameter values.

The accuracy landscape in **Fig. 13.6** has an interesting property: there is a range of parameter values that yield optimal classifier performance; furthermore, these equivalent points in parameter space fall along a "ridge" in parameter space. This phenomenon can be understood as follows. Consider a particular value of (γ, C). If we decrease the value of γ, this decreases the curvature of the decision boundary; if we then increase the value of C the decision boundary is forced to curve to accommodate the larger penalty for errors/margin errors. This is illustrated in **Fig. 13.7** for two-dimensional data.
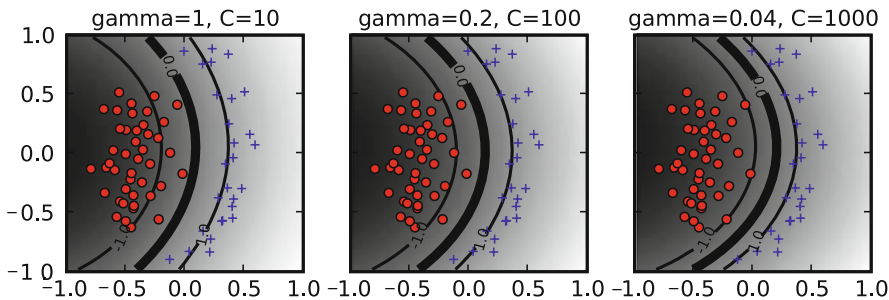
Fig. 13.7. Similar decision boundaries can be obtained using different combinations of SVM hyperparameters. The values of C and γ are indicated on each panel and the figure style follows **Fig. 13.3**.

7. SVMs for Unbalanced Data

Many datasets encountered in bioinformatics and other areas of application are unbalanced, i.e., one class contains a lot more examples than the other. Unbalanced datasets can present a challenge when training a classifier and SVMs are no exception – *see* (13) for a general overview of the issue. A good strategy for producing a high-accuracy classifier on imbalanced data is to classify any example as belonging to the majority class; this is called the majority-class classifier. While highly accurate under the standard measure of accuracy such a classifier is not very useful. When presented with an unbalanced dataset that is not linearly separable, an SVM that follows the formulation Equation [10] will often produce a classifier that behaves similarly to the majority-class classifier. An illustration of this phenomenon is provided in **Fig. 13.8**.

The crux of the problem is that the standard notion of accuracy (the success rate or fraction of correctly classified examples) is not a good way to measure the success of a classifier applied to unbalanced data, as is evident by the fact that the majority-class classifier performs well under it. The problem with the success rate is that it assigns equal importance to errors made on examples belonging to the majority class and the minority class. To correct for the
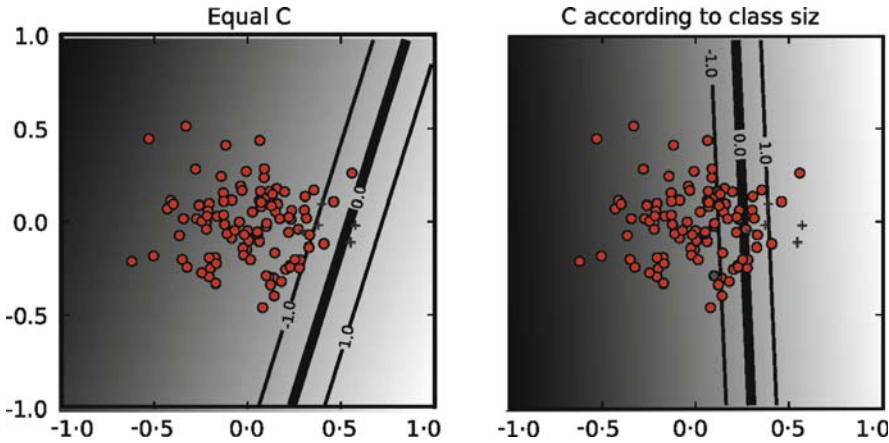
Fig. 13.8. When data are unbalanced and a single soft-margin is used, the resulting classifier (*left*) will tend to classify any example to the majority class. The solution (*right panel*) is to assign a different soft-margin constant to each class (see text for details). The figure style follows that of **Fig. 13.3**.

imbalance in the data, we need to assign different costs for misclassification to each class. Before introducing the balanced success rate, we note that the success rate can be expressed as

$$P(\text{success}|+)P(+) + P(\text{success}|-)P(-),$$

where $P(\text{success}|+)$ ($P(\text{success}|-)$) is an estimate of the probability of success in classifying positive (negative) examples, and $P(+)$ ($P(-)$) is the fraction of positive (negative) examples. The balanced success rate modifies this expression to

$$\text{BSR} = (P(\text{success}|+) + P(\text{success}|-))/2,$$

which averages the success rates in each class. The majority-class classifier will have a balanced-success-rate of 0.5. A balanced error-rate is defined as $1-\text{BSR}$. The BSR, as opposed to the standard success rate, gives equal overall weight to each class in measuring performance. A similar effect is obtained in training SVMs by assigning different misclassification costs (SVM soft-margin constants) to each class. The total misclassification cost, $C\sum_i \xi_i$ is replaced with two terms, one for each class:

$$C\sum_{i=1}^{n}\xi_i \to C_+\sum_{i\in I_+}\xi_i + C_-\sum_{i\in I_-}\xi_i$$

where $C_+(C_-)$ is the soft-margin constant for the positive (negative) examples and $I_+(I_-)$ are the sets of positive (negative) examples. To give equal overall weight to each class, we want the total penalty for each class to be equal. Assuming that the number of misclassified examples from each class is proportional to the number of examples in each class, we choose C_+ and C_- such that

$$C_+ n_+ = C_- n_-,$$

where $n_+(n_-)$ is the number of positive (negative) examples. Or in other words

$$C_+/C_- = n_+/n_-.$$

This provides a method for setting the ratio between the soft-margin constants of the two classes, leaving one parameter that needs to be adjusted. This method for handling unbalanced data is implemented in several SVM software packages, e.g., LIBSVM (14) and PyML.

8. Normalization

Linear classifiers are known to be sensitive to the way features are scaled (*see* e.g. (14) in the context of SVMs). Therefore, it is essential to normalize either the data or the kernel itself. This observation carries over to kernel-based classifiers that use non-linear kernel functions: the accuracy of an SVM can severely degrade if the data are not normalized (14). Some sources of data, e.g., microarray or mass-spectrometry data require normalization methods that are technology-specific. In what follows, we only consider normalization methods that are applicable regardless of the method that generated the data.

Normalization can be performed at the level of the input features or at the level of the kernel (normalization in feature space). In many applications, the available features are continuous values, where each feature is measured in a different scale and has a different range of possible values. In such cases, it is often beneficial to scale all features to a common range, e.g., by *standardizing* the data (for each feature, subtracting its mean and dividing by its standard deviation). Standardization is not appropriate when the data are sparse since it destroys sparsity since each feature will typically have a different normalization constant. Another way to handle features with different ranges is to bin each feature and replace it with indicator variables that indicate which bin it falls in.

An alternative to normalizing each feature separately is to normalize each example to be a unit vector. If the data are explicitly represented as vectors, you can normalize the data by dividing each vector by its norm such that $||\mathbf{x}|| = 1$ after normalization. Normalization can also be performed at the level of the kernel, i.e., normalizing in feature space, leading to $||\phi(\mathbf{x})|| = 1$ (or equivalently $k(\mathbf{x},\mathbf{x}) = 1$). This is accomplished using the *cosine* kernel, which normalizes a kernel $k(\mathbf{x},\mathbf{x}')$ to

$$k_{\text{cosine}}(\mathbf{x}, \mathbf{x}') = \frac{k(\mathbf{x}, \mathbf{x}')}{\sqrt{k(\mathbf{x}, \mathbf{x})k(\mathbf{x}', \mathbf{x}')}}. \qquad [13]$$

Note that for the linear kernel, cosine normalization is equivalent to division by the norm. The use of the cosine kernel is redundant for the Gaussian kernel since it already satisfies $k(\mathbf{x},\mathbf{x}) = 1$. This does not mean that normalization of the input features to unit vectors is redundant: our experience shows that the Gaussian kernel often benefits from it. Normalizing data to unit vectors reduces the dimensionality of the data by one since the data are projected to the unit sphere. Therefore, this may not be a good idea for low-dimensional data.

9. SVM Training Algorithms and Software

The popularity of SVMs has led to the development of a large number of special purpose solvers for the SVM optimization problem (15). One of the most common SVM solvers is LIBSVM (14). The complexity of training of nonlinear SVMs with solvers such as LIBSVM has been estimated to be quadratic in the number of training examples (15), which can be prohibitive for datasets with hundreds of thousands of examples. Researchers have therefore explored ways to achieve faster training times. For linear SVMs, very efficient solvers are available which converge in a time which is linear in the number of examples (16, 17, 15). Approximate solvers that can be trained in linear time without a significant loss of accuracy were also developed (18).

There are two types of software that provide SVM training algorithms. The first type is specialized software whose main objective is to provide an SVM solver. LIBSVM (14) and SVMlight (19) are two popular examples of this class of software. The other class of software is machine learning libraries that provide a variety of classification methods and other facilities such as methods for feature selection, preprocessing, etc. The user has a large number of choices, and the following is an incomplete list of environments that provide an SVM classifier: Orange (20), The Spider (http://www.kyb.tuebingen.mpg.de/bs/people/spider/), Elefant (21), Plearn (http://plearn.berlios.de/), Weka (22), Lush (23), Shogun (24), RapidMiner (25), and PyML (http://pyml.sourceforge.net). The SVM implementation in several of these are wrappers for the LIBSVM library. A repository of machine learning open source software is available at http://mloss.org as part of a movement advocating distribution of machine learning algorithms as open source software (7).

10. Further Reading

This chapter focused on the practical issues in using support vector machines to classify data that are already provided as features in some fixed-dimensional vector-space. In bioinformatics, we often encounter data that have no obvious explicit embedding in a fixed-dimensional vector space, e.g., protein or DNA sequences, protein structures, protein interaction networks, etc. Researchers have developed a variety of ways in which to model such data with kernel methods. *See* (2, 8) for more details. The design of a good kernel, i.e., defining a set of features that make the classification task easy, is where most of the gains in classification accuracy can be obtained.

After having defined a set of features, it is instructive to perform *feature selection*: remove features that do not contribute to the accuracy of the classifier (26, 27). In our experience, feature selection does not usually improve the accuracy of SVMs. Its importance is mainly in obtaining better understanding of the data – SVMs, like many other classifiers, are "black boxes" that do not provide the user much information on why a particular prediction was made. Reducing the set of features to a small salient set can help in this regard. Several successful feature selection methods have been developed specifically for SVMs and kernel methods. The Recursive Feature Elimination (RFE) method, for example, iteratively removes features that correspond to components of the SVM weight vector that are smallest in absolute value; such features have less of a contribution to the classification and are therefore removed (28).

SVMs are two-class classifiers. Solving multiclass problems can be done with multiclass extensions of SVMs (29). These are computationally expensive, so the practical alternative is to convert a two-class classifier to a multiclass. The standard method for doing so is the so-called one-vs-the-rest approach, where for each class a classifier is trained for that class against the rest of the classes; an input is classified according to which classifier produces the largest discriminant function value. Despite its simplicity, it remains the method of choice (30).

Acknowledgments

References

1. Boser, B.E., Guyon, I.M., and Vapnik, V.N. (1992) A training algorithm for optimal margin classifiers. In D. Haussler, editor, *5th Annual ACM Workshop on COLT*, pp. 144–152, Pittsburgh, PA. ACM Press.

2. Schölkopf, B., Tsuda, K., and Vert, J-P., editors (2004) *Kernel Methods in Computational Biology.* MIT Press series on Computational Molecular Biology.

3. Noble, W.S. (2006) What is a support vector machine? *Nature Biotechnology* **24**, 1564–1567.

4. Shawe-Taylor, J. and Cristianini, N. (2004) *Kernel Methods for Pattern Analysis.* Cambridge University Press, Cambridge, MA.

5. Schölkopf, B. and Smola, A. (2002) *Learning with Kernels.* MIT Press, Cambridge, MA.

6. Hsu, C-W., Chang, C-C., and Lin, C-J. (2003) *A Practical Guide to Support Vector Classification.* Technical report, Department of Computer Science, National Taiwan University.

7. Sonnenburg, S., Braun, M.L., Ong, C.S. et al. (2007) The need for open source software in machine learning. *Journal of Machine Learning Research*, **8**, 2443–2466.

8. Cristianini, N. and Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines.* Cambridge University Press, Cambridge, MA.

9. Hastie, T., Tibshirani, R., and Friedman, J.H. (2001) *The Elements of Statistical Learning.* Springer.

10. Bishop, C.M. (2007) *Pattern Recognition and Machine Learning.* Springer.

11. Cortes, C. and Vapnik, V.N. (1995) Support vector networks. *Machine Learning* **20**, 273–297.

12. Chapelle, O. (2007) Training a support vector machine in the primal. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines.* MIT Press, Cambridge, MA.

13. Provost, F. (2000) Learning with imbalanced data sets 101. In *AAAI 2000 workshop on imbalanced data sets.*

14. Chang, C-C. and Lin, C-J. (2001) *LIBSVM: a library for support vector machines.* Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

15. Bottou, L., Chapelle, O., DeCoste, D., and Weston, J., editors (2007) *Large Scale Kernel Machines.* MIT Press, Cambridge, MA.

16. Joachims, J. (2006) Training linear SVMs in linear time. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 217–226.

17. Sindhwani, V. and Keerthi, S.S. (2006) Large scale semi-supervised linear SVMs. In *29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 477–484.

18. Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005) Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research* **6**, 1579–1619.

19. Joachims, J. (1998) Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods: Support Vector Machines.* MIT Press, Cambridge, MA.

20. Demsar, J., Zupan, B., and Leban, J. (2004) *Orange: From Experimental Machine Learning to Interactive Data Mining.* Faculty of Computer and Information Science, University of Ljubljana.

21. Gawande, K., Webers, C., Smola, A., et al. (2007) ELEFANT user manual (revision 0.1). Technical report, NICTA.

22. Witten, I.H., and Frank, E. (2005) *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 2nd edition.

23. Bottou, L. and Le Cun, Y. (2002) *Lush Reference Manual.* Available at http://lush.sourceforge.net

24. Sonnenburg, S., Raetsch, G., Schaefer, C. and Schoelkopf, B. (2006) Large scale multiple kernel learning. *Journal of Machine Learning Research* **7**, 1531–1565.

25. Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., and Euler, T. (2006) YALE: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

26. Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L., editors. (2006) *Feature Extraction, Foundations and Applications.* Springer Verlag.

27. Guyon, I., and Elisseeff, A. (2003) An introduction to variable and feature selection. *Journal of Machine Learning Research* **3**, 1157–1182. MIT Press, Cambridge, MA, USA.

28. Guyon, I., Weston, J., Barnhill, S., and Vapnik, V.N. (2002) Gene selection for cancer classification using support vector machines. *Machine Learning* **46**, 389–422.

29. Weston, J. and Watkins, C. (1998) Multiclass support vector machines. *Royal Holloway Technical Report CSD-TR-98-04.*

30. Rifkin, R. and Klautau, A. (2004) In defense of one-vs-all classification. *Journal of Machine Learning Research* **5**, 101–141.

Hidden Markov Models in Biology

Claus Vogl and Andreas Futschik

Abstract

Markov and Hidden Markov models (HMMs) are introduced using examples from linkage mapping and sequence analysis. In the course, the forward–backward, the Viterbi, the Baum-Welch (EM) algorithm, and a Metropolis sampling scheme are presented.

Key words: Markov models, Hidden Markov models, linkage mapping, sequence analysis, Viterbi algorithm, forward–backward algorithm, Baum-Welch (EM) algorithm, metropolis sampling.

1. Introduction

Markov models and especially Hidden Markov models (HMMs) have become very important tools in sequence analysis, linkage mapping, population genetics, and generally in bioinformatics. They promise relatively simple probabilistic approaches to many phenomena at computational costs only marginally above those for ad-hoc models, such as sliding windows.

In biological modeling, Markov processes appear naturally in many contexts. Evolution, in particular, can be modeled as a Markov process, such that the probability distribution of individuals in the next generation depends only on individuals in the current generation. Unfortunately, we usually only have current data, but not from earlier times, when events actually happened. This may lead to rather thorny inference problems. Markov models are often used in evolutionary biology, and especially population genetics, but the literature is older than the one on sequence analysis and narrow-sense bioinformatics and often relies on diffusion approximations (e.g., 1). This will not be covered in this chapter.

Generally, modeling of states along a chromosome or a section of a chromosome, such as a gene, has become the most prominent application of Markov and Hidden Markov models especially in linkage mapping (2, 3) and sequence analysis. In sequence analysis, the book by Durbin and colleagues (4) has been enormously influential and seems to have helped to transform the field of bioinformatics from using ad-hoc methods to probabilistic modeling.

In the following, definitions of Markov and Hidden Markov models are given. Markov chains will then be illustrated using linkage mapping. There the sequence of events along the chromosome is modeled as a Markov chain as it is done in the following example, a very simple model for detecting CpG islands. In the context of this second example, HMMs and some basic analysis methods will be discussed. Finally, applications of HMMs in bioinformatics are pointed out.

1.1. Definitions

A **(homogeneous) Markov chain** is a sequence of random variables

$$\theta_1 \rightarrow \theta_2 \rightarrow \cdots \rightarrow \theta_{i-1} \rightarrow \theta_i \rightarrow \theta_{i+1} \rightarrow \cdots \theta_{N-1} \rightarrow \theta_N$$

with the following properties:
- Markov property: The future depends on the past only via the most recently observed state, i.e.,

$$\Pr(\theta_{i+1}|\theta_1, \ldots, \theta_i) = \Pr(\theta_{i+1}|\theta_i). \qquad [1]$$

- Time homogeneity: The transition probabilities do not change over time, i.e., the probability $P(\theta_{i+1} = s|\theta_i = s^*) = \pi(s, s^*)$ does not depend on i.

We will assume that the random variables θ_i have a finite number of possible values, also called states.

A **hidden Markov process** is a Markov process where the random variables θ_i cannot be observed directly, but influence a sequence of observable variables $(y_1, , y_N)$. (We will try to follow the convention that observable variables or data are denoted by Latin symbols and unobservable variables or parameters are denoted by Greek symbols). We can summarize a canonical hidden Markov model as follows:

$$
\begin{array}{l|ccccccc}
\text{hidden states} & \cdots \rightarrow & \theta_{i-1} & \rightarrow & \theta_i & \rightarrow & \theta_{i+1} & \rightarrow \cdots \\
\textit{emit} & & \downarrow & & \downarrow & & \downarrow & \\
\text{observed states} & \cdots & y_{i-1} & & y_i & & y_{i+1} & \cdots
\end{array}
$$
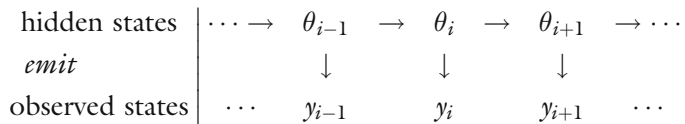
With hidden Markov models, the probability of the data y_i, at the ith step, only depends on the parameter θ_i, i.e.,

$$\Pr(y_i|\theta_1, \ldots, \theta_N) = \Pr(y_i|\theta_i). \qquad [2]$$

(More general models, where the dependence between y_{i-1} and y_i is not only over θ, are called Markov switching models.) For each i, the probability of the emitted data given the hidden states can then be represented as a vector. If the θ have, e.g., two possible states, then $\Pr(y_i|\theta) = (\Pr(y_i|\theta_i = 0), \Pr(y_i|\theta_i = 1))$. This probability is also called the emission probability, because the hidden states are thought to emit the observable data.

2. A Markov Chain in Linkage Mapping

In the following, basic concepts of Markov chains will be introduced using linkage mapping because there the transition probabilities often have a very simple form.

The most basic model in this context is for a backcross design, where two diploid inbred parental lines (*P1* and *P2*) are crossed with each other to produce a genetically homogenous filial generation *F1*, which has one chromosome from *P1* and the other from *P2* and is therefore heterozygous at every locus segregating between the parental lines. This *F1* is then backcrossed to one or the other parental line, e.g., to *P1*. Then, only the meiosis in the *F1* needs to be considered, because the other one produces only chromosomes of the line *P1*. According to Mendelian rules, the probability of a heterozygous state in the backcross, e.g., one allele from *P1* and the other from *P2* at a segregating locus *l*, is $1/2$ and that of a homozygous *P1* genotype is also $1/2$. Let us denote the heterozygous state (which we can only observe at segregating loci) with $\theta_l = 1$ and the homozygous stae with $\theta_l = 0$, such that we have $\Pr(\theta_l = 0) = \Pr(\theta_l = 1) = 1/2$. At the next segregating locus, e.g., at position $l + 1$, we again have two possible events and again their probabilities are equal. In general, the events at the neighboring loci will be correlated, such that the probability of observing the same configuration at both loci is higher than observing different ones.

Instead of going from one segregating locus to the next, let us consider the molecular level, and model sites, i.e., base pairs. Extending the above definition, we now set the hidden state to be $\theta_i = 1$, if the homologous nucleotides at a site come from different parental lines; otherwise, we set $\theta_i = 0$. If we start at the ith site, we expect Mendelian proportions of states of parental origin, as anywhere else in the genome, which we will denote with $\Pr(\theta_i = 0) = \Pr(\theta_i = 1) = 1/2$. Let us proceed to the next site $i + 1$. Obviously the probability of going from state 0 to state 1 must be the same as going from state 1 to state 0–otherwise the Mendelian rules would not hold at the next site. Hence the transition matrix is as follows:

$$T_{i,i+1} = \begin{pmatrix} \Pr(\theta_{i+1} = 0|\theta_i = 0)\Pr(\theta_{i+1} = 1|\theta_i = 0) \\ \Pr(\theta_{i+1} = 0|\theta_i = 1)\Pr(\theta_{i+1} = 1|\theta_i = 1) \end{pmatrix}$$
$$= \begin{pmatrix} 1 - \rho & \rho \\ \rho & 1 - \rho \end{pmatrix},$$

[3]

where ρ is the, generally unknown, recombination rate per base. If we assume that the recombination rate does not vary along the chromosome, we can leave away the subscript of the transition matrix T and obtain a homogeneous Markov chain. Proceeding along the chromosome we thus have a Markov chain satisfying:

$$\Pr(\theta_N) = (1/2, 1/2)T \cdots T$$
$$= (1/2, 1/2)T^N$$
$$= (1/2, 1/2) = P(\theta_1).$$

[4]

$(\Pr(\theta_N) = \Pr(\theta_1) = (1/2, 1/2)$ is the so-called stationary distribution of the Markov chain. If molecular markers are available at a segregating locus y_i, we can usually determine the site's state unequivocally in this simple cross.

Haldane's mapping function. It can be rather inconvenient to multiply the matrix T many times as markers may often be separated by several megabases. Diagonalizing T leads to the following approximate formula which is easy to evaluate:

$$T^N = 1/2 \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & (1 - 2\rho)^N \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$
$$\approx 1/2 \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & e^{-2\rho N} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 - r & r \\ r & 1 - r \end{pmatrix},$$

[5]

where $r = (1 - e^{-2\rho N})/2$. Illuminating its connection with Markov chains, this leads to the well-known Haldane mapping function (5), if recombination rates along a stretch of chromosome are approximately constant.

3. Hidden Markov Models

In more complex crossing designs, the data do not allow us to infer a unique state anymore. This more complicated situation occurs when more than one meiosis contributes to the next generation, e.g., when mating occurs within the *F1* generation. In such an intercross design, the probability of the four ordered genotypes (AA, Aa, aA, aa) is 1/4th, according to Mendel's rules and the independence of the two meioses. With bi-allelic codominant markers, the two homozygous genotypes can be distinguished

from the unordered heterozygote genotype. Thus, three of the four states can be distinguished by their phenotype. If markers are dominant, the heterozygote phenotype cannot be distinguished from one or the other homozygote phenotype.

Quantitative traits may also differ between the two lines. Such traits are, e.g., milk or meat yield in animals, or the number of leaf hairs per unit area in plants. If these traits are modeled as normally distributed, which is very common, the same continuum of phenotypes is possible for all genotypes, albeit with different probabilities.

Since there are four possible states (AA, Aa, aA, aa, from left to right), the transition matrix will be 4×4:

$$\mathbf{T}(\mathbf{l}, \mathbf{l+1}) = \begin{pmatrix} (1-r)^2 & (1-r)r & r(1-r) & r^2 \\ (1-r)r & (1-r)^2 & r^2 & r(1-r) \\ (1-r)r & r^2 & (1-r)^2 & r(1-r) \\ r^2 & (1-r)r & r(1-r) & (1-r)^2 \end{pmatrix} \qquad [6]$$

In this general case, the exact sequence of states (i.e., genotypes along the chromosome) is usually unknown, even with codominant markers, and we could thus use this example to illustrate the analysis techniques for HMMs. We will however use an even simpler model that is no more than a caricature of the underlying biological situation.

3.1. A Simple CpG Island Model

In mammalian genomes, GC or CG dinucleotides (usually written as CpG to distinguish them from the C–G base pair) are underrepresented. This is because the C nucleotide in CpG arrangements is often methylated, which in turn increases the probability of a C mutating into a T. In some relatively short regions of the genome (about a megabase or so) methylation is suppressed. In these regions, many more CpG dinucleotides are seen than elsewhere in the genome. We will model such "CpG islands" very simply below. (For a more realistic but also more complex model of CpG islands see, e.g., (4)).

The distinguishing feature of a CpG island is the relative richness in CpG dinucleotides. Hence we look at pairs of nucleotides sliding along the sequence and determine the number of CpG dinucleotides, which we contrast with all other dinucleotides. At each position i, the data can be in one of two states: $y_i = 1$ if the dinucleotide is CpG and $y_i = 0$ otherwise. The hidden variable θ_i indicates whether the ith dinucleotide is in a CpG island, i.e., $\theta_i = 1$ if it is, and $\theta_i = 0$ otherwise. By our model assumptions, the probability of $y_i = 1$, i.e., the emission probability of a CpG dinucleotide is greater if $\theta_i = 1$ than if $\theta_i = 0$. To be specific, we assume that $\Pr(y_i = 1 | \theta_i = 1) = 0.25$ and that $\Pr(y_i = 1 | \theta_i = 0) = 0.01$. Furthermore assume that the prior probability of the two states is $1/2$ and that the probability of transition between the hidden states is 0.01, i.e., the transition matrix is:

$$\mathbf{T} = \begin{pmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{pmatrix} \qquad [7]$$

With these assumptions, we simulated a sequence of length 2000, which we use for the subsequent analysis. In the following, three commonly used algorithms and a Metropolis sampler will be introduced, which serve (i) to determine the most probable path of hidden states through the sequence given the transition and emission probabilities (the **Viterbi** algorithm); (ii) to calculate the probability of the data given the transition and emission probabilities, thereby "summing out" the hidden parameters, which leads to the likelihood (the **forward–backward** algorithm); (iii) to calculate a (local) maximum likelihood estimate of the transition and emission probabilities (the **Baum-Welch** algorithm); and (iv) to provide a sample from the posterior distribution of the transition and emission probabilities using a **Metropolis** algorithm.

3.2. The Viterbi Algorithm

The **Viterbi** algorithm provides an efficient way to obtain the most probable path of hidden states

$$\theta^* = \operatorname{argmax}_\theta \Pr(y, \theta), \qquad [8]$$

assuming that all transition and emission probabilities are known. This dynamic programming algorithm proceeds as follows. First, one calculates recursively the quantities $v_k(i)$, which give the maximum probability of ending in the hidden state k at step i when observing $y_1, \ldots, y_i$. Obviously, $\max_k v_k(N)$ is then the probability of the most likely path and $\arg \max_k v_k(N)$ is the value of the last hidden state θ_N^* for the most likely path. The previous hidden states θ_i^* $(i < n)$ are obtained by backtracking.

Let p_{lk} denote the probability of switching from hidden state l to the hidden state k. The Viterbi algorithm can be summarized as follows:

- *Initialization (i=1):* Set $v_k(1) = \Pr(y_1|\theta_1 = k)\pi_k$ for all k, where π_k denotes the probability that the hidden Markov chain starts in state k.
- *Recursion (i = 2, ..., N):* Calculate for all $k : v_k(i) = \Pr(y_i|\theta_i = k) \max_l(v_l(i-1)p_{lk})$ and $\operatorname{ptr}_i(k) = \arg \max_l(v_l(i-1)p_{lk})$.
- *Termination:* $\Pr(y|\theta^*) = \max_k(v_k(N)); \theta_N^* = \arg \max_k(v_k(N))$
- *Traceback (i = N, ..., 1):* $\theta_{i-1}^* = \operatorname{ptr}_i(\theta_i^*)$

Numerical instabilities may occur. This can be avoided by taking the logarithm of the probabilities and summing. We note that the maximum will remain unaffected by the log-transformation.

In **Fig. 14.1**, we show an example of a data set simulated assuming the simple CpG model described above, the true hidden states and those inferred by the Viterbi algorithm.

Inferred States

True States
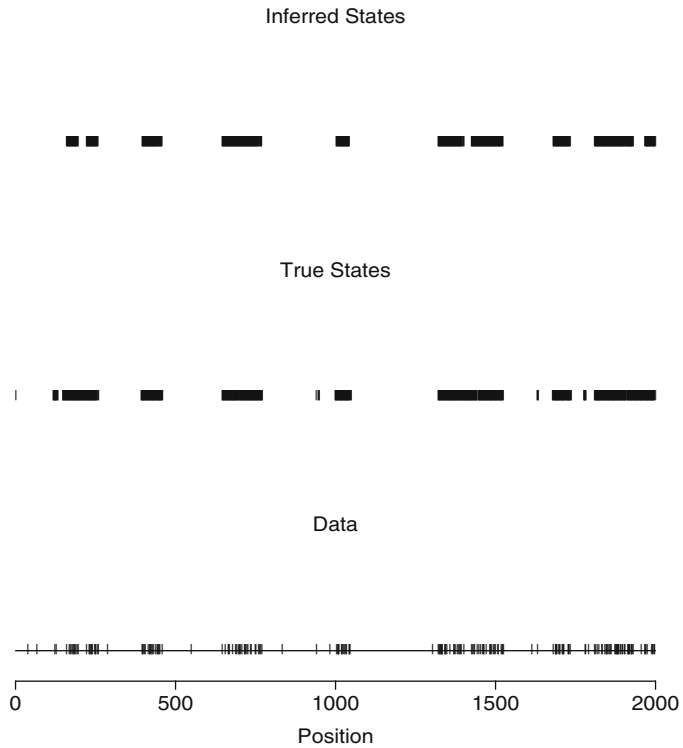
Data

Fig. 14.1. Plot of the data, the true hidden values θ, and the hidden states inferred by the Viterbi algorithm. The black bars denote true (resp. estimated) positions of CpG islands. There is generally good agreement between the true and inferred hidden states.

3.3. The Forward–
Backward algorithm

With the forward–backward algorithm, the probability of the data (y) given the transition and emission probabilities can be found. For this, the forward part of the forward–backward algorithm is sufficient. Furthermore, at each i, the probability of each hidden state can be calculated, for which the backward algorithm is necessary.

The algorithm is similar to the Viterbi algorithm. Again, we define (π_l) to be the probabilities of starting in state l.

The forward algorithm provides a recursive way of calculating $f_l(i+1) = \Pr(y_1, \ldots, y_i; \theta_{i+1} = l)$. By introducing (for technical reasons) a hidden end state θ_{N+1} for which no observed state is available, one finally obtains the probability of the whole observed sequence as $\Pr(y) = \sum_k f_k(N+1)$. The algorithm can be summarized as follows:

- *Initialization (i=1):* $f_l(1) = \pi_l$

Recursion (i = 1, , N);

$f_l(i+1) = \sum_k f_k(i) p_{kl} \Pr(y_i | \theta_i = k)$

- *Termination:* $\Pr(y) = \sum_k f_k(N+1)$

We note that the probability actually depends on the transition and emission probabilities, which may be abbreviated with $\mathbf{T}$ and $\mathbf{E}$, respectively, such that $\Pr(y|\mathrm{T}, \mathrm{E})$ is preferable. However, the probability does not anymore depend on the hidden states, which have been "summed out."

The backward algorithm provides a recursion for calculating $b_l(i) = \Pr(y_i, \ldots, y_N | \theta_i = l)$. Together with the forward algorithm, this permits to obtain the posterior probability for a hidden state at any position i, i.e., $\Pr(\theta_i = k|y)$ (which again depends on $\mathbf{T}$ and $\mathbf{E}$). Indeed

$$\Pr(\theta_i = k|y) = \frac{f_k(i)b_k(i)}{\Pr(y)}, \qquad [9]$$

where $\Pr(y)$ can be obtained either as $\Pr(y) = \sum_k f_k(i)\, b_k(i)$, or via the forward algorithm.

The algorithm can be summarized as follows.

- *Initialization (i = N + 1): $b_k(N + 1) = 1$ for all k*
- *Recursion (i = N,, 1): $b_k(i) = \sum_l p_{kl} b_l(i + 1)\, \Pr(y_i|\theta_i = k)$*

Both the forward and the backward algorithms may suffer from numerical instability, which is addressed, e.g., in (4).

In **Fig. 14.2**, we show the simulated CpG model data, the true hidden states and the posterior probabilities of the states inferred from the CpG algorithm.

Compared to the Viterbi algorithm, this way of calculating probable states is particularly useful, if many paths are about as likely as the most probable path. In addition, the forward–backward algorithm is an integral part of the Baum-Welch algorithm presented below.

3.4. The Baum-Welch or Expectation-Maximization algorithm

The Baum-Welch algorithm (6) is a special case of the Expectation-Maximization or EM algorithm (*see* (7), for more information on the EM algorithm). Hereby, the marginal likelihood (or posterior distribution) of the transition and emission probabilities is maximized. Contrary to the earlier assertions of always using Greek letters for unknown variables, we will continue using $\mathbf{T}$ and $\mathbf{E}$, for the now assumed unknown matrices, to not confuse the reader. The Baum-Welch algorithm can be used to obtain the maximizers both of the likelihood $\Pr(y|\mathrm{T}, \mathrm{E})$ and of the posterior $\Pr(\mathrm{T}, \mathrm{E}|y)$.

To motivate the Baum-Welch algorithm, we first assume that the path through the sequence of states is known, i.e., a known sequence of θ_is. For estimating the emission probabilities in the two states, we find in state $\theta = 0$ that a CpG dinucleotide is emitted in 6 out of 851 cases, such that the maximum likelihood estimate of $\Pr(y = 1|\theta = 0) \approx 0.007$ (compared to the true value of 0.01); in state $\theta = 1$ we observe the emission of a CpG dinucleotide in 293 out of 1,149 cases, such that the maximum likelihood estimate of $\Pr(y = 1|\theta = 1) \approx 0.26$ (compared to the true
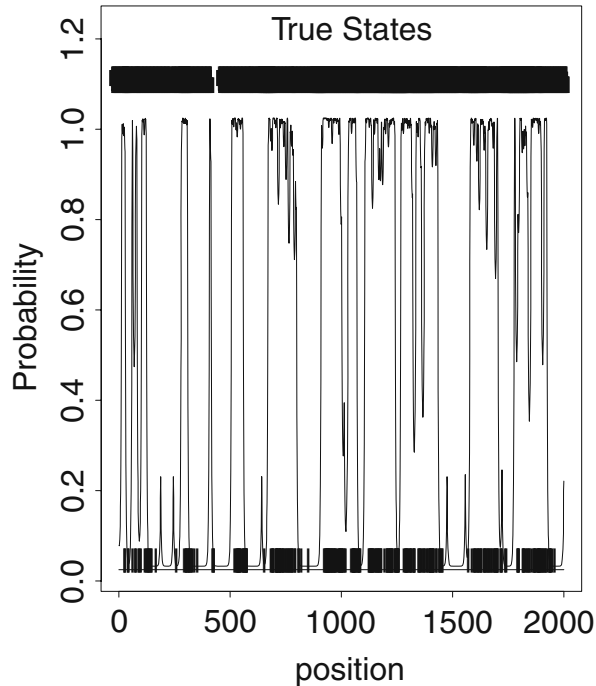
Fig. 14.2. Plot of the data (a vertical bar at the bottom denotes a CpG position in the data), the true hidden states θ_i (the black bars on top indicate CpG island positions), and the posterior probabilities of the CpG hidden state inferred by the forward–backward algorithm. It can be seen that the posterior probability for CpG tends to be high at the true CpG island positions, although the probabilities fluctuate considerably.

value of 0.25). Obviously, the error will get smaller, if more cases are observed. If only very few cases are expected, counts of 0 will be likely. It is then customary to add "pseudo-counts" to avoid estimates of 0. These pseudo-counts can be given an interpretation as Bayesian priors.

Generally, we do not know the true paths. We may then start by supplying initial guesses of transition and emission probabilities. Based on these probabilities and conditional on the observed data, we can calculate the expected number of transitions for each pair of hidden states. Furthermore, the expected number of emissions of a particular observed state can be obtained for each hidden state. Finally, one can obtain the expected frequency of each of the hidden states. Using these expectations, it is easy to re-estimate the unknown parameters. With these parameters, the above-mentioned expected values are recalculated and the process is iterated until convergence. Notice that all expected values are conditional on the data and therefore random.

The Baum-Welch algorithm may be summarized as follows:

- *Initialization:* Supply initial guesses for the transition probabilities, i.e., $\mathbf{T}_0$ and $\mathbf{E}_0$

- *Iteration, E-step:* Based on the forward and backward probabilities and given the old $\mathbf{T}_t$ and $\mathbf{E}_t$, calculate the expected number of transitions between the hidden states θ_i and θ_j as F_{kl}/F_k where

$$F_{kl} = \sum_i \Pr(\theta_i = k, \theta_{i+1} = l|y)$$

with

$$\Pr(\theta_i = k, \theta_{i+1} = l|y) = \frac{f_k(i)b_l(i+1)p_{kl}\Pr(y_i|\theta_i = k)}{\Pr(y)}.$$

Using (9), calculate furthermore the expected number of times the hidden state k is visited as $F_k \sum_i \Pr(\theta_i = k|y)$. Finally calculate the expected number of times being in state k and observing $\tilde{y}$ as $O_k(\tilde{y}) = \sum_i I(y_i = \tilde{y})\Pr(\theta_i = k|y)$.

- *Iteration, Estimation-step:*

  For all k and l, set $(\mathbf{T}_{t+1})_{k,l} = F_{kl}/F_k$. For all k and $\tilde{y}$, set $(\mathbf{E}_{t+1})_{k\tilde{y}} = O_k(\tilde{y})/F_k$.

- *Termination:* Terminate when the change from t to $t+1$ in the log likelihood or log posterior remains below a preset small value.

  This sequence of E- and M-steps converges to a local maximum of the marginal likelihood $\Pr(y|\mathbf{T}, \mathbf{E})$. If the process converges to the same local optimum from different initial values, this may provide empirical support for the global optimality of this point.

  In our example, the sequence was started from the true values and only the emission probabilities (instead of additionally the transition probabilities) were updated. After 50 iterations, the emission probabilities were 0.009 and 0.313 for the no CpG and the CpG state, respectively.

3.5. The Metropolis Sampler

The Baum-Welch algorithm provides the (local) maximum-likelihood (or posterior) estimator of the transition and emission probabilities. Yet it would also be interesting to gauge the reliability of the so-obtained estimates. With the Metropolis sampler and related Markov-chain Monte Carlo methods, (see, (7, 8, 9)), we can, in principle, sample from the posterior distribution of an estimator and thus obtain additional information on its precision.

The longer a Markov-chain Monte Carlo sampler runs, the better it will approximate the posterior distribution. For checking convergence, starting conditions should be overdispersed. Convergence can then be monitored by either visually inspecting the sequence of log-posterior values or with criteria in (7). The initially sampled items before convergence is reached – the so-called burn-in phase–should be discarded.

A Metropolis sampling scheme can be summarized as follows:

- *Initialization:* Supply guesses for the transition probabilities, i.e., $\mathbf{T}_0$ and $\mathbf{E}_0$ and calculate $\Pr(y|\mathbf{T}_0, \mathbf{E}_0)$ using the forward algorithm.

- *Iteration, proposal:* Suggest new randomly chosen transition and emission matrices $\mathbf{T}_p$ and $\mathbf{E}_p$ according to some proposal distribution.

- *Iteration, acceptance:* Calculate $\Pr(y|\mathbf{T}_p, \mathbf{E}_p)$ and determine the ratio:

$$a = \frac{\Pr(y|\mathbf{T}_p, \mathbf{E}_p)}{\Pr(y|\mathbf{T}_t, \mathbf{E}_t)} \qquad [10]$$

This formula is for the case of a uniform prior where the posterior is proportional to the likelihood. For other priors, one uses the product prior × likelihood (i.e., $\pi(\mathbf{T}, \mathbf{E}) \Pr(y|\mathbf{T}, \mathbf{E})$) both in the numerator and denominator instead. Accept the proposal if $a > 1$; if $a < 1$ accept with probability a.

- *Termination:* The more iterations the better the simulation approximates the posterior.

(Notice that there is a variant of the Metropolis algorithm known as the Gibbs sampler where the proposal distribution is identical to the conditional distribution of the parameter or parameters.)

In **Fig. 14.3**, we show results from the first 500 iterations of a Metropolis sampler started from the true values. Only the emission probabilities **E** were updated. In addition to estimates of the mean or median of the emission probabilities, we get an approximate sample from the posterior probability.

4. Discussion

A classical introduction to Hidden Markov models focussing on speech recognition is the paper by Rabiner (10). The use of Hidden Markov models has been proposed for the analysis of genome structure by Churchill (11). A more recent, but already classical reference on Hidden Markov models for biological sequence analysis is the book by Durbin and colleagues (4). The book by Koski (12) provides a mathematically more rigorous introduction to HMMs with applications to bioinformatics. They also discuss the use of profile HMMs for multiple (protein) sequence alignment. The use of HMMs for gene finding goes back to the work of Haussler and colleagues (13). A paper dealing with gene finding for eukaryotes is Burge and Carlin's (14). Hidden Markov models with continuous responses have been used for ion-channel modeling (e.g., 15). A large bibliography (period 1991–2000) on the use of HMMs in the biosciences, as well as in many other
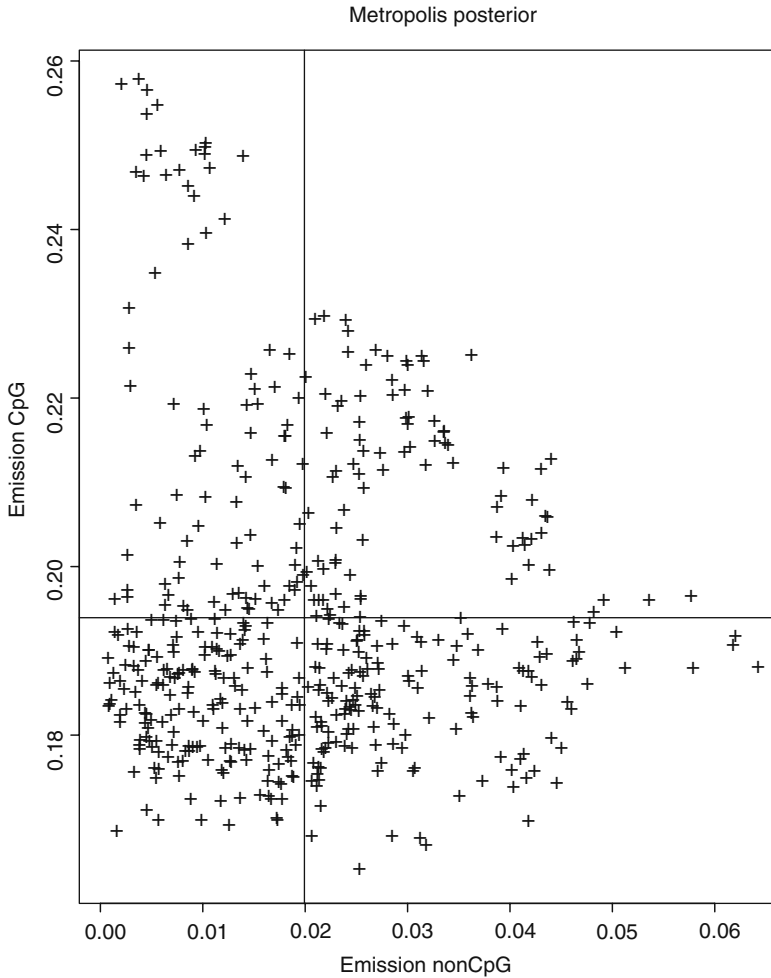
Fig. 14.3. Plot of the first 500 pairs of emission probabilities outside of CpG islands (x-axis) and within CpG islands (y-axis) in a Metropolis sampler started from the true values. Vertical and horizontal lines represent means.

fields can be found at the Webpage by Cappé "http://www.tsi.enst.fr/cappe/docs/hmmbib.html". A matlab toolbox for HMMs, pointers to other software, as well as to further online introductory material can be found at the Murphy's Webpage "http://www.cs.ubc.ca/murphyk/Software/HMM/hmm.html".

Furthermore, with the Bayesian approach (7) probabilistic submodels may be combined to provide comprehensive and, presumably, realistic models. With the computer intensive analysis methods presented herein, the posterior distribution of such models may be calculated or their modes approximated.

Acknowledgments

References

1. Ewens W (1979) Mathematical Population Genetics. Springer, New York.

2. Lander E, Green P (1987) Construction of multilocus genetic linkage maps in humans. PNAS 84:2363–2367. Genetics, Linkage map, multipointing.

3. Lander E, Botstein D (1989) Mapping mendelian factors underlying quantitative traits using rflp linkage maps. Genetics 121:185–199. Genetics, QTL.

4. Durbin R, Eddy S, Krogh A, Mitchison G (1998) Biological Sequence Analysis. Cambridge University Press, Cambridge.

5. Haldane J (1919) The combination of linkage values, and the calculation of distances between the loci of linked factors. J Genet 8:299–309.

6. Baum L (1972) An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. Inequalities 3:1–8.

7. Gelman A, Carlin J, Stern H, Rubin D (1995) Bayesian Data Analysis. Chapman & Hall, New York.

8. Liu J (2001) Monte Carlo Strategies in Scientific Computing. Springer, New York.

9. Kendall W, Liang F, Wang J-S (2005) Markov Chain Monte Carlo: Innovations and Applications. World Scientific, Singapore.

10. Rabiner L (1989) A tutorial on hidden markov models an selected applications in speech recognition. Proceedings of the IEEE 77: 257–286.

11. Churchill G (1992) Hidden markov chains and the analysis of genome structure. Comput and Chem 16:107–115.

12. Koski T (1992) Hidden Markov Models for Bioinformatics. Kluwer, Dordrecht.

13. Haussler D, Krogh A, Mian I, Sjolander K (1993) Protein modeling using hidden markov models: analysis of globins. Proc Twenty-Sixth Hawaii Int Conf on Syst Sci 1:792–802.

14. Burge C, Karlin S (1998) Finding the genes in genomic dna. Curr Opin Struct Biol 8:346–354.

15. Ball F, Rice J (1992) Stochastic models for ion channels: Introduction and bibliography. Math Biosci 112:189–206.

Section III

Database Annotations and Predictions

Chapter 15

Integrated Tools for Biomolecular Sequence-Based Function Prediction as Exemplified by the ANNOTATOR Software Environment

Georg Schneider, Michael Wildpaner, Fernanda L. Sirota, Sebastian Maurer-Stroh, Birgit Eisenhaber, and Frank Eisenhaber

Abstract

Given the amount of sequence data available today, in silico function prediction, which often includes detecting distant evolutionary relationships, requires sophisticated bioinformatic workflows. The algorithms behind these workflows exhibit complex data structures; they need the ability to spawn subtasks and tend to demand large amounts of resources. Performing sequence analytic tasks by manually invoking individual function prediction algorithms having to transform between differing input and output formats has become increasingly obsolete. After a period of linking individual predictors using ad hoc scripts, a number of integrated platforms are finally emerging. We present the ANNOTATOR software environment as an advanced example of such a platform.

Key words: sequence analysis, function prediction, visualization.

1. Introduction

Advances in sequencing technology have taken the number of sequences available in databases to unprecedented levels. Between the years 2000 and 2008, the number of sequences in Genbank has increased almost tenfold (1). Nevertheless, this apparent increase in the quantity of raw data has not been matched by a corresponding improvement in the ability to gain insights into the actual biological functions. Of the more than 6,000 sequences in the yeast *Saccharomyces cerevisae* which have been available since 1997 (2), there are still over 1,000 with uncharacterized function (3). In human, the situation is even worse, with more than half of the genes being functionally characterized only incompletely or not at all.

The reason for this widening gap is that the classical route to functional characterization involving experimental methods from the genetic and biochemical toolbox like specific knock-outs, targeted mutations, and a battery of biochemical assays is laborious, time consuming, and expensive. There is clearly a need for in silico methods that can be used for functional hypothesis generation to direct the subsequent experimental planning in the laboratory.

2. Integrated Sequence Analytic Tools

Recent years have seen a large number of new sequence analytic methods being added to the toolbox of the computational biologist. They are usually the offspring of an individual research project but, unfortunately, most of these tools do not have a standardized interface. Analyzing a particular sequence will, therefore, involve the invocation of a few command-line tools each of which with its own set of parameters, input formats, and result presentation. There are a number of Web-Services that have to be consulted, some of them requiring registration. Input and result formats are again not standardized, and it is difficult to further process data which is presented on an HTML page. The more advanced steps in sequence analysis typically involve the construction of a multiple alignment using a combination of an alignment program and a visual editor.

Putting everything together means handling numerous file formats, invocation parameters, and in the end sifting through Megabytes of textual (ASCII-type) information to come up with a functional characterization. A lot of time is lost converting from one sequence format to another or organizing result files.

Another concern is the issue of comparing results from different methods. As an example, a number of different approaches have been developed for the prediction of transmembrane regions (4–7). Having simultaneous access to the results of diverse algorithms can improve the sequence analysts' confidence in a certain prediction. Nevertheless, automatically applying rules for integrating results from various methods is only possible if the original results have been parsed into a standardized data model.

While the situation is tedious but can still be remedied by expanding a fair amount of time on an individual sequence, the process does not scale and it is impossible for a single expert to go beyond the analysis of a few sequences. In addition, the availability of sequence data from a wide range of organisms makes evolutionary relationships more traceable since previously "missing links" can serve as bridges between hitherto unconnected parts of the sequence universe. However, no researcher can "build"

these bridges manually. Many analysis protocols require the processing of an extended workflow with iterative steps (8). They collect large protein families and involve the repeated invocation of homology searches (9) as well as filtering for low-complexity (10) and coiled-coil regions (11). Handling these algorithms manually is out of the question since large families might require thousands of individual steps.

There are two basic approaches to automating sequence analytic tasks. The first one involves the creation of task-specific scripts with limited scope for reuse while a more advanced form of automation makes use of integrated frameworks to build complex workflows.

2.1. Ad Hoc Scripting

One step up from manually handling different programs and their outputs is the use of scripting languages like Perl or Python. These allow to stitch together individual tools into some form of a workflow. The task has been made easier by libraries like bioperl (12, 13) and its corresponding implementation for other programming languages (14), which encapsulate some fairly common operations like format conversion or result parsing.

The simplest approach to the development of a sequence analytic workflow, and the one overwhelmingly taken, is to look at the requirements of a specific project, define the input data, write a script that implements a certain heuristic, and output the results in some form of text file or maybe an HTML page.

This ad hoc approach confers a few undeniable advantages. First of all, it is fast and speed can be an issue when most of the sequence data is publicly available, since other researchers might make the same findings and publish them earlier. Second, scripting languages can be learned and deployed to a satisfactory degree in a fraction of the time it takes to grasp the intricacies of programming languages that allow to build more complicated but robust systems (Java, C++, Lisp, etc.). Flexibility to fast changing requirements might also be perceived as being higher, although this often turns out to be an illusion since it is traded for maintainability in the medium or long term.

The downside of this *quick and dirty* way is nevertheless considerable. The probability of having to reinvent the wheel for each sequence analytic project is quite high. Issues like internal representation, persistence, and user-interface have to be addressed over and over again. The format in which results are finally stored will also vary from project to project. The most common solution is to have a collection of flat files linger around in a folder hidden in the home directory of the researcher, gathering dust after the project terminated and a paper was published.

This means, a big opportunity for synergy is lost. Methods of data analysis developed for one project are often interesting in the context of another one, but the respective scripts might not run on

the current system setup, the input and output formats are incompatible and require complicated transformation steps or in the worst case, knowledge of the existence of the specific script has vanished with the departure of a researcher from the group.

The data itself could also be of interest in a new research context. A specific sequence showing up as the result of a candidate gene approach might have already formed part of the result of a search for orthologues run for another project by a different researcher. Nevertheless, this kind of reuse cannot be achieved with ad hoc scripts and requires the development of integrated frameworks.

2.2. Integrated Frameworks for Bioinformatic Workflows

Several strategies exist for the combination of distinct tools and algorithms into workflows using integrated frameworks. The most basic approach is to run individual command-line tools and link the respective inputs and outputs either manually or by writing a script. The EMBOSS Software Suite (15) is such an example.

Very large genome sequencing and annotation projects tend to build specialized systems that stitch together individual tools using a scripting language such as Perl. As an example, the annotation of the *Drosophila melanogaster* genome (16) was supported by an integrated computational pipeline (17). The pipeline consists of a database, a Perl module, and a job-management system. Analysis to be performed on the data sources is specified in a configuration file and progress monitored via an interactive command-line Perl interpreter or a Web front end. The results of the automatic analysis are presented to the curators, who then generate annotations manually.

GenDB (18) takes the concept a step further. Although it is still a genome annotation system that sequentially processes sequence data, there are a few features that set it apart from the previous two applications. Repetitive tasks can be handed over to so-called wizards, which are described as software agents that require complex and synchronized changes to several data objects. GenDB allows for the integration of arbitrary tools that create "observations" for a specific kind of region. Only a limited set of data about a specific result is actually stored in the database, with the rest being recomputed on demand.

A different approach is to design a system that initially serves no specific task at all, but presents a standardized data model and interface to the user. Tools are then incorporated by specifying their input parameters in a configuration file. Pise (19) is an early representative of such a system. A configuration file describing the command-line parameters of the available executables is used to automatically generate a (Web)–user-interface. After running a program, a menu appears showing all tools that would be able to operate on the result of the first one. This allows for the manual execution of a workflow.

Individual programs still run at the system's installation site, though. Soaplab (20) on the other hand goes beyond that by providing wrappers for command-line programs that allow them to be installed as Web-Services anywhere on the Internet. Communication between a client consuming the service and the server is via the standardized SOAP protocol (21). In this way, it is possible to programmatically build a pipeline with individual processing steps taking place on the Internet in a distributed manner.

Taverna (22) uses these Web-Services to build more elaborate workflows. It hides the complexities of calling the programs wrapped by Soaplab and chaining their respective inputs and outputs by providing a graphical user interface for the composition of workflows.

The problem of discovering which Web-Services are available in the first place and what data-types they require for input and in which format they produce their outputs, still remains though. BioMOBY (23) tries to remedy this by using a central registry for Web-Services. Service providers have to adhere to an ontology-based messaging structure that allows the client and server to look-up data types in the ontology and interpret and parse them correctly. This functionality was made available as an extension to the Taverna software system (24).

Orchestrating a workflow from distributed components nevertheless carries some serious disadvantages. Connectivity to the Web-Service can be problematic, as well as reliability of a service provided by somebody else (the service might even be discontinued at some point). On top of that, there could be concerns about sending sensitive data to an external entity. While simple workflows are well suited for this model, more complicated ones that include several iterative steps might run into difficulties regarding performance.

There are a number of software systems that offer workflow orchestration for components which are not implemented as Web-Services. Pegasys (25) is a Client-Server application where a "fat client" provides a graphical user-interface to create a workflow. The workflow is then sent to the server in an XML-representation, where it is converted into a directed acyclic graph. The application traverses the graph and schedules individual analysis on a compute-cluster, the results of which are then inserted into a relational database. Adapters are used to export the data for human interpretation or import into other applications.

Wildfire (26) is similar in that it also offers a graphical user-interface for workflow construction. It uses GEL (Grid Execution Language) (27) which runs the workflow directly or over the compute nodes of a cluster.

It remains to be seen if the composition of workflows from components in a graphical user-interface by nonprogrammers will be used extensively. It might be feasible to realize simple pipelines

in this way, but the implementation of sophisticated heuristics requiring access to internal data structures or performance optimization by multithreading is more likely to remain encapsulated in a plugin-style manner, offering all the advantages of a general purpose programming language.

3. The ANNOTATOR, an Example of an Integrated Sequence Analytic Tool

The ANNOTATOR software environment, which is being actively developed at the Bioinformatics Institute, A*Star (http://www.annotator.org), implements many of the features discussed above. Biological objects are represented in a unified data model and long-term persistence in a relational database is supplied by an object-relational mapping layer (*see* **Fig. 15.1**). Data to be analyzed can be provided in different formats ranging from Web-based forms, FASTA formatted flat files to remote import over a SOAP interface.

Import

Analysis

Visualisation

Persistence Layer
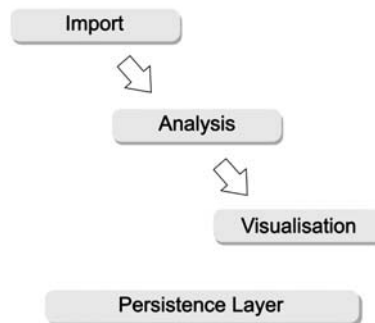
Fig. 15.1. Architecture of the ANNOTATOR software environment.

3.1. Algorithm Integration

Currently over 30 external sequence analytic algorithms are integrated using a plugin-style mechanism (*see* **Fig. 15.2**) and can be applied to uploaded sets of sequences. The display of applicable algorithms is such that it closely follows the standard procedure for segment based sequence analysis, which is based on the assumption that proteins are chains of functional units that can be analyzed independently with the overall function of the protein arising from the synthesis of the functions predicted for each individual module (28). The initial steps are aimed at finding (i) nonglobular regions and (ii) known globular domains. Available algorithms range among others from methods for identifying regions without intrinsically preferred structure or of low sequence complexity
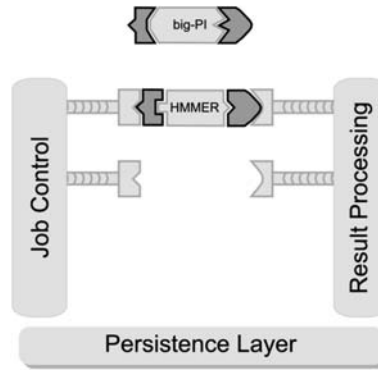
Fig. 15.2. Plugin mechanism for integrating external algorithms.

(29–31), sites of posttranslational modifications (32–37), targeting signals (7, 38), short sequence motif and globular domain pattern searches (39–44) to tools for detecting homology relationships (9, 45).

The last step of segment-based sequence analysis involves the identification of distantly related homologues to query sequence segments that remain without match in the preceding two analysis steps. While tools like PSI-BLAST (9) exist that provide a standard form of iterative family collection, it is often necessary to implement a more sophisticated heuristic to detect weaker links throughout the sequence space. The implementation of such a heuristic might require, among other tasks, the spawning of numerous external algorithms, the manipulation of alignments as well as the persistence of intermediate results.

It should be obvious that the mechanism of wrapping an external algorithm will not be sufficient in this case. While the logic of the heuristic could be implemented externally, it would still need access to internal data objects, as well as the ability to submit jobs to a compute-cluster. For this reason, an extension mechanism for the ANNOTATOR was devised which allows for the integration of algorithms that need access to internal mechanisms and data.

A typical example for using this extension mechanism to implement a sophisticated search heuristic is the FAMILY-SEARCHER, an integrated algorithm that is used to uncover homology relationships within large superfamilies of protein sequences. Applying this algorithm, the evolutionary relationship between classical mammalian lipases and the human adipose triglyceride lipase (ATGL) was established (8).

For large families, the amount of data produced when starting with one particular sequence as a seed can easily cross the Terabyte barrier. At the same time, the iterative procedure will spawn the execution of tens of thousands of individual homology searches. It is clearly necessary to have access to a cluster of compute nodes for

the heuristic and to have sophisticated software tools for the analysis of the vast output to terminate the task in a reasonable timeframe.

3.2. Visualization

The visualization of results is an important aspect of a sequence analysis system because it allows an expert to gain an immediate condensed overview of possible functional assignments. The ANNOTATOR offers specific visualizers both at the individual sequence as well as at the set level.

The visualizer for an individual sequence projects all regions that have been found to be functionally relevant onto the original sequence. The regions are grouped into panes and color-coded which makes it easy to spot consensus among a number of predictors for the same kind of feature (e.g., transmembrane regions). Zooming capabilities as well as rulers facilitate the exact localization of relevant amino acids.

The ability to analyze potentially large sets of sequences marks a qualitative step up from the focus on individual proteins. Alternative views of sets of proteins make it possible to find features that are conspicuously more frequent pointing to some interesting property of the sequence set in question. The *histogram view* in the ANNOTATOR is an example of such a view. It displays a diagram where individual features (e.g., domains) are ordered by their abundance within a set of sequences.

Another example is the *taxonomy view*. It shows the taxonomic distribution of sequences within a particular sequence set. It is then possible to apply certain operators that will extract a portion of the set that corresponds to a branch of the taxonomic tree which can then be further analyzed. One has to keep in mind that a set of sequences is not only created when a user uploads one but also when a particular result returns more than one sequence. Alignments from homology searches are treated in a similar manner and the same operators can be applied to them.

4. Conclusions

The enormous amount of sequence data available to biomolecular researchers makes the development of applications that can organize and detect patterns that relate sequences and functions an absolute necessity. At the same time, algorithms for predicting a particular function or uncovering distant evolutionary relationships (which ultimately allows transferring functional annotations) have become ever more demanding on resources. The output as well as intermediate results can no longer be assessed or reused manually and require sophisticated integrated frameworks.

The ANNOTATOR software provides crucial support for these tasks by supplying an infrastructure capable of applying a large array of sequence analytic methods to protein sequences, presenting the user with a condensed overview of possible functional assignments and, at the same time, allowing drill down to raw data from intermediate results for validation purposes.

References

1. Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., Wheeler, D. L. (2008) Gen-Bank. *Nucleic Acids Res* 36, D25–D30, 10.1093/nar/gkm929.

2. Blattner, F. R., Plunkett, G., Bloch, C. A., Perna, N. T., Burland, V., Riley, M., Collado-Vides, J., Glasner, J. D., Rode, C. K., Mayhew, G. F., et al. (1997) The Complete genome sequence of Escherichia coli K-12. *Science* 277, 1453–1462, 10.1126/science.277.5331.1453.

3. Peña-Castillo, L., Hughes, T. R. (2007) Why are there still over 1000 uncharacterized yeast genes? *Genetics* 176, 7–14, 10.1534/genetics.107.074468.

4. Cserzo, M., Eisenhaber, F., Eisenhaber, B., Simon, I. (2004) TM or not TM: transmembrane protein prediction with low false positive rate using DAS-TMfilter. *Bioinformatics* 20, 136–137.

5. Tusnády, G. E., Simon, I. (2001) The HMMTOP transmembrane topology prediction server. *Bioinformatics* 17, 849–850.

6. Krogh, A., Larsson, B., von Heijne, G., Sonnhammer, E. L. (2001) Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J Mol Biol* 305, 567–580, 10.1006/jmbi.2000.4315.

7. Käll, L., Krogh, A., Sonnhammer, E. L. L. (2004) A combined transmembrane topology and signal peptide prediction method. *J Mol Biol* 338, 1027–1036, 10.1016/j.jmb.2004.03.016.

8. Schneider, G., Neuberger, G., Wildpaner, M., Tian, S., Berezovsky, I., Eisenhaber, F. (2006) Application of a sensitive collection heuristic for very large protein families: evolutionary relationship between adipose triglyceride lipase (ATGL) and classic mammalian lipases. *BMC Bioinformatics* 7, 164, 10.1186/1471-2105-7-164.

9. Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25, 3389–3402.

10. Wootton, J. C. (1994) Non-globular domains in protein sequences: automated segmentation using complexity measures. *Comput Chem* 18, 269–285.

11. Lupas, A., Van Dyke, M., Stock, J. (1991) Predicting coiled coils from protein sequences. *Science* 252, 1162–1164, 10.1126/science.252.5009.1162.

12. Stajich, J. E., Block, D., Boulez, K., Brenner, S. E., Chervitz, S. A., Dagdigian, C., Fuellen, G., Gilbert, J. G. R., Korf, I., Lapp, H., et al. (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Res* 12, 1611–1618, 10.1101/gr.361602.

13. Stajich, J. E. (2007) An Introduction to BioPerl. *Methods Mol Biol* 406, 535–548.

14. Mangalam, H. (2002) The Bio* toolkits – a brief overview. *Brief Bioinform* 3, 296–302.

15. Rice, P., Longden, I., Bleasby, A. (2000) EMBOSS: the European molecular biology open software suite. *Trends Genet* 16, 276–277.

16. Misra, S., Crosby, M. A., Mungall, C. J., Matthews, B. B., Campbell, K. S., Hradecky, P., Huang, Y., Kaminker, J. S., Millburn, G. H., Prochnik, S. E., et al. (2002) Annotation of the *Drosophila melanogaster* euchromatic genome: a systematic review. *Genome Biol* 3, RESEARCH0083.

17. Mungall, C. J., Misra, S., Berman, B. P., Carlson, J., Frise, E., Harris, N., Marshall, B., Shu, S., Kaminker, J. S., Prochnik, S. E., et al. (2002) An integrated computational pipeline and database to support whole-genome sequence annotation. *Genome Biol* 3, RESEARCH0081.

18. Meyer, F., Goesmann, A., McHardy, A. C., Bartels, D., Bekel, T., Clausen, J., Kalinowski, J., Linke, B., Rupp, O., Giegerich, R., et al. (2003) GenDB – an open source genome annotation system for prokaryote genomes. *Nucleic Acids Res* 31, 2187–2195.

19. Letondal, C. (2001) A Web interface generator for molecular biology programs in Unix. *Bioinformatics* 17, 73–82.

20. Senger, M., Rice, P., Oinn, T. (2003) Soaplab – a unified Sesame door to analysis tools. In *Proceedings of the UK e-Science, All Hands Meeting*. Simon J Cox, pp. 509–513.

21. Gudgin, M., Hadley, M., Mendelsohn, N., Jean-Jaques, M., Nielsen, H. (2003) SOAP Version 1.2 Part 1: Messaging Framework. *W3C Recommendation*. Available at: http://www.w3.org/TR/soap12-part1.

22. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., et al. (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 20, 3045–3054, 10.1093/bioinformatics/bth361.

23. Wilkinson, M. D., Senger, M., Kawas, E., Bruskiewich, R., Gouzy, J., Noirot, C. (2008) Interoperability with Moby 1.0–It's better than sharing your toothbrush! *Brief Bioinformatics*, 10.1093/bib/bbn003, 10.1093/bib/bbn003.

24. Kawas, E., Senger, M., Wilkinson, M. D. (2006) BioMoby extensions to the Taverna workflow management and enactment software. *BMC Bioinformatics* 7, 523.

25. Shah, S. P., He, D. Y. M., Sawkins, J. N., Druce, J. C., Quon, G., Lett, D., Zheng, G. X. Y., Xu, T., Ouellette, B. F. F. (2004) Pegasys: software for executing and integrating analyses of biological sequences. *BMC Bioinformatics* 5, 40.

26. Tang, F., Chua, C. L., Ho, L., Lim, Y. P., Issac, P., Krishnan, A. (2005) Wildfire: distributed, Grid-enabled workflow construction and execution. *BMC Bioinformatics* 6, 69.

27. Lian, C. C., Tang, F., Issac, P., Krishnan, A. (2005) GEL: grid execution language. *J Parallel Distr Com* 65, 857–869.

28. Eisenhaber, F. (2006) Prediction of protein function. In *Discovering Biomolecular Mechanisms with Computational Biology*. Springer, US, pp. 39–54.

29. Promponas, V. J., Enright, A. J., Tsoka, S., Kreil, D. P., Leroy, C., Hamodrakas, S., Sander, C., Ouzounis, C. A. (2000) CAST: an iterative algorithm for the complexity analysis of sequence tracts. Complexity analysis of sequence tracts. *Bioinformatics* 16, 915–922.

30. Wootton, J. C. (1994) Non-globular domains in protein sequences: automated segmentation using complexity measures. *Comput Chem* 18, 269–285.

31. Dosztányi, Z., Csizmók, V., Tompa, P., Simon, I. (2005) The pairwise energy content estimated from amino acid composition discriminates between folded and intrinsically unstructured proteins. *J Mol Biol* 347, 827–839, 10.1016/j.jmb.2005.01.071.

32. Eisenhaber, B., Bork, P., Eisenhaber, F. (1999) Prediction of potential GPI-modification sites in proprotein sequences. *J Mol Biol* 292, 741–758, 10.1006/jmbi.1999.3069.

33. Eisenhaber, B., Wildpaner, M., Schultz, C. J., Borner, G. H. H., Dupree, P., Eisenhaber, F. (2003) Glycosylphosphatidylinositol lipid anchoring of plant proteins. Sensitive prediction from sequence- and genome-wide studies for Arabidopsis and rice. *Plant Physiol* 133, 1691–1701, 10.1104/pp.103.023580.

34. Eisenhaber, B., Schneider, G., Wildpaner, M., Eisenhaber, F. (2004) A sensitive predictor for potential GPI lipid modification sites in fungal protein sequences and its application to genome-wide studies for *Aspergillus nidulans*, *Candida albicans*, *Neurospora crassa*, *Saccharomyces cerevisiae* and *Schizosaccharomyces pombe*. *J Mol Biol* 337, 243–253, 10.1016/j.jmb.2004.01.025.

35. Maurer-Stroh, S., Eisenhaber, B., Eisenhaber, F. (2002) N-terminal N-myristoylation of proteins: prediction of substrate proteins from amino acid sequence. *J Mol Biol* 317, 541–557, 10.1006/jmbi.2002.5426.

36. Maurer-Stroh, S., Eisenhaber, B., Eisenhaber, F. (2002) N-terminal N-myristoylation of proteins: refinement of the sequence motif and its taxon-specific differences. *J Mol Biol* 317, 523–540, 10.1006/jmbi.2002.5425.

37. Maurer-Stroh, S., Eisenhaber, F. (2005) Refinement and prediction of protein prenylation motifs. *Genome Biol* 6, R55, 10.1186/gb-2005-6-6-r55.

38. Neuberger, G., Maurer-Stroh, S., Eisenhaber, B., Hartig, A., Eisenhaber, F. (2003) Prediction of peroxisomal targeting signal 1 containing proteins from amino acid sequence. *J Mol Biol* 328, 581–592.

39. Eddy, S. R. (1998) Profile hidden Markov models. *Bioinformatics* 14, 755–763.

40. Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., Cuche, B. A., de Castro, E., Lachaize, C., Langendijk-Genevaux, P. S., Sigrist, C. J. A. (2008) The 20 years of PROSITE. *Nucleic Acids Res* 36, D245–D249, 10.1093/nar/gkm977.

41. Schäffer, A. A., Wolf, Y. I., Ponting, C. P., Koonin, E. V., Aravind, L., Altschul, S. F.

(1999) IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices. *Bioinformatics* 15, 1000–1011.

42. Marchler-Bauer, A., Panchenko, A. R., Shoemaker, B. A., Thiessen, P. A., Geer, L. Y., Bryant, S. H. (2002) CDD: a database of conserved domain alignments with links to domain three-dimensional structure. *Nucleic Acids Res* 30, 281–283.

43. Letunic, I., Doerks, T., Bork, P. (2009) SMART 6: recent updates and new developments. *Nucleic Acids Res* 37, D229–D232, 10.1093/nar/gkn808.

44. Finn, R. D., Tate, J., Mistry, J., Coggill, P. C., Sammut, S. J., Hotz, H., Ceric, G., Forslund, K., Eddy, S. R., Sonnhammer, E. L. L., et al. (2008) The Pfam protein families database. *Nucleic Acids Res* 36, D281–D288, 10.1093/nar/gkm960.

45. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., Lipman, D. J. (1990) Basic local alignment search tool. *J Mol Biol* 215, 403–410, 10.1006/jmbi.1990.9999.

Chapter 16

Computational Methods for Ab Initio and Comparative Gene Finding

Ernesto Picardi and Graziano Pesole

Abstract

High-throughput DNA sequencing is increasing the amount of public complete genomes even though a precise gene catalogue for each organism is not yet available. In this context, computational gene finders play a key role in producing a first and cost-effective annotation. Nowadays a compilation of gene prediction tools has been made available to the scientific community and, despite the high number, they can be divided into two main categories: (1) ab initio and (2) evidence based. In the following, we will provide an overview of main methodologies to predict correct exon–intron structures of eukaryotic genes falling in such categories. We will take into account also new strategies that commonly refine ab initio predictions employing comparative genomics or other evidence such as expression data. Finally, we will briefly introduce metrics to in house evaluation of gene predictions in terms of sensitivity and specificity at nucleotide, exon, and gene levels as well.

Key words: gene prediction, gene finder, ab initio prediction, hidden Markov models, similarity searches, expression data, gene prediction accuracy.

1. Introduction

Technological improvements in high-throughput DNA sequencing are tremendously increasing the public availability of prokaryotic and eukaryotic genomes. Model organisms have been sequenced in both the plant and animal kingdoms and numerous new genomic sequences are daily added to the main primary databases. This growing amount of nucleotide sequence data requires also a concurrent development of adequate bioinformatics tools for a comprehensive understanding of the genetic information they encode as well as of their underlying biology. The genome sequence is in fact generally indicated a blueprint of an organism but deciphering all

instructions needed to express all typical biological traits is not a trivial issue. The complete sequence of the human genome, for instance, is freely available since the year 2001, but the entire set of encoded genes is not yet known with precision (1, 2). Computational gene finding, thus, represents a fertile field to develop ever better and more accurate tools and pipelines to get automatic annotation of uncharacterized DNA sequences.

Automatic gene prediction, however, raises the enigmatic question of what a gene in reality is. Insights from the human ENCODE project strongly support a new idea of gene that is far from being simply a genomic region coding a functional polypeptide (3, 4). Alternative splicing, noncoding RNAs involved in posttranscriptional and translational regulation and chimeric transcripts are only few examples explaining how our gene view is changing (5). Nevertheless, performing computational gene prediction needs the formulation of a practical even though limited definition of gene. Computational methods are in fact projected to execute commands according to specific statements, schemes, and rules and, thus, given a genomic sequence it is basic to have a clear idea of what to search for. Practically, an eukaryotic gene can be defined as a transcribed DNA region composed of exons and introns whose expression is regulated by *cis*-acting elements such as promoters mostly located upstream of the gene and other regulatory elements (e.g., enhancers) located also very far away form the transcription start site (TSS). Furthermore, specific sequences recognized by the splicing machinery are generally found between introns and exons, and inside introns.

After the transcription, the corresponding primary messenger is further processed by removing intronic regions and leading to the mature mRNA in which the coding region is flanked upstream and downstream from untranslated regions, 5′-UTR and 3′-UTR, respectively.

In this context, a gene prediction program works scanning an unknown genomic sequence to identify exact boundaries of many different signals common to most eukaryotic protein coding genes with the aim to automatically reconstruct a complete and reliable gene structure.

Although regions upstream and downstream of the protein coding genes can be very important for gene regulation, the lack of common sequence motifs likely make these regions refractory to prediction by current algorithms. It is well known that enhancers and silencers often lie many kilobases away from the gene TSS and are frequently not well conserved and sometimes cryptic. Consequently, most of gene prediction programs focus solely on identifying the protein coding regions of a gene. For this reason, in the following, we will provide an overview on the main approaches and systems for the identification of protein-coding eukaryotic genes, illustrating also their specific strength and weakness aspects.

Over the last 15 years and as new genome sequences are made available to the scientific community, several computational systems devoted to solve the hard task of predicting coding genes in eukaryotic genomes have been proposed (a comprehensive list is depicted in **Table 16.1**). In general, such automatic gene prediction systems

Table 16.1
A compilation of widespread ab initio and evidence-based gene prediction programs

Program	Web page	Evidence	Reference
GENSCAN	http://genes.mit.edu/GENSCAN.html	No	(16)
GENEID	http://www1.imim.es/geneid.html	No	(18)
SNAP	http://homepage.mac.com/iankorf/	No	(6)
GlimmerHMM	http://www.cbcb.umd.edu/software/GlimmerHMM/	No	(19)
GeneMark	http://exon.gatech.edu/GeneMark/eukhmm.cgi	No	(17)
AUGUSTUS	http://augustus.gobics.de/	ESTs, cDNAs, and proteins	(26, 52)
SGP2	http://genome.imim.es/software/sgp2/sgp2.html	TBLASTX hits	(48)
GENOMESCAN	http://genes.mit.edu/genomescan.html	BLASTX hits	(50)
TWINSCAN	http://mblab.wustl.edu/nscan/submit/	BLASTN hits and ESTs	(49)
GENOMINER	http://bl209.caspur.it/Gminer/	Complete genomes	(33)
ENSEMBL	http://www.ensembl.org/	ESTs, cDNAs, and proteins	(55)
N-SCAN	http://mblab.wustl.edu/nscan/submit/	ESTs, complete genomes	(8, 51)
EXOGEAN	http://www.biologie.ens.fr/dyogen/spip.php?rubrique4&lang=en	ESTs, cDNAs, and proteins	(43)
GENEWISE	http://www.ebi.ac.uk/Wise2/index.html	Proteins	(45)
ASPIC	http://t.caspur.it/ASPIC/	ESTs and cDNAs	(41, 42)
Eugène	http://www.inra.fr/mia/T/EuGene/	ESTs, cDNAs, and proteins	(20)
GAZE	http://www.sanger.ac.uk/Software/analysis/GAZE/	All available + ab initio	(59)
JIGSAW	http://www.cbcb.umd.edu/software/jigsaw/	All available + ab initio	(60)

can be divided into two main categories: (1) ab initio (or intrinsic) and (2) evidence-based (or extrinsic). Ab initio methods deal strictly with genomic sequences and make use of statistical approaches to search for coding regions and typical gene signals. In contrast, evidence-based methods, sometimes called also homology methods, attempt to find out genes using either similarity search procedures in the main databases or experimental data including expressed sequence tags (ESTs), full-length complementary DNAs (cDNAs), and even data from microarray hybridization experiments. Although evidence-based and ab initio methods can work independently, more accurate gene predictions have been obtained combining both the approaches. Ab initio computational tools in fact tend to over-predict coding exons misplacing start and stop codons. On the other hand, evidence-based tools fail in all cases in which genes are not experimentally supported or when no similarity can be found in the public databases.

Recently, a new category of gene predictors, defined sometimes consensus, has been described. Systems falling in such category take as input ab initio predictions from different gene finding programs, similarity search results, expression data (ESTs, cDNAs, and proteins) and combine all together in order to obtain a final consensus gene structure.

Anyway, researchers involved in genome annotation should beware of using gene prediction programs since they may not fit their expectations. Ab initio gene finders, for instance, predict poorly without adequate training sets or do not perform optimally in a foreign genome (6). Sometimes gene prediction tools do not correctly work with protein similarities against evolutionary distant organisms, and the use of cDNAs or ESTs from other species generally leads to inaccurate splice site and exon predictions (7, 8).

2. Methods

2.1. Ab Initio Gene Finding

After the completion of the primary sequence of a genome, the simplest and cost-effective approach is to carry out a genome-wide ab initio gene finding to annotate a complete set of exon–intron structures and infer all the potentially encoded proteins. Such programs in fact do not require experimental evidences or a priori knowledge of a specific genomic portion. They go through an uncharacterized DNA sequence searching for signals (such as start and stop codons, splice sites, polyadenylation sites, branch points) and features of coding regions, subsequently used to assemble a final gene structure.

Historically, the first ab initio gene finders were defined for prokaryotic genomes. The high gene density of these genomes

with the simpler and unbroken gene structure allowed the identification of two main regions, those coding for proteins (and thus translated) and those intergenic noncoding. Consequently, the issue of gene prediction was firstly limited in discerning coding from noncoding and sometimes open reading frames (ORFs) were simply identified looking for long genomic sequences (at least 300 bp) starting with the ATG codon and ending with a termination codon (TGA, TAA, TAG). However, it was not unlikely to find quite long unfunctional ORFs, especially in the antisense strand of the expressed ORFs (9).

Actually, different statistical approaches taking into account the codon usage or the nucleotide composition improved to global identification of coding regions (10, 11). Moreover, potential protein coding sequences show a period-3 bias due to the codon structure that can be modeled by appropriate mathematical techniques such as Fourier Transformation (12) or wavelets (13, 14). Recently, a new statistical approach based on Z-curve has also been proposed and described (15).

Although statistics methods revealed very useful in the case of microbial genomes, they are clearly inadequate to predict eukaryotic coding genes because of their more complex structures consisting of discontinuous genomic regions where short coding exons are joined together by noncoding and frequently very long introns. Many recent programs such as GENSCAN (16), GeneMark (17), or GeneID (18) distinguish coding from noncoding regions of eukaryotic genomes by means of an alternative coding measure based on hexanucleotide usage. Generally, six-mer distribution can be modeled through well-defined stochastic processes known as Markov chains in which the probability to observe a given nucleotide (A, C, G, or T) at a specific position depends only on the previous k nucleotides (where k is also indicated as the order of the Markov chain). For this reason, a Markov chain model enables the capturing of local dependencies between adjacent nucleotides in function of its order. From a practical point of view, for instance, a Markov model of the fifth order reflects the peculiar hexamer distribution. Since each element of a Markov model is a probability value, its estimation from real data sets depends on the number of available coding and noncoding sequences of the training sets. Although a higher order Markov chain may lead to a more accurate prediction, it generally requires a larger training set. The majority of currently used ab initio gene predictors including GENSCAN (16), GeneMark (17), or GeneID (18) discriminate coding from noncoding regions using Markov chains of order four or five. GENSCAN (16) and GeneMark (17) also adopt a more complex three-periodic Markov model in which single Markov models are used for each nucleotide of a codon. A different strategy is used by the program GlimmerHMM (19) in which different Markov models of order ranging from 0 to

k (k up to 8) are interpolated. The resulting interpolated Markov model (IMM), implemented also in Eugène (20) program, proved to overcome mistakes due to narrow training sets (21).

Unfortunately, Markov models alone are not able to correctly predict entire eukaryotic protein coding genes. They fail every time a short coding exon is encountered and the main reason of this failure is due to the low number of nucleotides that significantly reduces the power of whatever statistical method. Moreover, Markov models do not make assumptions about exon–intron borders as well as other characterized gene signals. Exons are in fact chained with introns and roughly 99% of exon–intron boundaries are functionally conserved (5′ GT-AG 3′) to allow the removal of introns by the posttranscriptional splicing mechanism. Splice sites (donors and acceptors), start and stop codons, intronic branch points, or polyadenylation sites are currently ab initio predicted by position weight matrices (PWMs) or related approaches. A PWM is simply a reference matrix to score each possible base at every position within a potential signal. Specific PWMs, graphically depicted as sequence logos, can be calculated on every gene signal from a large collection of genes. Matrices for splice donors, for instance, typically consider the three terminal nucleotides of an exon and the six starting nucleotides of the following intron, whereas matrices for splice acceptors generally take into account the last six nucleotides of an intron and the three first nucleotides of the following exon and sometimes include also the poly-pyrimidine tract (commonly found 15–30 nucleotides upstream the 3′ end of intron). Besides PWMs, more complex matrices such as weight array matrices (WAMs) are frequently implemented in modern gene finding tools in order to capture dependencies between adjacent positions.

Independent predictions of signal and coding regions normally yield a high number of false positives. Such a number is dramatically reduced combining signals and coding predictions. In this context, the program GeneID (18) is undoubtedly the best example. It is in fact designed following a simple hierarchical structure. First, eukaryotic gene signals (start and stop codons, splicing sites) are predicted and scored along an unknown genomic sequence using PWMs or WAMs. Subsequently, coding exons compatible with previously detected signals are assembled and assigned to one of the four categories: single, initial, internal, and terminal. Single exons begin with a start codon and end with stop codon; initial exons begin with a start codon and end with a donor site; internal exons begin with an acceptor site and end with a donor site; and terminal exons begin with an acceptor site and end with a termination codon. A specific score is also assigned to each exon and calculated as the sum of the scores of the defined signals plus the log-likelihood ratio for coding DNA according to a Markov model of order four or five. Finally, from the set of

predicted exons, the gene structure maximizing the sum of the score of its exons is assembled using a dynamic programming algorithm. GeneID in fact was the first gene predictor implementing an appropriate module (named GenAmic) to reconstruct full gene structures in linear time according to specific rules such as frame compatibility, exon order (an internal after an initial but never after a terminal), or simple gene and genomic structure requirements (a minimum intron length or a minimum intergenic distance) (22).

Although geneID is quite efficient in predicting eukaryotic genes, its accuracy is sometimes lower than the accuracy obtained using the new generation of ab initio gene finders based on hidden Markov models (HMMs) (23). HMMs are powerful statistical techniques firstly developed in the field of linguistics research, but they revealed highly useful also in many biological problems. They are currently used for finding periodicities and motifs in DNA, for producing reliable multiple alignments, or for predicting protein domains and secondary structures.

In the gene prediction context, an HMM can be thought as a stochastic process consisting of a discrete set of states (in which some of the details are unknown or *hidden*) and a set of transition probabilities to move from a state to another. Each state is a feature of the eukaryotic gene and while the HMM is in any particular state, it emits nucleotides which are visible and have got the same statistical properties of that state. Since the output of a regular HMM exhibits a length of one for each state within the hidden state space, the more complex generalized hidden Markov model (GHMM) has been developed. In this case, each state is a gene feature of arbitrary length such as exons, introns, splice sites, and poly-adenylation sites. Gene prediction programs including the very popular GENSCAN (16), Genie (24), HMMgene (25), SNAP (6), GlimmerHMM (19), and AUGUSTUS (26) model genomic sequences via a GHMM approach. Each program exhibits a specific state diagram even though all programs use the well-known Viterbi algorithm to produce a reliable gene architecture. This algorithm works like a directed acyclic graph where nodes are the sequences emitted by the GHMM.

Ab initio gene prediction systems as those previously cited are valuable in annotating newly sequenced genomes. Sometimes, they represent the only available approach to predict coding genes in genomic regions not yet supported by experimental evidence. However, since they are based on HMM or GHMM, all parameters of the model are probabilities that have to be inferred through an accurate training procedure. Occasionally, newly sequenced genomes may lack large enough samples of known genes from which to estimate model parameters. Consequently, they lead to incorrect gene predictions and a lot of false positives. At the same time, also predictions performed using ab initio gene

finders trained on another organism result generally unsuccessful. From a phylogenetic point of view, in fact, two organisms may be strongly related without sharing the same genomic features such as gene density, codon usage, or nucleotide composition. Recently, however, the idea of bootstrap parameter estimation has been introduced (6). According to this idea, a foreign gene finder is run on a novel genome and the resulting predictions are used to estimate the parameters for gene prediction for the novel genome (6). Anyway, it should be underlined that ab initio gene finders could work suboptimally every time the characteristics of a given genomic sequence do not fit those sampled in the training set.

2.2. Evidence-Based Gene Finding

In contrast with ab initio methods, gene prediction systems based on external evidences aim to identify gene structures using similarity search results and expression data (ESTs, cDNAs, and proteins). Generally, methods relying on evidence make use of a statistically significant similarity between an unknown genomic sequence and a protein or DNA sequence present in a database in order to determine transcribed and/or coding regions. Currently, the best approaches to detect similarities between sequences are based on local alignment methods either optimal such as Smith-Waterman algorithm (27) or heuristic as implemented in BLAST, FASTA, and BLAT (28–31) programs.

It has been estimated that almost 50% of the genes in a novel genome can be identified thanks to a sufficient similarity score with sequences stored in the main databases. However, even when good similarities are found, they do not ensure the accurate prediction of exact gene architectures such as the correct exon–intron boundaries. A part of predicted exons may be only partially identified. Nevertheless, the exponential growth of primary sequence databases makes similarity searches the key methodology to discriminate coding from noncoding regions. Over the last years, a huge compilation of approaches devoted to address this task has been proposed, including CRITICA (32) and GenoMiner (33). In particular, the latter bioinformatic tool is able to identify conserved sequence tags (CSTs) through the comparison of an unknown genomic region and one or more of the available complete genomes. The assessment of the coding or noncoding nature of each CST is performed through the computation of a coding potential score (CPS) based on the evaluation of the peculiar evolutionary dynamics of protein coding sequences at both the nucleotide and amino acid levels (33, 34).

Despite the usefulness of similarity search against protein and nucleic acid databases, all available software in this field suffers in predicting main gene signals. In many cases, some splice sites or small exons are completely missed. Such limitations of similarity-based tools have been actually accommodated in a new family of programs developed to optimally handle expression data. Full-length

cDNAs and ESTs, in fact, represent a tremendous source of evidence to identify exon–intron boundaries and complete gene structures. Programs as SIM4 (35), Spidey (36), GeneSeqer (37, 38), GMAP (39), and EST_Genome (40) have been recently released to perform a progressive alignment (sometimes also a mapping) of a genomic sequence against a cDNA database. The predominant strategy adopted is to align onto a genomic sequence one cDNA or EST sequence at a time using approximate algorithms and relaxing gap extension penalties to accommodate long introns. A part of these programs introduces also scoring schemes to improve the detection of exon–intron boundaries. However, many difficulties arise when a cDNA sequence differs from its corresponding genomic exons, due to polymorphisms, mutations, or sequencing errors. Sequencing errors in ESTs may be particularly misleading when they fall near exon–exon junctions, thus complicating the detection of correct splice sites. A way to reduce the errors due to the misalignments of expressed sequences is to perform a multiple EST sequence comparison and alignment against the genomic sequence. A method working in this direction has been implemented in the program ASPIC (41, 42). According to this algorithm, each EST or cDNA of a given cluster is split in a set of factors or pseudo-exons aligned onto the genomic sequence. Since repeats and short exons in the genomic sequence may produce ambiguous alignments (each factor may align in a number of equally probable ways), the correct exon–intron structure is obtained combining common pseudo-exons from different ESTs or cDNAs. In this methodology, the detection of canonical and noncanonical splice sites has been improved refining the exon–intron boundaries by a dynamic programming procedure. Moreover, compared to other available software, ASPIC exploits EST to genome multialignments to infer potential alternative transcripts through an ad hoc directed acyclic graph (41).

Differently from ASPIC, the recently introduced program Exogean (43) attempts to reconstruct exon–intron structures by aligning ESTs and cDNAs onto a genomic sequence using BLAT. After the alignment, different steps of directed acyclic graphs are followed to assemble complete gene structures and potential alternative splicing transcripts (43). Of course, Exogean and ASPIC are expected to fail in all cases in which ESTs cannot cover entire gene regions. However, Exogean should be more prone to errors since its alignments rely on BLAT that does not introduce specific corrections for ambiguously aligned splice sites. Anyway, BLAT ensures also the mapping and alignment of protein sequences even though this task has been specifically addressed in other programs such as Procrustes and GeneWise (44, 45). In particular, they work by cutting a query genomic sequence in all potential exons bordered by acceptor and donor sites and then build a full gene structure using the exons compatible with a given protein

sequence. However, inferences about not translated regions (5′ and 3′ UTRs) and noncoding exons or introns in these regions are generally precluded. Moreover, the accuracy of recovered gene structures is strongly related with the detected protein similarities (7). Therefore, it is expected that a higher protein similarity leads to a better gene prediction accuracy.

2.3. Combining Ab Initio and Evidence-Based Methods

Currently, many researchers tend to use independently ab initio and evidence-based methods, but more accurate and reliable gene predictions can be obtained combining both systems. The programs Doublescan (46) and SLAM (47), for instance, rely on sophisticated models of coding and noncoding DNA and splice signals, in addition to measures of sequence similarity. SLAM and Doublescan combine an optimal sequence alignment generated by Pair Hidden Markov Models (PHMMs) and modeling of the eukaryotic gene features adopting the so-called Generalized Pair HMMs (GPHMMs). In these methods, gene prediction and sequence alignment are obtained simultaneously.

A different class of programs adopt, instead, a more heuristic approach, and separate ab initio gene prediction from sequence similarity. In SGP2 (48), a query genomic sequence is contrasted against a known genome, sometimes called informant genome, using TBALSTX. Subsequently, high-scoring segment pairs (HSPs) are considered to increase the score of geneID predicted exons. In this way, exons predicted ab initio with low score but supported by sequence similarity can be included in the final gene model (48). A similar approach has been also implemented in the program TWINSCAN (49). In this case, however, HSPs obtained by a BLASTN search against an informant genome are converted into a representation called "conservation sequence." To each representation is then assigned a conservation probability used to modify the state probabilities of the GENSCAN model (49). This approach is reminiscent of that used in GENOMESCAN (50) to incorporate similarity to known proteins (via BLASTX) to modify the GENSCAN scoring scheme. In the more complex NSCAN (51) system, instead, the GENSCAN performance is improved by using the pattern of conservation from multiple informant genomes. Other existing programs such as AUGUSTUS-dual (52) or DOGFISH (53) work in a similar way.

All gene predictors based on sequence similarity harbor a theoretical advantage since they are not species-specific. In practice, however, the performance of these methods strongly depends on the evolutionary distance between compared sequences. Very large or very small distances, in fact, may prevent any significant improvement over ab initio systems. Moreover, the use of similarities from distantly related organisms could drive to inaccurate splice site predictions. A part of these limitations has been actually solved including the information from EST and cDNA

alignments. Significant improvements in gene prediction accuracy have been registered with TWINSCAN_EST (54), NSCAN_EST (8), and AUGUSTUS-EST (52) in which the predictions based on similarity searches are refined employing ESTs aligned to a query genomic sequence.

Besides methods previously described, a number of more or less complex pipelines have been also developed. ENSEMBL is of course one of the most important examples (55). In this pipeline, the automatic gene prediction is based on genomic information coming from four different sources: proteins and mRNAs from the corresponding species, proteins and mRNAs from other species, ESTs, and ab initio gene predictions supported by experimental data. A specific set of analysis tools is then used to handle with these different data sources. For example, protein similarities are considered by GeneWise, and cDNA and EST alignments by EST_Genome and Exonerate (56), where ab initio predictions are generated by GENSCAN. When ESTs and cDNAs data are sufficient, ENSEMBL is also able to predict alternative transcripts using the program ESTGenes (57) based on a directed acyclic graph.

The Pairagon + NSCAN-EST pipeline predicts eukaryotic coding genes in two steps. At the beginning, the Pairagon program based on PairHMM probability model is used to align native full-length cDNA against a query genomic sequence in order to produce a first set of reliable gene structures. Then, NSCAN_EST is applied to the remaining genomic regions of the query not covered by cDNAs leading to a second set of gene predictions.

Recently, a new pipeline called CEGMA (58) has been also proposed to provide an initial and reliable catalog of genes for newly sequenced genomes in absence of experimental data. In this case, conserved protein families occurring in a wide range of eukaryotes are mapped onto a novel genomic sequence in order to accurately identify the corresponding exon–intron structures. The core of the CEGMA pipeline includes the use of profile-HMMs.

Every time a novel genome is completely sequenced, the automatic annotation is a natural consequence. Given the high number of available gene predictors, however, selecting an appropriate tool is not a trivial issue. A common workflow is to produce a first annotation by ab initio gene predictors and then refine gene structures supported by experimental evidence. In the last part of the annotation, also computational systems-defined consensus or combiners can be used to generate more reliable gene structures. Examples of combiners are the programs GAZE or Jigsaw (59, 60). In practice, different sets of predictions are initially produced using a variety of gene finders either ab initio or evidence based. All inferred gene structures are then combined in a final set of nonredundant genes using algorithms based on dynamic programming (GAZE) or decision trees (Jigsaw). In case of Jigsaw, a

weight is assigned to each gene finder through a training procedure. However, prudence is always required in using these systems, since the inclusion of inconsistent data or gene models could lead to unsuccessful predictions.

2.4. Measuring Gene Finding Accuracy

Before the application of any prediction system or pipeline, it is good practice to test their accuracy on well-controlled benchmark data sets in order to choose the most appropriate system. To evaluate the accuracy of a gene prediction program, the gene structure predicted by the program is compared with the structure of the actual gene encoded in the sequence. The accuracy can be evaluated at different levels of resolution. Typically, these are the nucleotide, exon, and gene levels. These three levels offer complementary views of the accuracy of the program. At each level, there are two basic measures: sensitivity and specificity. Briefly, sensitivity (Sn) is the proportion of real elements (coding nucleotides, exons, or genes) that have been correctly predicted, while specificity (Sp) is the proportion of predicted elements that are correct. According to Burset and Guigò (61), Sn is defined as

$$Sn = \frac{TP}{TP + FN}$$

and Sp as

$$Sn = \frac{TP}{TP + FP}$$

where TP is the total number of coding elements correctly predicted, TN the number of correctly predicted noncoding elements, FP the number of noncoding elements predicted as coding, and FN the number of coding elements predicted as noncoding.

Both Sn and Sp take values from 0 to 1, and approach to 1 in a successful prediction. In cases where Sn is high but Sp is low, a gene over-prediction should be expected. On the other hand, if Sn is low while Sp is high, the prediction is overlay conservative and may miss a large number of genes.

However, neither Sn nor Sp alone constitutes good measures of global accuracy, since high sensitivity can be reached with low specificity and vice versa. For this reason, it is desirable in practice to use a single measure for accuracy. Generally in gene finding literature, it is defined as the average between Sn and Sp.

Recently, dedicated software has been developed to calculate main accuracy measures taking into account also alternative splicing. In this last case, all predicted gene models are compared to all annotated genes, and suitable Sn and Sp at transcript level are calculated.

The gene accuracy metrics are strongly related to the benchmark data set. Therefore, the choice of an appropriate reference set is a crucial step in the setup of a gene finding system. Current
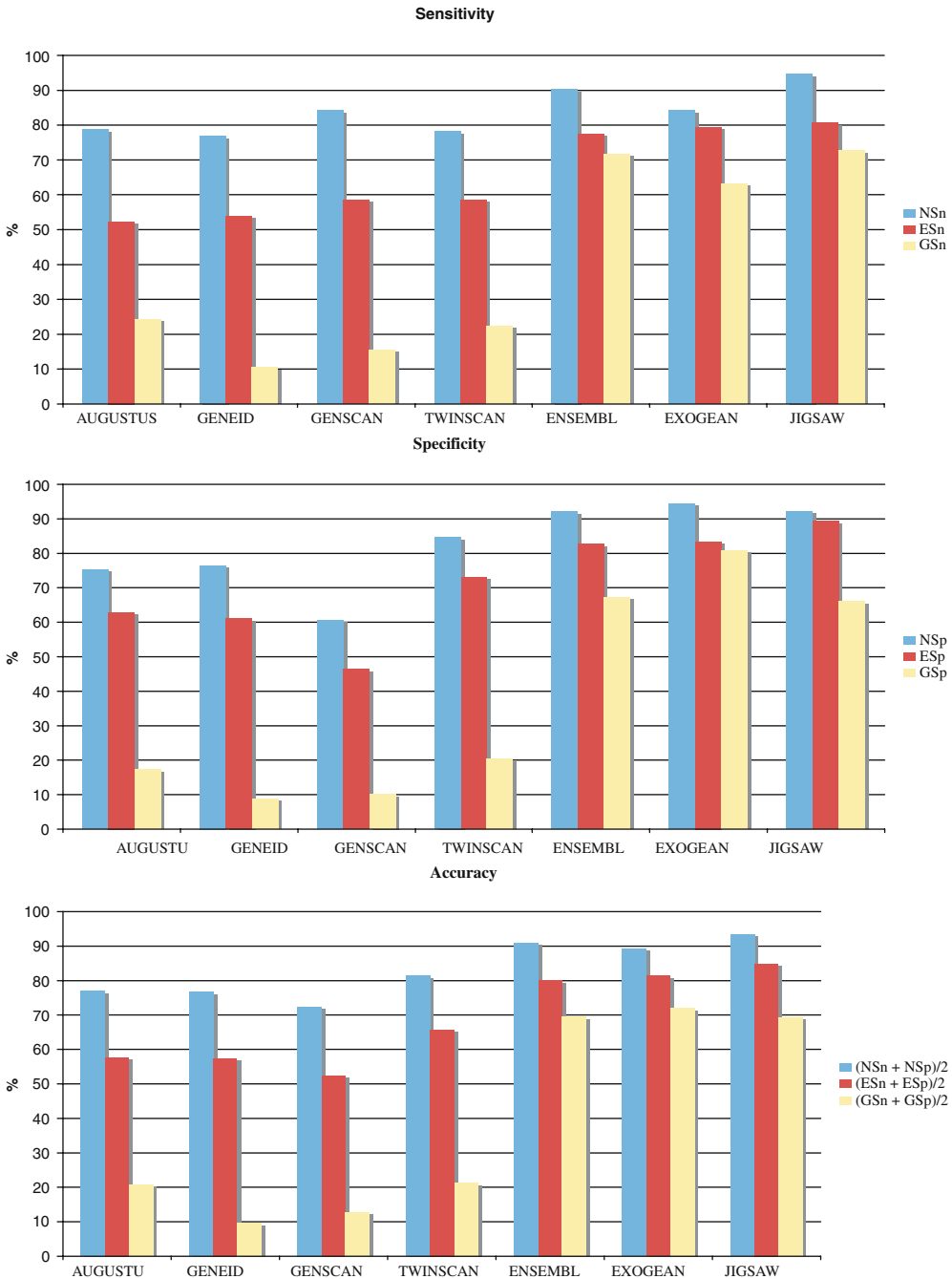
Fig. 16.1. Evaluation of gene predictions for gene finders falling into different categories. AUGUSTUS, GENEID, and GENSCAN work ab initio, TWINSCAN employs similarity from a related genome, ENSEMBL and EXOGEAN include expression data, and finally JIGSAW combines all available evidence. Sensitivity, specificity, and accuracy values are referred to human ENCODE regions according to the EGASP project (62). Abbreviations in the legends are as follow. NSn, ESn, and GSn stay for nucleotide, exon, and gene sensitivity. NSp, ESp, and GSp stay for nucleotide, exon, and gene specificity.

accuracy measures, in fact, have some limitation since predictions are generally tested on data sets made of short genomic sequences encoding a single gene with a simple gene structure. More reliable data sets, instead, should be a representative of the genome under study. The 44 genomic regions from ENCODE project spanning the 1% of the human genome are undoubtedly the best example of reference set (3). Since these regions have been manually annotated, they represent an optimal playground for assessing the accuracy of computational methods to predict eukaryotic genes. In this regard, the performance of different gene prediction tools has been evaluated through the EGASP project (62). Results from this assessment strongly indicate that programs using expression data outperform those ab initio or based on comparative genomics (**Fig. 16.1**). However, accurate results are also obtained combining pure ab initio gene predictions with available evidence (62) (*see* **Fig. 16.1**).

Acknowledgments

References

1. Wright, F. A., Lemon, W. J., Zhao, W. D., Sears, R., Zhuo, D., Wang, J. P., et al. (2001) A draft annotation and overview of the human genome. *Genome Biol* 2, RESEARCH0025.

2. McPherson, J. D., Marra, M., Hillier, L., Waterston, R. H., Chinwalla, A., Wallis, J., et al. (2001) A physical map of the human genome. *Nature* 409, 934–941.

3. ENCODE Project Consortium (2004) The ENCODE (ENCyclopedia Of DNA Elements) Project. *Science* 306, 636–640.

4. Gerstein, M. B., Bruce, C., Rozowsky, J. S., Zheng, D., Du, J., Korbel, J. O., et al. (2007) What is a gene, post-ENCODE? History and updated definition. *Genome Res* 17, 669–681.

5. Weinstock, G. M. (2007) ENCODE: more genomic empowerment. *Genome Res* 17, 667–668.

6. Korf, I. (2004) Gene finding in novel genomes. *BMC Bioinformatics* 5, 59.

7. Guigo, R., Agarwal, P., Abril, J. F., Burset, M., Fickett, J. W. (2000) An assessment of gene prediction accuracy in large DNA sequences. *Genome Res* 10, 1631–1642.

8. Arumugam, M., Wei, C., Brown, R. H., Brent, M. R. (2006) Pairagon+N-SCAN_-EST: a model-based gene annotation pipeline. *Genome Biol* 7 Suppl 1, S5 1–10.

9. Silke, J. (1997) The majority of long non-stop reading frames on the antisense strand can be explained by biased codon usage. *Gene* 194, 143–155.

10. Fickett, J. W. (1982) Recognition of protein coding regions in DNA sequences. *Nucleic Acids Res* 10, 5303–5318.

11. Staden, R. (1984) Measurements of the effects that coding for a protein has on a

DNA sequence and their use for finding genes. *Nucleic Acids Res* 12, 551–567.

12. Gerhard, D. S., Wagner, L., Feingold, E. A., Shenmen, C. M., Grouse, L. H., Schuler, G., et al. (2004) The status, quality, and expansion of the NIH full-length cDNA project: the Mammalian Gene Collection (MGC). *Genome Res* 14, 2121–2127.

13. Kotlar, D., Lavner, Y. (2003) Gene prediction by spectral rotation measure: a new method for identifying protein-coding regions. *Genome Res* 13, 1930–1937.

14. Lio, P. (2003) Wavelets in bioinformatics and computational biology: state of art and perspectives. *Bioinformatics* 19, 2–9.

15. Guo, F. B., Ou, H. Y., Zhang, C. T. (2003) ZCURVE: a new system for recognizing protein-coding genes in bacterial and archaeal genomes. *Nucleic Acids Res* 31, 1780–1789.

16. Burge, C., Karlin, S. (1997) Prediction of complete gene structures in human genomic DNA. *J Mol Biol* 268, 78–94.

17. Lukashin, A. V., Borodovsky, M. (1998) GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res* 26, 1107–1115.

18. Parra, G., Blanco, E., Guigo, R. (2000) GeneID in Drosophila. *Genome Res* 10, 511–515.

19. Majoros, W. H., Pertea, M., Salzberg, S. L. (2004) TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. *Bioinformatics* 20, 2878–2879.

20. Foissac, S., Bardou, P., Moisan, A., Cros, M. J., Schiex, T. (2003) EUGENE'HOM: A generic similarity-based gene finder using multiple homologous sequences. *Nucleic Acids Res* 31, 3742–3745.

21. Salzberg, S. L., Pertea, M., Delcher, A. L., Gardner, M. J., Tettelin, H. (1999) Interpolated Markov models for eukaryotic gene finding. *Genomics* 59, 24–31.

22. Guigo, R. (1998) Assembling genes from predicted exons in linear time with dynamic programming. *J Comput Biol* 5, 681–702.

23. Stormo, G. D. (2000) Gene-finding approaches for eukaryotes. *Genome Res* 10, 394–397.

24. Reese, M. G., Kulp, D., Tammana, H., Haussler, D. (2000) Genie–gene finding in Drosophila melanogaster. *Genome Res* 10, 529–538.

25. Krogh, A. (2000) Using database matches with for HMMGene for automated gene detection in Drosophila. *Genome Res* 10, 523–528.

26. Stanke, M., Waack, S. (2003) Gene prediction with a hidden Markov model and a new intron submodel. *Bioinformatics* 19 Suppl 2, ii215–225.

27. Smith, T. F., Waterman, M. S. (1981) Identification of common molecular subsequences. *J Mol Biol* 147, 195–197.

28. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., Lipman, D. J. (1990) Basic local alignment search tool. *J Mol Biol* 215, 403–410.

29. Kent, W. J. (2002) BLAT – the BLAST-like alignment tool. *Genome Res* 12, 656–664.

30. Pearson, W. R. (2000) Flexible sequence similarity searching with the FASTA3 program package. *Methods Mol Biol* 132, 185–219.

31. Karlin, S., Altschul, S. F. (1993) Applications and statistics for multiple high-scoring segments in molecular sequences. *Proc Natl Acad Sci USA* 90, 5873–5877.

32. Badger, J. H., Olsen, G. J. (1999) CRITICA: coding region identification tool invoking comparative analysis. *Mol Biol Evol* 16, 512–524.

33. Castrignano, T., De Meo, P. D., Grillo, G., Liuni, S., Mignone, F., Talamo, I. G., et al. (2006) GenoMiner: a tool for genome-wide search of coding and non-coding conserved sequence tags. *Bioinformatics* 22, 497–499.

34. Castrignano, T., Canali, A., Grillo, G., Liuni, S., Mignone, F., Pesole, G. (2004) CSTminer: a web tool for the identification of coding and noncoding conserved sequence tags through cross-species genome comparison. *Nucleic Acids Res* 32, W624–W627.

35. Florea, L., Hartzell, G., Zhang, Z., Rubin, G. M., Miller, W. (1998) A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Res* 8, 967–974.

36. Wheelan, S. J., Church, D. M., Ostell, J. M. (2001) Spidey: a tool for mRNA-to-genomic alignments. *Genome Res* 11, 1952–1957.

37. Usuka, J., Brendel, V. (2000) Gene structure prediction by spliced alignment of genomic DNA with protein sequences: increased accuracy by differential splice site scoring. *J Mol Biol* 297, 1075–1085.

38. Usuka, J., Zhu, W., Brendel, V. (2000) Optimal spliced alignment of homologous cDNA to a genomic DNA template. *Bioinformatics* 16, 203–211.

39. Wu, T. D., Watanabe, C. K. (2005) GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics* 21, 1859–1875.

40. Mott, R. (1997) EST_GENOME: a program to align spliced DNA sequences to unspliced genomic DNA. *Comput Appl Biosci* 13, 477–478.

41. Bonizzoni, P., Rizzi, R., Pesole, G. (2005) ASPIC: a novel method to predict the exon-intron structure of a gene that is optimally compatible to a set of transcript sequences. *BMC Bioinformatics* 6, 244.

42. Castrignano, T., Rizzi, R., Talamo, I. G., De Meo, P. D., Anselmo, A., Bonizzoni, P., et al. (2006) ASPIC: a web resource for alternative splicing prediction and transcript isoforms characterization. *Nucleic Acids Res* 34, W440–W443.

43. Djebali, S., Delaplace, F., Crollius, H. R. (2006) Exogean: a framework for annotating protein-coding genes in eukaryotic genomic DNA. *Genome Biol* 7 Suppl 1, S7 1–10.

44. Gelfand, M. S., Mironov, A. A., Pevzner, P. A. (1996) Gene recognition via spliced sequence alignment. *Proc Natl Acad Sci USA* 93, 9061–9066.

45. Birney, E., Clamp, M., Durbin, R. (2004) GeneWise and Genomewise. *Genome Res* 14, 988–995.

46. Meyer, I. M., Durbin, R. (2002) Comparative ab initio prediction of gene structures using pair HMMs. *Bioinformatics* 18, 1309–1318.

47. Pachter, L., Alexandersson, M., Cawley, S. (2002) Applications of generalized pair hidden Markov models to alignment and gene finding problems. *J Comput Biol* 9, 389–399.

48. Parra, G., Agarwal, P., Abril, J. F., Wiehe, T., Fickett, J. W., Guigo, R. (2003) Comparative gene prediction in human and mouse. *Genome Res* 13, 108–117.

49. Korf, I., Flicek, P., Duan, D., Brent, M. R. (2001) Integrating genomic homology into gene structure prediction. *Bioinformatics* 17 Suppl 1, S140–S148.

50. Yeh, R. F., Lim, L. P., Burge, C. B. (2001) Computational inference of homologous gene structures in the human genome. *Genome Res* 11, 803–816.

51. Gross, S. S., Brent, M. R. (2006) Using multiple alignments to improve gene prediction. *J Comput Biol* 13, 379–393.

52. Stanke, M., Tzvetkova, A., Morgenstern, B. (2006) AUGUSTUS at EGASP: using EST, protein and genomic alignments for improved gene prediction in the human genome. *Genome Biol* 7 Suppl 1, S11 11–18.

53. Carter, D., Durbin, R. (2006) Vertebrate gene finding from multiple-species alignments using a two-level strategy. *Genome Biol* 7 Suppl 1, S6 1–12.

54. Wei, C., Brent, M. R. (2006) Using ESTs to improve the accuracy of de novo gene prediction. *BMC Bioinformatics* 7, 327.

55. Curwen, V., Eyras, E., Andrews, T. D., Clarke, L., Mongin, E., Searle, S. M., et al. (2004) The Ensembl automatic gene annotation system. *Genome Res* 14, 942–950.

56. Slater, G. S., Birney, E. (2005) Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics* 6, 31.

57. Eyras, E., Caccamo, M., Curwen, V., Clamp, M. (2004) ESTGenes: alternative splicing from ESTs in Ensembl. *Genome Res* 14, 976–987.

58. Parra, G., Bradnam, K., Korf, I. (2007) CEGMA: a pipeline to accurately annotate core genes in eukaryotic genomes. *Bioinformatics* 23, 1061–1067.

59. Howe, K. L., Chothia, T., Durbin, R. (2002) GAZE: a generic framework for the integration of gene-prediction data by dynamic programming. *Genome Res* 12, 1418–1427.

60. Allen, J. E., Majoros, W. H., Pertea, M., Salzberg, S. L. (2006) JIGSAW, GeneZilla, and GlimmerHMM: puzzling out the features of human genes in the ENCODE regions. *Genome Biol* 7 Suppl 1, S9 1–13.

61. Burset, M., Guigo, R. (1996) Evaluation of gene structure prediction programs. *Genomics* 34, 353–367.

62. Guigo, R., Flicek, P., Abril, J. F., Reymond, A., Lagarde, J., Denoeud, F., et al. (2006) EGASP: the human ENCODE Genome Annotation Assessment Project. *Genome Biol* 7 Suppl 1, S2 1–31.

Chapter 17

Sequence and Structure Analysis of Noncoding RNAs

Stefan Washietl

Abstract

Noncoding RNAs (ncRNAs) are increasingly recognized as important functional molecules in the cell. Here we give a short overview of fundamental computational techniques to analyze ncRNAs that can help us better understand their function. Topics covered include prediction of secondary structure from the primary sequence, prediction of consensus structures for homologous sequences, search for homologous sequences in databases using sequence and structure comparisons, annotation of tRNAs, rRNAs, snoR-NAs, and microRNAs, de novo prediction of novel ncRNAs, and prediction of RNA/RNA interactions including miRNA target prediction.

Key words: noncoding RNAs, RNA secondary-structure prediction, homology search, gene prediction, snoRNAs, microRNAs, tRNAs, rRNAs.

1. Introduction

In the past few years, it has become evident that noncoding RNAs are much more abundant than previously thought. It is now widely acknowledged that ncRNAs are key players in the cell with important biological functions (1).

ncRNAs are a surprisingly inhomogeneous class of molecules. They can vary considerably in size, ranging from very short 22nt long micro RNAs (miRNAs) to polyadenlyated mRNA-like ncRNAs that can be many kilobases long. They also have diverse molecular functions. They can target other molecules by sequence-specific RNA/RNA interactions like small nucleolar RNAs (snoR-NAs) or miRNAs, they can be important structural components of large protein complexes like the RNA component of the signal recognition particle, or they can even have enzymatic function themselves as, for example, RNAseP or the spliceosomal RNAs.
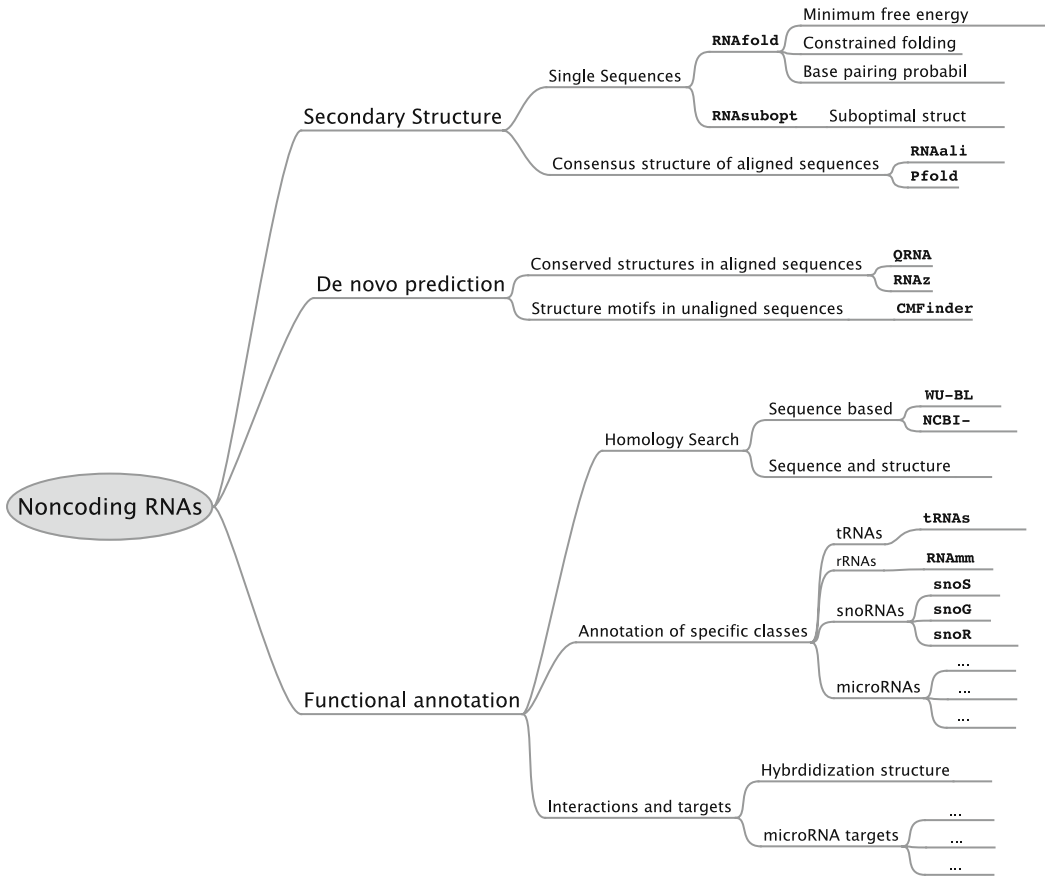
Fig. 17.1. Overview of the concepts and programs covered in this chapter.

In this chapter, we present a selection of important computational methods to analyze ncRNAs. **Figure 17.1** shows an overview of the concepts and programs used.

Secondary structure is a central key to understand ncRNA function. In the first part of this chapter (**Section** 3.2), we address the problem of predicting accurate secondary structure models from the primary sequence.

The prediction of novel ncRNA genes in genomic data is still a difficult problem. However, there has been recent progress and we present ncRNA gene-finding strategies in the second part of this chapter (**Section** 3.3).

The rest of the chapter presents techniques to classify and/or functionally annotate ncRNAs. These techniques can be used to annotate full genomes or analyze experimentally identified and/or computationally predicted ncRNAs. Topics covered are generic sequence/structure homology search (**Section** 3.4), specific classification algorithms for well-known ncRNA classes (**Section** 3.5), and prediction of possible target interactions (**Section** 3).

The goal of this chapter is to give an overview of the most important techniques and demonstrate the use of current state-of-the art programs. Due to space restrictions, we can only show the most basic usage of these programs and have had to leave out some programs altogether. Whenever possible, we try to give pointers to advanced analysis techniques and sources of information to enable the interested reader to delve more deeply into the subject.

1.1. Typographical Conventions

Lines starting with a "#" are commands and you should type them into your terminal window, followed by pressing return. The "#" sign stands for your command line prompt and may look different on your system. If a command is too long for one line in this book, it is separated by a backslash "\" and continues on the next line. You do not have to input the backslash, you can simply type the command on one line.

2. Materials

2.1. Hardware

All examples shown in this chapter can be run on a modern desktop or laptop computer in reasonable time. However, in a real-life scenario it is likely that more complex data sets are analyzed which would require more processing power, e.g., in the form of a computing cluster.

2.2. Software

We recommend using the Linux operating system. Alternatively, it is possible to use Mac OS X or any other UNIX like system. Some programs might also run under Microsoft Windows. However, installation and usage under Windows is generally more complicated and not covered here. Most of the analysis can also be carried out using public Web-servers. In this case, only a Web-browser is necessary.

Table 17.1 shows the software necessary to complete the examples in this chapter. All programs are freely available on the Web. The version numbers are the latest as of February 2008. Most of the programs can either be run locally or through a Web-server. Depending on the demands and knowledge of the user, either way might be preferable. Only QRNA and INFERNAL strictly require installation as there is no Web-server available. Pfold, on the other hand, is only available through the Web-server.

2.3. Example Files

Example files used in this chapter can be downloaded here: www.tbi.univie.ac.at/papers/SUPPLEMENTS/MiMB/.

2.4. Additional Information

On the site www.tbi.univie.ac.at/papers/SUPPLEMENTS/MiMB/, you can also find all links to software downloads, Web-servers, and additional documentation.

Table 17.1
Software used in this chapter

Software	Version	Download	Web-Server	Web-Site	Ref.
Vienna RNA	1.7	+	+	www.tbi.univie.ac.at/~ivo/RNA/	(2)
Pfold		−	+	www.daimi.au.dk/~compbio/ rnafold/	(3)
RNAz	1.1	+	+	www.tbi.univie.ac.at/~wash/RNAz	(4)
QRNA	2.0.3c	+	−	ftp://selab.janelia.org/pub/software/ qrna	(5)
CMFinder	0.2	+	+	bio.cs.washington.edu/yzizhen/ CMfinder/	(6)
INFERNAL	0.81	+	−	infernal.janelia.org	(7)

3. Methods

3.1. Software Installation

All software packages are distributed as TAR/GZIP compressed archives. Download the files ending in tar.gz or tgz to your machine and uncompress it, e.g.,

```
# tar -xzf ViennaRNA-1.7.tar.gz
```

Vienna RNA, RNAz, and INFERNAL use the standard GNU installation system. So you can easily install the software packages by running the following commands:

```
# ./configure
# make
# su
# make install
```

This requires root privileges and installs all files under the /usr/ local tree. The executables, for example, are installed to /usr/local/ bin and should be ready to run. If you do not have root privileges, if you want to install the programs in a different location, or if you experience other problems (e.g., gcc compiler not found) see **Note 1**. Repeat this process for RNAz and INFERNAL.

To install QRNA, run the following commands:

```
# tar -xzf qrna-2.0.3c.tar.gz
# cd qrna-2.0.3c
# cd squid
# make
# cd squid02
```

```
# make
# cd ..
# cd src
# make
```

This creates the executable file eqrna. You can make it accessible from everywhere from your system, for example, by creating a link to the executable in /usr/local/bin (requiring root privileges and assuming that you have extracted the folder to ~/ programs):

```
# su
# ln -s ~/programs/qrna-2.0.3c/src/eqrna \
/usr/local/bin/eqrna
```

In addition, you have to set the environment variable QRNADB pointing to your installation directory. If you are using a C shell type,

```
# setenv QRNADB ~/programs/qrna-2.0.3c/lib
```

if you are using BASH run,

```
# export QRNADB=~/programs/qrna-2.0.3c/lib
```

The other programs will be used through their Web-interfaces and we do not show their installation here.

3.2. Prediction of RNA Secondary Structure

Prediction of the secondary structure from the primary sequence is probably the most central problem in RNA analysis. Although the function of an RNA molecule is ultimately dependent on its tertiary structure, secondary structure can be seen as a coarse-grained approximation and it is a useful level on which to understand RNA function. Depending on the data available, different strategies can be used to obtain the best secondary structure models.

3.2.1. Structure Prediction for Single Sequences

The most widely used and generally most accurate way of predicting a secondary structure for a single sequence is thermodynamic folding algorithms, as implemented, for example, in the Vienna RNA package. The program RNAfold uses experimentally derived energy parameters and efficient algorithms to calculate the energetically optimal structure, or more precisely, the structure with minimum free energy (MFE). The sequence is given in a FASTA-like format (*see* **Fig. 17.2A**). As an illustrative example, we use a tRNA sequence that folds into the well-known clover-leaf structure. RNAfold reads the file from the standard input and writes the results to standard output:

A
```
>tRNA AF041468
GGGGGTATAGCTCAGTTGGTAGAGCGCTGCCTTTGCACGGCAGATGTCAGGGGTTCGAGTCCCCTTACCTCCA
```

B
```
>tRNA2 X54300
GGGGGUAUAGCUUAGUUGGUAGAGCGCUGCUUUUGCAAGGCAGAUGUCAGCGGUUCGAAUCCGCUUACCUCCA
..........................(((((xxxxx))))...............................
```

C
```
CLUSTAL W(1.81) multiple sequence alignment


AF041468        GGGGGTATAGCTCAGT-TGGTAGAGCGCTGCCTTTGCACGGCAGATGTCAGGGGTTCGAG
X54300          GGGGGTATAGCTTAGT-TGGTAGAGCGCTGCTTTTGCAAGGCAGATGTCAGCGGTTCGAA
L00194          GGGGCCATAGCTCAGT-TGGTAGAGCGCCTGCTTTGCAAG-CAGGTGTCGTCGGTTCGAA
AY017179        GGGCCGGTAGCTCAGCCTGGGAGAGCGTCGGCTTTGCAAGCCGAAGGCCCCGGGTTCGAA


AF041468        TCCCCTTACCTCCA
X54300          TCCGCTTACCTCCA
L00194          TCCGTCTGGCTCCA
AY017179        TCCCGGCCGGTCCA
```

D
```
>AF041468
GGGGGTATAGCTCAGT-TGGTAGAGCGCTGCCTTTGCACGGCAGATGTCAGGGGTTCGAGTCCCCTTACCTCCA
>X54300
GGGGGTATAGCTTAGT-TGGTAGAGCGCTGCTTTTGCAAGGCAGATGTCAGCGGTTCGAATCCGCTTACCTCCA
>L00194
GGGGCCATAGCTCAGT-TGGTAGAGCGCCTGCTTTGCAAG-CAGGTGTCGTCGGTTCGAATCCGTCTGGCTCCA
>AY017179
GGGCCGGTAGCTCAGCCTGGGAGAGCGTCGGCTTTGCAAGCCGAAGGCCCCGGGTTCGAATCCCGGCCGGTCCA
```

E
```
# STOCKHOLM 1.0

AF041468        GGGGGTATAGCTCAGT-TGGTAGAGCGCTGCCTTTGCACGGCAGATGTCAGGGGTTCGAG
X54300          GGGGGTATAGCTTAGT-TGGTAGAGCGCTGCTTTTGCAAGGCAGATGTCAGCGGTTCGAA
L00194          GGGGCCATAGCTCAGT-TGGTAGAGCGCCTGCTTTGCAAG-CAGGTGTCGTCGGTTCGAA
AY017179        GGGCCGGTAGCTCAGCCTGGGAGAGCGTCGGCTTTGCAAGCCGAAGGCCCCGGGTTCGAA
#=GC SS_cons    <<<<<<<..<<<<.........>>>>.<<<<<<.....>>>>>>.....<<<<<......

AF041468        TCCCCTTACCTCCA
X54300          TCCGCTTACCTCCA
L00194          TCCGTCTGGCTCCA
AY017179        TCCCGGCCGGTCCA
#=GC SS_cons    .>>>>>>>>>>>.
```

Fig. 17.2. File formats used in this chapter. (**A**) Input for RNAfold. The Vienna RNA package generally uses a FASTA-like format including a header starting with ">" and the sequence. The difference to standard FASTA is that the header is optional and that the sequence *must not* contain line breaks, i.e., it must be on one line. (**B**) Input for constrained folding (RNAfold -C). The second line shows the constraints on the structure like requiring specific base-pairs with "(" and ")" or unpaired regions with "x". (**C**) CLUSTAL W alignment format used for RNAalifold and RNAz. (**D**) Multiple alignment format in FASTA format. Sequences contain gaps and are all of the same length. Input format for Pfold and (if pairwise) for QRNA. (**E**) STOCKHOLM format used by INFERNAL. It is similar to CLUSTAL W. Note the annotated secondary structure in a dot/bracket notation with angular brackets.

```
# RNAfold < tRNA.fa
>tRNA AF041468
GGGGGUAUAGCUCAGUUGGUAGAGCGCUGCCUUUGCACGGCAGA
((((((((..((((........))))).(((((.......))))).
UGUCAGGGGUUCGAGUCCCCUUACCUCCA
....(((((.......))))))))))))).  (-31.10)
```

The predicted structures are given below the sequences in a "dot/bracket" notation. Each base-pair in the secondary structure is indicated by a pair of brackets, unpaired bases are shown as dots. Next to the structure you see the MFE of -31.10 in kcal/mol. RNAfold also creates a graphical representation of the structure prediction in Postscript format. The file is automatically named according to the sequence name, in our case tRNA_ss.ps. Under Linux, it can be viewed using ghostview

```
# gv tRNA_ss.ps
```

Under OS X you can run

```
# open tRNA_ss.ps
```

which automatically converts the postscript file to PDF and displays it.

One has to keep in mind that the RNA structure calculated by RNAfold is only a prediction. The accuracy heavily depends on the type of RNA studied. One can expect roughly 70% of the predicted bases to be correct, but in unfortunate cases the accuracy can be far below this. The accuracy also depends on the length of the sequence. Structures for short RNAs are predicted more reliably than for long RNAs.

Under natural conditions, an RNA molecule usually does not only fold into a single structure but forms a *thermodynamic ensemble* of structures including suboptimal structures. Using RNAsubopt it is possible to predict all suboptimal structures that are within a certain energy range of the MFE.

```
# RNAsubopt -e 1 -s < tRNA2.fa
```

This command calculates all structures that are at most 1 kcal/mol above the MFE and sorts them by their energy. In this example of another tRNA, the structure of minimum free energy at -27.90 kcal/mol is not the correct one. The correct fold has a free energy of -27.30 and appears as 11th suboptimal structure in the list.

You will notice that for long sequences there is a huge number of suboptimal structures. To get an overview over all structures in the ensemble, it is possible to calculate the pairing probabilities of all possible base pairs in a sequence:

```
# RNAfold -p < 5S.fa
```
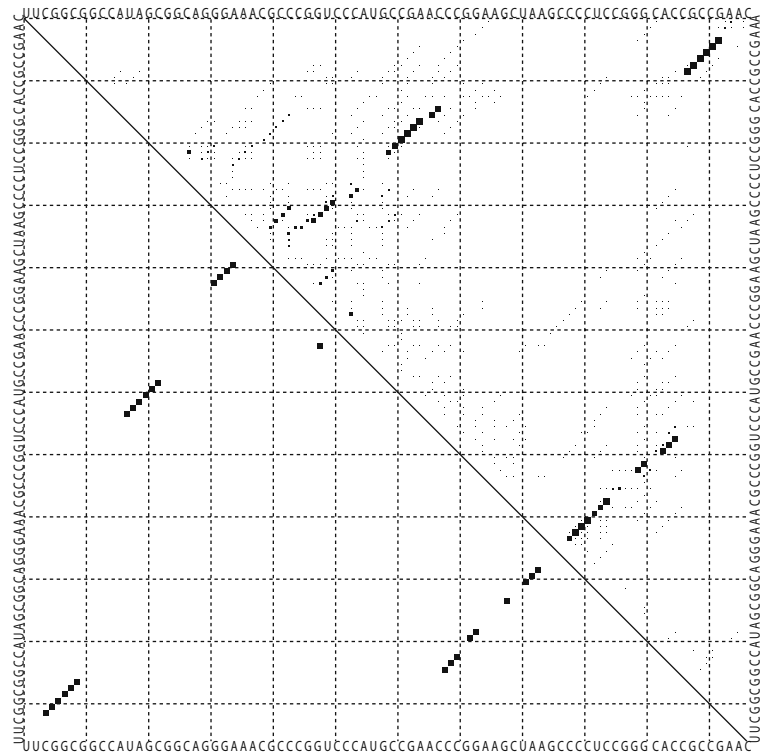
Fig. 17.3. Base-pairing probabilities of a 5S rRNA visualized by a "dot-plot." Each dot in the upper right triangle of the matrix corresponds to a base-pair. The size of a dot is proportional to the probability of the base-pair in the thermodynamical ensemble. The base-pairs in the minimum free energy structure are shown in the lower left triangle.

This command calculates the base-pairing probabilities of a 5S rRNA. The file 5S_dp.ps is created that visualizes the probabilities in a "dot-plot" (**Fig. 17.3**). In this example, the base-pairs of the MFE structure are the most likely, but you also see alternative base-pairs with lower probability that might be of interest although they are not part of the MFE.

If one has some information on the true secondary structure, e.g., from experimental data, one can improve the model by incorporating this information in the prediction. If we call RNA-fold with the option −C ("constrained folding"), it expects another line in the input file after the sequence that holds constraints on the structure (**Fig. 17.2B**). Here we require four base-pairs in the anticodon stem to be formed (indicated by "(" and ")"). On the other hand, we require the anticodon to be unpaired (indicated by "x").

```
# RNAfold -C < tRNA2_constraints.fa
```

The constraints result in a correctly predicted clover-leaf shape, while the unconstrained folding gives a completely different structure.

3.2.2. Consensus Structure Prediction for Homologous Sequences

The quality of a secondary structure model can be improved considerably if additional information from homologous sequences is incorporated. RNAalifold predicts a consensus secondary structure for aligned sequences. It extends the thermodynamic folding algorithm of RNAfold and incorporates covariation information (i.e., consistent and compensatory mutations). RNAalifold takes a CLUSTAL W formatted alignment (**Fig. 17.2C**) as input, otherwise the usage is similar to RNAfold:

```
# RNAalifold < tRNA.aln
4 sequences; length of alignment 74.
GGGGCUAUAGCUCAGU_UGGUAGAGCGCCGCCUUUGCAAGGCAGA
(((((((..(((.........)))).(((((.....))))))).
UGUCAGCGGUUCGAAUCCCCUUACCUCCA
...(((((.......))))))))))))).
minimum free energy = -30.98 kcal/mol (-27.72 + -3.25)
```

The command returns the consensus structure in dot/bracket notation and a "consensus MFE" that consists of an "energy term" -27.72 and a "co-variation term" that is -3.25 in our example. The energy term is essentially the average folding energy of the single sequences if forced to fold into a consensus structure. The covariation term is (roughly speaking) negative if there are many consistent and/or compensatory mutations supporting a structure and positive if there are many mutations inconsistent with the consensus structure.

RNAalifold generates the file alirna.ps that contains the secondary structure plot with covariations highlighted. Similar to RNAfold, RNAalifold can also be run with option −p, that produces a dot-plot of pair probabilities. More information on how to use RNAalifold can be found in another book of this series (**8**).

An alternative to RNAalifold is Pfold. It uses a probabilistic folding algorithm based on stochastic context free grammars and incorporates a model of the phylogenetic relationship between the sequences. Although Pfold gives good results, one limitation is that it is only available through a Web-server. Go to www.daimi.au.dk/~compbio/rnafold/ and paste the content of the file tRNA_aln.fa into the form. Pfold takes a FASTA formatted alignment as input (*see* **Fig. 17.2C**). Enter also your E-mail address and press "Fold RNA." Within a few minutes, a link to the results should be sent to your E-mail address. On the result page, you get secondary structure predictions in dot/bracket format together with an estimation on how reliable the predicted bases are. The results include a dot-plot, secondary structure plot, and the phylogenetic tree that was used to predict the structure.

3.3. De Novo Prediction of Structural ncRNAs

Unfortunately there is no general de novo "gene-finder" for ncRNAs as we know it for protein genes. No single algorithm can detect all of the diverse classes of molecules that we loosely refer to as "ncRNAs." However, one common characteristic of a large subset of ncRNAs is their secondary structure. Therefore, it is a promising approach to predict potentially functional secondary structures in genomic data. These structures can give hints to the presence of ncRNAs.

3.3.1. Single Sequences

RNAfold can be used to predict a structure for a given sequence. However, it predicts a structure for any sequence, be it an ncRNA, some other biological sequence, or even just a random sequence. To assess the significance of a folded structure one commonly calculates "stability z-scores" (9, 10). One compares the folding energy of the given sequence to the expected folding energies of random sequences of the same dinucleotide content and length. Let m be the MFE of the native sequence and μ and σ the mean and standard deviation of random sequences. The z-score is given by $z = (m - \mu)/\sigma$. Negative z-scores indicate that the given sequence is more stable than one would expect from chance. This approach is implemented in the latest version of RNAz that calculates a z-score for a sequence in FASTA format:

```
# RNAz --single tRNA.fa
```

Unfortunately the z-score is generally not significant enough to be used as basis for a gene finder. It is also difficult to give clear advise on how to interpret the value of a z-score for a single sequence. However, a z-score is preferable to using the absolute folding energy and it can give a first hint as to whether a fold might be functional or not.

3.3.2. Aligned Sequences

A much more accurate way to predict functional RNA structures is a combination of the z-score approach with evolutionary signals. The preferable input for RNAz is a multiple sequence alignment of 2–6 sequences. It estimates z-scores for the single sequences and, in addition, calculates a so-called structure conservation index (SCI). The SCI indicates whether the sequences share a common structure or not. The value lies usually between 0 and 1 and is calculated using the RNAfold/RNAalifold algorithms. RNAz then combines the z-score and the SCI to classify an alignment as "functional RNA" or "other." Sequences with low z-score (i.e., stable structures) and high SCI (i.e., high structural conservation) are predicted as potential ncRNAs. The input alignment needs to be in Clustal W (**Fig. 17.2C**) format or MAF alignment format. We can simply run

```
# RNAz tRNA.aln
```

Here are the relevant lines of the output:

```
Mean z-score: -2.68
Structure conservation index: 0.94
SVM RNA-class probability: 0.999672
Prediction: RNA
```

The *z*-score and SCI are calculated. From these values a "RNA class probability" is calculated. If this value is above 0.5 the prediction is "RNA" otherwise it is "other."

An alternative way to detect evolutionarily conserved RNA secondary structures is QRNA. It analyzes pairwise alignments and scores the data using probabilistic models. There are three alternative models: one model for structural RNAs, one for protein coding regions, and one for other genomic regions. The alignment is classified according to the model which is found to explain the data best. QRNA takes FASTA-formatted input alignments (**Fig. 17.2D**). tRNA_pair.fa contains the first two sequences of the previously used multiple alignment tRNA.aln. QRNA is run as follows:

```
# eqrna tRNA_pair.fa
```

The relevant lines of the output:

```
winner = RNA
sigmoidalOTH = -5.327 sigmoidalCOD = -9.302
sigmoidalRNA = 5.233
```

For each model, a so-called sigmoid score is shown, which is calculated from the score of this model and the two other models as null model. The model with the highest sigmoid score is the "winner." In our case, the RNA model scores best and the alignment is classified as RNA.

3.3.3. Unaligned Sequences

Both RNAz and QRNA require a sequence alignment as input. However, if the sequences are too diverged it can be difficult to get a reasonable alignment. Still the sequences can have common secondary structure motifs. CMFinder finds structured motifs in unaligned sequences. An advantage of CMFinder is that it can deal with long extraneous flanking regions, and cases when the motif is only present in a subset of sequences. CMFinder is based on an expectation-maximization algorithm using covariance models for motif description, heuristics for motif search, and a Bayesian framework for structure prediction combining folding energy and sequence covariation.

The Web-interface takes a FASTA-formatted sequence file. An important parameter is the "number of stem loops." It determines the complexity of the motifs to be found. Usually it is set to 1 or 2 for single stem-loop or double stem-loop motifs. CMFinder has many other parameters, which we cannot explain in detail here.

Please refer to the online documentation. In the simplest case use the default parameters, upload your file in FASTA format, input your E-mail address, and click "Submit." You can try it with the example file glms.fa that contains a bacterial ribozyme.

The results consist of a series of putative structure motifs. Depending on the input parameters, they consist of one or two stems or more complex motifs. The file ending in *. summary contains a table with all motifs, showing some interesting characteristics like folding energy, sequence identity, GC content, etc. For each motif, two files are generated: one is an alignment in Stockholm format (**Fig. 17.2E**) that contains the structurally aligned motifs, the other is a covariation model suitable for use with INFERNAL (*see* **Section** 3.4.2).

3.4. Searching Databases for Homologous Sequences

Homology search is currently the most promising way to get some clues on the function of an unknown ncRNA or to predict new ncRNAs of known families in genomic data. Searching databases for homologous sequences is a fundamental technique in bioinformatics and much has been written about it. However, rules to perform successful homology searches for structured ncRNAs can be quite different.

3.4.1. Sequence-Based Algorithms

The most commonly used program for homology searches of databases is the BLAST algorithm in its two incarnations WU-BLAST and NCBI-BLAST. Default parameters usually do not give the most accurate results for ncRNAs.

To get optimal sensitivity, the word size should be set lower than the default. A word size of 7 was shown to give good results (11) on a test set of various structured RNAs. This is the minimum word size for NCBI-BLAST. For WU-BLAST, it is possible to lower the word size to 3. This slightly increases the sensitivity but comes with a considerable increase in computing time.

The default scoring matrix for NCBI-BLAST is optimized for highly similar sequences (99% sequence identity). WU-BLAST defaults are optimized for more distant homologies (65–100%) and perform much better on structural ncRNAs. Therefore, NCBI should be used with parameters "−r 5 −q −4" to adjust match and mismatch scores.

The following command lines show useful parameters that gave good results on the benchmark in (11):

NCBI-BLAST:

```
# blastall -p blastn -d database.fa \
-i query.fa -W 7 -r 5 -q -4 -G 10 -E 10
```

WU-BLAST:

```
# blastn database.fa query.fa W=7
```

Depending on the problem at hand, these parameters can be a good starting point. NCBI-BLAST and WU-BLAST programs are freely available (http://blast.wustl.edu/, ftp://ftp.ncbi.nih.gov/blast/) together with extensive documentation on how to use them. Using the same parameters, both NCBI- and WU-BLAST usually give very similar results. WU-BLAST is, however, faster than NCBI-BLAST. Many sequence databases and genome projects have a BLAST interface for homology search. Also when using these services it is advisable to adjust the parameters if possible.

3.4.2. Sequence- and Structure-Based Algorithms

A combination of sequence and structure information can lead to much more sensitive homology searches. The tools of the INFERNAL package allow to build consensus RNA secondary structure profiles based on stochastic context-free grammars (profile SCFGs). Profile SCFGs include both sequence and RNA consensus secondary structure information.

To use INFERNAL, you need an alignment of RNA sequences and a consensus structure given in Stockholm format. The Stockholm format is similar to Clustal W (*see* **Fig. 17.2E**). In **Section 3.2.2**, it is explained how to obtain a consensus structure.

The file tRNA.sto contains our tRNA alignment example and a RNAalifold consensus structure. The following command builds the covariance model tRNA.cm from this alignment:

```
# cmbuild tRNA.cm tRNA.sto
```

The file database.fa contains a tRNA embedded in 3 kb of some viral DNA. The covariance model can be used to search this "database" for homologous sequences/structures:

```
# cmsearch tRNA.cm database.fa
```

This command calculates optimal local alignments of the query profile against the sequences in the database. The output is similar to BLAST output, indicating strand and location of the hit in the query and target. The secondary structure is shown in the first line in an extended notation. For example, < and > denote base pairs in simple stems, while "("and")" are base pairs closing a multifurcating loop as the acceptor stem in our tRNA. Refer to the excellent documentation of INFERNAL for more details on the output format.

```
Plus strand results:

Query = 1 - 73, Target = 1751 - 1824
 Score = 75.44, GC = 70


     (((((((,,<<<<___.____>>>>,
   1 GGggccguAGCucAGu.uGGuAgaGCG
     GGG:C::GUAGCUCAG UGG AGAGCG
1751 GGGCCGGUAGCUCAGCcUGGGAGAGCG
```

```
<<<<<<_____>>>>>>,,,,,<<<<<
ccGccuuuGcAaggCggAuGucgggGG
:CG:CUUUGCAAG:CG:A G C::GGG
UCGGCUUUGCAAGCCGAAGGCCCCGGG

_____>>>>>)))))))):
uucGAAuCCccccggcuCCA 73
UUCGAAUCCC::C:G:UCCA
UUCGAAUCCCGGCCGGUCCA 1824
```

The quality of a hit is measured by a bit score (in this example, 75.44). As a rule-of-thumb, bit scores higher than the log (base 2) of the database size are significant. The latest version of INFER-NAL also calculates *E*-values. You will have noticed that INFER-NAL is relatively slow and the calculation of *E*-values makes it even slower. The search without *E*-values took roughly 5 s on an Intel Centrino Core duo, while it took 2 min and 15 s including *E*-values.

Speed is a critical issue when using INFERNAL. It is impractic-able to search large databases or genomes on a single machine. The latest INFERNAL version provides heuristic speed-up strategies as well as ways to distribute the calculation on many CPUs. These advanced features are beyond the scope of this chapter and we have to refer the reader again to the INFERNAL documentation.

In our example, we have used cmsearch to match one model against a database of sequences. It is of course also possible to search one sequence against a database of many covariance models. The Rfam database provides covariance models of hundreds of structural RNA families (12). They can be downloaded here: ftp://ftp.sanger.ac.uk/pub/databases/Rfam/CURRENT/Rfam.tar.gz. Assuming that the Rfam models reside in a subdirectory rfam and a BASH shell is available, the following short loop searches the sequence tRNA.fa against all Rfam families and writes the output to rfam.out.

```
# for cm in rfam/*.cm;\
# do echo $cm >> rfam.out;\
# cmsearch $cm tRNA.fa >> rfam.out;\
# done
```

The Rfam database does not only provide covariance models. It is also possible download the raw sequences or multiple align-ments in various formats. Together with extensive and well-pre-sented annotation, it is a central resource for noncoding RNA analysis.

3.5. Annotation of Specific RNA Classes

If homology-based identification fails, functional annotation of ncRNAs is difficult. However, there are a few well-known classes of ncRNAs that can be identified on the basis of very general

Table 17.2
Programs to annotate specific RNA classes

Software	RNA class	DL	WS	Web site	Ref.
tRNAscan-SE	tRNA	+	+	lowelab.ucsc.edu/tRNAscan-SE	(13)
RNAmmer	rRNA	+	+	www.cbs.dtu.dk/services/RNAmmer/	(14)
snoScan	C/D box snoRNA	+	+	lowelab.ucsc.edu/snoscan	(15)
snoGPS	H/ACA box snoRNA	+	+	lowelab.ucsc.edu/snoGPS	(16)
snoReport	C/D and H/ACA snoRNA	+	−	www.tbi.univie.ac.at/~jana/software.html	(17)
BRUCE	tmRNA	+	+	130.235.46.10/ARAGORN1.1/HTML/ bruceindex.html	(18)
SRPscan	SRP RNA	−	+	bio.lundberg.gu.se/srpscan	(19)

DL, software available for download; WS, software available as Web-server.

structure and sequence characteristics. In the following, we present some programs used to classify and annotate important ncRNA classes (**Table 17.2**).

3.5.1. tRNAs

tRNAs can be annotated very reliably using tRNAscan-SE. The program is available for local use and as Web-server. It takes FASTA (and other sequence formats) as input. It returns an overall score and detailed annotation of each hit including secondary structure model and anticodon of the predicted tRNA. Specific models for eukaryotes, archaea, and bacteria are available. tRNAscan-SE is also capable of distinguishing true tRNAs from tRNA-derived pseudogenes. You can test with the file tRNA.fa.

3.5.2. rRNAs

The program RNAmmer annotates rRNAs (16s/18s, 23s/28s, and 5s but not 5.8 rRNAs). It uses profile HMMs and is mainly based on sequence homology. However, since it is specifically designed for rRNAs it provides more consistent and reliable annotations than generic homology search approaches. RNAmmer is available as Web-server and for download. It takes FASTA-formatted input sequence and you can test it with the file 5S.fa.

3.5.3. snoRNAs

There are two major types of snoRNAs: C/D box and H/ACA snoRNAs. The former mediate methylation modifications in rRNAs, while the latter guide pseudouridylation modifications. snoScan (15) and snoGPS (16) predict C/D box and H/ACA box snoRNAs, respectively. Both programs are available for local

use and as Web-server. snoScan and snoGPS use probabilistic models that also include the target sites in their prediction. This means that both programs need a list of potential target RNA sequences as input.

When the targets are unknown (e.g., so-called orphan snoR-NAs), the program snoReport (17) is more accurate. It uses a support vector machine approach that is based on structure and sequence features. snoReport is available for download and can be run under UNIX like systems.

3.5.4. miRNAs

Annotation of miRNAs is currently of particular interest. The basic idea is to predict the typical stem-loops of the pre-miRNA. It is challenging, however, to distinguish true miRNA stem-loops from "background" stem-loops that are abundant in all genomes. A plethora of methods have been proposed for this classification task. It is neither possible to present all of them in this chapter, nor would it make sense to pick out one or two specific methods since no "standard" approach currently exists. **Table 17.3** lists currently available programs to identify miRNAs. There are even more, but we have limited the list to methods which are either available for download or as a Web-server.

3.5.5. Other Classes

There are some other ncRNA classes for which specific software was developed to allow more reliable detection. Here we just want to mention SRPscan (19) for annotating signal recognition particle (SRP) RNAs and BRUCE (18) for detecting transfer-messenger RNAs (tmRNAs). Both programs are available as Web-server and BRUCE can also be downloaded for local use.

3.6. Interaction Partners

Many ncRNAs function through RNA/RNA interactions. The sequence-specific binding of miRNAs to UTRs of mRNAs is probably the best-known example. However, also in bacteria many small ncRNAs are known to interact with their target by hybridization (36).

3.6.1. Hybridization Structure of Two RNAs

RNAduplex can be used to find potential binding sites of a short RNA in a longer target RNA. It takes a FASTA formatted file (Vienna style without line breaks) containing two sequences. The first entry is the (long) target RNA, while the second is the (short) RNA for which a binding site should be found. The file "interaction.fa" contains the 3′-UTR of a mRNA and the microRNA mir-145. RNAduplex is run as follows:

```
# RNAduplex < interaction.fa
>NM_024615
 >hsa-miR-145
 .(((((.(((...((((((((((.&)))))))))))))))))).
34,57 : 1,19 (-21.80)
```

Table 17.3
Programs for microRNA prediction

Software	Description	DL	WS	Web site	Ref.
miRscan	Scores pairs of homologous hairpins; applied to nematodes and vertebrates; uses empirical rules	−	+	genes.mit.edu/mirscan	(20)
miRFinder[1]	Scores pairs of homologous hairpins; uses SVM; applied to drosophilids and vertebrates	+	−	www.bioinformatics.org/mirfinder	(21)
miPred[2]	SVM classification of hairpins; tested on animal, plant, and viral miRNAs.	+	−	web.bii.a-star.edu.sg/~stanley/Publications/Supp_ materials/06-002-supp.html	(22)
miPred[2]	Hairpin classification in human using machine-learning techniques	−	+	www.bioinf.seu.edu.cn/miRNA	(23)
triplet-SVM	SVM classification of hairpins; tested on animal, plant, and viral miRNAs	+	−	bioinfo.au.tsinghua.edu.cn/mirnasvm/	(24)
ProMir	Probabilistic learning of sequence and structure features of hairpins; applied to vertebrate miRNAs especially human	−	+	cbit.snu.ac.kr/~ProMiR2/	(25)
BayesmiRNAfind	Naive Bayes classifier of hairpins; applied to animal miRNAs	−	+	wotan.wistar.upenn.edu/miRNA/	(26)
RNAmicro	SVM classification of conserved hairpins; applied to animal miRNAs	+	−	www.tbi.univie.ac.at/~jana/software/RNAmicro.html	(27)
miRRim	Analyzes multiple alignments using a Hidden Markov model of conservation and structural features; tested on vertebrate genomes	+	−	mirrim.ncrna.org/	(28)

Table 17.3 (continued)

Software	Description	DL	WS	Web site	Ref.
microHARVESTER	Finds homologous mirRNAs to known miRNAs in plants using BLAST and structural filters	−	+	www-ab.informatik.uni-tuebingen.de/brisbane/tb/index.php	(29)
mirAlign	Finds homologous mirRNAs to known animal miRNAs using BLAST and structural filters	−	+	bioinfo.au.tsinghua.edu.cn/miralign	(30)
Srnaloop	Finds homologous mirRNAs to known miRNAs using a BLAST like algorithm and structural filters; applied to *C. elegans*	+	−	arep.med.harvard.edu/miRNA	(31)
Structure-Based miRNA analysis tool	Structural comparison of hairpin to known miRNA hairpins using RNAforester alignment tool	−	+	tagc.univ-mrs.fr/mirna/	(32)
findMiRNA	Prediction of miRNA precursor with targets in *Arabidopsis thaliana* transcripts	+	−	sundarlab.ucdavis.edu/mirna/	(33)
Microprocessor SVM, miRNA SVM	Classifies hairpins by predicting most likely Drosha processing site; uses SVM; applied to human	+	+	https://demo1.interagon.com/miRNA/	(34)

DL, software available for download; WS, software available as Web-server.
[1]There is another method called miRFinder (35). However, the software is not available for download or as Web-server.
[2]There are two programs named "miPred" by their authors.

Looking at the output, we find that that the most favorable binding has an interaction energy of −21.8 kcal/mol and pairs up pos. 34–57 of the UTR with pos. 1–19 of the miRNA. The hybridization structure is shown in dot/bracket notation. Note the "&" sign that separates the two molecules from each other. Using the option "−e," RNAduplex can also predict alternative (suboptimal) binding sites. For example, running RNAduplex−e 5 would list all binding sites within 5 kcal/mol of the best one. It is

Table 17.4
Programs for microRNA target prediction

Software	Description	DL	WS	Website	Ref.
PITA	Considers hybridization energy and thermodynamic target accessibility	+	+	genie.weizmann.ac.il/ pubs/mir07/ index.html	(38)
STarMir	Sfold-based algorithm that models target interaction as two-step reaction: nucleation at an accessible target site followed by hybrid elongation to disrupt local target secondary structure	−	+	sfold.wadsworth.org/ starmir.pl	(39)
TargetScanS	Searches perfect conserved seeds with small variations in the end	−	+	www.targetscan.org	(40)
miRNAda	Uses local alignment with emphasis on the seed and thermodynamic filter using modified RNAfold	+	+	www.microrna.org	(41)
PicTar	Predicts targets based on seed match and thermodynamic filter; calculates maximum likelihood by combining several predictions in one UTR	−	−	pictar.bio.nyu.edu	(42)
RNAhybrid	Finds target with best hybridization energy; provides BLAST like *E*-values	+	+	bibiserv.techfak.uni-bielefeld.de/ rnahybrid/	
Rna22	Extracts patterns of a set of miRNAs, finds regions that are likely to be targeted ("target islands"), finally finds best miRNA for the region	−	+	cbcsrv.watson.ibm.com/ rna22.html	(43)
Diana-microT	Scans UTRs in fixed window and calculates hybridization energy, which is compared to dinucleotide based shuffled random controls	−	+	diana.pcbi.upenn.edu	(44)
miTarget	SVM classifier using structural, thermodynamic, and position-based features	−	+	cbit.snu.ac.kr/~mi Target	(45)

DL, software available for download; WS, software available as Web-server.

important to note that RNAduplex only predicts *inter*molecular base pairs. For the more general case with *intra*molecular base-pairing allowed, the program RNAcofold can be used which is also part of the Vienna RNA package but is not covered in this chapter.

3.6.2. miRNA Target Prediction

While RNAduplex is a simple and useful method of calculating hybridization energies and structures, it cannot be recommended as general tool for miRNA target prediction. There are many specific tools for this task. For similar reasons as in **Section** 3, it is impossible, however, to recommend one of them or to review all in detail. Therefore, we want to point to recent reviews on this topic (37) and give a list of available methods (**Table 17.4**).

Most of the programs start with searching for short exact or nearly exact matches of the 5′-end of the miRNA ("seed"). Then the hybridization energy is evaluated to filter for valid targets. Variants include comparative analysis of multiple alignments and, more recently, consideration of the intramolecular secondary structure (accessibility) of the target.

4. Notes

4.1. Note 1

The installation process using ./configure and make should work on all UNIX-like systems. If you get error messages it may be necessary that you install additional "developer packages." On some Linux distributions, for example, there is no C-compiler installed by default. Also on OS X it may be necessary that you have installed the "XCode" tools.

If you do not have root privileges or want to install the programs into a different location than /usr/local/ (e.g., your home directory), you can use the following command:

```
# ./configure –prefix=/home/stefan
```

This installs the executable to /home/stefan/bin. Please note that the bin directory must be in your PATH of executables if you want to call the programs without specifying the complete path.

Acknowledgments

References

1. Bompfünewerer, A., Flamm, C., Fried, C., Fritzsch, G., Hofacker, I., Lehmann, J., Missal, K., Mosig, A., Müller, B., Prohaska, S., Stadler, B., Stadler, P., Tanzer, A., Washietl, S., Witwer, C. (2005) Evolutionary patterns of non-coding RNAs. *Theor Biosci* 123, 301–369.

2. Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M., Schuster, P. (1994) Fast folding and comparison of RNA secondary structures. *Monatsh Chem* 125, 167–188.

3. Knudsen, B., Hein, J. (2003) Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res* 31, 3423–3428.

4. Washietl, S., Hofacker, I. L., Stadler, P. F. (2005) Fast and reliable prediction of noncoding RNAs. *Proc Natl Acad Sci USA* 102, 2454–2459.

5. Rivas, E., Eddy, S. R. (2001) Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics* 2, 8–8.

6. Yao, Z., Weinberg, Z., Ruzzo, W. L. (2006) CMfinder – a covariance model based RNA motif finding algorithm. *Bioinformatics* 22, 445–452.

7. Eddy, S. R., Durbin, R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res* 22, 2079–2088.

8. Hofacker, I. L. (2007) RNA consensus structure prediction with RNAalifold. *Methods Mol Biol* 395, 527–544.

9. Washietl, S., Hofacker, I. L. (2004) Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics. *J Mol Biol* 342, 19–30.

10. Clote, P., Ferre, F., Kranakis, E., Krizanc, D. (2005) Structural RNA has lower folding energy than random RNA of the same dinucleotide frequency. *RNA* 11, 578–591.

11. Freyhult, E. K., Bollback, J. P., Gardner, P. P. (2007) Exploring genomic dark matter: a critical assessment of the performance of homology search methods on noncoding RNA. *Genome Res* 17, 117–125.

12. Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S. R., Bateman, A. (2005) Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res* 33, D121–D124.

13. Lowe, T. M., Eddy, S. R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res* 25, 955–964.

14. Lagesen, K., Hallin, P., Rodland, E. A., Staerfeldt, H. H., Rognes, T., Ussery, D. W. (2007) RNAmmer: consistent and rapid annotation of ribosomal RNA genes. *Nucleic Acids Res* 35, 3100–3108.

15. Lowe, T. M., Eddy, S. R. (1999) A computational screen for methylation guide snoRNAs in yeast. *Science* 283, 1168–1171.

16. Schattner, P., Decatur, W. A., Davis, C. A., , Fournier, M. J., Lowe, T. M. (2004) Genome-wide searching for pseudouridylation guide snoRNAs: analysis of the *Saccharomyces cerevisiae* genome. *Nucleic Acids Res* 32, 4281–4296.

17. Hertel, J., Hofacker, I. L., Stadler, P. F. (2008) SnoReport: computational identification of snoRNAs with unknown targets. *Bioinformatics* 24, 158–164.

18. Laslett, D., Canback, B., Andersson, S. (2002) BRUCE: a program for the detection of transfer-messenger RNA genes in nucleotide sequences. *Nucleic Acids Res* 30, 3449–3453.

19. Regalia, M., Rosenblad, M. A., Samuelsson, T. (2002) Prediction of signal recognition particle RNA genes. *Nucleic Acids Res* 30, 3368–3377.

20. Lim, L. P., Glasner, M. E., Yekta, S., Burge, C. B., Bartel, D. P. (2003) Vertebrate microRNA genes. *Science* 299, 1540.

21. Huang, T. H., Fan, B., Rothschild, M. F., Hu, Z. L., Li, K., Zhao, S. H. (2007) MiR-Finder: an improved approach and software implementation for genome-wide fast microRNA precursor scans. *BMC Bioinformatics* 8, 341.

22. Ng, K. L., Mishra, S. K. (2007) De novo SVM classification of precursor microRNAs from genomic pseudo hairpins using global and intrinsic folding measures. *Bioinformatics* 23, 1321–1330.

23. Jiang, P., Wu, H., Wang, W., Ma, W., Sun, X., Lu, Z. (2007) MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic Acids Res* 35, W339–W344.

24. Xue, C., Li, F., He, T., Liu, G. P., Li, Y., Zhang, X. (2005) Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics* 6, 310.

25. Nam, J. W., Shin, K. R., Han, J., Lee, Y., Kim, V. N., Zhang, B. T. (2005) Human microRNA prediction through a probabilistic co-learning model of sequence and structure. *Nucleic Acids Res* 33, 3570–3581.

26. Yousef, M., Nebozhyn, M., Shatkay, H., Kanterakis, S., Showe, L. C., Showe, M. K. (2006) Combining multi-species genomic data for microRNA identification using a Naive Bayes classifier. *Bioinformatics* 22, 1325–1334.

27. Hertel, J., Stadler, P. F. (2006) Hairpins in a Haystack: recognizing microRNA precursors in comparative genomics data. *Bioinformatics* 22, e197–e202.

28. Terai, G., Komori, T., Asai, K., Kin, T. (2007) miRRim: a novel system to find conserved miRNAs with high sensitivity and specificity. *RNA* 13, 2081–2090.

29. Dezulian, T., Remmert, M., Palatnik, J. F., Weigel, D., Huson, D. H. (2006) Identification of plant microRNA homologs. *Bioinformatics* 22, 359–360.

30. Wang, X., Zhang, J., Li, F., Gu, J., He, T., Zhang, X., Li, Y. (2005) MicroRNA identification based on sequence and structure alignment. *Bioinformatics* 21, 3610–3614.

31. Grad, Y., Aach, J., Hayes, G. D., Reinhart, B. J., Church, G. M., Ruvkun, G., Kim, J. (2003) Computational and experimental identification of C. elegans microRNAs. *Mol Cell* 11, 1253–1263.

32. Ritchie, W., Legendre, M., Gautheret, D. (2007) RNA stem-loops: to be or not to be cleaved by RNAse III. *RNA* 13, 457–462.

33. Adai, A., Johnson, C., Mlotshwa, S., Archer-Evans, S., Manocha, V., Vance, V., Sundaresan, V. (2005) Computational prediction of miRNAs in *Arabidopsis thaliana*. *Genome Res* 15, 78–91.

34. Helvik, S. A., and Saetrom, P. (2007) Reliable prediction of Drosha processing sites improves microRNA gene prediction. *Bioinformatics* 23, 142–149.

35. Bonnet, E., Wuyts, J., Rouze, P., Van de Peer, Y. (2004) Detection of 91 potential conserved plant microRNAs in *Arabidopsis thaliana* and *Oryza sativa* identifies important target genes. *Proc Natl Acad Sci USA* 101, 11511–11516.

36. Vogel, J., Wagner, E. G. (2007) Target identification of small noncoding RNAs in bacteria. *Curr Opin Microbiol* 10, 262–270.

37. Maziere, P., Enright, A. J. (2007) Prediction of microRNA targets. *Drug Discov Today* 12, 452–458.

38. Kertesz, M., Iovino, N., Unnerstall, U., Gaul, U., Segal, E. (2007) The role of site accessibility in microRNA target recognition. *Nat Genet* 39, 1278–1284.

39. Long, D., Chan, C. Y., Ding, Y. (2008) Analysis of microRNA-target interactions by a target structure based hybridization model. *Pac Symp Biocomput* 13, 64–74.

40. Lewis, B. P., Burge, C. B., Bartel, D. P. (2005) Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *Cell* 120, 15–20.

41. Enright, A. J., John, B., Gaul, U., Tuschl, T., Sander, C., Marks, D. S. (2003) MicroRNA targets in Drosophila. *Genome Biol* 5, R1.

42. Krek, A., Grun, D., Poy, M. N., Wolf, R., Rosenberg, L., Epstein, E. J., MacMenamin, P., da Piedade, I., Gunsalus, K. C., Stoffel, M., Rajewsky, N. (2005) Combinatorial microRNA target predictions. *Nat Genet* 37, 495–500.

43. Miranda, K. C., Huynh, T., Tay, Y., Ang, Y. S., Tam, W. L., Thomson, A. M., Lim, B., Rigoutsos, I. (2006) A pattern-based method for the identification of MicroRNA binding sites and their corresponding heteroduplexes. *Cell* 126, 1203–1217.

44. Kiriakidou, M., Nelson, P. T., Kouranov, A., Fitziev, P., Bouyioukos, C., Mourelatos, Z., Hatzigeorgiou, A. (2004) A combined computational-experimental approach predicts human microRNA targets. *Genes Dev* 18, 1165–1178.

45. Kim, S. K., Nam, J. W., Rhee, J. K., Lee, W. J., Zhang, B. T. (2006) miTarget: microRNA target gene prediction using a support vector machine. *BMC Bioinformatics* 7, 411.

Chapter 18

Conformational Disorder

Sonia Longhi, Philippe Lieutaud, and Bruno Canard

Abstract

In recent years it was shown that a large number of proteins are either fully or partially disordered. Intrinsically disordered proteins are ubiquitary proteins that fulfill essential biological functions while lacking a stable 3D structure. Despite the large abundance of disorder, disordered regions are still poorly detected. The identification of disordered regions facilitates the functional annotation of proteins and is instrumental in delineating boundaries of protein domains amenable to crystallization. This chapter focuses on the methods currently employed for predicting disorder and identifying regions involved in induced folding.

Key words: intrinsic disorder, intrinsically unstructured proteins, induced folding, prediction methods, disorder metaserver.

1. Introduction

In recent years there has been an increasing amount of experimental evidence pointing out the abundance of protein disorder within the living world. Recent computational studies have shown that the frequency and length of disordered regions increase with increasing organism complexity, with as much as one third of eukaryotic proteins containing long intrinsically disordered regions (1) and 12% of them being fully disordered (2). Intrinsically disordered proteins (IDPs) are functional proteins that fulfill essential biological functions while lacking highly populated constant secondary and tertiary structures under physiological conditions. Although there are IDPs that carry out their function while remaining disordered all the time (e.g., entropic chains), many of them undergo a disorder-to-order transition upon binding to their physiological partner(s), a process termed induced folding.

The functional relevance of disorder resides in an increased plasticity that enables the binding of numerous, structurally distinct targets. Accordingly, intrinsic disorder is a distinctive and common feature of "hub" proteins, with disorder serving as a determinant of protein promiscuity (3). As such, most IDPs are involved in functions that imply multiple partner interactions, such as molecular recognition, molecular assembly, cell cycle regulation, signal transduction, and transcription [for a recent review on IDPs see (4)].

The recognition of disordered regions has a practical interest in that it facilitates the functional annotation of proteins (5) and is instrumental for delineating protein domains amenable to crystallization.

Statistical analyses showed that amino acid sequences encoding disordered regions are significantly different from those of ordered proteins, thus allowing IDPs to be predicted with a rather good accuracy. Specifically, IDPs (i) have a biased amino acid composition, being enriched in G, S, P and depleted in W, F, I, Y, V, L, (ii) have a low secondary structure content, (iii) tend to have a low sequence complexity, (iv) are on average much more variable than ordered ones due to less restrictive amino acid substitutions.

Based on these peculiar sequence features, a number of disorder predictors have been developed in recent years, the majority of which are available on the Web [for reviews see (6–8)]. In this chapter, we focus on the various disorder predictors currently available and present a general suggested procedure for disorder prediction.

2. Materials

1. Computer connected to the Web

3. Methods

3.1. Running Individual Disorder Predictions

In the last decade a number of disorder predictors have been developed, which exploit the sequence bias of disordered proteins. Different types or "flavors" of protein disorder exist (9), differing in the extent (i.e., the amount of residual secondary and/or tertiary structure) and length of disorder. Since different predictors rely on different physico-chemical parameters, a given predictor can be more performant in detecting a given feature of a disordered protein. Hence, predictions good enough to decipher the modular organization of a protein can only be obtained by combining various predictors [for examples see (6, 7, 10–13)].

It is useful to distinguish two kinds of predictors: those that have been trained on data sets of disordered proteins and those that have not. Data sets of disorder are necessarily biased, since they contain relatively few disordered proteins. Indeed, the Dis-Prot (http://www.disprot.org/), which is the largest publicly available database of disordered proteins whose disorder has been experimentally assessed, contains only 523 entries (14), and regions of missing electron density in the PDB are generally short, as long regions generally prevent crystallization. While predictors trained on data sets of disordered regions identify disordered regions on the basis of the peculiar sequence properties that characterize them, the others identify disorder as lack of ordered 3D structure. The second group of predictors avoids the shortcomings and biases associated with the disordered data sets. Therefore, they are expected to perform better than the former methods on disordered proteins presently underrepresented in training data sets (i.e., fully or mostly disordered proteins).

3.1.1. Predictors Trained on Data Sets of Disordered Proteins

PONDR (Predictor of Natural Disordered Regions) (*see* http://www.pondr.com), a neural network based on local amino acid composition, flexibility, and other sequence features, was the first predictor to be developed (15). PONDR is available in various versions. While VSL1 performs better to identify short regions of disorder, VL3 should be preferred to delineate domains as it gives smoother predictions. Notably, VL-XT can highlight potential protein-binding regions, indicated by sharp drops in the middle of long disordered regions (*see* Section 3.3).

3.1.1.1. PONDR

1. Before you can use PONDR you will need to create a new user account (limited to 50 predictions for academic users).
2. Paste your sequence in raw format and click on "submit."
3. The result is provided as a gif file. The significance threshold above which residues are considered to be disordered is 0.5. Segments composed by more than 40 consecutive disordered residues are highlighted by a thick black line.

3.1.1.2. DisProt VSL2

As the accuracy of PONDR predictors is limited for short disordered regions (<30 residues), the group of Dunker has recently developed a new predictor, DisProt VSL2, which is intended to give accurate predictions regardless of the length of the disordered region (16). The VSL2 predictor is based on a support vector machine. The data set, obtained from both DisProt and PDB, has been split into two groups on the basis of the length of disorder (i.e., >30 and <30 residues). VSL2 turned out to behave well with both subgroups and to be able to identify short disordered regions

that were mis-predicted by the previous PONDR predictors. The publicly available VSL2 server (*see* http://www.ist.temple.edu/disprot/predictorVSL2.php) consists of two variants of the VSL2 predictor: VSL2B is the baseline model that uses only 26 features calculated from the amino acid sequence, while the more accurate VSL2P uses 22 additional features derived from PSI-BLAST profiles. The VSL2 predictor integrating the full set of different features (including residue features, PSI-BLAST profiles, and secondary structure PHD and PSIPRED predictions) can be downloaded from http://www.ist.temple.edu/disprot/predictorVSL2.php.

1. Paste your sequence in raw format, enter your E-mail address, and click on "submit."

2. The result is provided in another page and the plot can be saved (png format) by clicking on it with the mouse right button. The output also provides a table with disorder probabilities per residue. The significance threshold above which residues are considered to be disordered is 0.5.

3.1.1.3. DisProt (PONDR) VL3, VL3H, VL3E, and VL3P

VL3 uses several features from the previously introduced PONDR VL2 (9), but benefits from optimized predictor models and a slightly larger (152 versus 145) set of disordered proteins that were cleaned of mislabeling errors found in the smaller set. The VL3 predictor is based on an ensemble of feed-forward neural networks whose training stage is done using a data set obtained from both DisProt and PDB. PONDR VL3H uses the same method as VL3, but it utilizes homologues of the disordered proteins in the training stage, while PONDR VL3P uses attributes derived from sequence profiles obtained by PSI-BLAST searches (17, 18). These predictors are available at http://www.ist.temple.edu/disprot/predictor.php. Requests are limited to 100 per IP address per day and the maximum length of a query sequence is limited to 5,000 residues. For the VL3E predictor, which results from the combination of VL3H and VL3P, up to 10 queries no longer than 500 residues can be processed per IP address per day. Predictions for VL3E are sent by E-mail upon completion.

1. Chose the predictor to be run among VL2, VL3, VL3H, and VL3E.

2. Paste your sequence in raw format, enter your E-mail address, and click on "submit."

3. Prediction results are returned online and the plot can be saved (png format) by clicking on it with the mouse right button. The output also provides a table with disorder probabilities per residue. The significance threshold above which residues are considered to be disordered is 0.5.

3.1.1.4. Globplot 2   Globplot 2 (http://globplot.embl.de) uses the "Russell/Lind-ing" scale that expresses the propensity for a given amino acid to be in "random coil" or in "regular secondary structure" (19). It also provides an easy overview of modular organization of large proteins thanks to user-friendly, built-in SMART, PFAM, and low-complexity predictions. Note that in Globplot outputs, changes of slope often correspond to domain boundaries.

1. Paste your sequence in raw format or enter the SwissProt ID (or AC) in the foreseen field, enter Title (optional), and click on "GlobPlot now."

2. The result page provides a postscript (ps) file that can be down-loaded. Below the graph, the amino acid sequence of the pro-tein is given, with disordered residues colored in blue.

3.1.1.5. DisEMBL   DisEMBL (http://dis.embl.de) is based on a neural network and consists of three separate predictors, trained on separate data sets, that comprise respectively residues within "loops/coils," "hot loops" (loops with high B-factors – i.e., very mobile from X-ray crystal structure), or that are missing from the PDB X-ray struc-tures (called "Remark 465") (20). Among these, the only true disorder predictor is Remark 465, as the two others only predict regions devoid of regular secondary structure. DisEMBL also provides prediction of low sequence complexity (CAST predictor) and aggregation propensity (TANGO predictor).

1. Paste your sequence in raw format or enter the SwissProt ID (or AC) in the foreseen field, enter Title (optional), and click on "DisEMBL protein."

2. The result page provides a postcript file that can be down-loaded. Below the graph, the amino acid sequence of the protein is given, with residues in loops and hot loops being colored in blue and red, respectively. Disordered residues, as predicted by Remark 465, are shown in green.

3.1.1.6. Disopred2   Disopred2 (http://bioinf.cs.ucl.ac.uk/disopred) is based on sup-port vector machine classifiers trained on PSI-BLAST profiles (21). It therefore incorporates information from multiple sequence alignments since its inputs are derived from sequence profiles generated by PSI-BLAST. Hence, prediction accuracy is lower if there are few homologues.

1. Paste your sequence in raw format, enter Title (optional) and your E-mail address, and click on "predict."

2. Prediction results are sent by E-mail upon completion. Aster-isks represent disordered predictions and dots predictions of order. Links to disorder profile plots (ps, pdf, and jpg for-mats) and plain text files containing classifier outputs are given in the main body of the E-mail.

3.1.1.7. RONN

RONN (http://www.strubi.ox.ac.uk/RONN) uses an approach based on a bio-basis function neural network. It relies on the calculation of "distances," as determined by sequence alignment, from well-characterized prototype sequences (ordered, disordered, or a mixture of both). Its key feature is that amino acid side chain properties are not considered at any stage (22).

1. Paste your sequence in raw format and click on "send."

2. Prediction results are returned online and the plot can be saved as an image (png) format. Below the graph, the amino acid sequence of the protein is given, with disordered residues highlighted by an asterisk. Boundaries of disordered regions are also clearly indicated above the graph.

3.1.1.8. DISpro

DISpro (http://scratch.proteomics.ics.uci.edu/) is based on a neural network (23). It combines sequence profiles obtained by PSI-BLAST, secondary structure predictions and solvent accessibility. This predictor was trained on disordered sequences (i.e., regions of missing atomic coordinates) derived from the PDB.

1. Enter your E-mail address (required), the sequence name (optional), paste your sequence in raw format, select the prediction to be run by ticking the appropriate box, and click on "submit."

2. Prediction results are sent by E-mail. Residues predicted to be disordered or ordered are indicated by a "D" or an "O," respectively. Per residue disorder probabilities are also provided.

3.1.1.9. SPRITZ

The SPRITZ server (http://protein.cribi.unipd.it/spritz) takes into account sequence profiles obtained from PSI-BLAST and structure predictions. SPRITZ uses two separate predictors based on vector machines trained on different data sets (24). The training data set of short disordered regions (less than 45 residues) was derived from a subset of PDB sequences with short regions of missing density, while the training data set of long regions was derived from both DisProt and from a subset of the PDB (i.e., PDBselect25). This server allows the submission of several sequences at one time and offers the possibility of choosing between predictions of short or of long disordered regions.

1. Enter your E-mail address, the name of the query sequence (optional), paste your sequence in raw format.

2. Chose the type of disorder (i.e., "short" or "long") and click on "predict."

3. Prediction results are sent by E-mail. Residues predicted to be in regions other than helices and β strands are indicated by a "C." For a detailed explanation of the results, see http://distill.ucd.ie/distill/explanation.html#output_formats.

3.1.1.10. PreLink

PreLink (http://genomics.eu.org/prelink/) relies on amino acid composition and on low hydrophobic cluster content (25). In this respect, it is a derivative of HCA, a powerful approach that is discussed below. Prelink predicts regions that are expected to be unstructured in all conditions, regardless of the presence of a binding partner. Thus, it generally predicts as ordered disordered regions that have the potential to be ordered in the presence of a partner (i.e., to undergo induced folding).

1. Paste your sequence in raw format or upload a file (Fasta format) containing multiple sequences (click on "upload" to select the file on your computer and then on "upload"), and click on "submit."

2. Prediction results are returned online. The plot can be saved as an image (png format) by clicking on it with the mouse right button. Below the graph the amino acid sequence is given and disordered residues are shown in red.

3.1.1.11. Ucon

Ucon (http://www.predictprotein.org/submit_ucon.html) is a method that combines predictions for protein-specific contacts with a generic pairwise potential. This predictor was trained against the DisProt and the PDB. It performs well in predicting proteins with long unstructured regions (26).

1. Enter your E-mail address, tick the box "results on our website, NOT in e-mail," and enter the sequence name (optional).

2. Paste your sequence in raw format or click on "SRS6" to retrieve the sequence from a public database, and click on "submit/run prediction."

3. Upon completion of the prediction, an E-mail is sent with a link for accessing the results. Alternatively, the user can choose to be sent an E-mail with the results in html format.

3.1.1.12. OnD-CRF

OnD-CRF (http://babel.ucmp.umu.se/ond-crf/) predicts disorder using conditional random fields (CRF) (27).

1. Paste your sequence in raw or Fasta format or upload the query sequence from a file and click on "submit query" (you can also choose to be sent results by E-mail).

2. Prediction results are returned online. The plot can be saved as an image (png format) by clicking on it with the mouse right button. The threshold above which residues are considered as disordered is dynamic and indicated above the plot. Below the graph, boundaries of disordered regions are provided and the amino acid sequence is also given, with disordered residues shown in red. Disorder probabilities per residue are given upon positioning the pointer on the amino acid sequence shown below the graph.

3.1.1.13. POODLE-S   POODLE-S (http://mbs.cbrc.jp/poodle/poodle-s.html) pre-
dicts disordered regions from amino acid sequences by using
physico-chemical features and reduced amino acid set of a posi-
tion-specific scoring matrix (28). POODLE-S was trained against
the PDB and the DisProt database. Assessment of performance
was done on the data set of CASP7.

1. Paste your sequence in raw format, enter your E-mail address,
   choose the type of prediction ("missing residues" or "High
   B-Factor residues"), and click on "submit."

2. Prediction results are sent by E-mail, where a link to a gra-
   phical output is given. Residues with disorder probabilities
   higher than 0.5 are considered to be disordered. Probabilities
   per residue are given upon positioning the pointer on the
   disorder curve. The plot can be saved by using the "screen
   capture" option of the user's computer.

3.1.1.14. PrDOS   PrDOS (http://prdos.hgc.jp/cgi-bin/top.cgi) is composed of
two predictors: a predictor based on the local amino acid sequence
and one based on template proteins (or homologous proteins for
which structural information is available). The first part is imple-
mented using support vector machine (SVM) algorithm for the
position-specific score matrix (or profile) of the input sequence.
More precisely, a sliding window is used to map individual residues
into a feature space. A similar idea has already been used in a
secondary structure prediction, as in PSI-PRED. The second part
assumes the conservation of intrinsic disorder in protein families
and is simply implemented using PSI-BLAST and our own mea-
sure of disorder, as described later. The final prediction is done as
the combination of the results of the two predictors.

1. Paste your sequence in raw format, enter the sequence name
   and the E-mail address (optional), and click on "predict."

2. A new page appears where the estimated calculation time is
   indicated. The user is asked to confirm the submission by
   clicking on the OK button.

3. On the results page, the plot can be saved as an image (png
   format) by clicking on it with the mouse right button. Resi-
   dues with disorder probabilities higher than 0.5 are consid-
   ered to be disordered. Above the graph, the amino acid
   sequence is shown and disordered residues are shown in red.
   Disorder probabilities per residue can be obtained by clicking
   on the download button (below the graph), which yields an
   output in the casp or csv format.

3.1.1.15. DisPSSMP   DisPSSMP (http://biominer.bime.ntu.edu.tw/dispssmp/) is
based on the use of position-specific scoring matrices (PSSMs)
that takes into account the amino acid composition and residue

position, i.e., the environment around each residue. The specificity of DisPSSMP is that it condenses the PSSMs with respect to physico-chemical properties of amino acids (29).

1. Paste your sequence in raw format, enter the E-mail address (optional), and click on "predict."

2. A new page appears and the user is asked to click on "links."

3. Prediction results are returned online. The plot can be saved as an image (png format) by clicking on it with the mouse right button. Residues with disorder probabilities higher than 0.5 are considered to be disordered. Below the graph, disorder probabilities per residue are given in casp format.

3.1.2. Predictors That Have Not Been Trained on Disordered Proteins

3.1.2.1. The Charge/ Hydropathy Method and Its Derivative FoldIndex

The charge/hydropathy analysis is based on the elegant reasoning that folding of a protein is governed by a balance between attractive forces (of hydrophobic nature) and repulsive forces (electrostatic, between similarly charged residues) (30). Thus, globular proteins can be distinguished from unstructured ones based on the ratio of their net charge versus their hydropathy. The Mean Net Charge (R) of a protein is determined as the absolute value of the difference between the number of positively and negatively charged residues divided by the total number of amino acid residues. It can be calculated using the program ProtParam at the ExPASy server (http://www.expasy.ch/tools). The Mean Hydrophobicity (H) is the sum of normalized hydrophobicities of individual residues divided by the total number of amino acid residues minus 4 residues (to take into account fringe effects in the calculation of hydrophobicity). Individual hydrophobicities can be determined using the Protscale program at the ExPASy server, using the options "Hphob / Kyte & Doolittle," a window size of 5, and normalizing the scale from 0 to 1. The values computed for individual residues are then exported to a spreadsheet, summed and divided by the total number of residues minus four to yield (H). A protein is predicted as disordered if $H < [(R + 1.151) / 2.785]$.

A drawback of this approach is that it gives only a global (i.e., not positional) indication, not valid if the protein is composed of both ordered and disordered regions. It can be applied only to protein domains, implying that a prior knowledge of the modular organization of the protein is required.

A derivative of this method, FoldIndex (http://bip.weizmann.ac.il/fldbin/findex), solves this problem by computing the charge/hydropathy ratio using a sliding window along the protein (31). However, since the default sliding window is set to 51 residues, FoldIndex does not provide reliable predictions for the N- and C-termini and is therefore not recommended for proteins with less than 100 residues.

1. Paste your sequence in raw format and click on "process."

2. The results page shows a plot that can be saved as an image (png format) by clicking on it with the mouse right button. Disordered regions are shown in red and have a negative "foldability" value, while ordered regions are shown in green and have a positive value. Disorder statistics (number of disordered regions, longest disordered region, number of disordered residues and scores) are given below the plot.

3.1.2.2. NORSp

NORSp (No Ordered Regular Secondary structure predictor) (http://cubic.bioc.columbia.edu/services/NORSp/submit.html) generates multiple sequence alignments and relies on the principle that long regions predicted to be devoid of secondary structure and accessible to the solvent are generally unstructured (32). However, a few exceptions to this rule exist, namely the "loopy proteins" that are devoid of regular secondary structure and yet are ordered (33).

1. Enter your E-mail address, paste your sequence in raw format or upload a sequence file, and click on "submit/run prediction."

2. Upon completion of prediction, the user is sent an E-mail with a link to the result page. Boundaries of NORS regions are indicated before the annotated sequence in which solvent exposure, secondary structure elements, coils, and trans-membrane regions are also indicated.

3.1.2.3. IUPred

IUPred (http://iupred.enzim.hu) uses a novel algorithm that evaluates the energy resulting from inter-residues interactions (34). Although it was derived from the analysis of the sequences of globular proteins only, it allows the recognition of disordered proteins based on their lower interaction energy. This provides a new way to look at the lack of a well-defined structure, which can be viewed as a consequence of a significantly lower capacity to form favorable contacts, correlating with studies by the group of Galzitskaya [see (35)].

1. Enter the sequence name (optional), paste your sequence in raw format, chose the prediction type (long disorder, short disorder, structured regions), choose "plot" in output type and adjust the plot window size, and click on "submit."

2. Prediction results are promptly returned online and the plot can be saved (png format) by clicking on it with the mouse right button. The output also provides a table with disorder probabilities per residue. The significance threshold above which residues are considered to be disordered is 0.5.

3.1.2.4. FoldUnfold

The FoldUnfold predictor (http://skuld.protres.ru/~mlobanov/ogu/ogu.cgi) calculates the expected average number of contacts *per* residue from the amino acid sequence alone (35). The average

number of contacts per residue was computed from a data set of globular proteins. A region is considered as natively unfolded when the expected number of close residues is less than 20.4 for its amino acids and the region is greater or equal in size to the averaging window.

1. Paste your sequence in raw format or upload a sequence file, tick boxes "analyse regions also in those proteins which are predicted as fully disordered" and "write profile" and click on "send."

2. Prediction results are returned online. Boundaries of disordered regions are given. In the profile disordered residues are shown in red. The average contacts per residue are also given.

3.1.2.5. DRIP-PRED   DRIP-PRED (Disordered Regions In Proteins PREDiction) (http://www.sbc.su.se/~maccallr/disorder/) is based on search of sequence patterns obtained by PSI-BLAST that are not typically found in the PDB (http://www.forcasp.org/paper2127.html). If a sequence profile is not well represented in the PDB, then it is expected to have no ordered 3D structure. For a query sequence, sequence profile windows are extracted and compared to the reference sequence profile windows, and then an estimation of disorder is performed for each position. As a last step, the results of this comparison are weighted by PSI-PRED predictions. As predictions can take up to 8 hours, it is preferred to choose to be sent results by E-mail as well. In this latter case, the user is sent an E-mail with a link to the result page.

1. Enter your E-mail address (optional), paste your sequence in raw format, click on "submit," and give your job a name (optional).

2. Prediction results are shown in the amino acid sequence format with disordered residues underlined and a color code as a function of disorder probabilities. Per residue disorder probabilities are given below the amino acid sequence in the casp format.

3.1.3. Hydrophobic Cluster Analysis (HCA): A Nonconventional Disorder Predictor   Another non-automated method that is very useful for unveiling unstructured regions is HCA (36). HCA outputs can be obtained from http://bioserv.rpbs.jussieu.fr/RPBS/cgi-bin/ and from the MeDor metaserver (http://www.vazymolo.org/MeDor/). HCA provides a 2D helical representation of protein sequences in which hydrophobic clusters are plotted along the sequence (36). As such, HCA is not *stricto sensu* a predictor. Disordered regions are recognizable as they are depleted (or devoid) in hydrophobic clusters (*see* **Fig. 18.1**). HCA stands aside from other predictors since it provides a representation of the short-range environment of each amino acid, thus giving information not only on order/

disorder but also on the folding potential (*see* Section 3.2). Although HCA does not provide a quantitative prediction of disorder and rather requires human interpretation, it provides additional, qualitative information as compared to automated predictors. In particular, HCA highlights coiled-coils (*see* **Fig**. **18.1**), regions with a biased composition, regions with potential for induced folding, and very short potential globular domains (for examples see (6, 7, 11)). Finally, it allows meaningful comparison with related protein sequences and enables a better definition of the boundaries of disordered regions.
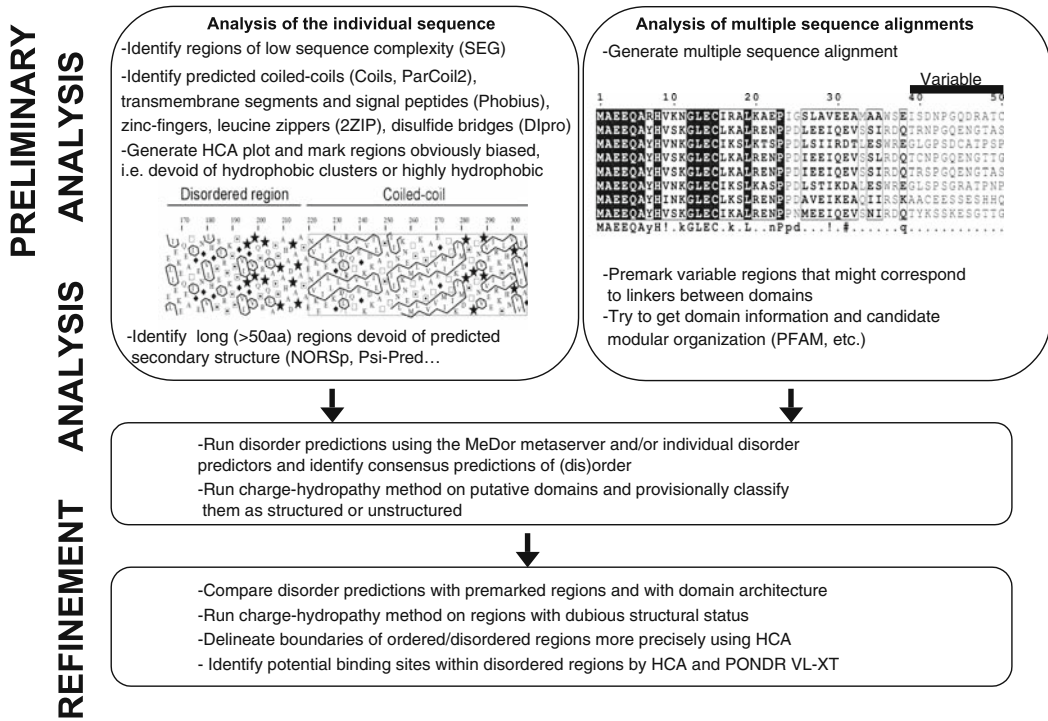
Fig. 18.1. General scheme for prediction of disordered regions in a protein.

3.1.4. Predictor Specificities

Some predictors, namely Disopred2, Prelink and DisEMBL Remark465, perform better on short disordered regions in the context of globally ordered proteins. These predictors have a good specificity (i.e., they predict relatively few ordered residues to be disordered), but a moderate sensitivity (i.e., they miss a significant number of disordered residues). Although recent reports suggest that progress has been made in predicting short (< 20 residues) disordered regions, it is noteworthy that the shorter the region of predicted disorder, the higher the probability that it corresponds to an ordered, yet devoid of regular secondary structure, protein

segment connecting α- or β-strands (Longhi et al., unpublished data). Finally, while IUPred and Ucon perform well for predicting long disordered segments, some predictors are "polyvalent" (e.g., RONN, PONDR VSL1, FoldIndex, and Globplot2).

3.2. Identifying Regions of Induced Folding

IDPs bind to their target(s) through "molecular recognition elements" (MoREs), where these latter are interaction-prone short segments that become ordered upon binding to partner(s) (37). It has been noticed (38) that PONDR VL-XT can highlight potential MoREs [for examples see (39, 40)]. The analysis of hydrophobic clusters and secondary structures is also instrumental for the identification of regions undergoing induced folding, because burying of hydrophobic residues at the protein–partner interface is often the major driving force in protein folding. In some cases, hydrophobic clusters are found within secondary structure elements that are unstable in the native protein, but can stably fold upon binding to a partner. Therefore, HCA can be very informative in highlighting potential induced folding regions (*see* **Fig**. **18.2A**).

1. Perform HCA on the query sequence using either the HCA server or the MeDor metaserver (*see* Section 3.4) and look for short hydrophobic clusters occurring within disordered regions.

2. Perform prediction using PONDR VL-XT and look for sharp (and short) drops in the middle of disorder predictions.

3.3. General Procedure for Disorder Prediction

As the performance of predictors is dependent on both the type of disorder they predict and on the type of disorder against which they were trained, multiple prediction methods need to be combined to improve the accuracy and specificity of disorder predictions (6, 7). **Figure 18.1** illustrates a general sequence analysis procedure that integrates the peculiarities of each method to predict disordered regions.

1. Retrieve the amino acid sequence of the protein of interest by entering the protein name at the NCBI home page (http://www.ncbi.nlm.nih.gov/) after selecting "protein" in the "search" field.

2. Perform an analysis of sequence composition using the ProtParam ExPASy server (http://www.expasy.ch/tools/protparam.html) and compare the results with the average sequence composition of proteins within the UniProtKB/Swiss-Prot database (*see* http://www.expasy.ch/sprot/relnotes/relstat.html).

3. Perform an analysis of sequence complexity using the SEG program (41). The SEG program can be downloaded from ftp://ftp.ncbi.nih.gov/pub/seg/seg, while simplified
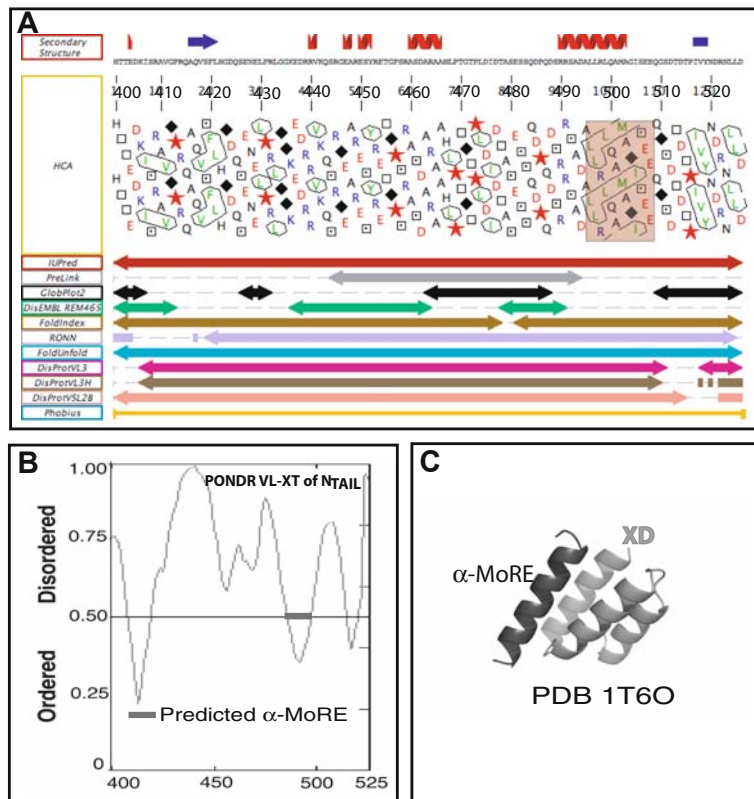
Fig. 18.2. (**A**) MeDor output of N$_{TAIL}$ (VAZyMolO accession number: VAZy90 (11); DisProt accession number: DP00160). Predicted secondary structure elements, as provided by the Pred2ary predictor, are shown above the N$_{TAIL}$ sequence. The HCA plot is shown below the amino acid sequence. Arrows below the HCA plot correspond to regions of predicted disorder. The induced folding region (α-MoRE) is highlighted. Note that the disordered state of N$_{TAIL}$ has been experimentally confirmed (39, 49). (**B**) Graphical output of the VL-XT prediction of the N$_{TAIL}$ domain with the predicted α-MoRE highlighted by a black bar. (**C**) Structure of the α-MoRE (dark grey α-helix) in complex with the C-terminal X domain of the measles virus P (50) (PDB code: 1T6O) confirming its involvement in partner-induced folding. The picture was obtained using Pymol. Note that a drop in the VL-XT output similar to that corresponding to the α-MoRE can also be observed for the 400–420 region (**B**) and that the HCA plot of this region shows the presence of a small hydrophobic cluster (**A**), consistent with an additional induced folding region. The folding potential of this region has been experimentally confirmed through spectroscopic studies in the presence of 20% TFE (51).

versions with default settings can be run at either http://men del.imp.univie.ac.at/METHODS/seg.server.html or http://www.ncbi.nlm.nih.gov/BLAST. The stringency of the search for low-complexity segments is determined by 3 user-defined parameters: trigger window length [W], trigger complexity [K(1)], and extension complexity [K(2)]. Typical parameters for disorder prediction of long non-globular domains are [W]=45, [K(1)]=3.4, and [K(2)]=3.75, while for short

non-globular domains are [W]=25, [K(1)]=3.0, and [K(2)]=3.3. Note however, that low-complexity regions can also be found in ordered proteins, such as coiled-coils and other non-globular proteins like collagen.

4. Search for (i) signal peptides and transmembrane regions using the Phobius server (http://phobius.sbc.su.se/index.html) (42), (ii) leucine zippers using the 2ZIP server (http://2zip.molgen.mpg.de/) (43), and (iii) coiled-coils using programs such as Coils (http://www.ch.embnet.org/software/COILS_form.html) (44). Note that the identification of coiled-coils is crucial since they can lead to mis-predictions of disorder [for examples see (6, 7)]. It is also recommended to use DIpro (http://contact.ics.uci.edu/bridge.html) (45) to identify possible disulfide bridges and search for possible metal-binding regions by looking for conserved Cys_3-His or Cys_2-His_2 motifs in multiple sequence alignments. Indeed, the presence of conserved cysteines and/or of metal-binding motifs prevents meaningful local predictions of disorder within these regions, as they may display features typifying disorder while gaining structure upon disulfide formation or upon binding to metal ions (30).

5. Run hydrophobic cluster analysis (HCA) (http://bioserv.rpbs.jussieu.fr/RPBS/cgi-bin/) to highlight regions devoid of hydrophobic clusters and with obvious sequence bias composition.

6. Search for long (>50 residues) regions devoid of predicted secondary structure using the PSI-PRED (http://bioinf.cs.ucl.ac.uk/psipred/psiform.html) (46) and PredictProtein (http://www.predictprotein.org/) servers.

7. Generate a multiple sequence alignment. A set of related sequences can be obtained by running PSI-BLAST (http://www.ncbi.nlm.nih.gov/blast/Blast.cgi). Click on the "get selected sequences" option and save them to a file in Fasta format. Use this file as input for building up a multiple sequence alignment using ClustalW (http://align.genome.jp/). Mark variable regions likely corresponding to flexible linkers or long disordered regions.

8. Analyze the sequence against the PFAM database (http://pfam.jouy.inra.fr/hmmsearch.shtml) to get domain information and candidate modular organization.

9. Run individual disorder predictions and identify a consensus of disorder. As a first approach, we suggest to perform predictions using the default parameters of each predictor, as they generally perform at best in terms of accuracy, specificity, and sensitivity. Once a gross domain architecture for the protein of interest is established, the case of domains whose

structural state is uncertain can be settled using the charge/hydropathy method, which has a quite low error rate. As a last step, boundaries between ordered and disordered regions can be refined and regions with propensity to undergo induced folding can be identified using HCA and PONDR VL-XT.

Since running multiple prediction methods is a time-consuming procedure, use of the MeDor metaserver (47) (*see* Section 3.4) can considerably speed up the procedure as it allows launching multiple, simultaneous disorder predictions.

3.4. Running the MeDor Metaserver for the Prediction of Disorder

MeDor (MEtaserver of DisORder) (http://www.vazymolo.org/MeDor/) (47) helps to identify protein disorder by providing a graphical interface with a unified view of the output of multiple disorder predictors (**Fig. 18.2A**). It allows fast, simultaneous analysis of a query sequence by multiple predictors and easy comparison of the prediction results. It also enables a standardized access to disorder predictors and allows meaningful comparisons among various query sequences. Beyond providing a graphical representation of the regions of predicted disorder, MeDor is also conceived to serve as a tool allowing to highlight specific regions of interest and to retrieve their sequence (47). Presently, the following programs are run by MeDor: a secondary structure prediction (SSP), based on the StrBioLib library of the Pred2ary program (48), HCA, IUPred, PreLink, RONN, FoldUnfold, DisEMBL, FoldIndex, GlobPlot2, DISPROT VSL2B, VL3, VL3H, and Phobius. While SSP and HCA do not require a Web connection, the other predictors are remotely launched through connection to the public Web servers. Additional predictors could be nevertheless easily implemented in MeDor in the future. Predictors to be run can be selected from the MeDor input frame.

MeDor provides a graphical output (**Fig. 18.2A**), in which the sequence query and the results of the various predictors are featured horizontally, with a scroll bar allowing progression from the N-terminus to the C-terminus. All predictions are drawn along the sequence that is represented as a single, continuous horizontal line. In addition, MeDor outputs can be saved and printed.

It is noteworthy that MeDor is not intended to provide a consensus of disorder prediction and is rather conceived to speed up the disorder prediction step by itself and to provide a global overview of predictions. As such, the identification of regions of disorder is presently done on a case-by-case basis and requires human analysis. Future developments implying the generation of an automated and reliable consensus of disorder are expected to further accelerate the identification of both structured and disordered regions.

1. Go to the MeDor home page (http://www.vazymolo.org/MeDor/).

2. Paste the sequence in either raw or Fasta format and optionally enter the sequence name.

3. Click on "Start MeDor."

4. Alternatively, MeDor can be downloaded (choose the appropriate version according to your computer environment). Using the downloaded version of MeDor instead of the applet version enables the user to (i) run DISPROT VL3, VL3H, and VSL2B predictions (in the limit of 100 requests per IP number), (ii) print the results, (iii) save the output as an image, (iv) save (and load) files in the MeDor format, (v) access the comment panel, and (vi) import a sequence by providing the SwissProt accession number.

Acknowledgments

References

1. Ward, J. J., Sodhi, J. S., McGuffin, L. J., Buxton, B. F., Jones, D. T. (2004) Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *J Mol Biol* 337(3), 635–645.

2. Bogatyreva, N. S., Finkelstein, A. V., Galzitskaya, O. V. (2006) Trend of amino acid composition of proteins of different taxa. *J Bioinform Comput Biol* 4(2), 597–608.

3. Haynes, C., Oldfield, C. J., Ji, F., Klitgord, N., Cusick, M. E., Radivojac, P., Uversky, V. N., Vidal, M., Iakoucheva, L. M. (2006) Intrinsic disorder is a common feature of hub proteins from four eukaryotic interactomes. *PLoS Comput Biol* 2(8), e100.

4. Radivojac, P., Iakoucheva, L. M., Oldfield, C. J., Obradovic, Z., Uversky, V. N., Dunker, A. K. (2007) Intrinsic disorder and functional proteomics. *Biophys J* 92(5), 1439–1456.

5. Lobley, A., Swindells, M. B., Orengo, C. A., Jones, D. T. (2007) Inferring function using patterns of native disorder in proteins. *PLoS Comput Biol* 3(8), e162.

6. Ferron, F., Longhi, S., Canard, B., Karlin, D. (2006) A practical overview of protein disorder prediction methods. *Proteins* 65(1), 1–14.

7. Bourhis, J., Canard, B., Longhi, S. (2007) Predicting protein disorder and induced folding: from theoretical principles to practical applications. *Curr Protein Peptide Sci* 8, 135–149.

8. Uversky, V. N., Radivojac, P., Iakoucheva, L. M., Obradovic, Z., Dunker, A. K. (2007) Prediction of intrinsic disorder and its use in functional proteomics. *Methods Mol Biol* 408, 69–92.

9. Vucetic, S., Brown, C., Dunker, K., Obradovic, Z. (2003) Flavors of protein disorder. *Proteins* 52, 573–584.

10. Karlin, D., Ferron, F., Canard, B., Longhi, S. (2003) Structural disorder and modular organization in Paramyxovirinae N and P. *J Gen Virol* 84(Pt 12), 3239–3252.

11. Ferron, F., Rancurel, C., Longhi, S., Cambillau, C., Henrissat, B., Canard, B. (2005) VaZyMolO: a tool to define and

classify modularity in viral proteins. *J Gen Virol* 86(Pt 3), 743–749.

12. Severson, W., Xu, X., Kuhn, M., Senutovitch, N., Thokala, M., Ferron, F., Longhi, S., Canard, B., Jonsson, C. B. (2005) Essential amino acids of the hantaan virus N protein in its interaction with RNA. *J Virol* 79(15), 10032–10039.

13. Llorente, M. T., Barreno-Garcia, B., Calero, M., Camafeita, E., Lopez, J. A., Longhi, S., Ferron, F., Varela, P. F., Melero, J. A. (2006) Structural analysis of the human respiratory syncitial virus phosphoprotein: characterization of an a-helical domain involved in oligomerization. *J Gen Virol* 87, 159–169.

14. Sickmeier, M., Hamilton, J. A., LeGall, T., Vacic, V., Cortese, M. S., Tantos, A., Szabo, B., Tompa, P., Chen, J., Uversky, V. N., Obradovic, Z., Dunker, A. K. (2007) DisProt: the database of disordered proteins. *Nucleic Acids Res* 35(Database issue), D786–D793.

15. Romero, P., Obradovic, Z., Li, X., Garner, E. C., Brown, C. J., Dunker, A. K. (2001) Sequence complexity of disordered proteins. *Proteins* 42(1), 38–48.

16. Obradovic, Z., Peng, K., Vucetic, S., Radivojac, P., Dunker, A. K. (2005) Exploiting heterogeneous sequence properties improves prediction of protein disorder. *Proteins.* 61(Suppl. 7), 176–182.

17. Obradovic, Z., Peng, K., Vucetic, S., Radivojac, P., Brown, C. J., Dunker, A. K. (2003) Predicting intrinsic disorder from amino acid sequence. *Proteins* 53(Suppl 6), 566–572.

18. Peng, K., Vucetic, S., Radivojac, P., Brown, C. J., Dunker, A. K., Obradovic, Z. (2005) Optimizing long intrinsic disorder predictors with protein evolutionary information. *J Bioinform Comput Biol* 3(1), 35–60.

19. Linding, R., Russell, R. B., Neduva, V., Gibson, T. J. (2003) GlobPlot: Exploring protein sequences for globularity and disorder. *Nucleic Acids Res* 31(13), 3701–3708.

20. Linding, R., Jensen, L. J., Diella, F., Bork, P., Gibson, T. J., Russell, R. B. (2003) Protein disorder prediction: implications for structural proteomics. *Structure (Camb)* 11(11), 1453–1459.

21. Ward, J. J., McGuffin, L. J., Bryson, K., Buxton, B. F., Jones, D. T. (2004) The DISOPRED server for the prediction of protein disorder. *Bioinformatics* 20(13), 2138–2139.

22. Yang, Z. R., Thomson, R., McNeil, P., Esnouf, R. M. (2005) RONN: the biobasis function neural network technique applied to the detection of natively disordered regions in proteins. *Bioinformatics* 21(16), 3369–3376.

23. Cheng, J., Sweredoski, M., Baldi, P. (2005) Accurate prediction of protein disordered regions by mining protein structure data. *Data Mining Knowledge Discov* 11, 213–222.

24. Pollastri, G., McLysaght, A. (2005) Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics* 21(8), 1719–1720.

25. Coeytaux, K., Poupon, A. (2005) Prediction of unfolded segments in a protein sequence based on amino acid composition. *Bioinformatics* 21(9), 1891–1900.

26. Schlessinger, A., Punta, M., Rost, B. (2007) Natively unstructured regions in proteins identified from contact predictions. *Bioinformatics* 23(18), 2376–2384.

27. Wang, L., Sauer, U. H. (2008) OnD-CRF: predicting order and disorder in proteins using [corrected] conditional random fields. *Bioinformatics* 24(11), 1401–1402.

28. Shimizu, K., Hirose, S., Noguchi, T. (2007) POODLE-S: web application for predicting protein disorder by using physicochemical features and reduced amino acid set of a position-specific scoring matrix. *Bioinformatics* 23(17), 2337–2338.

29. Su, C. T., Chen, C. Y., Ou, Y. Y. (2006) Protein disorder prediction by condensed PSSM considering propensity for order or disorder. *BMC Bioinformatics* 7, 319.

30. Uversky, V. N., Gillespie, J. R., Fink, A. L. (2000) Why are "natively unfolded" proteins unstructured under physiologic conditions? *Proteins* 41(3), 415–427.

31. Zeev-Ben-Mordehai, T., Rydberg, E. H., Solomon, A., Toker, L., Auld, V. J., Silman, I., Botti, S., Sussman, J. L. (2003) The intracellular domain of the Drosophila cholinesterase-like neural adhesion protein, gliotactin, is natively unfolded. *Proteins* 53(3), 758–767.

32. Liu, J., Rost, B. (2003) NORSp: Predictions of long regions without regular secondary structure. *Nucleic Acids Res* 31(13), 3833–3835.

33. Liu, J., Tan, H., Rost, B. (2002) Loopy proteins appear conserved in evolution. *J Mol Biol* 322(1), 53–64.

34. Dosztanyi, Z., Csizmok, V., Tompa, P., Simon, I. (2005) IUPred: web server for

the prediction of intrinsically unstructured regions of proteins based on estimated energy content. *Bioinformatics* 21(16), 3433–3434.

35. Galzitskaya, O. V., Garbuzynskiy, S. O., Lobanov, M. Y. (2006) FoldUnfold: web server for the prediction of disordered regions in protein chain. *Bioinformatics* 22(23), 2948–2949.

36. Callebaut, I., Labesse, G., Durand, P., Poupon, A., Canard, L., Chomilier, J., Henrissat, B., Mornon, J. P. (1997) Deciphering protein sequence information through hydrophobic cluster analysis (HCA): current status and perspectives. *Cell Mol Life Sci* 53(8), 621–645.

37. Vacic, V., Oldfield, C. J., Mohan, A., Radivojac, P., Cortese, M. S., Uversky, V. N., Dunker, A. K. (2007) Characterization of molecular recognition features, MoRFs, and their binding partners. *J Proteome Res* 6(6), 2351–2366.

38. Oldfield, C. J., Cheng, Y., Cortese, M. S., Romero, P., Uversky, V. N., Dunker, A. K. (2005) Coupled folding and binding with alpha-helix-forming molecular recognition elements. *Biochemistry* 44(37), 12454–12470.

39. Bourhis, J., Johansson, K., Receveur-Bréchot, V., Oldfield, C. J., Dunker, A. K., Canard, B., Longhi, S. (2004) The C-terminal domain of measles virus nucleoprotein belongs to the class of intrinsically disordered proteins that fold upon binding to their pohysiological partner. *Virus Res* 99, 157–167.

40. John, S. P., Wang, T., Steffen, S., Longhi, S., Schmaljohn, C. S., Jonsson, C. B. (2007) Ebola virus VP30 is an RNA binding protein. *J Virol* 81(17), 8967–8976.

41. Wootton, J. C. (1994) Non-globular domains in protein sequences: automated segmentation using complexity measures. *Comput Chem* 18(3), 269–285.

42. Kall, L., Krogh, A., Sonnhammer, E. L. (2007) Advantages of combined transmembrane topology and signal peptide prediction – the Phobius web server. *Nucleic Acids Res* 35(Web Server issue), W429–W432.

43. Bornberg-Bauer, E., Rivals, E., Vingron, M. (1998) Computational approaches to identify leucine zippers. *Nucleic Acids Res* 26(11), 2740–2746.

44. Lupas, A., Van Dyke, M., Stock, J. (1991) Predicting coiled coils from protein sequences. *Science* 252(5009), 1162–1164.

45. Baldi, P., Cheng, J., Vullo, A. (2004) Large-scale prediction of disulphide bond connectivity. *Adv Neural Inf Process Syst* 17, 97–104.

46. McGuffin, L. J., Bryson, K., Jones, D. T. (2000) The PSIPRED protein structure prediction server. *Bioinformatics* 16(4), 404–405.

47. Lieutaud, P., Canard, B., Longhi, S. (2008) MeDor: a metaserver for predicting protein disorder. *BMC Genomics* 9(Suppl 2), S25.

48. Chandonia, J. M. (2007) StrBioLib: a Java library for development of custom computational structural biology applications. *Bioinformatics* 23(15), 2018–2020.

49. Longhi, S., Receveur-Brechot, V., Karlin, D., Johansson, K., Darbon, H., Bhella, D., Yeo, R., Finet, S., Canard, B. (2003) The C-terminal domain of the measles virus nucleoprotein is intrinsically disordered and folds upon binding to the C-terminal moiety of the phosphoprotein. *J Biol Chem* 278(20), 18638–18648.

50. Kingston, R. L., Hamel, D. J., Gay, L. S., Dahlquist, F. W., Matthews, B. W. (2004) Structural basis for the attachment of a paramyxoviral polymerase to its template. *Proc Natl Acad Sci USA* 101(22), 8301–8306.

51. Morin, B., Bourhis, J. M., Belle, V., Woudstra, M., Carrière, F., BGuigliarelli, B., Fournel, A., Longhi, S. (2006) Assessing induced folding of an intrinsically disordered protein by site-directed spin-labeling EPR spectroscopy. *J. Phys. Chem. B* 110(41), 20596–20608.

Chapter 19

Protein Secondary Structure Prediction

Walter Pirovano and Jaap Heringa

Abstract

While the prediction of a native protein structure from sequence continues to remain a challenging problem, over the past decades computational methods have become quite successful in exploiting the mechanisms behind secondary structure formation. The great effort expended in this area has resulted in the development of a vast number of secondary structure prediction methods. Especially the combination of well-optimized/sensitive machine-learning algorithms and inclusion of homologous sequence information has led to increased prediction accuracies of up to 80%. In this chapter, we will first introduce some basic notions and provide a brief history of secondary structure prediction advances. Then a comprehensive overview of state-of-the-art prediction methods will be given. Finally, we will discuss open questions and challenges in this field and provide some practical recommendations for the user.

Key words: secondary structure, secondary structure prediction, multiple sequence alignment.

1. Introduction

1.1. How Do We Define a Secondary Structure?

In 1951, Pauling and Corey (1, 2) first suggested the existence of regular conformations of amino acid sequences in globular proteins. In their studies they put forward two possible stable conformations of the backbone: the α-helix and the β-strand. Stability of these elements is mainly gained by the formation of hydrogen bonds between the residue side chains. Later it appeared that some protein parts assume less regular folds. This third class of more unstructured regions is commonly referred to as coil or loop. A secondary structure element can be defined as a consecutive fragment of a protein sequence which corresponds to a local region in the associated protein structure showing distinct geometrical features (*see* **Fig. 19.1**). Generally, about 50% of all protein residues participate in α-helices and β-strands, while the remaining half is more irregularly structured.

Fig. 19.1. Cartoon representation of a protein structure containing all three secondary structure elements: helices, strands (displayed as arrows) and coils (displayed as ropes). The PDB-ID of the protein, a formyl transferase, is 1meo.

The organization in specific secondary structure elements is of crucial importance for the stability of proteins. It is commonly known that when globular proteins fold into their 3D structure, the hydrophobic residues tend to group together in the internal core of the protein (while the hydrophilic residues can mainly be found along the surface). Nonetheless, the hydrophilic nature of the main-chain forms an obstacle for correct folding. This is because the polar nitrogen and oxygen atoms of the main-chain need to form hydrogen bonds, also when located in the protein core where no solvent is available to engage in hydrogen bonding. Fortunately, α-helices and β-sheets perfectly suit this stability requirement, since each main-chain N-atom can associate with a complementary O-atom. In α-helices, hydrogen bonds can be conveniently formed between the main-chain residues of a successive turn, whereas β-strands require the formation of more complex 'parallel' or 'anti-parallel' β-sheets to yield H-bonded interaction between the polar main-chain atoms.

1.2. Importance of Secondary Structure Prediction

In the previous section we briefly described the importance of secondary structure in the protein-folding process. It has been observed that secondary structure elements are formed early on during folding. Their subsequent assembly results in the proteins' initial structural framework (3). As a consequence of this so-called framework model of protein folding, secondary

structure prediction techniques are often implemented in methods that infer protein 3D structures. One of these approaches, called threading, aims at the identification of a template structure that most closely matches a given query structure. Threading techniques thus follow the so-called *inverse* folding problem, where the question is not what three-dimensional structure a given protein sequence will adopt (the folding problem) but what sequence is compatible with a given three-dimensional structure. In most threading implementations a database of tertiary structures (for instance, the Protein Data Bank (PDB) (4)) is scanned and for each fold a pseudo-energy is computed to determine if it is a good match for the query sequence, often in conjunction with its predicted secondary structure (5–9). Also ab initio prediction, where sequence information is used for de novo prediction of a 3D model, has been shown to benefit significantly from reliably predicted secondary structure (10).

In addition to fold recognition, secondary structure prediction has also been successfully integrated in a number of further important bioinformatics tools. These include homology detection programs (11, 12), multiple sequence alignment routines (13, 14) and protein disorder prediction approaches (15). In all these cases, the common thread is that structure is more conserved than sequence. This applies particularly to more distantly related proteins, where evolutionary relatedness might not be discernible at the sequence level anymore but can still be detected at the structural level. In some applications, secondary structure information is used indirectly. For instance, a threading algorithm might not directly use information from secondary structure prediction but employ a technique for remote homology detection that incorporates secondary structure prediction.

1.3. Deciphering Prediction Rules from Residue Patterns

1.3.1. Information Retrieval

The most intuitive and direct way to derive amino acid residue patterns encoding distinct secondary structure topologies is to gather information from sequences for which the corresponding tertiary structure is known. In fact, many analyses using the Protein Data Bank, currently holding over 50,000 solved protein structures, have led to a much better understanding of the principles governing secondary structure formation. Other benefits have come from the use of multiple sequence alignments, which has been of great value for unravelling the evolutionary conservation patterns. Although prediction methods differ in the way they exploit these principles, all of them make use of the observed trends in α-helices, β-sheets, and coils. This means that modern methods all carry out a form of knowledge-based prediction, where prediction rules are learned from information gathered from databases. It should be kept in mind however that none of these rules is infallible and examples are abundant where controversial patterns dominate.

1.3.2. α-Helices   α-Helices are often positioned in proximity of the buried core of the protein. The inner buried face of the helix is therefore usually composed of hydrophobic residues, while the opposite side faces the solvent and thus consists of more hydrophilic residues providing polar interaction with the solvent (in **Fig. 19.2** an amino acid hydrophobicity scale is provided). Given that one complete turn around the central α-helical axis comprises 3.6 residues, the expected sequence pattern is hydrophobic and hydrophilic residues alternating every two residues (*see* **Fig. 19.3**). Variations on this theme can be observed in coiled-coil structures (e.g. so-called leucine zippers) where two or more helices twist along each other's axes. Here the periodicity is slightly different since hydrophobic residues are repeated every seven residues. Some residue types are not seen in helices or parts thereof. An example is proline, which is usually not observed in the central parts as its typical circular side-chain causes a disruption of the helical turn. Nevertheless prolines are able to form hydrogen bonds at least at the N-terminal side of the helix and this explains why they can participate more easily in the first turn of the helix. For similar reasons also other amino acids, such as glycine, serine, or tyrosine, tend to avoid α-helices as they do not contribute to a stable helical conformation.

hydrophobic →

ILE (I)	PHE (F)	VAL (V)	LEU (L)	TRP (W)	MET (M)	ALA (A)	GLY (G)	CYS (C)	TYR (Y)
1.38	1.19	1.08	1.06	0.81	0.64	0.62	0.48	0.29	0.26

→ **hydrophilic**

PRO (P)	THR (T)	SER (S)	HIS (H)	GLU (E)	ASN (N)	GLN (Q)	ASP (D)	LYS (K)	ARG (R)
0.12	−0.05	−0.18	−0.40	−0.74	−0.78	−0.85	−0.90	−1.50	−2.53
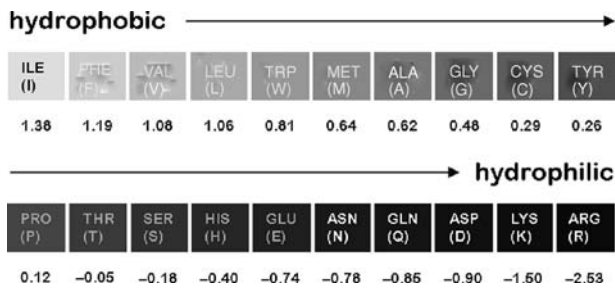
Fig. 19.2. Normalized hydrophobicity scale for amino acids as proposed by Eisenberg et al. (74).

α-helix   β-sheet

edge strand   ○ ● ○ ● ○ ● ○ ●

buried strand   ○ ○ ● ● ● ● ○ ○

○ ● ● ○ ○ ● ● ○ ○ ● ○ ○ ● ●
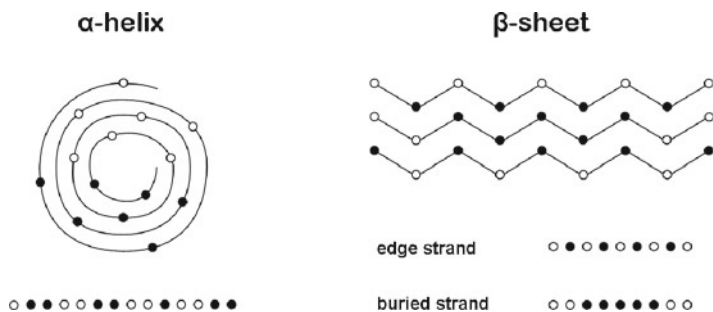
Fig. 19.3. Hydrophobicity patterns for different secondary structures types. Simplified structures of both the α-helix and the β-sheet are given along with a generalized amino acid scheme (white balls indicate hydrophilic amino acids, black balls hydrophobic).

1.3.3. β-Strands

The amino acid patterns observed in β-strands are of a different nature. A tight net, the β-sheet, is formed through hydrogen-bonded main-chain interaction between the strands. Depending on the orientation of the strands, the β-sheet is organized in a parallel or anti-parallel manner. A distinction can be made between strands located in the buried part of the sheet and those that reside at the edges. In buried strands (*see* **Fig. 19.3**), the main-chains typically form hydrogen bonds with the backbone of the neighbouring strands. The two strands at either edge of the β-sheet however have only one neighbouring strand at the inside and must therefore satisfy one half of their main-chain hydrogen bonding requirement with solvent or protein parts not involved in the β-sheet considered. An elegant solution for this problem is seen in the so-called β-barrel fold where the two edge strands are brought in close proximity such that their main-chains can now form mutual hydrogen bonds. Because of their position away from the solvent, buried β-strands typically consist of hydrophobic amino acids. Edge strands often have one face exposed to the solvent, giving rise to an alternating 1-to-1 hydropathical pattern (*see* **Fig. 19.3**) caused by the fact that their side-chains alternatingly protrude into the solvent or into the protein interior.

Unlike helices, strands are able to accommodate 'disruptive' amino acids (such as glycines) giving rise to so-called β-bulges (16, 17) that kink the main-chain but do not disrupt the tight structure formed by the hydrogen bonds. Proline residues do not occur in β-bulges as their main-chain conformation is not consistent with this structure (17). Prolines are preferred constituents of edge-strands, where they can form hydrogen bonds with the solvent.

1.3.4. Loops

As mentioned above, loop regions are mainly located at the protein surface. Unlike α-helices and β-strands, loops do not have a defined structure (in fact they are unstructured) and residue side-chains are usually not involved in hydrogen bond formation. These distinctive properties are partly guaranteed by the occurrence of polar residues such as alanine, glycine, serine, and tyrosine. Glycines are particularly preferred in loops because of their inherent flexibility. Furthermore, prolines are regularly observed in these regions because their rigid and unusual main-chain structure is not an obstacle for hydrogen bonding in loops as opposed to helices and strands.

1.4. Building Up Topology Models

The promising initial results of secondary structure prediction have encouraged many researchers to investigate mechanisms behind the higher order stages of the folding process. Unfortunately, tertiary structure prediction has not yet reached a generally satisfying level and as a result so-called structural genomics initiatives have been instigated to crystallize all representative members

of the protein-folding space. Nevertheless several principles based on secondary structure can be used to predict a helpful topology model of a protein.

- β-α-β motifs, repetitive elements of two β-strands alternated by an α-helix, in more than 95% of all cases give rise to a right-handed chirality.

- More in general, if proteins contain both α- and β-topologies, the helices often cover up a core of β-strands. For modelling purposes it can therefore be useful to consider that β-sheets tend to be 'sandwiched' in between α-helices.

- Transmembrane segments, where part of a protein sticks into the cell membrane, can serve well as anchoring points for a topology model of membrane proteins. Especially hydrophobicity patterns for transmembrane α-helices are quite well recognized and can be quite reliably predicted by one of the state-of-the-art methods. This topic will be described more extensively below.

- Residue conservation, displayed by multiple sequence alignments, can also be informative for a topology model. On the one hand, gapped alignment regions may indicate an unstructured region that will probably correspond to a loop. On the other hand also strongly conserved residues might be comprised in loops: these are then likely to be responsible for the catalytic function of enzymes. The grouping of these catalytic residues can help to reconstruct the active site of the model.

2. Overview of Secondary Structure Prediction Methods

2.1. The Early Methods

The first attempts to predict secondary structure were made in the 1970s and involved only single sequences. Most early methods basically relied on a straightforward statistical analysis of sequence composition underlying the three secondary structure elements. The main challenge in the early days was that only relatively few structures were experimentally verified, such that the statistics could only be gathered from relatively few observations. As a result, the early prediction accuracies were in between 50% and 60% (18). It is important to keep in mind that based on the overall distributions of helices, strands, and coils (of about 30%, 20% and 50%, respectively), a random predictor will give an accuracy of around 40%.

The early approach proposed by Nagano (19) explored the likelihood of residue pairs within specific secondary structures to show short-range interactions (up to six residues in sequence). The interactions were linearly combined to calculate interacting

residue propensities for each secondary structure element. Also the Chou–Fasman method (20) based its predictions on differences in residue type composition for the various secondary structure states. Neighbouring residues were checked for helices and strands and predicted types were selected according to the higher scoring preference. Lim (21) instead developed an extensive set of stereo-chemical prediction rules for α-helices and β-sheets, based on their observed frequencies in globular proteins. For some time this was the most accurate method (reaching 56% accuracy) due to the valuable hydrophobicity rules applied here (22). Nevertheless the Chou–Fasman method was more popular due to its public availability. A breakthrough in the field was obtained by the GOR method (23), which uses amino acid frequencies within a 17-residue window to discriminate between secondary structure types. Further refinement of the method in subsequent versions, including some post-processing steps (24), led to an accuracy of around 65%: a notable success for a method based solely on single sequence prediction.

2.2. A Note on Current Prediction Methods

Also recent methods adopt the window approach that includes a local stretch of amino acids around a central position to predict the secondary structure state at that position. Training algorithms, when properly applied, then help to decipher the prediction rules. Whereas early methods relied on a straightforward statistical analysis of sequence composition underlying the three secondary structure elements, modern methods adopt more sophisticated machine learning protocols for gleaning the sequence signals associated with the secondary structure types. However, the powerful combination of a large number of crystallized protein structures for better training techniques and the use of multiple sequence alignments have been of great advantage in recent prediction methods. The latter idea was first exploited by Zvelebil et al. in 1987 (25) and its success was later also confirmed by Levin et al. (26) and Rost and Sander (27). As a consequence, nowadays all state-of-the-art methods (including those described below) use multiple alignment information to better incorporate the evolutionary signals of residue and secondary structure conservation. Another crucial development in the field concerns the usage of computational neural networks for secondary structure prediction. The earliest published method (28) appeared when neural network computing was in its infancy – only 2 years after the initial publication by Rumelhart et al. (29). After the first successful neural network implementation by Qian and Sejnowski, prediction algorithms based on other computational formalisms were developed, the most important of which include k-Nearest-Neighbour approaches, Hidden Markov Model (HMM) methods and Consensus approaches. Although each of these techniques have their own distinct advantages, over the past several years neural

nets have turned out to be the most successful. As a result alternative approaches also have converged on neural nets, for instance by merging them into their original strategy.

The next section gives an overview of state-of-the-art methods that are exclusively or partly based on neural networks. For most of them a Web server is available of which the addresses are given in **Table 19.1**.

Table 19.1
Overview of state-of-the-art secondary structure predictors and their Web sites

Name	Web site
PHD/PHDpsi	http://www.predictprotein.org/
PSIPRED	http://bioinf.cs.ucl.ac.uk/psipred/
PROF (king)	http://www.aber.ac.uk/~phiwww/prof/
SSpro	http://scratch.proteomics.ics.uci.edu/
Porter	http://distill.ucd.ie/porter/
APSSP2	http://www.imtech.res.in/raghava/apssp2/
SAM-T06	http://www.soe.ucsc.edu/research/compbio/SAM_T06/T06-query.html/
YASPIN	http://www.ibi.vu.nl/programs/yaspinwww/
Jpred (v3)	http://www.compbio.dundee.ac.uk/jpred/

2.3. State-of-the-Art Methods: The Power of Neural Networks

2.3.1. Principles of Neural Networks

Neural networks are complex machine-learning algorithms that are based upon non-linear statistics. They are organized as interconnected layers of input and output units and can also contain intermediate unit layers (for a review, see Ref. (30)). Each unit in a layer receives information from one or more other connected units, as if it receives an electrical stimulus being a live neuronal cell, and determines its output signal based on the weights of the input signals. A neural net can be regarded as a black box, which operates as a result of the specific weights of the internal connections connecting the units and the function used in each unit to convert the input signals into an output signal. In practice, often a simple step function associated with a threshold value is used to determine the output signal. The training protocol followed to optimize the grouping of a set of input patterns into a set of output patterns by adjusting the weights is therefore crucial. Training normally starts with a set of random weights, after which in a so-called *forward pass* the outputs are calculated and the error at the output units determined. Then, in a *backward pass*, the output unit error is used to alter the weights on the output units. The error at the hidden nodes is calculated by *backpropagating* the error at the

output units through the weights, while the weights on the hidden nodes are adjusted in turn using these values. For each data pair to be learned, a forward pass and backward pass are performed. This scenario is iterated until the error is at a low enough level (or a maximum number of iterations is reached). Generally, neural net-based secondary structure prediction methods employ a sliding window in order to train the network with signals corresponding to a single secondary structural element (*see* **Fig. 19.4**). Care must be taken not to overtrain the network, as this leads to a network that is not able to extrapolate the patterns it has seen during the training phase to new unseen sequences. As stated above, methods that incorporate neural networks got the upper hand but also other techniques which attempted to combine neural networks with other powerful schemes will be described.
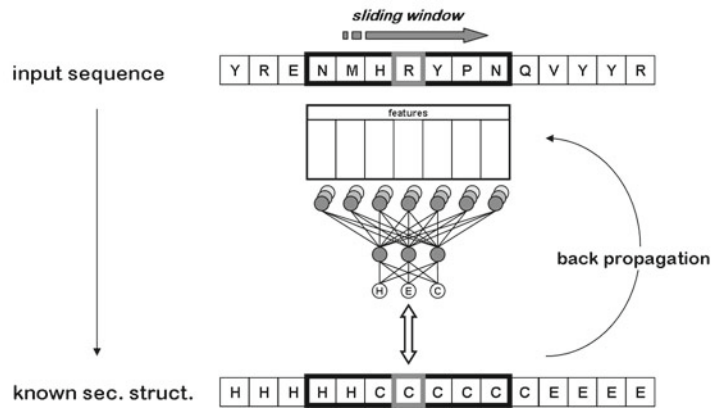
Fig. 19.4. Schematic representation of a sliding window approach used to train a neural network. A window, for which each time the middle position will be predicted, is slid over the sequence. The trained neural network converts the window information (here 'NMHRYPN') into a prediction (here 'C' representing coil) for the middle residue (here 'R'). The neural network depicted represents a single sequence prediction. Modern methods typically use a multiple sequence alignment as input, which is then converted into a profile comprising a frequency table of the amino acids appearing at each alignment position.

2.3.2. PHD/PHDpsi

These two methods share the same basic neural network technique and are therefore covered together in this section. The initial PHD method (27) was the first approach that combined database searching and prediction from multiple sequence alignment using the neural network formalism, and these benefits resulted in an accuracy for the first time surpassing 70%. In the classical PHD approach, first the query sequence is fed to BLAST (31) in order to retrieve homologous sequences from the SWISS-PROT (32) database. These are subsequently aligned using the MAX-HOM alignment program (33). The resulting multiple sequence alignment is then converted to a profile, which is passed on to the

core of the method: a three-layered neural network. In the first layer a 13-residue window slides over the multiple alignment profile and produces a three-state raw secondary structure prediction. The next network layer re-evaluates the raw output now using a 17-residue sliding window and attempts to correct unrealistic solutions provided by the previous network. A consequence of the neural network training protocol is that prediction errors might arise due to training biases rendering adjustment of the three-state probabilities crucial. The third network layer in the PHD method therefore represents a so-called jury network, where the predictions of a number of independently trained two-layered networks are converted into a final prediction.

The more recent PHDpsi method (34) exploits the advantages coming from increased protein database sizes and improved homology detection methods into the original approach. Following other methods in the field (see below), the PSI-BLAST method (35, 36) is used here to find more distantly related sequences, resulting in improved multiple alignment reliability.

2.3.3. PSIPRED

Another popular method is PSIPRED (37). Similarly to PHDpsi it invokes PSI-BLAST to gather additional sequence information for the given query sequence. However the position-specific sequence profiles (PSSMs) obtained by PSI-BLAST are directly taken as input for the neural network. The neural network architecture itself is less complex that that of PHD and only consists of two layers. First a PSI-BLAST sequence profile of window length 15 is passed to the first layer for an initial prediction. Then a second layer filters these raw results and determines the final output. Both the usage of local sequence alignments in the first step and a rather straightforward but effective training protocol for the neural networks has shown to be of great advantage for the prediction accuracy. As a consequence, the wide popularity of the method is mainly due to its easy usage and high overall prediction performance.

2.3.4. PROF (King)

The PROF (king) algorithm (38) follows a complicated scheme where multi-staged classifiers are used as input to a three-layered neural network architecture combined with linear discrimination. The idea behind the approach is to use as much relevant information as possible using independently trained algorithms. As a first step of the PROF (king) method, GOR formalisms (see above) are 'learned' and used as classifiers for the neural network. As a next step, homologous sequences to the query are obtained using PSI-BLAST. Both the PSSM containing local information and profiles obtained after global alignment constitute a second group of classifiers for the neural net. As a third step, the outputs are given to a subsequent neural network layer which is combined with linear discriminant analysis. In the final step, additional attributes are used to produce new classifiers: the hydrophobic moment in helices or strands, the

fractions of several hydrophilic residues, and the secondary structure fractions determined in the preceding step. A last neural network layer then predicts the final secondary structure states.

2.3.5. SSpro/Porter

SSpro (39) is among the leading secondary structure prediction algorithms in terms of accuracy. The method takes advantage of a sophisticated protocol coined bi-directional recurrent neural network (BRNN). Of particular interest is the effort made in this method to include long-range interactions, which is considered a challenging task for prediction techniques. A solution provided by the BRNN architecture is the use of a basic sliding window plus an additional two windows that slide in from opposite sides (each covering the entire sequence) at each basic window position. Updated versions of the program (40) benefit from the use PSI-BLAST profiles as input for the BRNN whereas the original version used BLAST profiles. As another update, an additional SSpro flavour emerged (SSpro8) able to predict eight-state secondary structure instead of the classical three-state prediction, where the grouping in eight states follows the DSSP-concept (see Section 3).

In this section also the prediction program Porter (41) should be mentioned, which is a further development of the SSpro toolkit. The new elements implemented in Porter include an extension of the amino acid alphabet describing the input profiles, improved incorporation of long-range distance information, a more sophisticated filtering, and the use of more extensive training sets. Although only a slight improvement is observed relative to SSpro, Porter is the current top performer, attaining an accuracy of about 80% according to the EVA assessment server (see below).

2.3.6. PSSP/APSSP/ APSSP2

An alternative to neural networks are *k*-nearest-neighbour (kNN) techniques. Here an initial training phase is applied on sequence fragments from a database containing solved structures. For each database fragment, also called 'exemplar', the central residue is labelled according to the observed secondary structure state. Next, a sliding window approach is applied to the query sequence: each window is compared with the exemplars and the *k* most similar fragments are defined. From all selected *k* secondary structure labels (typically between 25 and 100) three-state propensities are calculated resulting in the final assessment. Neural network methods currently outperform kNN approaches and perhaps the best explanation for this is the reduced predictive power of kNNs in cases where no closely related solved structures are available. A number of kNN-based approaches integrate neural network systems in an attempt to obtain a synergic effect from this combination.

An interesting method that combines a kNN strategy with neural networks is PSSP (42). As a first step of the method, a 'classical' nearest-neighbour classifier searches for the most similar exemplars in the database. However, given the huge number of

protein structures that has become available, this step would be extremely slow. To alleviate this problem, only short sequence stretches (of three residues) that are identical to the database sequence are considered. As a second step, a standard neural network is implemented, which complements the limited fold coverage of the kNN. The third step evaluates the prediction probabilities coming from the first two steps and determines the final secondary structure state. A final refinement is subsequently performed using a so-called structure to structure approach. Two advanced versions of the program, APSSP (43) and APSSP2 (44), have had some modifications effected as compared to the original method: the original kNN step is in both these methods replaced by an example-based learning (EBL) technique. Furthermore, instead of the neural network approach adopted in the original protocol, the APSSP algorithm incorporates the method Jnet (see Section 2.3.9) as a second prediction step, whereas the APSSP2 method maintains the neural network approach, albeit the input is a multiple sequence alignment (created with PSI-BLAST) replacing the single sequence approach of the original PSSP method.

2.3.7. SAM-T99/SAM-T02/ SAM-T06

The SAM package contains a collection of sequence analysis tools that take advantage of Hidden Markov Models (HMMs). The SAM-T series includes a structure prediction method that combines HMMs with neural nets to reliably assign secondary structure. An HMM is a probabilistic model that contains a series of states linked together by state transitions. Compared to the standard Markov model, the internal parameters are unknown (hidden) and the challenge is to unravel these from the observed output. HMMs are implemented in a variety of areas, including speech recognition and weather forecasting. A successful application in bioinformatics is HMM-based remote homology searching (45, 46). This is also the first step of the SAM-T99 (46, 47) structure prediction protocol, where an HMM is used to search iteratively for related sequences in a protein database. From the sequences found, a proper multiple alignment is constructed which is then fed as input to a neural network (as seen in a number of related approaches). In SAM-T02 (48), the follow-up, the original protocol was improved by including predicted secondary structure features in the scoring functions of other techniques included in the package. The more recent version, SAM-T06 (49), refined the iterative strategies and the local structure prediction method. Moreover, the package now also includes contact prediction and full-coordinate 3D structure prediction, which can be performed in a fully automated manner.

2.3.8. YASPIN

The YASPIN method (50) also employs HMMs and neural networks, though using a rather different strategy. The method uses a single feed-forward perceptron network which receives (1) a 15-residue PSSM window obtained from PSI-BLAST and (2) an

additional unit indicating protein chain terminations. These are passed to the output layer that provides a seven-state prediction: helix beginning (Hb), helix (H), helix end (He), strand beginning (Eb), strand (E), strand end (Ee) and coil (C). The helical and stranded flanking regions are taken as separate entities because these regions often exhibit discernible sequence patterns. The strength of the neural net implementation of YASPIN resides both in its simplicity and in the effort to capture position-specific signals relating to capping regions of the structural elements. Finally, the seven-state output of the neural network is filtered through an HMM, which applies the Viterbi algorithm to optimally predict a three-state secondary structure for the given query. A specific strength of the method is its ability to predict β-strands with high accuracy.

2.3.9. Jpred (Jnet)   A different class of methods are consensus methods, which were a consequence of the realization that no single strategy will be able to outperform all others for the whole protein range. On the one hand, the different methodologies and systems behind each method will yield alternative predictions, where it is likely that some strategies will be more effective than others depending on the particular case considered. On the other hand, methods are trained using different techniques and training sets which can cause an undesired bias in a methods' performance. A solution to these problems is provided by consensus approaches which basically combine the outcomes of several state-of-the-art methods. A simple majority voting scheme can be adopted to determine for each position the most likely (i.e. the most observed) secondary structure state. The idea is analogous to the classical use of three clocks on board of historic navy vessels, such that a failing clock could be detected whenever the other two continued to work properly. It should be stressed however that consensus methods can become biased easily if they cover the methodology space in an unbalanced manner, for instance by including a number of similar methods that produce correlated results. Nevertheless, consensus approaches are promising in this field as they are less affected by chance effects involved in single methods.

A widely used consensus method for secondary structure prediction is Jpred. The original method was proposed in 1998 (51). Homologous sequences to the query were retrieved using BLAST and subsequently aligned with ClustalX (52). The resulting multiple sequence alignment was then fed to several methods, including PHD (53), PREDATOR (54) and NNSSP (55). A simple majority voting scheme determined the final output, albeit if no winning prediction could be declared, the PHD result was taken. A few years after the introduction of the original Jpred method, the Jnet algorithm was incorporated (56) in place of the various prediction methods included in the original Jpred implementation, inducing a substantial change in the method. The Jnet algorithm itself can be considered a consensus method but only of assorted neural

networks, akin to the PHD method (see above), rather than of different prediction techniques. This development made the Jpred method less dependent on the developments of other structure predictors. As a first step in the new Jpred approach, for each query sequence PSI-BLAST is run after which a filtering step removes redundant sequences. In the second step, alignment profiles are calculated using different strategies and given to the neural network ensemble. Recently an updated version of the prediction server has become available, Jpred 3 (57), which incorporates the Jnet v2.0 algorithm. The strategy now only uses PSI-BLAST PSSMs and hidden Markov model profiles from HMMER (45), albeit the complexity of the neural network has been increased.

3. Assessing Prediction Accuracy

3.1. How Do We Assess Predictions Using Experimentally Solved Protein Structures

At present, the main source of reference secondary structures is derived from the Protein Data Bank (PDB). The PDB is a continually updated database of all available experimentally derived three-dimensional protein structures. The PDB data is in the form of three-dimensional coordinate files, which can be parsed to extrapolate the secondary structure elements. The most commonly used secondary structure assignment program is DSSP program, which is based upon hydrogen bonding patterns between main-chain atoms. The program is used to produce the DSSP database (Dictionary for Secondary Structure of Proteins) (58). A more recent development is the DSSPcont protocol (59), which for each residue position provides a secondary structure likelihood. Another popular assignment program is STRIDE (60), which incorporates a knowledge-based assignment technique.

There are two ways in which one can approach the issue of protein structure prediction accuracy. The first way is from the developer's point of view, where the interest is in how well a method can react to a challenging problem. This question is addressed in the CASP and CAFASP meetings. The second way is from the user's point of view, so mainly molecular biologists. Here the interest is in which method is overall better, so that misleading results can be minimized. In this case, the EVA team has set up a server (http://cubic.bioc.columbia.edu/eva) that continually evaluates the accuracy of prediction programs that are registered to it.

3.2. CASP and CAFASP

The CASP experiments (Critical Assessment of techniques for protein Structure Prediction) are organized to assess all types of methods for predicting protein structure and discuss the current advances in the field as well as required future directions and improvements for problematic areas in the field. The first CASP

meeting was held in 1994 and has since been held bi-annually in different locations around the globe. The most recent experiment was CASP7 in November 2006 (for an overview see (61)). The CASP experiments put together protein sets of which the solved structural information has not been released yet and challenge all methods that take part to do their best predictions. This way, every 2 years the best methods are evaluated using newly solved proteins that have not been seen by any of the contenders. In addition, closely linked to CASP are the CAFASP experiments (Critical Assessment of Fully Automated Structure Prediction), which use the CASP protein sets to test automatic prediction servers that are available online for researchers to use. The fifth and latest CAFASP experiment was held together with CASP 7 in 2006. These experiments are mainly aimed to give an assessment of what online automatic tools are currently available to researchers and to determine how good they are by assessing them on equal terms.

3.3. The EVA Automatic Evaluation of Prediction Methods

The EVA server (EValuation of Automatic protein structure prediction) is a Web-based assessment tool that has been performing evaluations of the accuracies of its member structure prediction servers since June 2000 (62). The assessment comprises four different categories of structure prediction: (a) comparative modelling, (b) fold recognition and threading, (c) secondary structure prediction and (d) inter-residue contact prediction. The EVA server updates its reference secondary structure data sets on a daily basis by retrieving the most up-to-date experimentally determined structures from the PDB and employing the DSSP program (58) to parse the 3D coordinates into secondary structure chains. The amino acid sequences of the newly acquired proteins are then submitted to the member secondary structure prediction servers and their predictions are evaluated with reference to those generated by the DSSP program. At present (June 2008), 19 secondary structure prediction server-members are assessed by EVA and freely available on the EVA Web site at Columbia University (http://cubic.bioc.columbia.edu/eva/) and also mirrored at the UCSF (http://eva.compbio.ucsf.edu/~eva/) and at the CNB Madrid (http://pdg.cnb.uam.es/eva/).

4. Methods for Transmembrane Topology Prediction

Membrane proteins form a distinct topological class due to the presence of one or more transmembrane (TM) sequence segments. In contrast to globular proteins where all possible mutual orientations of individual structural elements are in principle possible, the TM segments of membrane proteins are subjected to severe restrictions imposed by the lipid bilayer of the cell membrane.

There is a considerable lag in structures available for membrane proteins compared to the large and vastly growing numbers of soluble proteins due to limitations of current crystallization techniques. In fact not even 2% of all solved structures deposited in the PDB database show a membrane topology (4, 63), although they constitute around 20–30% of the total number of sequenced proteins (64). The most frequently observed secondary structure in transmembrane segments is the α-helix, but also transmembrane structures based on β-strands that constitute a β-barrel have been encountered.

Fortunately, the location of the transmembrane segments in the primary sequence is relatively easy to predict due to the rather strong tendency of certain hydrophobic amino acid types with their special physico-chemical properties to occur in membrane-spanning regions. A number of methods are available for prediction of α-helical TM segments, achieving accuracies above 70%. Nonetheless, just like in globular proteins, it is a hard task to determine the exact location of the structural element compared to the number of segments. Two widely used methods that employ HMMs for TM topology prediction are HMMTOP (65) and TMHMM (66). Another HMM-based method, called Phobius (67), is able to also discriminate between TM segments and signal peptides. This method performs rather well on proteins containing a signal peptide since other methods often confuse signal peptide patterns with transmembrane patterns. PolyPhobius (68) follows the Phobius protocol but now includes homology-extended information from PSI-BLAST for prediction. The MEMSAT (69) method also exploits evolutionary information from PSI-BLAST, albeit it implements neural networks. The claim here is that, akin to secondary structure prediction methods, the evolutionary information represented in a sequence alignment is incorporated more optimally in neural nets. There are only a few prediction methods for β-barrel TM segments. Among these is TBBpred (70), a neural network-based method which also incorporates a support vector machine. A major problem for training is the severely limited number of experimentally solved β-barrel TM structures available to train on.

5. Integrating Secondary Structure Information into Multiple

5.1. Sequence Alignment

An important application of secondary structure prediction is the integration of secondary structure information in multiple sequence alignment techniques. Given the gap in between the availability of sequence and structure data, most sequence analysis protocols have to resort to predicted structural data. In the case of multiple alignment, incorporating secondary structure information

relies on the fact that in divergent evolution molecular structures are more conserved than their corresponding coding sequences. This implies that secondary structures can be used to pinpoint homologous sequence regions if the evolutionary traces in the sequences themselves have become obscure. Owing also to the fact that modern secondary structure prediction techniques approach 80% in prediction accuracy, yielding four out of five residues correctly predicted on average, the evolutionary advantage of using secondary structure information now outweighs the chance of misprediction.

Following the multiple alignment method PRALINE (13, 71, 72), the SPEM (14) method and the recently developed PROMALS (73) technique are able to complement progressive alignment with secondary structure prediction in an attempt to improve the alignment accuracy. While PRALINE and SPEM use a standard progressive alignment protocol, PROMALS adopts Bayesian consistency to fill its library with the posterior decoding of a pair hidden Markov model.

Further, the PRALINE method incorporates an iterative scheme to optimize alignment and secondary structure prediction, where secondary structure prediction can be performed using an initial multiple alignment, after which the next multiple alignment is created with the help of the predicted secondary structure. Alternatively, for each sequence, homologous sequences can be obtained by using PSI-BLAST, followed by secondary structure prediction.

6. Challenges for the Field and Practical Considerations

6.1. Is There an Upper Limit?

The current state-of-the-art sustained prediction accuracy is 80% as attained by the Porter method (see above). The most important feature to which this top accuracy can be attributed is the incorporation of long-range interactions by means of two extra windows slid over the sequence in opposite directions. This mechanism is likely to particularly aid the prediction of β-strands, which generally turn out to be the least predictable secondary structure element due to their context dependency. Since β-strands embedded in the core of a β-sheet need two adjacent β-strands to satisfy the hydrogen bonding requirements (see above), the side-chains of such a β-strand together with those of its neighbouring strands need to accommodate the arrangement of the three strands in either a parallel or anti-parallel fashion. Sequence patterns associated with such interactions can be effectively traced by the triplet window approach as implemented in the Porter method. Some other local protein structures might also be amenable to this technique. An example is interacting α-helices in the β-barrel fold, where each helix interacts with two

neighbouring helices. However, other structures that cannot be approximated by three colinear stretches of protein sequence are not likely to be predicted accurately by the Porter prediction scheme.

It is interesting to note that the available prediction programs do not assess the likelihood of the predicted secondary structure with respect to the tertiary structure. For example, a single predicted β-strand is highly unlikely, and so is a predicted alternating β/α structure starting with a N-terminal helix.

Further improvements in accuracy beyond 80% might result from more flexible schemes to incorporate long-range interactions, while gains might also be anticipated from techniques that combine secondary structure prediction and higher-order structure akin to the framework model of protein folding (see above), possibly in an iterative fashion.

6.2. Prediction Accuracy and Multiple Alignment Quality

A crucial factor in the prediction of secondary structure is the quality of the input multiple sequence alignment. In fact, where 15 years of development of secondary structure prediction techniques has lead to an increase of about 7% in prediction accuracy, alternative alignments obtained from different alignment programs lead to secondary structure prediction accuracies varying easily over 20%. For optimal alignment it is important to carefully select an appropriate set of orthologous sequences that can be trusted to fold into the same secondary structural elements. The PSI-BLAST homology searching program is currently the most widely used method for gathering a set of sequences orthologous to a given query sequence. Sources of error are overly similar sequences that will lead to biases in the sequence profile abstracted from the multiple alignment, while distantly related sequences might lead to misalignment. Another complicating factor for assembling a set of orthologous sequences is the occurrence of orphan sequences that have no relatives in current sequence databases. A consequence is reduced evolutionary information in the multiple alignment that can be reduced to a single sequence in extreme cases, or the inclusion of non-homologous sequences that are unlikely to have similar tertiary structures and comprise a different secondary structure.

It should be stressed that even if an optimal set of orthologous sequences is assembled and aligned exactly according to their evolutionary relationships, slight differences in the length of secondary structure elements in the various orthologous sequences will lead to alignment where the flanking regions of matched helices and strands will be rugged, resulting in noise that will negatively affect the delineation of the secondary structure elements of the query sequence. The intricate evolutionary relationships between orthologous sequences and associated structural variation is believed to be a major burden for optimal prediction

based upon multiple alignment information. The development of filtering techniques for homologous regions that will lead to more consistently matched flanking regions might aid the prediction accuracy. However, for higher order structure prediction based upon assembling predicted secondary structure elements, it is more permissible to mispredict the flanking regions of the secondary structures than to miss a single complete helix or strand.

6.3. Practical Recommendations

A number of prediction techniques report so-called reliability indices accompanying the putative secondary structure elements, indicating to what extent the user might trust the predictions. This allows the user to select the most reliably predicted secondary structure elements, which generally are predicted with higher accuracy than regions with lower reliability indices.

It is also important to include a number of different state-of-the-art methods in the prediction of secondary structure. In case of inconsistencies it is advisable to assemble a consensus prediction where preference might be given to the most reliable prediction methods. Reliability indices can aid the effective weighting of prediction accuracies of the methods included. It is also recommended to carefully select a set of putative orthologous sequences and attempt a number of different multiple alignment techniques. As mentioned above, multiple alignment differences can lead to more varying predictions than obtained as a result of different prediction techniques.

Before finalising a prediction, the structural implications of the predicted secondary structure elements should be checked. It can be insightful to include threading techniques in this process, as those algorithms might aid the tertiary structural and functional annotation of a given query sequence, and in turn provide information as to the most crucial secondary structure elements.

References

1. Pauling, L., Corey R. B., Branson, H. R. (1951) The structure of proteins; two hydrogen-bonded helical configurations of the polypeptide chain. *Proc Natl Acad Sci USA* 37, 205–211.
2. Pauling, L., Corey, R. B. (1951) Configurations of polypeptide chains with favored orientations around single bonds: two new pleated sheets. *Proc Natl Acad Sci USA* 37, 729–740.
3. Goldenberg, D. P., Frieden, R. W., Haack, J. A., Morrison, T. B. (1989) Mutational analysis of a protein-folding pathway. *Nature* 338, 127–132.
4. Berman, H. M., et al. (2000) The protein data bBank. *Nucl Acids Res* 28, 235–242.
5. Russell, R. B., Copley, R. R., Barton, G. J. (1996) Protein fold recognition by mapping predicted secondary structures. *J Mol Biol* 259, 349–365.
6. Rost, B., Schneider, R., Sander, C. (1997) Protein fold recognition by prediction-based threading. *J Mol Biol* 270, 471–480.
7. Koretke, K. K., Russell, R. B., Copley, R. R., Lupas, A. N. (1999) Fold recognition using sequence and secondary structure information. *Proteins* Suppl 3, 141–148.
8. Zhou, H., Zhou, Y. (2004) Single-body residue-level knowledge-based energy score combined with sequence-profile and secondary structure information for fold recognition. *Proteins* 55, 1005–1013.

9. Jones, D. T. (1999) GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J Mol Biol* 287, 797–815.

10. Skolnick, J., Kolinski, A., Ortiz, A. R. (1997) MONSSTER: a method for folding globular proteins with a small number of distance restraints. *J Mol Biol* 265, 217–241.

11. Hargbo, J., Elofsson, A. (1999) Hidden Markov models that use predicted secondary structures for fold recognition. *Proteins* 36, 68–76.

12. Soding, J. (2005) Protein homology detection by HMM-HMM comparison. *Bioinformatics* 21, 951–960.

13. Simossis, V. A., Heringa, J. (2005) PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. *Nucl Acids Res* 33, W289–W294.

14. Zhou, H., Zhou, Y. (2005) SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures. *Bioinformatics* 21, 3615–3621.

15. Ward, J. J., et al. (2004) Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *J Mol Biol* 337, 635–645.

16. Richardson, J. S., Getzoff, E. D., Richardson, D. C. (1978) The beta bulge: a common small unit of nonrepetitive protein structure. *Proc Natl Acad Sci USA* 75, 2574–2578.

17. Chan, A. W., Hutchinson, E. G., Harris, D., Thornton, J. M. (1993) Identification, classification, and analysis of beta-bulges in proteins. *Protein Sci* 2, 1574–1590.

18. Kabsch, W., Sander, C. (1983) How good are predictions of protein secondary structure? *FEBS Lett* 155, 179–182.

19. Nagano, K. (1973) Logical analysis of the mechanism of protein folding. I. Predictions of helices, loops and beta-structures from primary structure. *J Mol Biol* 75, 401–420.

20. Chou, P. Y., Fasman, G. D. (1974) Conformational parameters for amino acids in helical, beta-sheet, and random coil regions calculated from proteins. *Biochemistry* 13, 211–222.

21. Lim, V. I. (1974) Structural principles of the globular organization of protein chains. A stereochemical theory of globular protein secondary structure. *J Mol Biol* 88, 857–872.

22. Schulz, G. E. (1988) A critical evaluation of methods for prediction of protein secondary structures. *Ann Rev Biophys Biophys Chem* 17, 1–21.

23. Garnier, J., Osguthorpe, D. J., Robson, B. (1978) Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *J Mol Biol* 120, 97–120.

24. Garnier, J., Gibrat, J. F., Robson, B. (1996) GOR method for predicting protein secondary structure from amino acid sequence. *Methods Enzymol* 266, 540–553.

25. Zvelebil, M. J., Barton, G. J., Taylor, W. R., Sternberg, M. J. (1987) Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *J Mol Biol* 195, 957–961.

26. Levin, J. M., Pascarella, S,, Argos, P., Garnier, J. (1993) Quantification of secondary structure prediction improvement using multiple alignments. *Protein Eng* 6, 849–854.

27. Rost, B., Sander, C. (1993) Prediction of protein secondary structure at better than 70-percent accuracy. *J Mol Biol* 232, 584–599.

28. Qian, N., Sejnowski, T. J. (1988) Predicting the secondary structure of globular-proteins using Neural Network Models. *J Mol Biol* 202, 865–884.

29. Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986) Learning representations by back-propagating errors. *Nature* 323, 533–536.

30. Minsky, M., Papert, S. (1988) *Perceptrons*. MIT Press, Cambridge, MA, USA.

31. Altschul, S. F., et al. (1990) Basic local alignment search tool. *J Mol Biol* 215, 403–410.

32. Bairoch, A., Boeckmann, B. (1991) The SWISS-PROT protein sequence data bank. *Nucl Acids Res* 19(Suppl), 2247–2249.

33. Sander, C., Schneider, R. (1991) Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins* 9, 56–68.

34. Przybylski, D., Rost, B. (2002) Alignments grow, secondary structure prediction improves. *Proteins* 46, 197–205.

35. Altschul, S. F., Koonin, E. V. (1998) Iterated profile searches with PSI-BLAST – a tool for discovery in protein databases. *Trends Biochem Sci* 23, 444–447.

36. Altschul, S. F., et al. (1997) Gapped BLAST and PSI-BLAST, a new generation of protein database search programs. *Nucl Acids Res* 25, 3389–3402.

37. Jones, D. T. (1999) Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol* 292, 195–202.

38. Ouali, M., King, R. D. (2000) Cascaded multiple classifiers for secondary structure prediction. *Protein Sci* 9, 1162–1176.

39. Baldi, P., et al. (1999) Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics* 15, 937–946.

40. Pollastri, G., Przybylski, D., Rost, B., Baldi, P. (2002) Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins* 47, 228–235.

41. Pollastri, G., McLysaght, A. (2005) Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics* 21, 1719–1720.

42. Raghava, G. P. S. (2000) in *CASP 4*, pp. 75–76.

43. Raghava, G. P. S. (2002) in *CASP 5*, p. 132.

44. Raghava, G. P. S. (2002) in *CASP 5*, p. 133.

45. Eddy, S. R. (1998) Profile hidden Markov models. *Bioinformatics* 14, 755–763.

46. Karplus, K., Barrett, C., Hughey, R. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics* 14, 846–856.

47. Karplus, K., et al. (1999) Predicting protein structure using only sequence information. *Proteins* Suppl 3, 121–125.

48. Karplus, K., et al. (2003) Combining local-structure, fold-recognition, and new fold methods for protein structure prediction. *Proteins* 53(Suppl 6), 491–496.

49. Shackelford, G., Karplus, K. (2007) Contact prediction using mutual information and neural nets. *Proteins* 69(Suppl 8), 159–164.

50. Lin, K., Simossis, V. A., Taylor, W. R., Heringa, J. (2005) A simple and fast secondary structure prediction method using hidden neural networks. *Bioinformatics* 21, 152–159.

51. Cuff, J. A., et al. (1998) JPred: a consensus secondary structure prediction server. *Bioinformatics* 14, 892–893.

52. Thompson, J. D., et al. (1997) The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucl Acids Res* 25, 4876–4882.

53. Rost, B. (1996) PHD: predicting one-dimensional protein structure by profile-based neural networks. *Methods Enzymol* 266, 525–539.

54. Frishman, D., Argos, P. (1997) Seventy-five percent accuracy in protein secondary structure prediction. *Proteins* 27, 329–335.

55. Salamov, A. A., Solovyev, V. V. (1995) Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *J Mol Biol* 247, 11–15.

56. Cuff, J. A., Barton, G. J. (2000) Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins* 40, 502–511.

57. Cole, C., Barber, J. D., Barton, G. J. (2009) The Jpred 3 secondary structure prediction server. *Nucl Acids Res* 36, W197–W201.

58. Kabsch, W., Sander C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22, 2577–2637.

59. Andersen, C. A., Palmer, A. G., Brunak, S., Rost, B. (2002) Continuum secondary structure captures protein flexibility. *Structure* 10, 175–184.

60. Heinig, M., Frishman, D. (2004) STRIDE: a web server for secondary structure assignment from known atomic coordinates of proteins. *Nucl Acids Res* 32, W500–W502.

61. Moult, J., et al. (2007) Critical assessment of methods of protein structure prediction-Round VII. *Proteins* 69(Suppl 8), 3–9.

62. Koh, I. Y., et al. (2003) EVA: evaluation of protein structure prediction servers. *Nucl Acids Res* 31, 3311–3315.

63. Tusnady, G. E., Dosztanyi, Z., Simon, I. (2005) PDB_TM: selection and membrane localization of transmembrane proteins in the protein data bank. *Nucl Acids Res* 33, D275–D278.

64. Wallin, E., von Heijne, G. (1998) Genome-wide analysis of integral membrane proteins from eubacterial, archaean, and eukaryotic organisms. *Protein Sci* 7, 1029–1038.

65. Tusnady, G. E., Simon, I. (2001) The HMMTOP transmembrane topology prediction server. *Bioinformatics* 17, 849–850.

66. Krogh, A., Larsson, B., von Heijne, G., Sonnhammer, E. L. (2001) Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J Mol Biol* 305, 567–580.

67. Kall, L., Krogh, A., Sonnhammer, E. L. (2004) A combined transmembrane topology and signal peptide prediction method. *J Mol Biol* 338, 1027–1036.

68. Kall, L., Krogh, A., Sonnhammer, E. L. (2005) An HMM posterior decoder for sequence feature prediction that includes homology information. *Bioinformatics* 21(Suppl 1), i251–i257.

69. Jones, D. T. (2007) Improving the accuracy of transmembrane protein topology prediction using evolutionary information. *Bioinformatics* 23, 538–544.

70. Natt, N. K., Kaur, H., Raghava, G. P. (2004) Prediction of transmembrane regions of beta-barrel proteins using ANN- and SVM-based methods. *Proteins* 56, 11–18.

71. Simossis, V. A., Kleinjung, J., Heringa, J. (2005) Homology-extended sequence alignment. *Nucl Acids Res* 33, 816–824.

72. Pirovano, W., Feenstra, K. A., Heringa, J. (2008) PRALINETM: a strategy for improved multiple alignment of transmem-
brane proteins. *Bioinformatics* 24, 492–497.

73. Pei, J., Grishin, N. V. (2007) PROMALS: towards accurate multiple sequence alignments of distantly related proteins. *Bioinformatics* 23, 802–808.

74. Eisenberg, D., Schwarz, E., Komaromy, M., Wall, R. (1984) Analysis of membrane and surface protein sequences with the hydrophobic moment plot. *J Mol Biol* 179, 125–142.

Chapter 20

Analysis and Prediction of Protein Quaternary Structure

Anne Poupon and Joel Janin

Abstract

The quaternary structure (QS) of a protein is determined by measuring its molecular weight in solution. The data have to be extracted from the literature, and they may be missing even for proteins that have a crystal structure reported in the Protein Data Bank (PDB). The PDB and other databases derived from it report QS information that either was obtained from the depositors or is based on an analysis of the contacts between polypeptide chains in the crystal, and this frequently differs from the QS determined in solution.

The QS of a protein can be predicted from its sequence using either homology or threading methods. However, a majority of the proteins with less than 30% sequence identity have different QSs. A model of the QS can also be derived by docking the subunits when their 3D structure is independently known, but the model is likely to be incorrect if large conformation changes take place when the oligomer assembles.

Key words: oligomeric proteins, protein molecular weight, biomolecule, molecular assembly, protein–protein docking, threading, modeling.

1. Introduction

Most proteins are made of not one, but several polypeptide chains, and their assembly constitutes a quaternary structure (QS). QS was first identified in hemoglobin in the mid 1920s, when its molecular weight was determined by sedimentation in the ultracentrifuge. Mammalian hemoglobins have two α and two β chains. They are heterotetramers, 'hetero' referring to the different amino acid sequences of α and β, but other animal species have hemoglobins that are monomers, homodimers (in which the two chains have the same sequence), or more complex assemblies. All are homologous and their subunits have essentially the same fold: the globin fold. They also have the same basic function, binding oxygen, but the diversity of their QS plays a major role in adapting that function to the physiology of the organism in which they occur.

For many years, the time table of sequence, crystal structure, and QS determination was more or less the same, and their status also (1). With the advent of large-scale DNA sequencing, many amino acid sequences became available and the focus was placed on predicting from the sequence (the primary structure), the presence of α-helices and β-sheets (the secondary structure), and ultimately the fold of the polypeptide chain (the tertiary structure). QS was largely forgotten until, in recent years, genome-wide genetic and biochemical studies indicated that most gene products are not autonomous entities. Rather, they are elements of multimolecular assemblies that range from small oligomers (proteins with few subunits) to huge molecular machines like the ribosome or the nuclear pore (2). In all, QS is essential to function, and it is now recognized that this must be established along with the protein sequence and fold. Relevant databases and Web sites are listed in **Table 20.1**.

Table 20.1
Databases and Web servers

General	
PDB	http://www.rcsb.org/pdb/
SCOP	http://scop.mrc-lmb.cam.ac.uk/scop/
ExPASy	http://www.expasy.ch
PFAM	http://www.sanger.ac.uk/Software/Pfam
Quaternary structure databases	
3D Complex	http://3dcomplex.org
PiQSi	http://www.piqsi.org/
ProtBuD	http://dunbrack.fccc.edu/ProtBuD/
QS prediction from the PDB	
PQS	http://pqs.ebi.ac.uk/
PITA	http://www.ebi.ac.uk/thornton-srv/databases/pita
PISA	http://www.ebi.ac.uk/msd-srv/prot_int
QS prediction from sequence	
InterPreTS	http://www.russell.embl.de/cgi-bin/interprets2
3DID	http://gatealoy.pcb.ub.es/3did/
Prediction of interacting surfaces	
ASEdb	http://nic.ucsf.edu/asedb
ConSurf	http://consurf.tau.ac.il

Table 20.1 (continued)

Intervor	http://cgal.inria.fr/Intervor/
PIbase	http://alto.compbio.ucsf.edu/pibase/
Protein–protein docking	
CAPRI	http://capri.ebi.ac.uk/
Servers	
ClusPro	http://nrc.bu.edu/cluster/
MultiDock	http://www.sbg.bio.ic.ac.uk/docking/multidock.html
GRAMM-X	http://vakser.bioinformatics.ku.edu/resources/gramm/grammx
HADDOCK	http://haddock.chem.uu.nl/
PatchDock	http://bioinfo3d.cs.tau.ac.il/PatchDock
SymmDock	http://bioinfo3d.cs.tau.ac.il/SymmDock/
SKE-Dock	http://www.pharm.kitasato-u.ac.jp/biomoleculardesign/files/SKE_DOCK.html
SmoothDock	http://structure.pitt.edu/servers/smoothdock/
RosettaDock	http://rosettadock.graylab.jhu.edu/
PRISM	http://prism.ccbb.ku.edu.tr/prism/

2. QS, Symmetry, and Crystal Structures

The QS is defined by the chain content of the protein ($\alpha_2\beta_2$ in hemoglobin) and the way the chains are arranged, especially its symmetry (3). An oligomeric protein with *n* identical subunits, each of which may comprise more than one polypeptide chain, can have the symmetries of one of the following point groups:

- *Cn* (cyclic *n*) with a *n*-fold axis ($360°/n$ rotation)
- *Dm* (dihedral *m*) for even $n = 2m$, with a *m*-fold axis and *m* twofold axes orthogonal to it
- Cubic *T*, *O* or *I*, with twofold, threefold, and either fourfold (in *O*) or fivefold (in *I*) rotation axes.

The most common is *C2* in homodimers, but dihedral symmetry is the rule when *n* is an even number. Thus, *D2* tetramers are more common than *C4*, and *D3* hexamers than *C6*. Membrane proteins are an exception, because of the inherent asymmetry of biological membranes, compatible only with *Cn*. Cubic symmetry is illustrated by the capsids of icosahedral viruses.

The QS of a protein may change with the conditions, the presence of a ligand for instance. Thus, many transcription factors oligomerize when they bind DNA. Similarly, the symmetry may be

exact or approximate. Examples are the HIV protease, a homo-dimer that loses its *C2* symmetry when it binds a peptide substrate, or hemoglobin, which has an exact *C2* symmetry, and also an approximate *D2* symmetry if the difference between α and β chains is ignored. In general, the symmetry of an oligomeric protein is known only after its atomic structure is determined. The major tools for that are X-ray crystallography and NMR, but crystallography is surprisingly poor at establishing the QS. A protein crystal is a multimolecular assembly held together by the same forces as the QS. In a crystal, intermolecular contacts coexist with subunit contacts, and telling the two apart may not be trivial; methods to do so are discussed below.

By convention, the Protein Data Bank (PDB) (4) reports atomic coordinates for the crystal asymmetric unit (ASU). This may include more than one polypeptide chain irrespective of the QS. Thus, a monomeric protein can yield crystals with two or more chains in the ASU, and crystallographers commonly refer to the 'dimer in the asymmetric unit' whether or not the protein is dimeric in solution. In counterpart, an oligomeric protein can give crystals with one chain in the ASU, the other chains being related to it by crystal symmetries. A dimeric protein can even crystallize with three chains in the ASU: a dimer and a half, the other half being generated by a crystal symmetry. For this reason, the header of many PDB entries includes 'biomolecule' records (described in 4.1 below) that relate the ASU to the QS. NMR structures are determined in solution, but comparatively few NMR entries in the PDB report oligomeric structures because of their larger size, and also of their symmetry that causes ambiguities when assigning resonances.

3. Mining the Literature

Experimentally, the subunit composition of a protein is assigned by measuring its molecular weight (Mw) and comparing to the Mw of the polypeptide chain(s), measured or calculated from their amino acid sequence and corrected for posttranslational modifications if needed. Less frequently, the subunit composition is determined by introducing chemical crosslinks between the subunits and analyzing the products by gel electrophoresis under denaturing conditions. Publications that report crystal or NMR structures usually refer to such data when they exist, but they do not do so in a systematic way, and the information is often buried in the text. A literature search must then explore the publications in full and use keywords related to methods for Mw determination as well as to the QS. Mass spectrometry, the most powerful method of all, is

not yet a standard tool for establishing QS, because conventional sample desorption procedures break the noncovalent bonds between subunits. A protein Mw in solution can be reliably estimated by careful studies of static light scattering, small-angle X-ray scattering (SAXS), or by equilibrium analytical centrifugation, all relatively expensive experiments. Parameters related to Mw, such as the diffusion coefficient, can be determined by NMR and other biophysical methods. Gel filtration on a molecular sieve (also called size exclusion chromatography) and dynamic light scattering (DLS) are less demanding in terms of equipment and the protein sample. These methods measure parameters, the Stokes radius for molecular sieves and the diffusion coefficient for DLS, that depend on the shape of the protein as well as its Mw. Both are commonly used, but in the absence of other data, a sentence such as 'the protein migrates as a dimer under gel filtration' should only be taken as an indication of the QS.

Curated sets of proteins of known X-ray and oligomeric structures have been obtained by carrying out manual surveys of the biochemical literature and identifying data that establish their QS in solution (5–7). The sets are of a very limited size: 96 monomers and 76 homodimers in (5), 188 monomers and 122 homodimers in (7), and they represent only a small fraction of the PDB. Recently, Levy (8) performed a large-scale literature search to assign the QS of 3,214 proteins, which may cover one-quarter or more of the PDB by extension to close homologs in which the QS is very likely conserved. The search used keywords related to the QS ('monomer', 'dimer') and to methods for Mw determination. The results are accessible through the PiQSi database (*see* **Section 4.2** below). The QS assignments in that study are in nearly perfect agreement with the manually curated data sets, but they often disagree with the PDB biomolecule. The discrepancies concern 15% of the proteins in the whole database, and 27% in a nonredundant subset (8).

4. QS in Databases

4.1. QS in the Protein Data Bank

PDB entries contain information on the biomolecule as reported by their authors, coded in the records REMARK 300 that defines its relationship to the ASU, and REMARK 350 that gives the symmetry operations needed to build it from the ASU.

Let us take some examples.

4.1.1. Entry 1mkb

REMARK 300 BIOMOLECULE: 1

REMARK 300 THIS ENTRY CONTAINS THE CRYSTALLOGRAPHIC ASYMMETRIC UNIT

REMARK 300 WHICH CONSISTS OF 2 CHAIN(S). SEE REMARK 350 FOR REMARK 300

INFORMATION ON GENERATING THE BIOLOGICAL MOLECULE(S)....

```
REMARK 350 BIOMOLECULE: 1
REMARK 350 APPLY THE FOLLOWING TO CHAINS: A, B
REMARK 350 BIOMT1 1 1.000000 0.000000 0.000000 0.00000
REMARK 350 BIOMT2 1 0.000000 1.000000 0.000000 0.00000
REMARK 350 BIOMT3 1 0.000000 0.000000 1.000000 0.00000
```

REMARK 300 tells us that there is one biomolecule and the ASU contains two chains; REMARK 350 cites their chain codes followed by an identity matrix, which implies that the biomolecule is the same as the ASU. Checking the SEQRES records shows that the two chains have the same sequence. Therefore, the protein is a homodimer.

4.1.2. Entry 1otp   The ASU contains one chain according to REMARK 300, but REMARK 350 has two matrices: the identity matrix and one that generates a second subunit. The protein is also a homodimer.

4.1.3. Entry 1r56   REMARK 300 reports two biomolecules, and eight chains in the ASU; REMARK 350 gives 4 chain codes followed by the identity matrix, to each biomolecule; SEQRES quotes 8 identical sequences. Taken together, these data show that the protein is a homotetramer, with two copies of it in the ASU.

It should be noted that the word 'dimer' or 'tetramer' appears nowhere in these entries. Thus, oligomers in the PDB cannot be extracted simply by searching for keywords. A script has to be written to compare the number of chains in the ASU to that of biomolecules in REMARK 300 and of transformation matrices in REMARK 350, and then build the biomolecule(s) using these matrices.

REMARK 300 is absent from entries deposited before 1999, and if present, it may just state that 'the biological unit of this protein is unknown'. The examples above indicate that the information on the QS in PDB entries is not easily interpretable. Moreover, it is never documented: REMARK 300 cites no data, and an entry will not necessarily be updated if new data become available after its deposition.

4.2. QS in Databases Derived from the PDB

4.2.1. Biounit   Aware of the difficulty in interpreting the QS information in PDB entries, the curators of the PDB at the Research Collaboratory for Structural Bioinformatics (RCSB) have created a derived database called Biounit. Biounit is based on the REMARK 300/350 records for entries posterior to 1999, and on supporting information from the authors, SwissProt or PQS (*see* **Section 4.3**) for earlier entries. It is accessible through, and at present only through, the RCSB PDB interface. For each entry, the interface displays images of the biomolecule and the ASU content, and it allows downloading their coordinates.

4.2.2. ProtBuD

The Protein Biological Unit Database (ProtBuD) (9) compares the QS derived from the PDB to that in PQS. The two agree in 82% of the entries and differ from the ASU in 52%. The ProtBuD hierarchy is based on SCOP (structural classification of proteins) (10) and, when a PDB or a SCOP entry code is entered, its user interface shows the SCOP classification of each chain before displaying the ProtBuD page proper. This page describes the content of the ASU and the biomolecule as defined in the PDB file and in PQS. ProtBuD also lists all the pairwise interfaces and gives access to coordinates for each assembly. DNA, RNA, and other ligands are cited when present.

4.2.3. D Complex

3D Complex (11) uses the QS information from the PDB Biounit. Version V2.0, based on the 1.73 release of SCOP, will eventually be extended to the whole PDB, omitting the nonprotein components. 3D Complex offers a hierarchical classification of protein assemblies (which it calls 'complexes') based on the domain assignments in SCOP and a graph representation of domain–domain contacts. Like SCOP, it has a top level 'topology' above a level of 'families' that accounts for evolutionary relationships. The topology depends on the number of subunits, the symmetry, and the pattern of contacts represented by a graph; an efficient graph-matching procedure is used to assign and compare QS topologies. A QS family contains assemblies of a given topology, and in which the chains making equivalent contacts belong to the same SCOP superfamily. Lower levels are labeled QSx, where x is the percent sequence identity between equivalent chains in related assemblies (e.g., 30% sequence identity for QS30).

The interface allows the user to browse through the hierarchy of 3D Complex and build a custom query (**Fig. 20.1**). It makes it for example, very easy to list all the homotetramers at less than 30% sequence identity. Each molecular assembly appears in a visually friendly graph representation, where the color and shape of the nodes mark homologous and nonhomologous subunits, and edges represent pairwise contacts. Labels near edges report the number of residues in the interface as an estimate of its size; clicking on the nodes yields additional information on the subunits and access to external links.

4.2.4. PiQSi

Protein quaternary structure investigation (PiQSi) (8) is an annotated database derived from 3D Complex and interlinked with it. As mentioned above, PiQSi stems from a literature search yielding QS annotations that were extended to proteins with high sequence identity (>90%) in a second step. The QS in PiQSi differs from that in the PDB or PQS in about 15% of the cases. At present, the database contains over 10,000 entries, which can be accessed by entering a PDB code or a protein sequence. The interface then displays the graph representation of the assembly and other items carried over from 3D Complex. It also cites as a clickable PubMed

A. 3D Complex

					Same graph topology	Subunits with same structure (SCOP Superfamily)	Within-complex subunit sequence identity	Across-complex subunit sequence similarity
			Totals	Terminology				
			192	QS Topologies	✓	✗	✗	✗
985	1003	160	184	89 •••	3151	QS families	✓	✓	✗	✗
985	1057	165	189	93 •••	3226	QS	✓	✓	✓	✗

Code	Picture	Function	No. Chains	Symmetry	QS	QS30	QS90	QS100	All data
1luq		Biotin binding	4	D2	1	1	2	23	117
1j40_1		Oxygen transport	4	D2	2	5	13	35	79
4crx		DNA Binding	4	C4	2	3	3	8	16

B. PiQSi

Protein Reference

	Code	Picture	Error?	Sym	No. Chains	SProt	Weight	Ref	H	E	Organism	Function
	4crx		NO	C4	4	-	178	P	H	E	Bacteriophage P1	DNA Ligase

Homologs

%id	Code	Picture	Error?	Sym	No. Chains	SProt	Weight	Ref	H	E	Organism	Function
99	1kbu		NO	C4	4	P06956	178	P	H	E	Bacteriophage P1	DNA Ligase
99	1crx		YES	NS	2	P06956	322	P	H	E	Bacteriophage P1	DNA Ligase
99	1f44		NO	C3	3	P06956	313	P	H	E	Bacteriophage P1	DNA Ligase

Fig. 20.1. Quaternary structures in the 3D Complex and PiQS databases. (**A**) The 3D Complex database (11) reports the QS of PDB biomolecules. The database has a hierarchic structure with 192 'topologies' and 3151 'families'. The square tetramer topology comprises 160 families; clicking yields a list of nonredundant representatives. Among those, the Cre recombinase (4crx) is a homotetramer with the (relatively rare) *C4* symmetry, when it is in complex with DNA forming a four-way Holliday junction. (**B**) A second click connects to 4crx in the PiQSi database (8) that reports QS information based on the literature. PiQSi has annotations for several PDB entries for the Cre recombinase or its homologs. The PDB biomolecule is a tetramer in 4crx and 1kbu, a dimer in 1crx, and a trimer in 1f44. The literature agrees with the PDB in the case of the 1f44 trimer, which is in complex with a three-way Holliday junction, but not the 1crx dimer.

ID, the reference used to annotate the QS. A tag indicates whether the biomolecule in the PDB is thought to be correct, incorrect, or uncertain, and in each case, a comment explains the annotator's opinion. The whole database can be downloaded through the Web site, for instance, to serve as a training set for prediction methods.

A specificity of PiQSi is that its users can submit new annotations in a Wiki spirit. They will be processed by the curators and eventually propagated in the database. Thus, PiQSi initiates a community effort to manually curate the QS information in the PDB.

4.3. Deriving the QS from the Atomic Coordinates: PQS, PITA, PISA

The Macromolecular Structure Database (MSD) Group at the European Bioinformatics Institute has developed tools that analyze geometric and physical-chemical properties of the protein–protein contacts in a crystal or NMR structure and attempt to derive the QS from those. They do not use the information in REMARK 300 and may disagree with it. Servers offer access to both the tools (by submitting coordinates in the PDB format) and databases that contain the results of their application to all entries in the PDB.

4.3.1. PQS

The Probable Quaternary Structure (PQS) algorithm (12) was the first of its type, and the others are derived from it. PQS applies crystal symmetries to each molecule in the ASU, generates neighbors, and gives each pairwise interface a score based on the area buried at the interface plus a solvation energy term. It then builds the QS iteratively by retaining only interfaces that achieve a given score. The 'probable' QS cited in the database is the one with the best score. A query to the database, done by submitting a PDB entry code or a set of atomic coordinates, returns a line of values that includes the assembly type (monomeric, dimeric, etc.). It also gives access to a text output that contains additional information, and to a file containing atomic coordinates of the assembly in PDB format.

4.3.2. PITA

Protein InTerfaces and Assemblies (PITA) (13) is similar to PQS, except that a statistical potential replaces the solvation energy, and that all assemblies are kept above a certain score.

4.3.3. PISA

In Protein Interfaces, Surfaces and Assemblies (PISA) (14), the iterative construction of the QS is replaced by a graph exploration procedure that surveys all the assemblies that can be formed in the crystal. The procedure handles nonprotein components ligands (DNA, RNA, small molecules, and ions), and it can detect large assemblies that PQS or PITA would miss. For each assembly, PISA calculates a free energy of dissociation, ΔG^{diss}, that includes a number of physical-chemical terms. Given a PDB code or a set of atomic coordinates, the user interface returns tables with

information on each chain, each pairwise interface, and all the assemblies that have a positive ΔG^{diss}. Selecting a chain, interface, or assembly in the tables gives access to atomic coordinates and to additional values concerning the selected item, and opens a 3D interactive graphic window.

5. Predicting QS from the Amino Acid Sequence

Determining the QS of a protein is no straightforward task experimentally in solution, and it can be error-prone even when an X-ray structure is available. Thus, there should be a strong incentive to predict the QS from the amino acid sequence yet we are still in early stages of such a prediction.

Given the sequence of a protein, one may first ask whether it is oligomeric or not. Oligomers contain large subunit interfaces that have physical–chemical properties resembling the protein interior more than its surface (15). This affects their amino acid composition and must be reflected in the sequence. The Quaternary Structure Explorer (16) uses an empirical combination of parameters derived from the sequence to predict whether a polypeptide chain is a monomer or part of an oligomer. However, the reported accuracy is only 70%, and a prediction based on just the amino acid composition performs at least as well (17).

Beyond that point, QS is generally predicted by homology assuming that it is conserved in evolution (18). Albeit safe for close homologs (hemoglobins from different mammals), the assumption breaks down at a certain level of divergence (fish hemoglobins, myoglobin, etc.), and this may explain why standard tools for homology modeling generally ignore the QS. How far in evolution is QS conserved? Aloy et al. (19) report that the interaction between two Pfam domains is almost invariably conserved at 30–40% sequence identity or more. However, Lévy et al. (20) find that, at this identity level, 30% of the proteins in PiQSi have different QSs, and that below 30% identity, the QS changes in half of the homologs. Thus, the reliability of a QS prediction made by homology is questionable below 40% sequence identity.

5.1. InterPreTS

Interaction Prediction through Tertiary Structure (InterPreTS) (21) creates models of protein–protein interfaces derived from a database of domains known to interact. Given two query sequences, InterPreTS identifies Pfam domains in them, and looks up the database to find if the domains interact. This database, now called 3DID, is derived from iPfam (22), and it records 4,814 domain–domain interactions, of which 85% are interchain (23). If the Pfam domain pair is present, the query sequences are aligned

with the closest homologs and the resulting interaction is evaluated with an empirical potential. This procedure was used to carry out structural predictions on 102 protein complexes in yeast, yielding at least partial models of half of the complexes (24).

5.2. Threading

Threading methods aim to detect structural homology at low levels of sequence identity. M-TASSER (25, 26) is a threading procedure designed to build dimers. It uses a library of 1,838 templates, mostly homodimers, selected in the PDB on the basis of the biomolecule record and checked against PQS; a literature search done on a small subset suggests that it still contains 10–15% of nondimers. A query sequence is first threaded with TASSER (27) to generate models of the monomers, which are structurally aligned on the dimers in the library and refined with the TASSER force field. An all-against-all comparison was performed on the dimers in the library. Excluding templates with >30% sequence identity, the structural alignment identified a correct template in about half of the cases, and the refined models had an average RMSD of 5.9 Å relative to the native structures. Threading the sequences directly on the dimers in the library yielded fewer templates, but 80% showed a weak sequence identity and the final RMSD was the same.

6. Protein–Protein Docking

Docking procedures predict the structure of a complex based on those of its components. A number of methods to dock two proteins have been developed in recent years (28–33). They generally operate in two steps: exploration and scoring. The exploration step moves one component as a rigid body relative to the other, aiming to bring in contact regions of the two protein surfaces that are complementary in shape and physical–chemical properties. The search yields thousands of models of the complex, most of which are false positives, along with a few near-native solutions, or possibly none if large conformation changes occur in the components. The scoring step evaluates and ranks these models in order to identify the near-native solutions and perform further refinement on them.

6.1. Docking Algorithms

Exploration algorithms often use a simplified representation of the molecules rather than the atomic coordinates, in order to remove high-resolution details and speed up the search. Thus, the protein model in ATTRACT (34) has only 1 to 3 pseudo-atoms per residue. In the commonly used FFT (fast Fourier transform) correlation algorithm, the two protein structures are mapped onto a cubic grid, and each grid point is given a weight that marks its position relative to the molecule (outside, inside, on the surface),

and codes for some physical-chemical properties (electrostatics, pair potentials). The correlation between the weights associated to the two proteins is used to score a position. It is efficiently calculated by FFT for all grid translations of one protein relative to the other, but the calculation must be repeated for each orientation. Procedures based on the FFT correlation algorithm are ZDOCK (35) and DOT (36), implemented in the ClusPro and SmoothDock servers, FTDOCK (37) implemented in the MultiDock server, and GRAMM-X (38). PatchDock (39) uses a very efficient computer vision procedure in which the two protein surfaces are divided into concave, convex, and flat patches, and complementary patches are brought together by a geometric hashing algorithm.

All these algorithms perform exhaustive, or at least extensive, translation/rotation searches. Other algorithms randomly generate starting positions ('decoys') and use heuristic procedures to optimize them. In RosettaDock (40, 41) and several related procedures (42), thousands of decoys are generated and refined. RosettaDock uses two-steps of Monte-Carlo minimization to do that, the first step with a simplified protein model and force field, the second, with explicit atoms and the very successful Rosetta force field, originally developed for folding predictions.

6.2. Scoring and Refinement

Scoring aims at identifying near-native models among those issued from the search. Scoring schemes often combine energy, geometric complementarity, propensities, and other terms in a single scoring function that is optimized through machine learning procedure. External information may be available on the complex, for instance, as an interface prediction (43), sequence conservation, or biological data on mutants. Such information can be used in the first step to limit the search space, or be taken care of during scoring. In the case of HADDOCK, external information is used to drive the search itself. This algorithm treats a variety of data including NMR data if they exist, as 'ambiguous interaction restraints' in an energy minimization procedure (44, 45).

The top scores yield candidate solutions that must be refined to optimize the position and orientation of the components in the modeled complex, and also to account for conformation changes. Commonly used protocols perform energy minimization and/or molecular dynamics simulations. Side-chain flexibility is a minimum requirement at that stage, but changes in backbone conformation and larger movements such as domain hinge rotations must also be considered, as they commonly occur when two proteins interact. HADDOCK or RosettaDock handle conformation changes by leaving free some main-chain dihedral angles. Other procedures do it by generating a number of alternative conformations in a first step, and then 'cross-docking' the conformers pairwise (46, 47). In both cases, flexibility greatly increases the size of the calculation and the number of false positives.

6.3. Generating Oligomers by Docking

Up to now, docking has mostly been used to model binary complexes between proteins for which an X-ray or NMR structure has been determined independently. As oligomeric proteins are obligate assemblies with few exceptions, no experimental structure of the isolated monomer is available for docking. This may nevertheless be performed on models generated in silico by homology or threading (48). A remarkable example is the structural model recently proposed for the nuclear pore (49). To build it, a fold type was assigned to each domain in all the 456 constituent proteins, the domains were modeled, and the models assembled by optimizing a score function under a large number of restraints derived from experiment. The *C16* symmetry of the nuclear pore played a major role in that operation.

Cyclic (or dihedral) symmetry is a severe constraint in modeling oligomeric proteins, and it has been incorporated in some of the docking procedures originally designed to model binary complexes (50, 51). The algorithm implemented in the SymmDock server is a version of PatchDock that limits all rotations to those compatible with *Cn* symmetry (39). When multicomponent docking is performed by adding one component at a time, the constraints due to the occupied space also play a major role in limiting the search (52, 53).

6.4. Assessing Docking Predictions: CAPRI

The CAPRI (Critical Assessment of PRedicted Interactions (54)) experiment has been designed to test the performance of protein–protein docking in blind predictions, as CASP (Critical Assessment of Techniques for Protein Structure Prediction) does for protein fold prediction. Early attempts to include QS predictions in CASP were not pursued in recent rounds (55) even though many CASP targets are oligomeric. CAPRI targets are protein–protein complexes; their X-ray structure is known but still unpublished, and their components (or at least close homologs) are in the PDB. CAPRI predictors dock the components and submit models that are assessed against the experimental structure of the complexes.

Because the component structures must be known, most CAPRI targets are transient complexes rather than oligomeric proteins (56, 57). In six years of the experiment, only three targets have been oligomers, all three homodimers: one in which the subunits take two different orientations, another that had a monomeric homolog in the PDB, and a viral envelope protein that changes from a dimeric form to a trimer during cell infection (58); the structure of the trimer had to be predicted from the previously known dimer. In all cases, symmetry should have helped in finding correct docking solutions, but there were large conformation changes and domain movements that the prediction procedures failed to reproduce. In the viral envelope protein, the movements did not affect the trimer interface, and HADDOCK produced a solution that correctly reproduced the geometry of the X-ray structure (59). In general, the capacity to predict and

simulate conformation changes appears to be crucial to the success of docking methods, and this is even more obvious when the target is an oligomeric protein.

7. Conclusion

The quaternary structure of proteins is highly relevant to their function, and its importance is now fully recognized. Because QS must be determined in solution and can remain uncertain even when an X-ray structure is known, the biomolecule assignment in PDB entries is often in error, and a number of derived databases implement methods to correct it. QS can also be predicted in the absence of a detailed structure by homology, threading, or protein–protein docking. Whereas the reliability of such predictions remains questionable, they have been proved to be extremely useful in cases where the constraints derived from experimental information sets are sufficient to guide the modeling procedure.

Acknowledgments

References

1. Darnall, D. W., Klotz, I. M. (1975) Subunit constitution of proteins: a table. *Arch Biochem Biophys* 166, 651–682.

2. Alberts, B. (1998) The cell as a collection of protein machines: preparing the next generation of molecular biologists. *Cell* 92, 291–294.

3. Goodsell, D. S., Olson, A. J. (2000) Structural symmetry and protein function. *Annu Rev Biophys Biomol Struct* 29, 105–153.

4. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., Bourne, P. E. (2000) The Protein Data Bank. *Nucleic Acids Res* 28, 235–242.

5. Ponstingl, H., Henrick, K., Thornton, J. M. (2000) Discriminating between homodimeric and monomeric proteins in the crystalline state. *Proteins* 41, 47–57.

6. Ponstingl, H., Kabir, T., Gorse, D., Thornton, J. M. (2005) Morphological aspects of oligomeric protein structures. *Prog Biophys Mol Biol* 89, 9–35.

7. Bahadur, R. P., Chakrabarti, P., Rodier, F., Janin, J. (2003) Dissecting subunit interfaces in homodimeric proteins. *Proteins* 53, 708–719.

8. Levy, E. D. (2007) PiQSi: protein quaternary structure investigation. *Structure* 15, 1364–1367.

9. Xu, Q., Canutescu, A., Obradovic, Z., Dunbrack, R. L., Jr. (2006) ProtBuD: a database of biological unit structures of protein families and superfamilies. *Bioinformatics* 22, 2876–2882.

10. Murzin, A. G., Brenner, S. E., Hubbard, T., Chothia, C. (1995) SCOP: a structural

classification of proteins database for the investigation of sequences and structures. *J Mol Biol* 247, 536–540.

11. Levy, E. D., Pereira-Leal, J. B., Chothia, C., Teichmann, S. A. (2006) 3D complex: a structural classification of protein complexes. *PLoS Comput Biol* 2, e155.

12. Henrick, K., Thornton, J. M. (1998) PQS: a protein quaternary structure file server. *Trends Biochem Sci* 23, 358–361.

13. Ponstingl, H., Kabir, T., Thornton, J. M. (2003) Automatic inference of protein quaternary structure from crystals. *J Appl Cryst* 36, 1116–1122.

14. Krissinel, E., Henrick, K. (2007) Inference of macromolecular assemblies from crystalline state. *J Mol Biol* 372, 774–797.

15. Janin, J., Miller, S., Chothia, C. (1988) Surface, subunit interfaces and interior of oligomeric proteins. *J Mol Biol* 204, 155–164.

16. Garian, R. (2001) Prediction of quaternary structure from primary structure. *Bioinformatics* 17, 551–556.

17. Carugo, O. (2007) A structural proteomics filter: prediction of the quaternary structural type of hetero-oligomeric proteins on the basis of their sequences. *J Appl Cryst* 40, 986–989.

18. Aloy, P., Pichaud, M., Russell, R. B. (2005) Protein complexes: structure prediction challenges for the 21st century. *Curr Opin Struct Biol* 15, 15–22.

19. Aloy, P., Ceulemans, H., Stark, A., Russell, R. B. (2003) The relationship between sequence and interaction divergence in proteins. *J Mol Biol* 332, 989–998.

20. Levy, E. D., Erba, E. B., Robinson, C. V., Teichmann, S. A. Assembly reflects evolution of protein complexes (*submitted*)

21. Aloy, P., Russell, R. B. (2003) InterPreTS: protein interaction prediction through tertiary structure. *Bioinformatics* 19, 161–162

22. Finn, R. D., Marshall, M., Bateman, A. (2005) iPfam: visualization of protein-protein interactions in PDB at domain and amino acid resolutions. *Bioinformatics* 21, 410–412.

23. Stein, A., Russell, R. B., Aloy, P. (2005) 3did: interacting protein domains of known three-dimensional structure. *Nucleic Acids Res* 33, D413–D417.

24. Aloy, P., Böttcher, B., Ceulemans, H., Leutwein, C., Mellwig, C., Fischer, S., Gavin, A. C., Bork, P., Superti-Furga, G., Serrano, L., Russell, R. B. (2004) Structure-based assembly of protein complexes in yeast. *Science* 303, 2026–2029.

25. Grimm, V., Zhang, Y., Skolnick, J. (2006) Benchmarking of dimeric threading and structure refinement. *Proteins* 63, 457–465.

26. Chen, H., Skolnick, J. (2008) M-TASSER: An Algorithm for Protein Quaternary Structure Prediction. *Biophys J* 94, 918–928.

27. Zhang, Y., Skolnick, J. (2004) Automated structure prediction of weakly homologous proteins on a genomic scale. *Proc Natl Acad Sci USA* 101, 7594–7599.

28. Smith, G. R., Sternberg, M. J. (2002) Prediction of protein-protein interactions by docking methods. *Curr Opin Struct Biol* 12, 28–35.

29. Halperin, I., Ma, B., Wolfson, H., Nussinov, R. (2002) Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins* 47, 409–443.

30. Gray, J. (2006) High-resolution protein-protein docking. *Curr Op Struct Biol* 16, 150–169.

31. Méndez, R., Leplae, R., Lensink, M. F., Wodak, S. J. (2005) Assessment of CAPRI predictions in rounds 3–5 shows progress in docking procedures. *Proteins* 60, 150–169.

32. Lensink, M. F., Méndez, R., Wodak, S. J. (2007) Docking and scoring protein complexes: CAPRI 3rd Edition. *Proteins* 69, 704–718.

33. Wiehe, K., Peterson, M. W., Pierce, B., Mintseris, J., Weng, Z. (2007) Protein-protein docking: overview and performance analysis. *Methods Mol Biol* 413, 283–314.

34. May, A., Zacharias, M. (2007) Protein-protein docking in CAPRI using ATTRACT to account for global and local flexibility. *Proteins* 69, 774–780.

35. Wiehe, K., Pierce, B., Tong, W. W., Hwang, H., Mintseris, J., Weng, Z. (2007) The performance of ZDOCK and ZRANK in rounds 6–11 of CAPRI. *Proteins* 69, 719–725.

36. Mandell, J. G., Roberts, V. A., Pique, M. E., Kotlovyi, V., Mitchell, J. C., Nelson, E., Tsigelny, I., Ten Eyck, L. F. (2001) Protein docking using continuum electrostatics and geometric fit. *Protein Eng* 14, 105–113.

37. Gabb, H. A., Jackson, R. M., Sternberg, M. J. (1997) Modelling protein docking using shape complementarity, electrostatics and biochemical information. *J Mol Biol* 272, 106–120.

38. Tovchigrechko, A., Vakser, I. A. (2006) GRAMM-X public web server for protein-protein docking. *Nucleic Acids Res* 34, W310–W314.

39. Schneidman-Duhovny, D., Inbar, Y., Nussinov, R., Wolfson, H. J. (2005) PatchDock and SymmDock: servers for rigid and symmetric docking. *Nucleic Acids Res* 33, W363–W367.

40. Chaudhury, S., Sircar, A., Sivasubramanian, A., Berrondo, M., Gray, J. J. (2007) Incorporating biochemical information and backbone flexibility in RosettaDock for CAPRI rounds 6–12. *Proteins* 69, 793–800.

41. Wang, C., Schueler-Furman, O., Andre, I., London, N., Fleishman, S. J., Bradley, P., Qian, B., Baker, D. (2007) RosettaDock in CAPRI rounds 6–12. Proteins. 69, 758–763.

42. Heifetz, A., Pal, S., Smith, G. R. (2007) Protein-protein docking: progress in CAPRI rounds 6–12 using a combination of methods: the introduction of steered solvated molecular dynamics. *Proteins* 69, 816–822.

43. Qin, S., Zhou, H. X. (2007) A holistic approach to protein docking. *Proteins* 69, 743–749.

44. Dominguez, C., Boelens, R., Bonvin, A. M. (2003) HADDOCK: a protein-protein docking approach based on biochemical or biophysical information. *J Am Chem Soc* 125, 1731–1737.

45. de Vries, S. J., van Dijk, A. D., Krzeminski, M., van Dijk, M., Thureau, A., Hsu, V., Wassenaar, T., Bonvin, A. M. (2007) HADDOCK versus HADDOCK: new features and performance of HADDOCK2.0 on the CAPRI targets. *Proteins* 69, 726–733.

46. Krol, M., Chaleil, R. A., Tournier, A. L., Bates, P. A. (2007) Implicit flexibility in protein docking: cross-docking and local refinement. *Proteins* 69, 750–757.

47. Schneidman-Duhovny, D., Inbar, Y., Nussinov, R., Wolfson, H. J. (2005) Geometry-based flexible and symmetric protein docking. *Proteins* 60, 224–231.

48. Tovchigrechko, A., Wells, C. A., Vakser, I. A. (2002) Docking of protein models. *Protein Sci* 11, 1888–1896.

49. Alber, F., Dokudovskaya, S., Veenhoff, L. M., Zhang, W., Kipper, J., Devos, D., Suprapto, A., Karni-Schmidt, O., Williams, R., Chait, B. T., Sali, A., Rout, M. P. (2007) The molecular architecture of the nuclear pore complex. *Nature* 450, 695–701.

50. Berchanski, A., Segal, D., Eisenstein, M. (2005) Modeling oligomers with Cn or Dn symmetry: application to CAPRI target 10. *Proteins* 60, 202–206.

51. Pierce, B., Tong, W., Weng, Z. (2005) M-ZDOCK: a grid-based approach for Cn symmetric multimer docking. *Bioinformatics* 21, 1472–1478.

52. Inbar, Y., Benyamini, H., Nussinov, R., Wolfson, H. J. (2005) Prediction of multimolecular assemblies by multiple docking. *J Mol Biol* 349, 435–447.

53. Inbar, Y., Benyamini, H., Nussinov, R., Wolfson, H. J. (2005) Combinatorial docking approach for structure prediction of large proteins and multi-molecular assemblies. *Phys Biol* 2, S156–S165.

54. Janin, J., Henrick, K., Moult, J., Eyck, L. T., Sternberg, M. J., Vajda, S., Vakser, I., Wodak, S. J. (2003) Critical Assessment of PRedicted Interactions. CAPRI: a Critical Assessment of PRedicted Interactions. *Proteins* 52, 2–9.

55. Moult, J., Fidelis, K., Kryshtafovych, A., Rost, B., Hubbard, T., Tramontano, A. (2007) Critical assessment of methods of protein structure prediction-Round VII. *Proteins* 69 S8, 3–9.

56. Janin, J. (2005) The targets of CAPRI rounds 3–5. *Proteins* 60, 170–175.

57. Janin, J. (2007) The targets of CAPRI rounds 6–12. *Proteins* 69, 699–703.

58. Bressanelli, S., Stiasny, K., Allison, S. L., Stura, E. A., Duquerroy, S., Lescar, J., Heinz, F. X., Rey, F. A. (2004) Structure of a flavivirus envelope glycoprotein in its low-pH-induced membrane fusion conformation. *EMBO J* 23, 728–738.

59. van Dijk, A. D., de Vries, S. J., Dominguez, C., Chen, H., Zhou, H. X., Bonvin, A. M. (2005) Data-driven docking: HADDOCK's adventures in CAPRI. *Proteins* 60, 232–238.

Chapter 21

Prediction of Posttranslational Modification of Proteins from Their Amino Acid Sequence

Birgit Eisenhaber and Frank Eisenhaber

Abstract

If posttranslational modifications (PTMs) are chemical alterations of the protein primary structure during the protein's life cycle as a result of an enzymatic reaction, then the motif in the substrate protein sequence that is recognized by the enzyme can serve as basis for predictor construction that recognizes PTM sites in database sequences. The recognition motif consists generally of two regions: first, a small, central segment that enters the catalytic cleft of the enzyme and that is specific for this type of PTM and, second, a sequence environment of about 10 or more residues with linker characteristics (a trend for small and polar residues with flexible backbone) on either side of the central part that are needed to provide accessibility of the central segment to the enzyme's catalytic site. In this review, we consider predictors for cleavage of targeting signals, lipid PTMs, phosphorylation, and glycosylation.

Key words: posttranslational modifications, GPI lipid anchor, myristoylation, prenylation, farnesylation, geranylgeranylation, phosphorylation, glycosylation, peroxisomal localization, protein function prediction.

1. Introduction

A posttranslational modification (PTM) of a protein is the chemical alteration of its primary structure after translation. Generally, it is required that the considered type of covalent modification has to be a general feature of proteins from different families, not just that of a group of sequentially very similar ones. PTMs include both the formation of covalent cross-links of intra- or intermolecular (with a ligand/another protein) nature and the cleavage of covalent bonds including the breakage of a peptide bond and the removal of groups from single amino acid types. A typical protein appears to undergo several PTMs during its life time. Currently, more than

100 PTMs are known, the variety of PTMs is still growing and modifications that were considered obscure only a decade ago have become a mainstream target of research (1).

PTMs have a great impact on protein size, hydrophobicity, and its other physico-chemical properties. With PTMs, proteins can go beyond the limitations of chemical structure imposed by the set of 20 natural amino acid monomers and, therefore, may assume a much larger variety of complementary functional properties. PTMs can change, enhance, or block specific protein activities, or target a protein to another subcellular localization. Consequently, they allow the regulation of a protein's function.

2. General Consideration for PTM Predictor Construction and Evaluation

2.1. PTMs as a Result of an Enzymatic Process

From the view point of protein function prediction (2), PTMs that originate from the action of posttranslationally modifying enzymes (in contrast to spontaneous PTMs, for example, as a result of long-term exposure to pathologically increased metabolite concentrations) are especially important. On the one hand, these PTMs are part of an information transfer in pathways and, thus, of special biological importance. On the other hand, these PTMs are introduced into and only into substrate proteins that carry a sequence motif that is recognized by the posttranslationally modifying enzyme. Regardless of the specific nature of the PTM, the general structure of the sequence motif region has common properties: Except for a minority of cases (PTM before folding or after unfolding, at sites on large loops, etc.), PTM sites are embedded in nonglobular regions without inherent structural preference (3, 4). As a trend, the residues in these regions are polar, small, and with flexible backbone. A small central motif region (typically, about 5 residues) that is specific for the PTM and that enters the catalytic cleft of the enzyme is surrounded by segments with linker characteristics (with more than 10 residues on either side) that make the PTM site mechanically accessible to the enzyme. It are the enzyme-generated PTMs that will be considered in the following text.

During the last decade, the number of high-throughput sequencing projects has been dramatically increased. Quite a few of whole genomes including the human one are available; protein sequence databases have reached an enormous size. At the same time, there are a huge number of functionally uncharacterized genes and proteins. Especially in the case of otherwise uncharacterized proteins, PTM prediction is a valuable tool for limiting the range of the protein's possible function. For example, a target for

glycosylphosphatidylinositol (GPI) lipid anchoring, a complex lipid PTM has a defined subcellular localization (inner surface of the endoplasmic reticulum, Golgi, or the outer leaflet of the plasmalemma) and a limited range of functions (as extracellular enzyme, receptor, surface antigen, or transporter). The knowledge of a protein's PTMs is not less important for proteins with some annotational features. This further characterization with a predicted PTM, possibly, can link the protein to another pathway or a new biological mechanism. From the biological point of view, it is important to know which posttranslationally modifying enzyme is the source of the PTM for a given substrate protein since this relationship carries a pathway information. To conclude, the computer-aided prediction of the possibility of a protein's posttranslational modification from amino acid sequence is an important task that is critical for the biological interpretation of proteome data.

2.2. General Considerations for PTM Predictors

Whereas PTM prediction from sequence was a relatively obscure area of research 10 years ago, a superficial study of literature databases reveals a flood of papers in recent years. Unfortunately, many of these predictors have not been properly validated and the significance of new predictions beyond the range of what is solidly known from experiments cannot be reasonably assessed. Indeed, it is insufficient to collect some type of learning set, apply an automatic learning procedure, and throw the "predictor" on the market. This will not lead to biologically meaningful new predictions. What are the problems in the area of PTM predictor construction? Below, we will consider the most important aspects relevant for the evaluation of prediction tools (1, 3, 4).

The most basic issue is motif size. Many prediction efforts process only the putative central part of the recognition motif in their score function, maybe, with inclusion of a couple of surrounding residues. With a motif length of about a handful of residues, the motif description is not very discriminative and false-positive predictions (with more incorrect than correct predictions) become the major problem that devalues the significance of the prediction in the eyes of experimentalists. The traditional approach of characterizing short motifs with PROSITE-like notations (5–7) emphasizes positional information isolatedly; it ignores inter-positional correlations and is not really applicable for motif regions that are rather characterized by physical property conservation of amino acid side chains (in contrast to amino acid type conservation) (4). Consideration of the requirement of motif embedding in a linker-type region (3, 8) enlarges the total motif length to something like 25–30 residues and dramatically adds discriminative power to the scoring function.

The second critical issue is the question of proper biological context. It is important to build enzyme-specific scoring function for predictors. Thus, the learning set should include only substrates that are collected from one posttranslationally modifying enzyme or from a class of enzymes that essentially recognize the same motif in substrate proteins. For example, there is no biological sense for an acetylation predictor (9) that recognizes an "average" motif ("averaged" over several enzymes with possibly overlapping, yet vastly nonidentical specificity) since there is no enzyme that recognizes this "average" motif. The high rate of false predictions by enzyme-non-specific tools render them not useful for advice in experimental strategy planning and for proteome-wide scans.

Yet, it must be noticed that, very often, the link between the information on the type of PTM in a substrate protein and the information about the enzyme gets lost. For example, there are a number of mass-spectrometric data sets on phosphorylated proteins available, but the nature of the kinases behind these phosphorylation sites remains generally unknown. With this background, databases such as Phospho.ELM (10) with detailed information on substrates of explicitly named kinases are the more valuable. One might think that only predictors following the recognition motif concept enter the recent scientific literature; surprisingly, the nonbiological concept of relating PTMs (and subcellular localization) with overall amino acid composition is still not dead [see for example (11–17)]. There is no biological mechanism known that recognizes total amino acid composition; yet, the attachment of a nonglobular segment to the target protein carrying the motif for a PTM executing enzyme or a translocation receptor add the respective property (capability for PTM or translocation) to the protein, essentially without changing the amino acid composition.

The third key question in predictor development is the quality of underlying data – the amount and redundancy of example substrate protein sequences (or information about model peptides) and the level of authenticity. The UniProt (Swiss-Prot/TrEMBL) database is one of the best curated protein sequence databases. It strives to provide a high level of annotation, a minimal level of redundancy, and high level of integration with other databases (http://ca.expasy.org/sprot). Unfortunately, many PTMs are annotated based on sequence similarity to known substrates or other more obscure considerations but not on first-hand experimental evidence. Such assumptions raise the level of noise in the dataset. To emphasize, homologues do not need to have the same type of PTM. For example, human nyctalopin is GPI lipid anchored and the mouse orthologue is membrane bound with a transmembrane helix, although there is considerable sequence identity among the two homologues (18).

Finally from the mathematical point of view, there is a large variety of formal methods to describe a regression between sequence features of a recognition motif and the PTM outcome. The application of automated learning procedures such as neural networks (NN) or support vector machines (SVM) appears straightforward; yet, the large amount of parameters involved in the scoring function renders them not optimal for the task when learning sets are small and when they might contain false examples.

A technically cheaper approach that tries to understand sequence determinants that affect productive binding of the substrate protein segment with the binding site of the modifying enzyme generates considerably fewer parameters for physical property terms that have been proven very successful in rejecting improper sequence queries (1, 4, 19, 20). Interestingly, small subsets of apparently wrong examples in the learning set become eye-catching as a result of their discordance with the binding site requirements. For example, in the case of the development of MyrPS/NMT (20, 21), a prediction tool for *N*-terminal *N*-myristoylation, this approach revealed a striking difference with regard to six *N*-terminally overly hydrophobic proteins that were annotated as NMT substrates in the sequence database and, thus entered the learning set: the physical binding model distinguishes them as clearly different from the learning set and possibly unsuitable targets (20, 21), whereas a neural network approach does not see this difference (22) and treats all learning set examples equally reliable.

2.3. General Issues with Prediction Rate Accuracy

A good sequence-based predictor should be able to clarify two questions:

1. Is the query protein a potential target for the PTM considered? Both the positive and negative answers need to be reliable and risk of a false positive or false negative prediction should be quantified probabilistically.

2. If the sequence is a potential target for the PTM, what are the likely sequence positions in the query protein that may harbor the modification?

The ideal predictor is characterized by a high sensitivity and a very low false-positive prediction rate. Any real predictor is always a compromise between these two oppositional requests. On the one hand, it is relatively easy to create a program which predicts a selected training (learning) set with nearly 100% by choosing not very stringent parameters. But such a parameter set will result in a giant rate of false positive predictions. On the other hand, very stringent parameters warrant a low false-positive prediction rate but decrease the sensitivity (true-positive prediction rate) of the program. In a practical context of an

experimental laboratory and for proteome scans, the issue of a low false-positive prediction rates might even become more important for the predictor than the recognition of 100% of the known cases.

Unfortunately, many of the existing predictors do not find the compromise with false-positive prediction rates in the race for full coverage of known sequence examples. The user of a prediction program is encouraged to cast a critical eye over the prediction algorithm and the prediction results:

(1) On the collection of the training/learning set(s): Did the authors analyze their training set(s) carefully? What is the quality of the data? Are they properly experimentally verified? Are the descriptions of the sequence entries in agreement with recent literature? Are there taxonomic differences and is the specificity of orthologues of the post-translationally modifying enzymes unchanged? Are their mutation analysis data available? How specific is the training set; e.g., a PTM can be introduced by different enzymes or different enzyme complexes with different requirements for substrate specificity. Is the training set specific for only one type of PTM pathway?

(2) What are the parameters of the prediction function? How many parameters does the prediction function include and how much learning data is backing them? Are there any biological mechanisms that can be used to explain the role of the parameters in the prediction model?

(3) Critical analysis of the prediction results: It is a good idea to test some experimentally verified examples with the predictor (as well as sequentially similar antiexamples, e.g., from mutations that abolish the capability for the PTM) and to analyze the quality of prediction. An important question is whether the prediction result is in agreement with the existing knowledge about the analyzed protein(s) and the mechanism of the respective PTM; e.g., only a restricted number of amino acids can serve as a cleavage and/or attachment site.

In the following text, we try summarize the state of the art of prediction of various PTMs from sequence. Emphasis is given to prediction tools that perform reliably, especially with low false-positive rates that have also been proven useful in finding new biological insight. A good predictor should have coverage of well-verified examples of 90% or better and, at the same time, generate false-positive prediction with a rate of, maximally, a few percent. Occasionally, we will turn to comparative analysis of predictors for the same PTM and highlight some of the points discussed above that distinguish good predictors from not so recommendable tools.

3. Predictors for Specific PTMS

3.1. Recognition of Targeting Signals with and Without Proteolytic Cleavage Sites

A number of N-terminal targeting signals, most notably the N-terminal signal peptide coding for extracellular export, are cleaved after recognition by the respective enzyme complex during the translocation process. In these cases, signal recognition occurs cotranslationally at the unfolded protein; thus, the concept of embedding in linker-type segments is not applicable here. The state of the art is represented by SignalP (23), Phobius (24), and SPOCTOPUS for the signal peptide and TargetP (25) for the organelle-specific predictors.

Interestingly, the concept of a central motif surrounded by linkers does work also for translocation signals that are recognized by receptor proteins posttranslationally such as the peroxisomal translocation signal 1 (PTS1) (26, 27).

The respective WWW servers can be found at the following sites:

SignalP (23)	http://www.cbs.dtu.dk/services/SignalP
ChloroP (25)	http://www.cbs.dtu.dk/services/ChloroP
TargetP (25)	http://www.cbs.dtu.dk/services/TargetP
Phobius (24)	http://phobius.sbc.su.se
PeroPS/ PTS1 (26)	http://mendel.imp.ac.at/mendeljsp/sat/ pts1/PTS1predictor.jsp

3.2. Lipid PTMs

The number of enzymes responsible for several lipid PTMs (GPI lipid, farnesyl, geranylgeranyl, and myristoyl anchors) among eukaryotes is relatively small and, typically, there is just one enzyme species for a given modification. Thus, all known targets can be assigned to this enzyme, a considerable shortcut in the assessment of the available data on lipid modified proteins. Quite reliable prediction tools are available for GPI lipid anchoring (19, 28–32), myristoyl (20, 21, 33–35) and prenyl anchors (36–39).

The range of palmitoyltransferases is less well understood and the assignment of protein substrates to the various enzymatic entities is only at the beginning. At best, heuristic rules are reasonably applied to palmitoylation sites prediction without great reliability (40).

The respective WWW servers can be found at the following sites:

big-PI/GPI animals (19)	http://mendel.imp.ac.at/gpi/gpi_server. html
big-PI/GPI plants (31)	http://mendel.imp.ac.at/gpi/plant_ser ver.html
big-PI/GPI fungi (32)	http://mendel.imp.ac.at/gpi/fungi_ser ver.html
MyrPS/NMT (20)	http://mendel.imp.ac.at/myristate/ SUPLpredictor.html
PrePS (36)	http://mendel.imp.ac.at/sat/PrePS
CSS_palm (40)	http://bioinformatics.lcd-ustc.org/ css_palm

The GPI lipid anchor prediction algorithm is the first out of a series of PTM prediction efforts based on the simplified binding site model (3, 28, 32). The animal protein big-PI predictor (19) and its taxonomic derivatives for fungi (32) and plants (31) are a good starting point to evaluate alternative and more recent predictor developments in the light of the requirements described in the sections above.

The aim of the programs is to predict suitable substrate protein candidates for glycosyl-phosphatidylinositol (GPI) anchoring and, in the case of a hit, to predict the attachment site of the anchor. GPI lipid anchoring is a PTM that tethers eukaryotic proteins and their viruses to the extracellular leaflet of the cell membrane. The biosynthesis of GPI anchored proteins is a very complex, multilevel process that takes place in the endoplasmatic reticulum (ER) (3, 41). A transamidase complex cleaves a C-terminal propeptide from the substrate protein and attaches a pre-synthesized GPI moiety to the cleavage site. The cleavage and attachment site is called ω-site. Typically, the whole construct is exported to the extracellular side of the cell membrane, but it is not excluded that some of the GPI anchored proteins remain in the ER or other compartments of the vesicular system during their whole lifecycle.

The GPI-SOM predictor (42) relies on automatic learning from a complex set of sequences annotated as transamidase substrates. This learning set has dramatic deficiencies:

(i) The authors claim that the positive training and evaluation sets consist of proteins that had been experimentally shown to be GPI anchored. These include 110 proteins regardless of taxa from the whole eukaryote kingdom selected via Entrez from Genbank, supplemented with a set of 248 GPI proteins from *Arabidopsis thaliana* (43). Unfortunately, the data are of different quality with respect to the experimental verification. It should be noted that it is not sufficient to know the fact of GPI anchoring of a protein (typically it is only verified by a phospholipase C release test), but it is also important to know the exact ω-cleavage site. At least for the Borner *A. thaliana* set, this is not the case. The authors of GPI-SOM ignore

the fact that there are taxonomic differences in the C-terminal GPI sequence motif (44) that are distinguished by the transamidase complexes of various taxa. (ii) There is a large body of data available from GPI motif mutation analyses spread over the literature and the majority of which has been compiled in two surveys for metazoa (19) and fungi (32). These data contain both an experimentally verified positive collection and a verified negative sequence set, both with more than 100 cases. The mutations are an especially demanding test for the GPI lipid anchor predictor since a single point mutation can abolish the capability for GPI lipid anchoring completely.

It was already mentioned that automated learning procedures require learning sets of especially high quality since they are unable to recognize errors in the sequence collections. The final GPI-SOM prediction program was implemented as a Kohonen SOM (neural network of the Kohonen type, also termed self-organizing map) with an input layer of 44 neurons (thus, introducing hundreds of adjustable parameters). Neuronal networks are a very common technique for classifying data sets but, with noise in the learning set, the predictor will generate systematic errors at the output layer. Due to their technical approach, the authors could not make use of any mechanistic insights that govern the recognition process of substrates by the transamidase. The following examples highlight the problems:

(1) Human complement decay-accelerating factor (CD55 or DAF, accession number P08174) is a protein that was tested to be GPI anchored even by different research groups. It was used for mutational studies (44, 45), the ω-site is well known. GPI-SOM predictor is not able to classify this protein as a GPI lipid modified protein and does not find even a trend among the various mutations.

(2) If one tests the 248 *A. thaliana* sequences (43), which are a part of the positive training set for GPI-SOM, then 245 out of the 248 sequences are listed as GPI-SOM positive predictions (240 as GPI-anchored proteins in combination with SignalP prediction). But a closer analysis of the prediction results reveals that most of the predicted ω-cleavage sites are wrong. Only a limited set of amino acid residues (Ala, Cys, Asp, Gly, Asn, and Ser) is known to be capable to serve as an ω-site (43, 45, 46). In the case of the predicted 245 GPI candidates, only 97 (39.6%) predicted cleavage sites are compatible with this rule. This means that the majority (at least 60.4%) of the ω-sites is wrongly predicted by GPI-SOM. Interestingly, GPI-SOM favors proline at the ω-site position (37 proteins), followed by serine (35) and glycine (25). According to mutational studies [see mutation database (19)], proline is absolutely disfavored from occupying the ω-site position. Thus, GPI-SOM predictions are in conflict with basic enzymology of the process considered.

(3) The GPI-SOM authors predict almost all of the proteins of a SWISS-PROT data set to be GPI anchored (**Table 2**-e in their paper). What they did not discuss is that this data set contains quite a number of doubtfully annotated protein sequences [**Table I** in Ref. (28)] and one might rightfully expect that these proteins will not carry a GPI lipid anchor.

(4) The authors of GPI-SOM found that, among transmembrane proteins with an N-terminal signal leader peptide plus a hydrophobic C-terminus – the proteins most closely resembling GPI-anchored ones – the false-positive prediction rate is around 30%, a rate that disqualifies the tool for experimental planning.

The authors of GPI-SOM combine their own prediction module with the SignalP (47) program (surprisingly, without acknowledgment of the SignalP authors). Using the SignalP program, the GPI-SOM authors reduce their prediction rate in *S. cerevisiae* from 438 positives (out of 5,864 proteins) to only 121 positives. The remaining 307 hits are ruled out as positives with the argument that a protein cannot be GPI anchored without having an N-terminal signal leader peptide. In the general context, this conclusion is very problematic. First, alternative export mechanisms to the ER appear to exist and example proteins exported via the alternative pathway with a GPI lipid anchor are known (48). Second, SignalP is a prediction program having its own error rates. Third, the quality of the sequence data in large sequence sets, especially in high-throughput sequence data, is limited. It is possible that the N-termini and/or C-termini of sequences are missing (49). To put in a nutshell, the GPI-SOM predictor is not a useful tool for analyzing large-scale data because the false-positive prediction rate is in an unacceptable range. The tool can also not be used for the prediction of ω-sites because it is incapable to do so.

There is another recently published GPI predictor called FragAnchor (50). It is based on the tandem use of a neuronal network and a hidden Markov model. We have tested both our mutation set containing proteins and mutated proteins for which GPI anchoring has been experimentally shown [positive mutation set – 188 sequences (19)] and a respective negative mutation set containing protein mutations, which were shown to be not compatible with GPI anchoring [108 sequences (19)] with the FragAnchor program. The prediction of the positive mutation set appears very reasonable, only 10 sequences (5.3%) are not predicted as true candidates for GPI lipid modification. But the prediction of the negative mutation set seems to be problematic. Only 22 out of 108 sequences (20.4%) were predicted as not GPI anchored; thus, the false-positive prediction rate is in the order of 80%. Indeed, it is relatively easy to create a predictor with high sensitivity, but the real challenging task is to keep the rate of false-positive predictions in a low range.

The most recent GPI-anchor predictor, PredGPI (51), is an especially negative example for using machine learning methods (in this case, with SVM) due to its very poor training set. PredGPI is trained on a dataset of only 26 protein sequences just ignoring the plenty of data that is available these days. But the paper is also remarkable in other respects. If one enters the small learning set of 26 sequences into the PredGPI Web server interface (http://gpcr2.biocomp.unibo.it/predgpi), one will get a really surprising result that greatly differs from what the authors show in **Table 3** in their paper. According to that table, PredGPI predicts all of the 26 proteins as GPI anchored. Unfortunately, the Web server comes up with a different result: seven of the proteins are predicted as GPI candidates with high probability, 13 proteins are predicted as probable GPI anchored proteins, one protein is indicated as suitable for GPI anchoring with low probability but five proteins are predicted as not GPI anchored at all.

3.3. Phosphorylation Prediction Tools

Protein phosphorylation is one of the most common and one of the most studied posttranslational modifications. In the case of eukaryotic proteins, kinases, the enzymes that catalyze protein phosphorylation, transfer a phosphate (PO_4) group from adenosine triphosphate (ATP) to serine, threonine, or tyrosine amino acids, thereby generating adenosine diphosphate (ADP). In addition, phosphorylation of basic amino acids (histidine, arginine, lysine) is possible in prokaryotic proteins. Today, more than 500 protein kinases have been identified in the human genome, and they modify at least one-third of all human protein species (52).

Reversible phosphorylation is a ubiquitous regulatory mechanism that controls a large variety of biological processes such as cell growth and differentiation, proliferation, and apoptosis. Typically, protein kinases, phosphatases (dephosphorylation enzymes), and their substrates are organized in very complex regulatory networks. Proteins can work as a trigger or a switch in such a system depending on their phosphorylation status.

Over the past few years, phosphoproteomic mass spectrometry (MS) has replaced the traditional methods for measuring protein phosphorylation (mutational analysis, Edman degradation chemistry on phosphopeptides). Thousands of phosorylation sites have been identified for thousands of proteins. But in most cases, the information which kinase is responsible for the phosphorylation of a certain amino acid in a certain protein is missing. Of course, this fact impedes the development of powerful predictors for protein phosphorylation sites. There are several databases such as Phospho.ELM (10), PhosphoSitePlus (www.phosphosite.org), and NetworKIN (53) trying to reduce the gap between the number of experimentally identified phosphorylation sites and the number of phosphorylation sites for which the modifying kinase is known.

Unfortunately, most of the phosphorylation predictors described in the literature are not created for substrates of a specific kinase but for quite a number of unrelated protein substrates (or, in the worst case, for all possible Ser-, Thr- or Tyr-kinases). Since the kinases have largely differing substrate specificities, naturally, these predictors struggle with the problem of predicting a huge number of potential phosphorylation sites (many of them being false positives). This leads to the situation that most of these predicted sites are misclassified. Therefore, these prediction tools cannot be recommended for planning of experiments or proteome-wide in silico scans. It is a fundamental quality feature of a predictor to aim at predicting substrates for specific kinases or a class of kinases with essentially identical substrate specificity.

There are quite a number of different phosphorylation site predictors available and the list below is far from complete; yet, only a few can be recommended for usage:

pkaPS (54)	http://mendel.imp.ac.at/sat/pkaPS
DIPHOS (8)	http://www.ist.temple.edu/DIPHOS
KinasePhos2.0 (55)	http://KinasePhos2.mbc.nctu.edu.tw
NetPhosK (56)	http://www.cbs.dtu.dk/services/NetPhosK
NetPhos (57)	http://www.cbs.dtu.dk/services/NetPhos
NetPhosYeast (58)	http://www.cbs.dtu.dk/services/NetPhosYeast
PredPospho (59)	http://pred.ngri.re.kr/PredPhospho.htm
GPS2.0 (60)	http://bioinformatics.lcd-ustc.org/gps2/down.php
Predikin (61)	http://predikin.biosci.uq.edu.au
Scansite2.0 (62)	http://scansite.mit.edu

The overarching problem in phosphorylation site prediction is the false-positive rate and only the more recent developments approach this problem seriously. Exact measurement of the selectivity is not an easy issue either (see (1, 4, 54, 63) for discussion).

NetPhos (57) was one of the first approaches which outperformed simple PROSITE-like (6) searches, but this is a neural network predictor for substrates of any kinase, a situation contradicting the known biological mechanisms of protein substrate selection. NetPhosK (56) is a kinase-specific extension to the general NetPhos method. Scansite 2.0 (62) uses position-specific scoring matrixes (PSSM) to predict phosphorylation motifs for 62 different kinases. In this site model, only the central motif that directly interacts with

the active site of the kinase is considered for discrimination from nonsites; it appears that this is a major reason for the high false-positive rate. Using support vector machines (SVMs), PredPhospho (59) and KinasePhos2.0 (55), attempts to predict phosphorylation sites and the type of kinase that acts at each site.

Typically, prediction algorithms which are based on PROSITE-like patterns generate a huge amount of false-positive predictions. The same is true for profile searches with short motifs. In the case of GPS2.0 (60), the authors even increase the possibility of a false-positive prediction by reducing the strength of the profile they search with (to achieve higher coverage of known examples). The final score for a query (phosphorylation) site is calculated as an average value of substitution scores which are obtained from the comparison of the query motif to all known phosphorylation sites in the training set. The substitution scores rely on a slightly modified BLOSUM62 substitution matrix which contains negative values for disfavored substitutions, e.g., replacing alanine with arginine is evaluated by -2, and positive (or zero) values for more advantageous substitutions, e.g. replacing alanine with serine is scored with +1. Surprisingly, the authors of GPS2.0 simplify the substitution scores in that way that they ignore negative matrix values (just adding zero instead) and, therefore, do not penalize the occurrence of disfavored amino acid types in the query motif at all.

pkaPS (54) is an algorithm for the prediction of protein kinase A (PKA) phosphorylation sites. The pkaPS authors studied the PKA recognition motif in great detail and created a capable prediction module (sensitivity ~ 96% at a specificity ~94%) based on a simplified substrate protein-binding model. Comparison of the prediction performances of pkaPS to the previously mentioned phosphorylation site predictors (PKA prediction only) shows that pkaPS is the most powerful standalone predictor at present (54). DIPHOS (8) is a predictor that is conceptionally most closely related to pkaPS. The authors of the predictor programs try to improve the discrimination between phosphorylation and non-phosphorylation sites by using disorder information (sequence complexity, hydrophobicity, net charge) in addition to position-specific amino acid frequencies. The authors rely on the hypothesis that protein phosphorylation predominantly occurs within intrinsically disordered protein regions.

3.4. Glycosylation   Many proteins in eukaryotic cells are glycoproteins. Glycosylation is an enzymatic process that covalently links oligosaccharide chains to certain amino acids. In contrast, glycation is a nonenzymatic reaction that adds sugar molecules, such as glucose or fructose, to lysine residues (56). Glycosylation is known to be important for a large variety of functions such as protein folding and stability, manipulation of a protein's cellular localization, and trafficking as well as cell–cell interactions.

There are four main categories of glycosylation: (i) *N*-linked glycosylation (addition of the sugars to the amino group (NH_2) of an asparagine of secreted or membrane-bound proteins), (ii) *O*-linked glycosylation (addition of the sugar to the hydroxyl group (OH) of a serine or a threonine), (iii) C-mannosylation (addition of a mannose sugar to tryptophan), and (iv) GPI anchors (see lipid anchor modifications) (56).

The number of glycosyltransferases is huge; many of them are not explored and, compared with the situation of phosphorylation substrates, the relationship between the specific glycosyltransferases and their substrates is generally even less studied than for the kinases. As a result, glycosylation prediction remains in a very unsatisfactory state and the computerized protein sequence-based predictors have low predictive power. Among the published literature and the WWW-available sites, the following servers belong to the more reasonable ones:

NetNGlyc (64)	http://www.cbs.dtu.dk/services/NetNGlyc
NetOGlyc (65)	http://www.cbs.dtu.dk/services/NetOGlyc
OGPET (no reference)	http://ogpet.utep.edu/OGPET
CKSSAP_OGlySite (66)	http://bioinformatics.cau.edu.cn/zzd_lab/CKSAAP_OGlySite
Oglyc (67)	http://www.biosino.org/Oglyc
DictyOGlyc (68)	http://www.cbs.dtu.dk/services/DictyOGlyc
YinOYang (64)	http://www.cbs.dtu.dk/services/YinOYang
NetCGlyc (69)	http://www.cbs.dtu.dk/services/NetCGlyc
NetGlycate (70)	http://www.cbs.dtu.dk/services/netGlycate

Traditionally, *N*-glycosylation is understood as a process of adding a glycan structure to the substrate protein in the endoplasmic reticulum. The NetNglyc server predicts *N*-glycosylation sites in human proteins using artificial neural networks that examine the sequence context of Asn-Xaa-Ser/Thr motifs (64).

O-glycosylation is more diverse from the viewpoint of subcellular localizations of targets. They can be secreted or become transmembrane proteins, also at Golgi vesicle membranes. In addition, *O*-glycosylation has been shown to occur in the nucleus and cytoplasm of cells (56). NetOGlyc(65) produces neural

network predictions of mucin-type GalNAc *O*-glycosylation sites in mammalian proteins. It is a combination of predictions from the best overall network and the best isolated (single) site network. The best overall network relies on amino acid composition, averaged surface accessibility predictions together with substitution matrix profile encoding of the protein sequence. Interestingly, the authors find that a glycosylated serine or threonine is less likely to be precisely conserved than a nonglycosylated one. Unfortunately, the predictive power is not great. The authors say that the addition of GalNAc to serine and threonine is mediated by at least 14 different UDP-GalNAc:polypeptide *N*-acetyl-galactosaminyl-transferase (71) with, most likely, differing specificity. Another WWW server for the same mucin-type glycosylation is OGPET (see above for WWW URL) with a simple pattern-matching procedure.

The authors of CKSSAP_OGlySite (66) suggest that the usage of the composition of k-spaced amino acid pairs evaluated with an SVM reaches prediction rates that are better than those of OGlyc (67), another SVM-based tool that evaluates the presence of 188 physical properties in the motif region; yet, the problem of glycosyltransferase preference for certain types of protein substrates remains untouched in both papers. When we submitted EGFR_-HUMAN (P00533, annotated as *N*-linked glycoprotein), the tool NetOGlyc does not predict any sites (apparently the correct prediction), OGPET generates one prediction (serine 921), CKSAAP delivers even 11 sites (residues 1025, 1026, 1028, 1032, 1036, 1037, 1039, 1041, 1045, 1141, 1145), and Oglyc finds 6 sites (threonine 273, serines 921, 924, 1036, and 1042, threonine 1046). It should be noted that OGlyc reports the results shifted by one residue, apparently a small bug. CSKAAP requires the user to wait (in our case, a day) to receive an E-mail for the prediction results.

DictyOGlyc (68) is a specialized predictor for *O*-α-GlcNAc sites in *Dictostelium discoideum*-based neuronal networks that explore the immediate sequence context and the predicted surface accessibility of potential sites. Unfortunately, there are only 39 sequences in the learning set.

The YinOYang neuronal network produces predictions for *O*-βGlcNAc attachment sites in eukaryotic protein sequences. This tool should be applied in parallel with NetPhos. The idea of "Ying-Yang"-sites suggests that glycosylation and phosphorylation of the same serine or threonine hydroxyl group are possible and it should be predicted by both methods.

C2-mannosylation of tryptophane is a recently discovered new form of glycosylation of proteins (72) that is introduced into the unfolded protein, apparently, during import into the endoplasmic reticulum. There are almost 70 experimentally verified examples with the WxxW (or the WxxC) motif. The NetCGlyc (69)

neuronal network explores an 21-residue sequence window as input window with the potentially modified tryptophane at the center. The NetGlycate (70) neuronal network attempts to find sites for nonenzymatic glycation sites.

4. Conclusions

The experience shows that the prediction tools function best for those PTMs, for which the biology is best understood and where the prediction algorithm can mimic the recognition of the substrate by the modifying enzyme. In the development of several lipid anchor prediction tools [big-PI (19, 31, 32), MyrPS (20), PrePS (36)], such considerations had been central and, not surprisingly, the predictions have become similarly reliable as the collection of sequence families within the homology concept. Other tools that can be recommended for general use are SIGNALP (23) and, with somewhat more reservations, pkaPS (54). The PeroPS (26, 27) tool for the recognition of the PTS1 translocation signal is constructed following similar principles and appears also safe for usage.

All other PTM prediction tools have considerable problems with false-positive predictions and are problematic for the application in the context of uncharacterized sequence studies. The enormous growth of the number of bioinformatics teams and researchers and the pressure on them to publish has led to a proliferation of journals, inflation of the number of published articles and a dramatic, collective decline of impact factors for specialized bioinformatics journals. Over the field as a whole, the standard of prediction tool development manuscripts has not improved over the last decade neither with respect to biological considerations nor technical perfection; even standards that were generally accepted such as justification of the amount of parameters by the amount of data, proper cross-validation tests, or at least the attempt of collecting all available experimental data for learning of the scoring function's parameters are frequently not complied with.

At the end, we wish to highlight the problem between the occurrences of sequence regions in a substrate protein that are recognized by the prediction tools as fit for receiving a specific PTM and the observation of this PTM in vivo for the same protein. In the first situation, the focus is whether the protein can be the target of a productive interaction with the modifying enzyme and this question can, for example, be studied in an in vitro assay. It is another question whether the two reaction partners ever come together in vivo (and this might depend on physiological or

pathological conditions, be influenced by mutations, etc.). Once this conception is agreed upon, one might predict the occurrence of sites that are fully functional for a PTM (or a translocation signal); yet, they remain silent in normal physiology. Indeed, such sites do exist as was shown in an experimental test (63).

References

1. Eisenhaber, F., Eisenhaber, B., Maurer-Stroh, S. (2003) Prediction of Post-translational modifications from amino acid sequence: problems, pitfalls, methodological hints. In Andrade, M. M. (ed.), *Bioinformatics and Genomes: Current Perspectives.* Horizon Scientific Press, Wymondham, pp. 81–105.

2. Eisenhaber, F. (2006) Prediction of protein function: two basic concepts and one practical recipe. In Eisenhaber, F. (ed.) *Discovering Biomolecular Mechanisms with Computational Biology*, 1st edition.Landes Biosciences and Eurekah.com, Georgetown. Chapter 3, pp. 39–54.

3. Eisenhaber, B., Eisenhaber, F. (2007) Post-translational modifications and subcellular localization signals: indicators of sequence regions without inherent 3D structure? *Curr Protein Pept Sci* 8, 197–203.

4. Eisenhaber, B., Eisenhaber, F., Maurer-Stroh, S., Neuberger, G. (2004) Prediction of sequence signals for lipid post-translational modifications: insights from case studies. *Proteomics* 4, 1614–1625.

5. Puntervoll, P., Linding, R., Gemund, C., Chabanis-Davidson, S., Mattingsdal, M., Cameron, S., Martin, D. M., Ausiello, G., Brannetti, B., Costantini, A., et al. (2003) ELM server: A new resource for investigating short functional sites in modular eukaryotic proteins. *Nucleic Acids Res* 31, 3625–3630.

6. Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., de, C. E., Langendijk-Genevaux, P. S., Pagni, M., Sigrist, C. J. (2006) The PROSITE database. *Nucleic Acids Res* 34, D227–D230.

7. Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., Cuche, B. A., de, C. E., Lachaize, C., Langendijk-Genevaux, P. S., Sigrist, C. J. (2008) The 20 years of PROSITE. *Nucleic Acids Res* 36, D245–D249.

8. Iakoucheva, L. M., Radivojac, P., Brown, C. J., O'Connor, T. R., Sikes, J. G., Obradovic, Z., Dunker, A. K. (2004) The importance of intrinsic disorder for protein phosphorylation. *Nucleic Acids Res* 32, 1037–1049.

9. Kiemer, L., Bendtsen, J. D., Blom, N. (2005) NetAcet: prediction of N-terminal acetylation sites. *Bioinformatics* 21, 1269–1270.

10. Diella, F., Gould, C. M., Chica, C., Via, A., Gibson, T. J. (2008) Phospho.ELM: a database of phosphorylation sites–update 2008. *Nucleic Acids Res* 36, D240–D244.

11. Liu, H., Yang, J., Wang, M., Xue, L., Chou, K. C. (2005) Using fourier spectrum analysis and pseudo amino acid composition for prediction of membrane protein types. *Protein J* 24, 385–389.

12. Cai, Y. D., Chou, K. C. (2006) Predicting membrane protein type by functional domain composition and pseudo-amino acid composition. *J Theor Biol* 238, 395–400.

13. Chou, K. C., Cai, Y. D. (2005) Prediction of membrane protein types by incorporating amphipathic effects. *J Chem Inf Model* 45, 407–413.

14. Wang, M., Yang, J., Liu, G. P., Xu, Z. J., Chou, K. C. (2004) Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition. *Protein Eng Des Sel* 17, 509–516.

15. Cai, Y. D., Zhou, G. P., Chou, K. C. (2003) Support vector machines for predicting membrane protein types by using functional domain composition. *Biophys J* 84, 3257–3263.

16. Cai, Y. D., Liu, X. J., Chou, K. C. (2001) Artificial neural network model for predicting membrane protein types. *J Biomol Struct Dyn* 18, 607–610.

17. Chou, K. C., Elrod, D. W. (1999) Prediction of membrane protein types and subcellular locations. *Proteins* 34, 137–153.

18. O'Connor, E., Eisenhaber, B., Dalley, J., Wang, T., Missen, C., Bulleid, N., Bishop, P. N., Trump, D. (2005) Species specific membrane anchoring of nyctalopin, a small leucine-rich repeat protein. *Hum Mol Genet* 14, 1877–1887.

19. Eisenhaber, B., Bork, P., Eisenhaber, F. (1999) Prediction of potential GPI-

modification sites in proprotein sequences. *J Mol Biol* 292, 741–758.

20. Maurer-Stroh, S., Eisenhaber, B., Eisenhaber, F. (2002) N-terminal N-myristoylation of proteins: prediction of substrate proteins from amino acid sequence. *J Mol Biol* 317, 541–557.

21. Maurer-Stroh, S., Eisenhaber, B., Eisenhaber, F. (2002) N-terminal N-myristoylation of proteins: refinement of the sequence motif and its taxon-specific differences. *J Mol Biol* 317, 523–540.

22. Bologna, G., Yvon, C., Duvaud, S., Veuthey, A. L. (2004) N-Terminal myristoylation predictions by ensembles of neural networks. *Proteomics* 4, 1626–1632.

23. Bendtsen, J. D., Nielsen, H., von, H. G., Brunak, S. (2004) Improved prediction of signal peptides: SignalP 3.0. *J Mol Biol* 340, 783–795.

24. Kall, L., Krogh, A., Sonnhammer, E. L. (2007) Advantages of combined transmembrane topology and signal peptide prediction – the Phobius web server. *Nucleic Acids Res* 35, W429–W432.

25. Emanuelsson, O., Brunak, S., von, H. G., Nielsen, H. (2007) Locating proteins in the cell using TargetP, SignalP and related tools. *Nat Protoc* 2, 953–971.

26. Neuberger, G., Maurer-Stroh, S., Eisenhaber, B., Hartig, A., Eisenhaber, F. (2003) Prediction of peroxisomal targeting signal 1 containing proteins from amino acid sequence. *J Mol Biol* 328, 581–592.

27. Neuberger, G., Maurer-Stroh, S., Eisenhaber, B., Hartig, A., Eisenhaber, F. (2003) Motif refinement of the peroxisomal targeting signal 1 and evaluation of taxon-specific differences. *J Mol Biol* 328, 567–579.

28. Eisenhaber, B., Bork, P., Eisenhaber, F. (1998) Sequence properties of GPI-anchored proteins near the omega-site: constraints for the polypeptide binding site of the putative transamidase. *Protein Eng* 11, 1155–1161.

29. Eisenhaber, B., Bork, P., Yuan, Y., Loffler, G., Eisenhaber, F. (2000) Automated annotation of GPI anchor sites: case study C. elegans. *Trends Biochem Sci* 25, 340–341.

30. Eisenhaber, B., Bork, P., Eisenhaber, F. (2001) Post-translational GPI lipid anchor modification of proteins in kingdoms of life: analysis of protein sequence data from complete genomes. *Protein Eng* 14, 17–25.

31. Eisenhaber, B., Wildpaner, M., Schultz, C. J., Borner, G. H., Dupree, P., Eisenhaber, F. (2003) Glycosylphosphatidylinositol lipid

anchoring of plant proteins. Sensitive prediction from sequence- and genome-wide studies for Arabidopsis and rice. *Plant Physiol* 133, 1691–1701.

32. Eisenhaber, B., Schneider, G., Wildpaner, M., Eisenhaber, F. (2004) A sensitive predictor for potential GPI lipid modification sites in fungal protein sequences and its application to genome-wide studies for Aspergillus nidulans, Candida albicans, Neurospora crassa, Saccharomyces cerevisiae and Schizosaccharomyces pombe. *J Mol Biol* 337, 243–253.

33. Maurer-Stroh, S., Eisenhaber, F. (2004) Myristoylation of viral and bacterial proteins. *Trends Microbiol* 12, 178–185.

34. Maurer-Stroh, S., Gouda, M., Novatchkova, M., Schleiffer, A., Schneider, G., Sirota, F. L., Wildpaner, M., Hayashi, N., Eisenhaber, F. (2004) MYRbase: analysis of genome-wide glycine myristoylation enlarges the functional spectrum of eukaryotic myristoylated proteins. *Genome Biol* 5, R21.

35. Benetka, W., Mehlmer, N., Maurer-Stroh, S., Sammer, M., Koranda, M., Neumuller, R., Betschinger, J., Knoblich, J. A., Teige, M., Eisenhaber, F. (2008) Experimental testing of predicted myristoylation targets involved in asymmetric cell division and calcium-dependent signalling. *Cell Cycle* 7, 3709–3719.

36. Maurer-Stroh, S., Eisenhaber, F. (2005) Refinement and prediction of protein prenylation motifs. *Genome Biol* 6, R55.

37. Maurer-Stroh, S., Koranda, M., Benetka, W., Schneider, G., Sirota, F. L., Eisenhaber, F. (2007) Towards complete sets of farnesylated and geranylgeranylated proteins. *PLoS Comput Biol* 3, e66.

38. Benetka, W., Koranda, M., Maurer-Stroh, S., Pittner, F., Eisenhaber, F. (2006) Farnesylation or geranylgeranylation? Efficient assays for testing protein prenylation in vitro and in vivo. *BMC Biochem* 7, 6.

39. Benetka, W., Koranda, M., Eisenhaber, F. (2006) Protein prenylation: an (almost) comprehensive overview on discovery history, enzymology and significance in physiology and disease. *Chemical Monthly* 137, 1241–1281.

40. Ren, J., Wen, L., Gao, X., Jin, C., Xue, Y., Yao, X. (2008) CSS-Palm 2.0: an updated software for palmitoylation sites prediction. *Protein Eng Des Sel* 21, 639–644.

41. Eisenhaber, B., Maurer-Stroh, S., Novatchkova, M., Schneider, G., Eisenhaber, F.

(2003) Enzymes and auxiliary factors for GPI lipid anchor biosynthesis and post-translational transfer to proteins. *Bioessays* 25, 367–385.

42. Fankhauser, N., Maser, P. (2005) Identification of GPI anchor attachment signals by a Kohonen self-organizing map. *Bioinformatics* 21, 1846–1852.

43. Borner, G. H., Lilley, K. S., Stevens, T. J., Dupree, P. (2003) Identification of glycosylphosphatidylinositol-anchored proteins in Arabidopsis. A proteomic and genomic analysis. *Plant Physiol* 132, 568–577.

44. Moran, P., Caras, I. W. (1994) Requirements for glycosylphosphatidylinositol attachment are similar but not identical in mammalian cells and parasitic protozoa. *J Cell Biol* 125, 333–343.

45. Caras, I. W., Weddell, G. N., Williams, S. R. (1989) Analysis of the signal for attachment of a glycophospholipid membrane anchor. *J Cell Biol* 108, 1387–1396.

46. Udenfriend, S., Kodukula, K. (1995) How glycosylphosphatidylinositol-anchored membrane proteins are made. *Annu Rev Biochem* 64, 563–591.

47. Emanuelsson, O., Brunak, S., von, H. G., Nielsen, H. (2007) Locating proteins in the cell using TargetP, SignalP and related tools. *Nat Protoc* 2, 953–971.

48. Howell, S., Lanctot, C., Boileau, G., Crine, P. (1994) A cleavable N-terminal signal peptide is not a prerequisite for the biosynthesis of glycosylphosphatidylinositol-anchored proteins. *J Biol Chem* 269, 16993–16996.

49. Nagy, A., Hegyi, H., Farkas, K., Tordai, H., Kozma, E., Banyai, L., Patthy, L. (2008) Identification and correction of abnormal, incomplete and mispredicted proteins in public databases. *BMC Bioinformatics* 9, 353.

50. Poisson, G., Chauve, C., Chen, X., Bergeron, A. (2007) FragAnchor: a large-scale predictor of glycosylphosphatidylinositol anchors in eukaryote protein sequences by qualitative scoring. *Genomics Proteomics Bioinformatics* 5, 121–130.

51. Pierleoni, A., Martelli, P. L., Casadio, R. (2008) PredGPI: a GPI-anchor predictor. *BMC Bioinformatics* 9, 392.

52. Johnson, S. A., Hunter, T. (2005) Kinomics: methods for deciphering the kinome. *Nat Methods* 2, 17–25.

53. Linding, R., Jensen, L. J., Pasculescu, A., Olhovsky, M., Colwill, K., Bork, P., Yaffe, M. B., Pawson, T. (2008) NetworKIN: a resource for exploring cellular

phosphorylation networks. *Nucleic Acids Res* 36, D695–D699.

54. Neuberger, G., Schneider, G., Eisenhaber, F. (2007) pkaPS: prediction of protein kinase A phosphorylation sites with the simplified kinase-substrate binding model. *Biol Direct* 2, 1.

55. Wong, Y. H., Lee, T. Y., Liang, H. K., Huang, C. M., Wang, T. Y., Yang, Y. H., Chu, C. H., Huang, H. D., Ko, M. T., Hwang, J. K. (2007) KinasePhos 2.0: a web server for identifying protein kinase-specific phosphorylation sites based on sequences and coupling patterns. *Nucleic Acids Res* 35, W588–W594.

56. Blom, N., Sicheritz-Ponten, T., Gupta, R., Gammeltoft, S., Brunak, S. (2004) Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence. *Proteomics* 4, 1633–1649.

57. Blom, N., Gammeltoft, S., Brunak, S. (1999) Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. *J Mol Biol* 294, 1351–1362.

58. Ingrell, C. R., Miller, M. L., Jensen, O. N., Blom, N. (2007) NetPhosYeast: prediction of protein phosphorylation sites in yeast. *Bioinformatics* 23, 895–897.

59. Kim, J. H., Lee, J., Oh, B., Kimm, K., Koh, I. (2004) Prediction of phosphorylation sites using SVMs. *Bioinformatics* 20, 3179–3184.

60. Xue, Y., Ren, J., Gao, X., Jin, C., Wen, L., Yao, X. (2008) GPS 2.0: Prediction of kinase-specific phosphorylation sites in hierarchy. *Mol Cell Proteomics* 7, 1598–1608.

61. Saunders, N. F., Brinkworth, R. I., Huber, T., Kemp, B. E., Kobe, B. (2008) Predikin and PredikinDB: a computational framework for the prediction of protein kinase peptide specificity and an associated database of phosphorylation sites. *BMC Bioinformatics* 9, 245.

62. Obenauer, J. C., Cantley, L. C., Yaffe, M. B. (2003) Scansite 2.0: Proteome-wide prediction of cell signaling interactions using short sequence motifs. *Nucleic Acids Res* 31, 3635–3641.

63. Neuberger, G., Kunze, M., Eisenhaber, F., Berger, J., Hartig, A., Brocard, C. (2004) Hidden localization motifs: naturally occurring peroxisomal targeting signals in non-peroxisomal proteins. *Genome Biol* 5, R97.

64. Gupta, R., Brunak, S. (2002) Prediction of glycosylation across the human proteome and the correlation to protein function. *Pac Symp Biocomput* 310–322.

65. Julenius, K., Molgaard, A., Gupta, R., Brunak, S. (2005) Prediction, conservation analysis, and structural characterization of mammalian mucin-type O-glycosylation sites. *Glycobiology* 15, 153–164.

66. Chen, Y. Z., Tang, Y. R., Sheng, Z. Y., Zhang, Z. (2008) Prediction of mucin-type O-glycosylation sites in mammalian proteins using the composition of k-spaced amino acid pairs. *BMC Bioinformatics* 9, 101.

67. Li, S., Liu, B., Zeng, R., Cai, Y., Li, Y. (2006) Predicting O-glycosylation sites in mammalian proteins by using SVMs. *Comput Biol Chem* 30, 203–208.

68. Gupta, R., Jung, E., Gooley, A. A., Williams, K. L., Brunak, S., Hansen, J. (1999) Scanning the available Dictyostelium discoideum proteome for O-linked GlcNAc glycosylation sites using neural networks. *Glycobiology* 9, 1009–1022.

69. Julenius, K. (2007) NetCGlyc 1.0: prediction of mammalian C-mannosylation sites. *Glycobiology* 17, 868–876.

70. Johansen, M. B., Kiemer, L., Brunak, S. (2006) Analysis and prediction of mammalian protein glycation. *Glycobiology* 16, 844–853.

71. Wang, H., Tachibana, K., Zhang, Y., Iwasaki, H., Kameyama, A., Cheng, L., Guo, J., Hiruma, T., Togayachi, A., Kudo, T. et al. (2003) Cloning and characterization of a novel UDP-GalNAc:polypeptide N-acetylgalactosaminyltransferase, pp-Gal-NAc-T14. *Biochem Biophys Res Commun* 300, 738–744.

72. Furmanek, A., Hofsteenge, J. (2000) Protein C-mannosylation: facts and questions. *Acta Biochim Pol* 47, 781–789.

Chapter 22

Protein Crystallizability

Pawel Smialowski and Dmitrij Frishman

Abstract

Obtaining well-diffracting crystals remains a major challenge in protein structure research. In this chapter, we review currently available computational methods to estimate the crystallization potential of a protein, to optimize amino acid sequences toward improved crystallization likelihood, and to design optimal crystal screen conditions.

Key words: protein crystallization, construct optimization, crystallization conditions, crystallization screen.

1. Introduction

1.1. Protein Crystallization

The study of structural properties of biological macromolecules is one of the most important avenues of contemporary biology. The availability of three-dimensional structures is an important prerequisite for understanding protein function (1) and ultimately for elucidating the inner workings of the living cell.

Protein crystallization followed by X-ray diffraction data collection is the method of choice for protein structure research. Upon crystallization, molecules form an ordered, solid array. Crystals typically start growing from supersaturated solutions in a process called "nucleation." Further crystal growth depends on the presence of a high number of identical or nearly identical molecules. For this reason the protein under study has to be sufficiently stable to be present in only one structural form. Physical properties of the molecular surface must allow the formation of a defect-free repetitive crystal lattice. Depending on the surface properties proteins can form crystals/lattices of different geometries even under the same crystallization conditions (2).

Except for very few examples (e.g., crystallines in the eye), cellular systems do not require protein crystallization to function properly. Considering the high protein concentration in the cell, it is feasible that evolution would select against unspecific protein–protein interactions that lead to aggregation or crystallization (3). Therefore, it is not surprising that protein crystallization under lab conditions has a relatively low success rate. The standard approach to crystallization requires sampling of physical and chemical conditions including temperature, pH, and ionic strength. Usually a vast number of different buffers, salts, and precipitating agents have to be tested (4). Small molecular cofactors or inhibitors can also play a crucial role in the crystallization process (5).

While the standard approach to crystallization is to search for successful crystallization conditions (2), it is also common to optimize the protein construct used for crystallization or replace the protein of interest by an ortholog with a higher crystallization probability. Many proteins which were recalcitrant to crystallization in wild-type form become tractable after mutating their sequence (6–13). As a consequence of construct optimization (e.g., removal of flexible loops, etc.), many structures deposited in the PDB [the databank of protein structures, (14)] cover only protein fragments or domains.

1.2. Structural Genomics

Structural genomics/proteomics is an international coordinated effort to determine atomic resolution three-dimensional structures of proteins at large scale in a high-throughput fashion.

The structural genomics pipeline consists of successive experimental stages from cloning up to structure determination and data deposition. The number of recalcitrant instances at each step is very high: on average, from 100 selected proteins only ∼3 yield three-dimensional structures deposited with the PDB databank (**Fig. 22.1**). The statistics obtained by structural genomics projects is also a good estimate of the success rate in structural biology in general.

This notoriously low success rate of structure determination stimulated the development of bioinformatics methods to select potentially tractable proteins, the so-called low hanging fruit. The ability to estimate a priori the prospect of a given protein to be experimentally tractable cannot be over appreciated. Even a minimal advance in this direction, improving the experimental success rate by just a few percentage points, would cause significant reduction of cost and possibly yield dozens of additional structures.

The systematic approach to data collection taken by structural genomics consortia gave rise to abundance of both positive and negative experimental data from all stages of the protein structure determination pipeline. This quickly growing corpus of experimental success and failure data creates a unique opportunity for retrospective data mining.
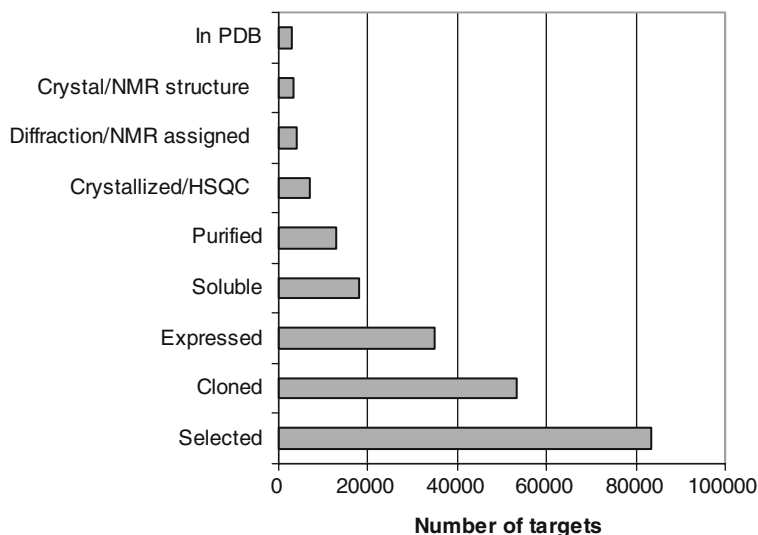
Fig. 22.1. The number of proteins surviving successive stages of structure determination [data from TargetDB (72) (Status: 10/March/2006)]. Out of the 83,596 initially selected targets only 3.4% (2830) have reached the PDB.

Systematic characterization of the protein features influencing solubility, crystallization, or more generally structural determination success rate began around year 2000 (15, 16) when high-throughput structural proteomics consortia accumulated enough experimental data to start the first round of retrospective evaluation. Until that time, a number of rules of thumb, describing the experimental behavior of proteins had been known for years: transmembrane proteins are hard to express, solubilize, and crystallize; long proteins are hardly accessible by nuclear magnetic resonance (NMR); prokaryotic proteins are generally easier to work with than eukaryotic ones; and proteins from thermophilic organisms are more stable.

In this chapter, we review different approaches to predict the crystallization behavior of proteins from their amino acid sequences and present publicly available methods and tools.

2. Methods

There are still very few methods capable of estimating the probability of protein crystallization or of the overall success in structure determination. Although the early data mining efforts did not result in publicly accessible Web-servers or software, they still succeed in elucidating the dependencies between sequence features and experimental behavior of proteins. Therefore, learning

Table 22.1
Predictive methods and databases for protein crystallization

Method	Url	Property
SECRET	http://webclu.bio.wzw.tum.de:8080/secret	Predicting protein crystallizability for highly soluble proteins of 46–200 residues length (23)
OB-Score	http://www.compbio.dundee.ac.uk/obscore/	Probability of success in structure determination by crystallizability (19)
XtalPred	http://ffas.burnham.org/XtalPred	Predict protein crystallizability and suggest bacterial orthologs (26)
SERp	http://nihserver.mbi.ucla.edu/SER/	Suggest protein construct optimization by point mutations (13)
CrysPres	http://www.ruppweb.org/cryspred/default.html	Designing crystal screening conditions
ConSeq	http://conseq.bioinfo.tau.ac.il/	Highlight protein residues important for function and structure (33)
XtalGrow	http://jmr.xtal.pitt.edu/xtalgrow/	Helps to construct and manage custom factorial crystallization tests (52)
BMCD	http://wwwbmcd.nist.gov:8080/bmcd/bmcd.html	Biological Macromolecule Crystallization Database. It includes information about the crystallization conditions and crystal data (53)
MPCD	www.crmcn.univ-mrs.fr/mpcd/	Marseille Protein Crystallization Database – compilation of two crystallization databases, CYCLOP and BMCD (v2.0). (73)

about the results of these early efforts should be useful for the reader. All databases and publicly available methods described below are summarized in **Table 22.1**.

Based on the performed task, methods can be divided into three groups: those that score protein amenability for structure determination or crystallization, those that help to optimize the protein construct, and those that guide crystallization condition screens.

2.1. Crystallization Target Selection

Working with proteins that do not yield crystals under standard test conditions can be futile (17). Therefore, structural genomics projects often resort to alternative targets sharing function and high sequence similarity with the original protein of interest, but having higher chances of crystallization. Orthologs from thermostable organisms were frequently used in the early days of structural genomics (15) as it was believed that thermostable proteins are generally more promising crystallization targets. In contrast to

the simplistic target selection strategy guided solely by the organism of origin, currently available methods can score the probability of protein crystallization based on a large body of success/failure data from high-throughput structure determination efforts.

2.1.1. Overall Structural Determination Success (from Cloning to Structure)

The overall success in structure determination is defined by the percentage of initially selected targets that survived all successive experimental stages from cloning to structure deposition in PDB. It is not always equivalent to protein crystallization since the second most popular method for structure determination at atomic resolution is nuclear magnetic resonance (NMR). Sample quality requirements for both methods partially overlap. In particular, the protein has to be structurally stable and highly soluble in aqueous solution. For the scope of this paragraph, we define overall structural determination success without differentiating between the NMR and X-ray methods.

Canaves et al. (18) attempted resolving three-dimensional structures of the entire protein complement of the hypothermophilic bacteria *Thermatoga maritima*. Out of 1,877 gene products encoded in this organism, 539 were purified and 465 of them crystallized. They described differences between the whole proteome and those proteins that yielded structures by crystallization. The successful set was depleted in proteins containing hydrophobic regions predicted to be transmembrane helices and low-complexity regions, with very few crystallized targets having more then 41 residues in such regions (18). The average length of a successful protein was 274 residues, notably lower than the 311 residues in the entire proteome. Very long (over 560 residues) and very short (fewer than 80 amino acids) proteins were shown to crystallize less frequently. Isoelectric point distributions for both sets were similar and bi-modal, with the minimum at 7.5 (physiological pH of *T. maritima*) and two maxima at 5.8 and 9.6. For crystallizable proteins, the second maximum was slightly shifted from 9.6 to 9.3. Moreover, success rate analysis showed that the probability of crystallization is elevated (32–36%) for the proteins having pI between 5.1 and 7.5. Hydrophobicity measured by GRAVY index (grant average of hydropathy) was also found to be a very potent feature. The distribution of the GRAVY index values for the subset of successful proteins was mono-modal, centered at -0.3, while for the entire proteome it was bi-modal with a second peak centered about 0.7. As a result of this divergence, proteins with GRAVY between -1 and 0.2 crystallized with the probability of $\sim$17% and those with values higher than 0.4 or lower than -1 almost never. Furthermore, amino acid composition was shown to be a very important determinant of structural genomics success rate. Similar to the GRAVY index, the distribution of charged residue occurrence (Glu, Asp, Lys, Arg, His) in the proteome was bi-modal while for the crystallizable subset it was mono-modal with a peak at 30%.

There were practically no crystallizing proteins with the content of charged residues below 24% (18). Interestingly, a two-dimensional drawing of GRAVY against isoelectric point revealed the presence of areas with a higher density of successful instances as well as other areas with a lower probability of success. The region restricted by the pI values 4.3–7.5 and GRAVY –0.67–0.33 was highly enriched in tractable proteins, containing 75% of all crystallized proteins and only 60% of the entire proteome. On the other hand, the proteins with pI higher then 9.1 and GRAVY higher then 0.53 were almost exclusively not crystallizable.

The idea of building a simple predictor of structure determination success based on pI and GRAVY values sparked by Canaves et al. (18) was further developed by Overton et al. (19). A classifier (www.compbio.dundee.ac.uk/obscore) was constructed comparing 5,454 PDB sequences against the UniRef50 data using a Z-score based statistical test in the pI, GRAVY space, resulting in a matrix of differential Z-score values. The UniRef50 dataset was derived from UniProt (20) by sequence clustering such that no two sequences share more than 50% identity. The method calculates the pI, GRAVY, and Z-score (called here OB-score) values for the query sequence using the precalculated differential Z-score matrix. Proteins with an OB-score ≥ 5 were shown to have higher relative probability of success. Since the method does not take into account NMR-derived structures, it essentially evaluates only the probability of structure determination by crystallization, but because the method is trained on the contrast between the PDB and UniProt sequences, without distinguishing individual stages of structure determination, it still predicts the overall success.

Goh and coworkers (21) identified the following factors that correlate with the overall success rate of structure determination: membership in an orthologous family defined in the COG database (22), higher percentage of acidic (DE > 9.7%) and nonpolar (GAVLI > 31.7%) amino acids as well as the lower content of cysteine (C < 1.8%) and the higher content of sulfur or oxygen-containing residues (SCTM>10%). Annotation with a COG family in this case reflects the fact that the given protein is already functionally characterized and thus presumably constitutes a more tractable experimental target.

2.1.2. Probability of Protein Crystallization

All currently available methods to predict crystallization propensity attempt to relate the query sequence to the body of known experimental results. The first and the most straightforward method to evaluate the chances of a protein being crystallizable is to check whether its homologs had been already crystallized. In some cases, this simple approach can also provide hints for construct optimization.

More sophisticated methods go one step further and relate the query sequences not directly to the experimental instances but to the statistical probabilistic models generalizing over the observed

data. Based on the analysis of structural genomics data, it was demonstrated that proteins determined structurally by X-ray or NMR have different amino acid composition in comparison to those that reached only the "purified" stage. The proteins unsuccessful at the structure determination stage (X-ray or NMR) have low content of alanine (A < 8.5%) and high percentage of hydrophobic residues (GAVLI > 26.7%) while successful targets are characterized by higher alanine frequency (21). Christendat et al. (15) found that 18 out of 25 crystallizable proteins, but only one out of 39 noncrystallizable proteins have asparagine composition below 3.5%. These values can be used for threshold-based estimation of success chances.

The method developed by our group is based on the frequencies of single amino acids, doublets, and triplets used as input for the two-layer SVM and Naive Bayes classifier (23). To learn specific features of crystallizable protein we explore the difference between two sets of proteins: those whose structures were determined only by NMR and not having any sequence similarity to proteins with known X-ray structures (used as a negative training set) and those with X-ray structures (used as a positive training set). This approach was inspired by previous work of Valfar et al. (24) and also by the fact that NMR is frequently being used by structural genomics consortia as a complementary technique to determine structures of proteins that did not yield to crystallographic attempts. Using as input the frequencies of one, two, and three amino acid stretches (optionally grouped by amino acid properties such as hydrophobicity), we built a two-layer classifier with a number of SVMs as primary classifiers and a Naive Bayes classifier as a result integrator. Employing ten time cross-validation, we achieved the accuracy of 67% (65% on the positive (crystallizable) and 69% on the negative (noncrystallizable) class) (23). The crystallization predictor is accessible as a Web-server (http://webclu.bio.wzw.tum.de:8080/secret). The limitation of the method is the sequence length limit of sizes between 46 and 200 amino acids.

Analysis of high-throughput experiments (TargetDB database extended with some internal data from PSI (Protein Structure Initiative) participants) and protein structures deposited in the PDB allowed Slabinski et al. (25, 26) to extract features decisive for crystallization. They found that the probability of protein crystallization correlates with sequence length, isoelectrical point, GRAVY hydrophobicity index, instability index, the number of residues predicted to be in coiled-coil [as calculated by COILS (27)], the length of the longest disordered region [as calculated by DISOPRED2 (28)], and sequence conservation (measured as a percentage of insertions in sequence when aligned with homologs from a nonredundant database). Based on those features calculated for crystallizable and noncrystallizable

structural genomics targets, they derived a probabilistic feasibility score using logarithmic opinion pool method (29). Targets with top and bottom 20% scores were successful in 57 and 10% of the cases, respectively. The authors offer a Web-server (XtalPred, http://ffas.burnham.org/XtalPred) categorizing proteins according to the feasibility score into optimal, suboptimal, average, difficult, and very difficult. Additionally, XtalPred is capable of providing close bacterial homologs which are supposed to be more likely to crystallize then the original protein. The main limitation of this method is the absence of an appropriate statistical evaluation on a protein set not used to formulate the rules (so-called withhold data set).

2.2. Construct Optimization

Complementary to scoring and selecting the most crystallizable proteins there exist a number of procedures, both experimental and computational, to improve protein constructs. This includes theoretical methods to detect domain boundaries (30, 31) and fold types (32), the presence of conserved or functionally crucial regions or residues (33, 34), loops, unstructured (28, 35) or low-complexity regions (36), secondary structure elements (37), high entropy, or hydrophobic patches on predicted protein surface (6, 38). There is also an array of experimental techniques helping to measure protein stability (DSC – differential scanning calorimetry), aggregation state (DLS – dynamic light scattering, size exclusion chromatography), the presence of flexible elements [NMR (39); DXMS – deuterium exchange mass spectrometry (40)] and domain boundaries [proteolytic mass spectrometry (41)]. All these standard tools serve to trim and modify the protein sequence in order to make it more structurally stable without affecting domains, active/binding sites, or conserved regions of interest. Because many of the computational methods listed above are covered in other chapters of this book, in this paragraph we will focus primarily on methods for improving putative crystal contact interfaces.

Crystal's nucleation and growth can be hindered by high entropy of the protein surface. Quite often removing surface loops or unstructured regions leads to improved crystallization behavior. But not only loops can be the source of unfavorable surface flexibility. Derewenda and coworkers (6–8) showed that a substantial improvement in crystallization behavior can be achieved by engineering crystal contacts.

Working with proteins of unknown structure, it is not possible to identify which residues will build the crystal contacts. The Derewenda method (13) detects clusters of nonconserved, solvent-exposed residues with high-conformational entropy (lysine, glutamine, glutamine acid), which can impede the formation of crystal contacts. These residues are then substituted by smaller, low-entropy amino acids such as alanine, histidine, tyrosine, or

threonine (8). Among crystallization-promoting substitutions, alanine was first reported. Currently it seems that tyrosine, threonine, serine, and histidine can be equally sufficient (8). In many cases, the latter substitutions are superior over alanine as they do not interfere with protein solubility and for some proteins (e.g., RhoGDI) they result in better crystal quality.

Selection of amino acids types to be replaced is based on the observed lower frequency of lysine, glutamine, and glutamic acid at the protein–protein interaction interfaces (42, 43). Hence by analogy their presence at the crystallization interface should be also avoided. The choice of substituting amino acids is motivated by the amino acid occurrence in interaction interfaces, where tyrosine, histidine, and serine are more frequent (42, 44, 45). Other amino acids (alanine and threonine) are used primary because of their small size, low entropy, and limited hydrophobicity.

Upon building for each protein a spectrum of constructs harboring mutations on different high-entropy patches, the Derewenda group reported improved crystallization and better crystal diffraction for almost all tested proteins (6–8). Interestingly, they also observed that mutated proteins crystallized in greater variety of conditions which brings us to the next topic.

2.3. Optimizing Initial Conditions

It is generally accepted that certain proteins will readily crystallize in a wide range of different conditions, while others are less amenable to crystallization and will require extensive optimization of conditions (17, 46). Nevertheless, screening a wide variety of chemical and physical conditions remains currently the most common approach to crystallization optimization. Various strategies are used to screen conditions for crystallization. Those include simplified rational approaches (screening at the pI), highly regimented approaches (successive grid screening) (47), and analytical approaches (incomplete factorials, solubility assays, perturbation, sparse-matrix) (48–50).

The incomplete factorial method was pioneered by Carter and Carter (48). It is based on random permutation of specific aspects of the crystallization conditions (e.g., pH, precipitant, additives). Random sampling is supposed to provide a broad coverage of the parameter space. The follow-up of this approach is the so-called sparse-matrix method proposed by Jancarik and Kim (49, 51). It has arguably become the most popular approach for initial crystallization screening. In the sparse-matrix method, the parameters of crystallization conditions are constrained to the value ranges known to crystallize proteins. To further limit the number of tests, those combinations of parameters that can be partially represented by other conditions were removed, resulting in the final number of 50 unique conditions. Thanks to a limited number of conditions, the sparse-matrix method requires the least amount of samples. Most of the commercially available screens are based on

either the sparse-matrix or the grid method. The choice of the strategy should be based on the a priori knowledge about a protein.

If you need to design a nonstandard screen you can use one of the publicly available programs. For example, XtalGrow (http://jmr.xtal.pitt.edu/xtalgrow/) (52) based on the Bayesian method extends the Jancarik and Kim work (49) and can be used to calculate a factorial matrix setup guided by the protein properties and functions or based on the range of chemical parameters provided by the user. One of the assumptions made by the XtalGrow authors is that similar macromolecules crystallize in clusters of similar experimental conditions. The guidelines for specific types of molecules (proteins were organized hierarchically according to function) embedded into XtalGrow are based on the crystallization data gathered in the Biological Macromolecular Crystallization Database [BMCD, (53)].

The complexity of the screening procedure can be further extended by using two different buffers: one to mix with the protein and a second one to fill the reservoir (54, 55). The same crystallization conditions over different reservoir solutions were shown to lead to different crystallization/precipitation behavior of the protein. Optimizing the reservoir solution can lead to a substantial improvement in success rate.

Although the importance of crystallization condition's pH is well known, it remains a subject of intense debate whether pH optimal for crystallization can be deduced from the protein pI (56–58). Optimizing protein buffering conditions for increased solubility can lead to higher success rates in subsequent crystallization test as demonstrated by Izaac et al. (59). By adjusting the formulation of the protein solution, they increased the appearance of crystals for 8 out of 10 tested proteins.

A very promising approach was presented by Anderson and coworkers (60). They performed multiple solubility experiments to derive phase diagrams for each protein separately. Equipped with this knowledge they were able to design protein-specific crystallization screens leading to successful crystallization for 9 out of 12 proteins, most of which failed on traditional screens.

Many groups try to define the smallest subset of conditions capable of crystallizing the maximum number of proteins. Kimber et al (17) studied crystallization behavior of 755 proteins from 6 organisms using the sparse-matrix screen described in Jancarik and Kim (49). They suggested that it will be reasonable to reduce the number of different conditions even further than originally proposed by Jancarik and Kim. Kimber and coworkers derived 3 minimal sparse screens with 6, 12, and 24 conditions covering 61, 79, and 94% of successful crystallizations relative to the full sparse screen with 48 conditions. **Table 22.2** contains the formulation of the minimal sparse screen with 12 conditions from

Table 22.2
Minimal sparse screen with 12 conditions from Kimber et al. (17). It covers 79% of the crystals produced by the standard 48 conditions of the Jancarik and Kim screen (49)

Numbers according to Jancarik and Kim (49)	Salt	Buffer	Precipitant
4		0.1 M Tris–HCl, pH 8.5	2 M NH$_4$ Sulfate
6	0.2 M MgCl$_2$	0.1 M Tris–HCl, pH 8.5	30% PEG 4000
10	0.2 M NH$_4$ Acetate	0.1 M Na Acetate, pH 4.6	30% PEG 4000
17	0.2 M Li Sulfate	0.1 M Tris–HCl, pH 8.5	30% PEG 4000
18	0.2 M Mg Acetate	0.1 M Na Cacodylate, pH 6.5	20% PEG 8000
30	0.2 M (NH$_4$)$_2$SO$_4$		30% PEG 8000
36		0.1 M Tris–HCl, pH 8.5	8% PEG 8000
38		0.1 M Na HEPES, pH 7.5	1.4 M Na Citrate
39		0.1 M Na HEPES, pH 7.5	2% PEG 400 2 M NH$_4$ Sulfate
41		0.1 M Na HEPES, pH 7.5	10% 2-Propanol 20% PEG 4000
43			30% PEG 1500
45	0.2 M Zn Acetate	0.1 M Na Cacodylate pH 6.5	18% PEG 8000

Kimber et al. (17). Using argumentation similar to Jancarik and Kim (49), they conclude that minimal screens are more practical and economical than the original screen which was found to be over-sampled toward high molecular weight PEGs (polyethylglycols).

Page and coworkers (61, 62) proposed a 67-condition screen based on the expertise gathered by the Joint Center for Structural Genomics (JCSG) and the University of Toronto during structural studies on bacterial targets. They indicated that such limited subset can outperform typical sparse-matrix screens in identifying initial conditions. The same group also showed that 75% of diffracting crystals can be obtained directly from initial coarse screens indicating that less then 25% of them required fine screening (62). In a similar effort, Gao and coworkers (63) derived a simplified screen based on the BMCD database which allowed them to reduce the total number of conditions and to crystallize proteins which failed with commercial screens.

Another optimization venue is the search for the optimal inhibitor/substrate stabilizing protein structure. This approach requires extensive experimental testing using libraries of putative

compounds to find the one with sufficient affinity to the protein. Usually, researchers tend to employ virtual ligand screening coupled with subsequent experimental measurements of the binding strength (for example fluorometry, calorimetry, or NMR). This protocol proved to be very successful in stabilizing proteins for crystallization and resulted in crystallization of previously unsuccessful targets (5, 64).

Because of the size limits, this paragraph covers only a small fraction of work done toward crystallization condition optimization. For further reading please refer to specialized reviews (65) or textbooks (4).

3. Notes

Considering protein properties leading to overall tractability in the structure determination pipeline, one should not forget that often different protein properties are pivotal for success at different stages along the experimental pipeline. Examples of such cases can be found above or in Smialowski et al. (66).

Considering construct optimization, one potential problem is that removing loops and unstructured regions can prevent proper protein folding and lead to aggregation and formation of inclusion bodies. A possible way around this obstacle is to conduct expression and purification on the longer construct and then to remove the unstructured region using engineered cleavage sites (67), nonspecific enzymatic cleavage (68), or even spontaneous protein degradation (69).

The quality of commercially available crystallization screens still requires attention as even identical formulations from different manufacturers can yield dramatically different results (70).

3.1. Data

One of the major constraints of the methods for predicting experimental tractability of proteins is the limited amount of available data. A particularly difficult challenge is the scarceness of negative experimental data. Data deficiency is the main reason why there are so few studies considering transmembrane proteins. Every set of rules or classifier is a form of statistical generalization over the data at hand. Hence, it is possible that a new protein will be sufficiently different from the data set used for training to render useless attempts of predicting its experimental tractability (e.g., crystallizability). Obviously this problem diminishes with the accumulation of experimental data but nevertheless it will never disappear. Applying rules and using predictors described in this chapter, one has to consider the similarity of the query proteins to the sequences used to construct algorithms. Another consequence of the low

amount of data is that the available methods are quite general and do not take into account specific properties of the protein under study. They are built based on the assumption that protein crystallization is governed by general rules and is not, for example, fold specific. In fact it seems sensible to expect that different rules will apply to proteins having very different folds even if they are all nontransmembrane proteins. Crystallization of some of the types of proteins underrepresented in the current data can be driven by different rules and therefore not well predicted by general protein crystallization algorithms. It remains to be investigated whether protein crystallization is prevalently governed by the universal rules or whether it is rather fold specific. Symptomatic is the experimental behavior of transmembrane proteins and the fact that none of the methods described above apply to transmembrane proteins.

3.2. Methods

An important limitation of the methods and studies described in this chapter is that except for the work of Hennessy et al. (52) all of them consider proteins in isolation and do not take into account chemical crystallization conditions. Such focus on the amino acid sequence is based on the experimental reports suggesting that individual proteins tend to either crystallize under many different conditions, or not at all (17). Nevertheless it is also well documented that the presence of posttranslational modifications (71) or addition of cofactors and inhibitors (5) can dramatically affect protein crystallization. Additionally none of the methods consider physical crystallization setup.

Crystallization prediction methods do not anticipate progress in crystallization production methods. It is conceivable that a protein that failed to crystallize 20 years ago can be easily crystallized nowadays. Rapid improvement of crystallization methods quickly makes earlier predictions based on previously available data obsolete.

References

1. Laskowski, R. A., J. M. Thornton (2008), Understanding the molecular machinery of genetics through 3D structures. *Nat Rev Genet* 9, 141–151.

2. McPherson, A. (1999), Crystallization of biological macromolecules. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory Press. IX, 586.

3. Doye, J. P., A. A. Louis, M. Vendruscolo (2004), Inhibition of protein crystallization by evolutionary negative design. *Phys Biol* 1, P9–P13.

4. Bergfors, T. (1999), Protein crystallization: techniques, strategies, tips. Iul Biotechnology Series. Uppsala: International University Line.

5. Niesen, F. H., H. Berglund, M. Vedadi (2007), The use of differential scanning fluorimetry to detect ligand interactions that promote protein stability. *Nat Protoc* 2, 2212–2221.

6. Derewenda, Z. S. (2004), Rational protein crystallization by mutational surface engineering. *Structure (Camb)* 12, 529–535.

7. Derewenda, Z. S. (2004), The use of recombinant methods and molecular engineering in protein crystallization. *Methods* 34, 354–363.

8. Cooper, D. R., T. Boczek, K. Grelewska, M. Pinkowska, M. Sikorska, M. Zawadzki, Z. Derewenda (2007), Protein crystallization

by surface entropy reduction: optimization of the SER strategy. *Acta Crystallogr D Biol Crystallogr* 63, 636–645.

9. Braig, K., Z. Otwinowski, R. Hegde, D. C. Boisvert, A. Joachimiak, A. L. Horwich, P. B. Sigler (1994), The crystal structure of the bacterial chaperonin GroEL at 2.8 A. *Nature* 371, 578–586.

10. Lawson, D. M., P. J. Artymiuk, S. J. Yewdall, J. M. Smith, J. C. Livingstone, A. Treffry, A. Luzzago, S. Levi, P. Arosio, G. Cesareni, et al. (1991), Solving the structure of human H ferritin by genetically engineering intermolecular crystal contacts. *Nature* 349, 541–544.

11. McElroy, H. H., G. W. Sisson, W. E. Schottlin, R. M. Aust, J. E. Villafranca (1992), Studies on engineering crystallizability by mutation of surface residues of human thymidylate synthase. *J Cryst Growth* 122, 265–272.

12. Yamada, H., T. Tamada, M. Kosaka, K. Miyata, S. Fujiki, M. Tano, M. Moriya, M. Yamanishi, E. Honjo, H. Tada, T. Ino, H. Yamaguchi, J. Futami, M. Seno, T. Nomoto, T. Hirata, M. Yoshimura, R. Kuroki (2007), 'Crystal lattice engineering,' an approach to engineer protein crystal contacts by creating intermolecular symmetry: crystallization and structure determination of a mutant human RNase 1 with a hydrophobic interface of leucines. *Protein Sci* 16, 1389–1397.

13. Goldschmidt, L., D. R. Cooper, Z. S. Derewenda, D. Eisenberg (2007), Toward rational protein crystallization: A Web server for the design of crystallizable protein variants. *Protein Sci* 16, 1569–1576.

14. Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, P. E. Bourne (2000), The Protein Data Bank. *Nucleic Acids Res* 28, 235–242.

15. Christendat, D., A. Yee, A. Dharamsi, Y. Kluger, A. Savchenko, J. R. Cort, V. Booth, C. D. Mackereth, V. Saridakis, I. Ekiel, G. Kozlov, K. L. Maxwell, N. Wu, L. P. McIntosh, K. Gehring, M. A. Kennedy, A. R. Davidson, E. F. Pai, M. Gerstein, A. M. Edwards, C. H. Arrowsmith (2000), Structural proteomics of an archaeon. *Nat Struct Biol* 7, 903–909.

16. Burley, S. K. (2000), An overview of structural genomics. *Nat Struct Biol* 7 Suppl, 932–934.

17. Kimber, M. S., F. Vallee, S. Houston, A. Necakov, T. Skarina, E. Evdokimova, S. Beasley, D. Christendat, A. Savchenko, C. H. Arrowsmith, M. Vedadi, M. Gerstein, A. M. Edwards (2003), Data mining crystallization databases: knowledge-based approaches to optimize protein crystal screens. *Proteins* 51, 562–568.

18. Canaves, J. M., R. Page, I. A. Wilson, R. C. Stevens (2004), Protein biophysical properties that correlate with crystallization success in *Thermotoga maritima*: maximum clustering strategy for structural genomics. *J Mol Biol* 344, 977–991.

19. Overton, I. M., G. J. Barton (2006), A normalised scale for structural genomics target ranking: the OB-Score. *FEBS Lett* 580, 4005–4009.

20. Apweiler, R., A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, L. S. Yeh (2004), UniProt: the Universal Protein knowledgebase. *Nucleic Acids Res* 32 Database issue, D115–D119.

21. Goh, C. S., N. Lan, S. M. Douglas, B. Wu, N. Echols, A. Smith, D. Milburn, G. T. Montelione, H. Zhao, M. Gerstein (2004), Mining the structural genomics pipeline: identification of protein properties that affect high-throughput experimental analysis. *J Mol Biol* 336, 115–130.

22. Tatusov, R. L., M. Y. Galperin, D. A. Natale, E. V. Koonin (2000), The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res* 28, 33–36.

23. Smialowski, P., T. Schmidt, J. Cox, A. Kirschner, D. Frishman (2006), Will my protein crystallize? A sequence-based predictor. *Proteins* 62, 343–355.

24. Valafar, H., J. H. Prestegard, F. Valafar (2002), Datamining protein structure databanks for crystallization patterns of proteins. *Ann N Y Acad Sci* 980, 13–22.

25. Slabinski, L., L. Jaroszewski, A. P. Rodrigues, L. Rychlewski, I. A. Wilson, S. A. Lesley, A. Godzik (2007), The challenge of protein structure determination–lessons from structural genomics. *Protein Sci* 16, 2472–2482.

26. Slabinski, L., L. Jaroszewski, L. Rychlewski, I. A. Wilson, S. A. Lesley, A. Godzik (2007), XtalPred: a web server for prediction of protein crystallizability. *Bioinformatics* 23, 3403–3405.

27. Lupas, A., M. Van Dyke, J. Stock (1991), Predicting coiled coils from protein sequences. *Science* 252, 1162–1164.

28. Ward, J. J., L. J. McGuffin, K. Bryson, B. F. Buxton, D. T. Jones (2004), The DISOPRED server for the prediction of protein disorder. *Bioinformatics* 20, 2138–2139.

29. Genest, C. (1984), Aggregation opinions through logarithmic pooling. *Theory and Decision* 17, 61–70.

30. Bateman, A., E. Birney, R. Durbin, S. R. Eddy, K. L. Howe, E. L. Sonnhammer (2000), The Pfam protein families database. *Nucleic Acids Res* 28, 263–266.

31. Liu, J., B. Rost (2004), Sequence-based prediction of protein domains. *Nucleic Acids Res* 32, 3522–3530.

32. Orengo, C. A., A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, J. M. Thornton (1997), CATH–a hierarchic classification of protein domain structures. *Structure* 5, 1093–1108.

33. Berezin, C., F. Glaser, J. Rosenberg, I. Paz, T. Pupko, P. Fariselli, R. Casadio, N. Ben-Tal (2004), ConSeq: the identification of functionally and structurally important residues in protein sequences. *Bioinformatics* 20, 1322–1324.

34. Thibert, B., D. E. Bredesen, G. del Rio (2005), Improved prediction of critical residues for protein function based on network and phylogenetic analyses. *BMC Bioinformatics* 6, 213.

35. Dosztanyi, Z., V. Csizmok, P. Tompa, I. Simon (2005), IUPred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content. *Bioinformatics* 21, 3433–3434.

36. Wootton, J. C., S. Federhen (1996), Analysis of compositionally biased regions in sequence databases. *Methods Enzymol* 266, 554–571.

37. Pollastri, G., A. McLysaght (2005), Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics* 21, 1719–1720.

38. Adamczak, R., A. Porollo, J. Meller (2004), Accurate prediction of solvent accessibility using neural networks-based regression. *Proteins* 56, 753–767.

39. Rehm, T., R. Huber, T. A. Holak (2002), Application of NMR in structural proteomics: screening for proteins amenable to structural analysis. *Structure* 10, 1613–1618.

40. Hamuro, Y., L. Burns, J. Canaves, R. Hoffman, S. Taylor, V. Woods (2002), Domain organization of D-AKAP2 revealed by enhanced deuterium exchange-mass spectrometry (DXMS). *J Mol Biol* 321, 703–714.

41. Cohen, S. L., A. R. Ferre-D'Amare, S. K. Burley, B. T. Chait (1995), Probing the solution structure of the DNA-binding protein Max by a combination of proteolysis and mass spectrometry. *Protein Sci* 4, 1088–1099.

42. Bordner, A. J., R. Abagyan (2005), Statistical analysis and prediction of protein-protein interfaces. *Proteins* 60, 353–366.

43. Ofran, Y., B. Rost (2003), Analysing six types of protein-protein interfaces. *J Mol Biol* 325, 377–387.

44. Fellouse, F. A., C. Wiesmann, S. S. Sidhu (2004), Synthetic antibodies from a four-amino-acid code: a dominant role for tyrosine in antigen recognition. *Proc Natl Acad Sci USA* 101, 12467–12472.

45. Lo Conte, L., C. Chothia, J. Janin (1999), The atomic structure of protein-protein recognition sites. *J Mol Biol* 285, 2177–2198.

46. Dale, G. E., C. Oefner, A. D'Arcy (2003), The protein as a variable in protein crystallization. *J Struct Biol* 142, 88–97.

47. Cox, M., P. C. Weber (1988), An investigation of protein crystallization parameters using successive automated grid search (SAGS). *J. Cryst. Growth* 90, 318–324.

48. Carter, C. W., Jr., C. W. Carter (1979), Protein crystallization using incomplete factorial experiments. *J Biol Chem* 254, 12219–12223.

49. Jancarik, J., S. H. Kim (1991), Sparse matrix sampling: a screening method for crystallization of proteins. *J Appl Cryst* 24, 409–411.

50. Stura, E. A., G. R. Nemerow, I. A. Wilson (1991), Strategies in protein crystallization. *J Cryst Growth* 110, 1–12.

51. McPherson, A. (1992), Two approaches to the rapid screening of crystallization conditions. *J Cryst Growth* 122, 161–167.

52. Hennessy, D., B. Buchanan, D. Subramanian, P. A. Wilkosz, J. M. Rosenberg (2000), Statistical methods for the objective design of screening procedures for macromolecular crystallization. *Acta Crystallogr D Biol Crystallogr* 56, 817–827.

53. Gilliland, G. L., M. Tung, D. M. Blakeslee, J. E. Ladner (1994), Biological Macromolecule Crystallization Database, Version 3.0: new features, data and the NASA archive for protein crystal growth data. *Acta Crystallogr D Biol Crystallogr* 50, 408–413.

54. Newman, J. (2005), Expanding screening space through the use of alternative reservoirs in vapor-diffusion experiments. *Acta Crystallogr D Biol Crystallogr* 61, 490–493.

55. Dunlop, K. V., B. Hazes (2005), A modified vapor-diffusion crystallization protocol that uses a common dehydrating agent. *Acta Crystallogr D Biol Crystallogr* 61, 1041–1048.

56. Kantardjieff, K. A., B. Rupp (2004), Distribution of pI versus pH provide prior information for the design of crystallization screening experiments: Response to

comment on "Protein isoelectric point as a prediction for increased crystallization screening efficiency". *Bioinformatics* 20, 2171–2174.

57. Kantardjieff, K. A., B. Rupp (2004), Protein isoelectric point as a predictor for increased crystallization screening efficiency. *Bioinformatics* 20, 2162–2168.

58. Page, R., S. K. Grzechnik, J. M. Canaves, G. Spraggon, A. Kreusch, R. C. Stevens, S. A. Lesley (2003), Shotgun crystallization strategy for structural genomics: an optimized two-tiered crystallization screen against the Thermotoga maritima proteome. *Acta Crystallogr D* 59, 1028–1037.

59. Izaac, A., C. A. Schall, T. C. Mueser (2006), Assessment of a preliminary solubility screen to improve crystallization trials: uncoupling crystal condition searches. *Acta Crystallogr D Biol Crystallogr* 62, 833–842.

60. Anderson, M. J., C. L. Hansen, S. R. Quake (2006), Phase knowledge enables rational screens for protein crystallization. *Proc Natl Acad Sci USA* 103, 16746–16751.

61. Page, R., R. C. Stevens (2004), Crystallization data mining in structural genomics: using positive and negative results to optimize protein crystallization screens. *Methods* 34, 373–389.

62. Page, R., A. M. Deacon, S. A. Lesley, R. C. Stevens (2005), Shotgun crystallization strategy for structural genomics II: crystallization conditions that produce high resolution structures for T. maritima proteins. *J Struct Funct Genomics* 6, 209–217.

63. Gao, W., S. X. Li, R. C. Bi (2005), An attempt to increase the efficiency of protein crystal screening: a simplified screen and experiments. *Acta Crystallogr D Biol Crystallogr* 61, 776–779.

64. Gileadi, O., S. Knapp, W. H. Lee, B. D. Marsden, S. Muller, F. H. Niesen, K. L. Kavanagh, L. J. Ball, F. von Delft, D. A. Doyle, U. C. Oppermann, M. Sundstrom (2007), The scientific impact of the Structural Genomics Consortium: a protein family and ligand-centered approach to medically-relevant human proteins. *J Struct Funct Genomics* 8, 107–119.

65. Durbin, S. D., G. Feher (1996), Protein crystallization. *Annu Rev Phys Chem* 47, 171–204.

66. Smialowski, P., A. J. Martin-Galiano, J. Cox, D. Frishman (2007), Predicting experimental properties of proteins from sequence by machine learning techniques. *Curr Protein Pept Sci* 8, 121–133.

67. Mikolajka, A., X. Yan, G. M. Popowicz, P. Smialowski, E. A. Nigg, T. A. Holak (2006), Structure of the N-terminal domain of the FOP (FGFR1OP) protein and implications for its dimerization and centrosomal localization. *J Mol Biol* 359, 863–875.

68. Dong, A., X. Xu, A. M. Edwards, C. Chang, M. Chruszcz, M. Cuff, M. Cymborowski, R. Di Leo, O. Egorova, E. Evdokimova, E. Filippova, J. Gu, J. Guthrie, A. Ignatchenko, A. Joachimiak, N. Klostermann, Y. Kim, Y. Korniyenko, W. Minor, Q. Que, A. Savchenko, T. Skarina, K. Tan, A. Yakunin, A. Yee, V. Yim, R. Zhang, H. Zheng, M. Akutsu, C. Arrowsmith, G. V. Avvakumov, A. Bochkarev, L. G. Dahlgren, S. Dhe-Paganon, S. Dimov, L. Dombrovski, P. Finerty, Jr., S. Flodin, A. Flores, S. Graslund, M. Hammerstrom, M. D. Herman, B. S. Hong, R. Hui, I. Johansson, Y. Liu, M. Nilsson, L. Nedyalkova, P. Nordlund, T. Nyman, J. Min, H. Ouyang, H. W. Park, C. Qi, W. Rabeh, L. Shen, Y. Shen, D. Sukumard, W. Tempel, Y. Tong, L. Tresagues, M. Vedadi, J. R. Walker, J. Weigelt, M. Welin, H. Wu, T. Xiao, H. Zeng, H. Zhu (2007), In situ proteolysis for protein crystallization and structure determination. *Nat Methods* 4, 1019–1021.

69. Ksiazek, D., H. Brandstetter, L. Israel, G. P. Bourenkov, G. Katchalova, K. P. Janssen, H. D. Bartunik, A. A. Noegel, M. Schleicher, T. A. Holak (2003), Structure of the N-terminal domain of the adenylyl cyclase-associated protein (CAP) from *Dictyostelium discoideum*. *Structure* 11, 1171–1178.

70. Wooh, J. W., R. D. Kidd, J. L. Martin, B. Kobe (2003), Comparison of three commercial sparse-matrix crystallization screens. *Acta Crystallogr D Biol Crystallogr* 59, 769–772.

71. Kim, K. M., E. C. Yi, D. Baker, K. Y. Zhang (2001), Post-translational modification of the N-terminal His tag interferes with the crystallization of the wild-type and mutant SH3 domains from chicken src tyrosine kinase. *Acta Crystallogr D Biol Crystallogr* 57, 759–762.

72. Chen, L., R. Oughtred, H. M. Berman, J. Westbrook (2004), TargetDB: a target registration database for structural genomics projects. *Bioinformatics* 20, 2860–2862.

73. Charles, M., S. Veesler, F. Bonnete (2006), MPCD: a new interactive on-line crystallization data bank for screening strategies. *Acta Crystallogr D Biol Crystallogr* 62, 1311–1318.

SUBJECT INDEX