In Silico Technologies in Drug Target Identification and Validation

Drug Discovery Series

Series Editor

Andrew Carmen

Johnson & Johnson PRD, LLC
San Diego, California, U.S.A.

1. Virtual Screening in Drug Discovery, *edited by Juan Alvarez and Brian Shoichet*
2. Industrialization of Drug Discovery: From Target Selection Through Lead Optimization, *edited by Jeffrey S. Handen, Ph.D.*
3. Phage Display in Biotechnology and Drug Discovery, *edited by Sachdev S. Sidhu*
4. G Protein-Coupled Receptors in Drug Discovery, *edited by Kenneth H. Lundstrom and Mark L. Chiu*
5. Handbook of Assay Development in Drug Discovery, *edited by Lisa K. Minor*
6. *In Silico* Technologies in Drug Target Identification and Validation, *edited by Darryl León and Scott Markel*

Drug Discovery Series/6

In Silico Technologies in Drug Target Identification and Validation

Edited by

Darryl León
Scott Markel

Taylor & Francis
Taylor & Francis Group
Boca Raton London New York

CRC is an imprint of the Taylor & Francis Group, an informa business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-10: 1-57444-478-6 (Hardcover)
International Standard Book Number-13: 978-1-57444-478-0 (Hardcover)
Library of Congress Card Number 2006005722

Library of Congress Cataloging-in-Publication Data

In silico technologies in drug target identification and validation / edited by Darryl León and Scott Markel.
p. ; cm.
Includes bibliographical references and index.
ISBN-13: 978-1-57444-478-0 (Hardcover : alk. paper)
ISBN-10: 1-57444-478-6 (Hardcover : alk. paper)
1. Drug development. 2. Drug development-Computer simulation.
[DNLM: 1. Drug Delivery Systems-methods. 2. Chemistry, Pharmaceutical. 3. Computational Biology. WB 340 I35 2006] I. León, Darryl. II. Markel, Scott.

RM301.25.I48 2006
615'.19--dc22 2006005722

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Foreword

When asked to write the foreword for this book in the early phases of planning, we wondered how much of the planned scope and depth would be possible to achieve. León and Markel had selected a broad range of possible topics, which became "must use" tools and technologies in the early drug-development process. In addition, they included topics about emerging technologies that have not made it to a broader audience and user community today, but have the potential to become essential in the next few years. They had compiled a list of world-renowned experts from the industrial sector as well as from research organizations and universities, asked them to provide a chapter, and then ably shepherded the process to completion.

León and Markel are both well-known experts in the field of life science informatics and both have collected in-depth knowledge and expert insight while affiliated with academic institutions and commercial entities. They have an outstanding knowledge about tools and technologies used and applied in the life science informatics area. In industry, they both played a major role in the architectural direction and the scientific design and content of major software solutions. In addition, they are involved in teaching bioinformatics and in the establishment and further development of standards.

After having the result in hand, we must say that it is a remarkable collection of in-depth chapters covering almost all major *in silico* techniques used today in the early drug development process. Each chapter is a valuable information source for the reader, whether an expert in the field, a user of the technology, or simply a researcher interested in understanding some of the technologies lying outside of his or her direct expertise.

Over the last 20 years, and especially in the last years of the 20th century, we saw some hype and "dot-com" behavior in the field of computational biology. This has certainly "cleared" by now, and much of the hype and promises have been taken down to more solid ground. We saw a remarkable expansion in the use of computational methods, and completely new applications areas have emerged. We can foresee that this will be the case for future decades. Undoubtedly, computational methods and technologies are core prerequisites in today's drug-development process, and efficient and innovative use of these technologies will be a key success factor for the development of drugs in the future. The application areas include the basic information technology challenges (e.g., data acquisition, storage, and retrieval), more advanced topics (e.g., analysis pipelines), and emerging technologies (e.g., text mining and pathway analysis).

This book will prepare the reader perfectly for the different "-omics" data floods, which we will face in the coming years. We hope you will have the same pleasure with the book as we have had.

Dr. Reinhard Schneider
Dr. Hartmut Voss
Heidelberg, Germany

Preface

In 2004, Merck voluntarily withdrew the arthritis medication Vioxx® (Rofecoxib) from the worldwide market because patients taking it were observed to have an increased risk for heart attack and stroke. In the months that followed, additional COX-2 selective drugs were flagged as having serious side effects. These concerns about prescription drugs have made the pharmaceutical industry more focused on designing highly selective medications. In the current drug discovery process, selecting the appropriate drug target can be as important as optimizing the chemical entity that binds to that target.

Today, identifying and validating a potential drug target involves not only numerous well-designed experiments, but also the incorporation of several *in silico* approaches. These *in silico* analyses are often predictive, offering cheaper and faster alternatives to *in vitro* or *in vivo* procedures. This book addresses the *in silico* technologies used in target identification and validation by describing how the available computational tools and databases are used within each step.

The book is divided into four main sections. The first section addresses target identification and covers the areas of pattern matching, functional annotation, polymorphisms, and mining gene expression data. The second section covers target validation, which includes text mining, pathways, molecular interactions, subcellular localization, and protein structures. The third section focuses on recent trends such as comparative genomics, pharmacogenomics, and systems biology. The final section discusses computational infrastructure resources needed to support target identification and validation, including database management, bioIT hardware and architecture, data pipelining, and ontologies.

We hope that many people from various scientific and computational backgrounds will find this book useful.

Editors

Darryl León, Ph.D., is currently director of bioinformatics marketing at SciTegic in San Diego, California, where he provides the vision and software requirements for bioinformatics-related products. He is also on the Bioinformatics Advisory Committee for the University of California San Diego Extension. Previously, he was director of life sciences at LION Bioscience, and was a bioinformatics scientist at NetGenics, DoubleTwist, and Genset. He was a faculty member at California Polytechnic State University, San Luis Obispo, and has authored several papers. He is a co-author, with Scott Markel, of *Sequence Analysis in a Nutshell: A Guide to Common Tools and Databases*. He has also taught at the University of California Santa Cruz Extension and at other colleges in northern California. Dr. Leon received his Ph.D. in biochemistry from the University of California–San Diego, and he did his postdoctoral research at the University of California–Santa Cruz.

Scott Markel, Ph.D., is the principal bioinformatics architect at SciTegic, a division of Accelrys. In this role he is responsible for the design and implementation of SciTegic's bioinformatics products. He is a member of the Board of Directors of the International Society for Computational Biology. He was most recently a research fellow and principal software architect at LION Bioscience, where he was responsible for providing architectural direction in the development of software for the life sciences, including the use and development of standards. He was a member of the Board of Directors of the Object Management Group and co-chair of the Life Sciences Research Domain Task Force. Prior to working at LION, Scott worked at NetGenics, Johnson & Johnson Pharmaceutical Research & Development, and Sarnoff Corporation. He has a Ph.D. in mathematics from the University of Wisconsin–Madison. He is a co-author, with Darryl León, of *Sequence Analysis in a Nutshell: A Guide to Common Tools and Databases*.

Acknowledgments

We would like to recognize several people for their contributions to this book. First, our biggest thanks go to all of the contributors who took time out of their busy schedules to write such insightful chapters. Their combined experience at biotechnology and pharmaceutical companies, research institutes, and universities has produced an outstanding collection of chapters.

A special thank you goes to Reinhard Schneider and Hartmut Voss for writing the foreword.

We also thank our Acquisitions Editor, Anita Lekhwani, for her patience and commitment to this project, and Patricia Roberson, our Project Coordinator, who kept us organized and on track.

We want to express our gratitude to Matt Hahn, David Rogers, J. R. Tozer, and Michael Peeler for the opportunity to work and learn at a leading-edge scientific software company.

From Darryl: First, I would like to thank my co-editor, Scott Markel, for joining me in this stimulating project. His contributions and dedication made this book a reality. Next, I want to thank my family for their continued encouragement. And, most importantly, I would like to thank my loving wife, Alison, who always supports my various writing and teaching endeavors.

From Scott: Thanks to Darryl León for inviting me to join this adventure. It's always a pleasure to work with him, either at work (NetGenics, LION, and now SciTegic) or on books. My children (Klaudia, Nathan, and Victor) keep me grounded, reminding me that there are books to be read as well as written. Finally, to Danette, the love of my life, thanks and appreciation for her never-ending support and encouragement (Proverbs 9:10; Philippians 4:13).

Contributors

Alex L. Bangs
Entelos, Inc.
Foster City, California

Michael R. Barnes
GlaxoSmithKline
Harlow, Essex, United Kingdom

Alvis Brazma
European Bioinformatics Institute
Cambridge, United Kingdom

Jaume M. Canaves
University of California
San Diego, California

Aedin Culhane
Dana-Farber Cancer Institute
Boston, Massachusetts

David de Juan
Centro Nacional de Biotecnología
Madrid, Spain

Michael Dickson
RLX Technologies
The Woodlands, Texas

Bruce Gomes
AstraZeneca Pharmaceuticals
Waltham, Massachusetts

William Hayes
Biogen-Idec
Cambridge, Massachusetts

Tad Hurst
ChemNavigator, Inc.
San Diego, California

Arek Kasprzyk
European Bioinformatics Institute
Hinxton, Cambridge, United Kingdom

Bahram Ghaffarzadeh Kermani
Illumina, Inc.
San Diego, California

Darryl León
SciTegic, Inc.
San Diego, California

Scott Markel
SciTegic, Inc.
San Diego, California

Robin A. McEntire
GlaxoSmithKline
King of Prussia, Pennsylvania

Seth Michelson
Entelos, Inc.
Foster City, California

Philip Miller
University of California
San Diego, California

Eric Minch
Merck Research Laboratories
West Point, Pennsylvania

Rajesh Nair
Columbia University
New York, New York

Seán I. O'Donoghue
Mandala IT
Heidelberg, Germany

Bruce Pascal
BioSift, Inc.
Watertown, Massachusetts

Michael Peeler
SciTegic, Inc.
San Diego, California

Raf M. Podowski
Oracle
Burlington, Massachusetts

Vinodh N. Rajapakse
Biomics, LLC
Boston, Massachusetts

Ana Rojas
Centro Nacional de Biotecnología
Madrid, Spain

Burkhard Rost
Columbia University
New York, New York

Robert B. Russell
European Molecular Biology Laboratory
Heidelberg, Germany

Andrea Schafferhans
Lion Bioscience AG
Heidelberg, Germany

Didier Scherrer
Entelos, Inc.
Foster City, California

Reinhard Schneider
European Molecular Biology Laboratory
Heidelberg, Germany

Christopher Sears
BioSift, Inc.
Watertown, Massachusetts

Viviane Siino
BioSift, Inc.
Watertown, Massachusetts

Damian Smedley
European Bioinformatics Institute
Hinxton, Cambridge, United Kingdom

Robert Stevens
The University of Manchester
Manchester, United Kingdom

Alfonso Valencia
Centro Nacional de Biotecnología
Madrid, Spain

Ivayla Vatcheva
German Cancer Research Center
Heidelberg, Germany

Hartmut Voss
Dievini Gmbh
Heidelberg, Germany

Contents

1 Introduction

Darryl León
SciTegic, Inc.

CONTENTS

1.1 THE DRUG-DEVELOPMENT LANDSCAPE

The pharmaceutical and biotechnology industries have encountered increasing research and development costs while facing a decreasing number of new molecular entities [1]. Even though the explosion of genomics-based drug discovery approaches has led to a large collection of potential drug targets, culling the "druggable" targets from the potential ones is the real challenge. Although the drug-discovery process is different for each company, several common steps are used within the industry. These steps include target identification and validation; lead identification and validation; and preclinical studies, with the ultimate goal of successful clinical studies (fig. 1.1). The identification and validation of targets for the pharmaceutical and biotechnology industries are complex processes that include laboratory techniques, outsourcing approaches, and informatics methodologies. However, some basic issues are addressed before a target is even selected for screening. These issues include such topics as how a target's RNA expression correlates with protein expression and a disease hypothesis, how a target is involved in a metabolic pathway or molecular interaction network, whether a target is druggable, and how a target's genomic locus can be correlated with a genetic marker. Of course, the challenges of target identification and validation can be accelerated using *in silico* technologies.

1.2 HISTORICAL PERSPECTIVE

In silico technology for target identification and validation has been used for several decades, although it was not always referred to as an *in silico* approach. In the 1970s,

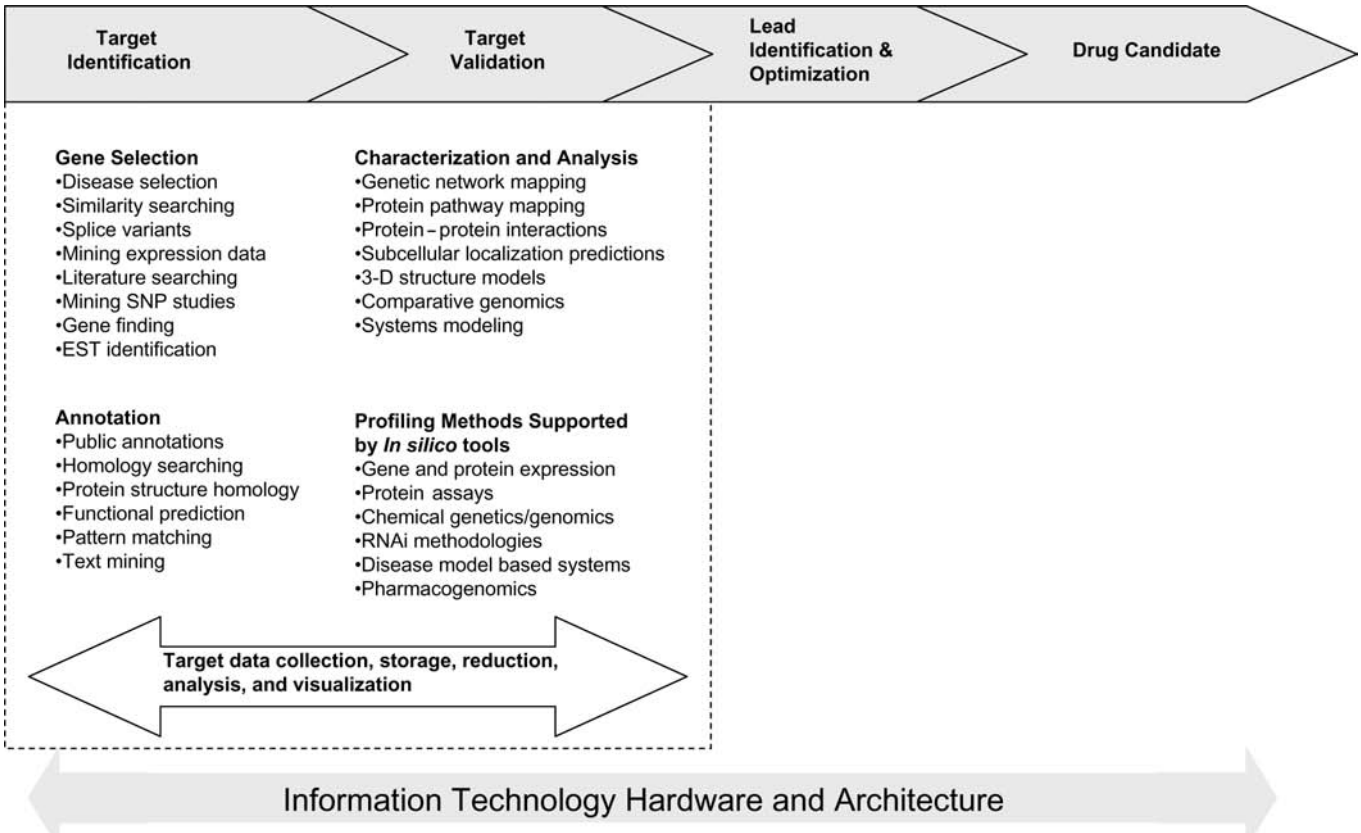


FIGURE 1.1 *In silico* technologies and supported methods for drug-target identification and validation.

the Brookhaven Protein Data Bank was formed to store information about crystal structures of proteins [2]. As more protein sequences were being determined, scientists needed a way to compare sequences from two different samples. The Needleman–Wunsch algorithm for sequence comparison was developed, and it was used to help the study of evolution. In computer science, the concept of the Internet was born, a development which contributed to the spread of bioinformatics tools at the end of the 20th century.

In the 1980s, laboratory techniques in molecular biology allowed for the first complete gene sequence of an organism. Another major milestone in the 1980s was the published physical map of *E. coli* [2]. Having this first genomic sequence of an organism was the start of a revolution in the field of genomics, which required the need for further computational tools in biology. The creation of the Swiss-Prot database allowed for the availability of hand-annotated protein sequences and later became a key database for target identification. To help molecular biologists compare sequences faster than using a global algorithm, the Smith–Waterman and FASTP algorithms were developed. These algorithms focused on local alignments between two sequences and have been used successfully for target identification.

The U.S. government also saw the importance of biological information and computational biology. Congress enacted legislation creating the National Center for Biological Information (NCBI) at the National Library of Medicine, and later supported the Human Genome Project (HGP) to sequence the entire human genome. At this same time, several other biologists with a talent for computer programming began writing simple programs to help the search and alignment of DNA and protein sequences. One significant program was called the Basic Local Alignment Search Tool (BLAST). It was designed to search sequence databases quickly by performing a local alignment of a query sequence against each sequence in a database. This program has become a mainstay in any *in silico* target identification pipeline. In the mid-1980s, a few people began selling their analysis programs and curated databases to discovery departments at drug and biotechnology companies. For example, the Genetics Computer Group became well known for their sequence analysis suite of tools called the Wisconsin Package (now called Accelrys GCG).

In the early 1990s, Incyte Pharmaceuticals began selling various databases of DNA sequence fragments known as expressed sequence tags (ESTs) to help research identify splice variants, which were believed to be associated with disease states. Scientists in the 1990s also began using tools from the computer science world and started using Perl (Practical Extraction Report Language) to help parse the various DNA and protein databases that were stored in a flat file format. During this time, the field of computational biology expanded into the field of bioinformatics, and there was considerable buzz surrounding this new field in the world of genomics. Computer scientists began learning more about biology, and more lab scientists were learning how to program.

The explosion of the World Wide Web in the late 1990s paralleled the continued breakthroughs in high-throughput genomic sequencing and the new algorithms being developed for sequence analysis. The need for *in silico* analysis of DNA and protein sequences was so large in the pharmaceutical and biotechnology industries, many bioinformatics start-up companies began appearing to meet the demand. Some of

these start-ups rode the entrepreneurial wave of the Internet becoming dot-com companies to position themselves as the future for *in silico* drug discovery. DoubleTwist was one of first companies to fully annotate a draft version of the human genome, but the daily dump of sequence data into NCBI's GenBank made it nearly impossible for them to keep their annotated version of the human genome up-to-date.

Sequence data were not the only information being stored, searched, and analyzed in the 1990s. The advent of the microarray plate (also known as the gene chip) contributed to another burst in biological information that needed help from *in silico* tools. Because the genome only contained the blueprint of life, scientists began using microarrays to understand how the genes are coexpressed and coregulated under various conditions. By the beginning of 2000, there were hundreds of public databases and several commercial ones as well. There were also so many open-source bioinformatics programs available to meet this need, and freely available on the Internet that it was difficult for companies selling *in silico* tools to generate revenue for their businesses. However, when Celera and the HGP announced the first complete draft sequence of the human genome, the excitement was palpable. When the human genome sequence was nearly complete, it was made available via the World Wide Web, and several public and commercial *in silico* tools were used to annotate it, visualize it, and navigate within it. With this international announcement, the identification of potential drug targets became the main focus of many drug companies. The technology used for *in silico* target identification and validation became ever more important and continues to be a major contributor to drug discovery.

1.3 TARGET IDENTIFICATION

As the number of sequenced genomes increases with each passing month, the pharmaceutical and biotechnology industries have access to more potential targets than they ever thought possible. Although many targets derived from sequenced genomes have been identified, there is a desire to understand which *in silico* methodologies are available for identifying other new, potential targets. These methodologies include gene selection, gene and protein annotation, and prioritization.

Gene selection can be broken down into three main areas: computational approach, database searching, and data mining. The computational approach includes such analyses as similarity searching, gene finding, EST identification, and splice variant construction. Database searching relies heavily on the quality and amount of data available. The data may come from commercial, public, or internal databases. Mining of this data by domain experts is also very important. In the data-mining approach, expert scientists may include disease studies, differential gene expression data, differential protein expression data, literature searches, and single nucleotide polymorphism studies when identifying a potential target. After selecting a gene target, the next step may include annotating the encoded protein. The annotation step may include protein structure prediction, functional predictions, public annotations, pattern matching, and homology searches.

The last step in target identification typically involves some type of assessment or prioritization. The decision makers may use a combination of computational and laboratory results in addition to a biologist's intuition when examining issues of

novelty, disease association, and drugability. After organizing the target list based on particular criteria, the decision maker may begin looking at known compounds or inhibitors that bind, tissue distribution, known catalytic activity, and potential disease correlations to create a final prioritized target list.

The first section of this book covers the various aspects of target identification, and each chapter focuses on important *in silico* technologies used in the early stages of target identification.

Chapter 2—Pattern Matching: Using a variety of pattern-matching approaches is essential for any basic nucleotide sequence analysis and protein function determination. Markel and Rajapakse review the key motif, domain, and pattern databases and tools commonly used in drug target identification projects.

Chapter 3—Tools for Computational Protein Annotation and Function Assignment: Functional annotation or assignment can be determined experimentally in the laboratory or computationally, but there is strong feedback between these two areas of research. Canaves discusses how computational findings can help laboratory biologists and chemists with experimental design, and how in turn their findings can suggest new directions for computational biologists. He also points out in detail the variety of tools and databases available for functional annotation and assignment.

Chapter 4—The Impact of Genetic Variation on Drug Discovery and Development: Barnes describes the various types of genetic variants (e.g., polymorphisms) and explains how these play a key role in target identification and validation. He also covers the tools and databases available to researchers studying genetic variants.

Chapter 5—Mining of Gene-Expression Data: Microarray data analysts are beginning to utilize complex data normalization, statistics, and other mathematical methods to understand how genes are regulated under various conditions. Brazma and Culhane begin by reviewing the advantages and disadvantages of microarrays, and they detail the methodologies and software tools available to analyze complex microarray data.

1.4 TARGET VALIDATION

Currently, in the drug-discovery process, the major bottleneck is target validation. If this process can be accelerated with computational tools, the target validation step will speed up significantly. The target-validation process includes determining if the modulation of a target's function will yield a desired clinical outcome, specifically the improvement or elimination or a phenotype. *In silico* characterization can be carried by using approaches such as genetic-network mapping, protein-pathway mapping, protein–protein interactions, disease-locus mapping, and subcellular localization predictions. Initial selection of a target may be based on the preliminary results found between cellular location and disease/health condition, protein expression, potential binding sites, cross-organism confirmation, or pathways involved in a disease/health condition.

Most important to the target validation step is the application of *in vitro* and *in vivo* profiling approaches. During this step, scientists attempt to confirm gene regulation or protein regulation; to understand mechanism, activity, or cellular location; to confirm activity in a model organism; and to be able to find a direct association of a target with a disease or health condition. Of course, these approaches can be supported with data collection, storage, reduction, analysis, and visualization; the final desired outcome is a short list of target candidates to begin screening with small molecules. The second section of this book covers several aspects of target validation, and each chapter focuses on important *in silico* technologies used during target validation.

Chapter 6—Text Mining: Gomes, Hayes, and Podowski review the fundamental concepts of text mining and discuss the various types of approaches of extracting information from life science literature. They also examine the challenges of text mining and give examples of how it can be used for drug-target discovery.

Chapter 7—Pathways and Networks: Minch and Vatcheva discuss the different types of pathway data, and they detail the various techniques of pathway analysis. They also include a useful summary of common public and commercial pathway tools and databases.

Chapter 8—Molecular Interactions: Learning from Protein Complexes: The spectrum of interactions is critical to comprehending the dynamics of a living system, and understanding it can help to develop methodology for future studies in other systems. Rojas, de Juan, and Valencia review the current state of experimental and computational methods for the study of protein interactions, including prospects for future developments.

Chapter 9—*In Silico* siRNA Design: León reviews the basic concepts of siRNA design and discusses how siRNA methodologies are being used in research and as therapeutic tools. He also gives a brief overview of the public and commercial databases and programs available and includes useful destinations on the Web where you can find related information.

Chapter 10—Predicting Protein Subcellular Localization Using Intelligent Systems: In their chapter, Nair and Rost point out that, despite the challenges in correctly assessing the accuracy of subcellular localization prediction methods, there have been many improvements made in this area. Future improvements are likely to include the use of integrated prediction methods that combine the output from several programs to provide a comprehensive prediction of subcellular localization.

Chapter 11—Three-Dimensional Structures in Target Discovery and Validation: The chapter by O'Donoghue, Russell, and Schafferhans begins with a short review of how the structure of a protein target is determined experimentally or theoretically. Next, the authors describe the importance of secondary structures in a drug target and review some of the key databases and visualization tools relevant to protein structure. They also discuss how finding a binding site on a protein is important for the drug-discovery process.

1.5 RECENT TRENDS

A target can be analyzed further using computational techniques such as three-dimensional (3D) comparative genomics, *in silico* chemical genomics, and systems modeling. Another area of interest to drug-discovery companies and the Food and Drug Administration (FDA) is pharmacogenomics. The third section of this book covers several aspects of these recent trends, and each chapter focuses on important *in silico* technologies used in target identification and validation.

Chapter 12—Comparative Genomics: Genome-sequencing efforts have become rather commonplace, and these projects have resulted in new genomes being published almost monthly. Siino, Pascal, and Sears examine how comparative genomes can facilitate the finding of potential drug targets by detecting genomic correlations among several organisms.

Chapter 13—Pharmacogenomics: Kermani discusses the scientific and computational challenges of pharmacogenomics and how personalized medicine has several sociological obstacles to overcome before it becomes a mainstay diagnostic for medical therapies.

Chapter 14—Target Identification and Validation Using Human Simulation Models: Michelson, Scherrer, and Bangs describe how the physiologic implications of a potential target's function fit in the context of the disease and its progression. They discuss the complexity of systems biology and describe the issues facing predictive approaches.

Chapter 15—Using Protein Targets for *In Silico* Structure-Based Drug Discovery: Validating the importance of a drug target can include its affinity for druglike compounds. Hurst briefly reviews the common approaches used in *in silico* drug screening and how virtual screening can be used with protein structures to identify potential druglike molecules.

1.6 COMPUTATIONAL INFRASTRUCTURE

Data integration has been a focus for many people involved with computational biology, bioinformatics, and *in silico* technologies over the last decade. Using the concept of genomics, scientists began attaching "-omics" to many other areas of research, including proteomics, metabolomics, and transcriptomics. The current challenge for *in silico* technologies in target validation is data integration. Because most of these databases are created from unrelated sources such as unstructured data, structured data, and tabular data, making use of this disparate information has been and still is a major task for bioinformaticists. The hardware giant, IBM, became interested in the life sciences, and they announced the Blue Gene project, which is designed to calculate the predicted 3D structure of a protein from only its amino acid sequence. The fourth section of this book covers several aspects of the informatics infrastructure needed to support *in silico* technologies and target identification and validation.

Chapter 16—Database Management: Recent developments in high-throughput technologies have significantly increased the amount of data being stored, managed, and retrieved for target identification and validation. Kasprzyk and Smedley examine how database management systems are necessary to control data quality, and they review the numerous database management systems available for biological information.

Chapter 17—BioIT Hardware Configuration: Computing power and infrastructure will continue to be essential in drug-target discovery. Miller explores the various components of BioIT and how these support *in silico* research efforts.

Chapter 18—BioIT Architecture: Software Architecture for Bioinformatics Research: Dickson explains the variety of requirements that the research process imposes on any BioIT architecture used in target identification and validation. He focuses on the types of components (e.g., architecture, environment, and services) that are necessary to support the target discovery process.

Chapter 19—Workflows and Data Pipelines: As the fields of computational biology and bioinformatics have become more mature, there has been increasing agreement about how biological data should be processed. These accepted methodologies have allowed for the automation of many types of workflows and pipelines. Peeler compares and contrasts the technologies and approaches that address the areas of data pipelining and workflows.

Chapter 20—Ontologies: McEntire and Stevens provide a review of the challenges of creating and implementing ontologies and how ontologies assist in the drug-target discovery process. They give a good review of the numerous tools, initiatives, projects, and standards being pursued in the world of ontologies, and they discuss how the concept of the semantic web will enhance the life sciences.

1.7 THE FUTURE OF *IN SILICO* TECHNOLOGY

Drug discovery will continue to rely heavily on computational methods to help accelerate the identification and validation of potential drug targets. However, the industry will see a shift from "classical" bioinformatics (e.g., genomic and protein annotations) to more complex computational problems. Data integration and high-throughput computing will still be necessary, but there will be a call for improved statistical tools for gene expression and proteomics. Scientists who study model organisms and use them in target validation will get a boost from comparative genomics, because the ever-growing sequence information generated daily from the public sector can be used to find common gene regulators in similar organisms. The development of more informative metabolic pathways will also result from the numerous model organisms being sequenced. Predictive protein–protein interactions will need more databases and more reliable association algorithms to support target identification and validation. It will be important to integrate this information from text mining and experimental results. Text mining in the life sciences will be critical; however, better natural language processing algorithms and browsing tools will be

needed to support target validation efforts. It will be essential to integrate this information with protein–protein interactions and patent searches. Although structural biology (i.e., structural genomics) is making steady progress, it will continue to be essential for drug discovery. The industry will see more representative structures being solved for all of the major protein families. The promise of pharmacogenomics is very exciting, and this area of research is becoming more interesting to the FDA. This is the beginning of personalized medicine (or targeted medicine), but it is too early to determine how personalized medicine might be used in the physician's office, for the direct and immediate benefit of the patient. Nonetheless, statistical software and visualization applications will be needed to assist health care workers in understanding the results from genetic tests. Finally, the most intriguing new approach in supporting target identification and validation is systems biology. Once all the data are gathered from a model organism and the predicted parameters and simulated environments have been computed, we should be able to predict the behavior of a complex biological system under various conditions. This predictive approach will open the door to the creation of virtual *in silico* patients and may someday reduce the number of years for clinical trials.

ACKNOWLEDGMENTS

I thank Alison Poggi León at Illumina for her comments and review.

REFERENCES

1. Suter-Crazzolara, C., and V. Nedbal. 2003. Finding druggable targets faster. LION Target Engine White Paper, LION Bioscience.
2. Richon, A. B. Network Science. Available at http://www.netsci.org/Science/Bioinform/feature06.html

Part I

Target Identification

2 Pattern Matching

Scott Markel
SciTegic, Inc.

Vinodh N. Rajapakse
Biomics, LLC

CONTENTS

2.1 INTRODUCTION

Many of our colleagues and customers work in pharmaceutical or academic research departments that include a focus on drug-target identification and validation. If you were to walk down the hallways of these organizations and look inside the offices and conference rooms at the whiteboards, you would likely see drawings representing nucleotide or protein sequences and their associated subsequences of interest. Many of the subsequences are represented by patterns. Nucleotide sequences are typically drawn as straight lines, with the patterns drawn as boxes, ovals, or underlines. See figure 5.5, showing the functional anatomy of a gene, for a good example of this. Protein sequences are often shown as two-dimensional objects based on their known or conjectured secondary structure.

In this chapter we cover pattern databases, tools you can use to discover novel patterns, and software you can use to create your own tools.

2.2 HISTORICAL BACKGROUND

Patterns are of interest in many domains and are both observed and inferred. Whenever something of interest is seen to occur more than once, we attempt to describe it. Architects use patterns to describe buildings. Software developers use patterns to describe useful bits of functionality. Patterns are not typically an end unto themselves. They are merely a way for fellow practitioners to note something of interest that may be useful to others. The patterns also become a way of communicating this information, a kind of shorthand [1,2].

Patterns are ubiquitous in science. Weather, seasons, and planetary and celestial motion were all observed by ancient civilizations. The golden ratio (golden mean,

golden section), approximately equal to 1.618, is a pattern frequently found in nature—for example, in shell growth.

Within molecular biology there are other patterns. Three nucleotides, also called a *codon*, are translated into a single amino acid. Early examples of sequencelike patterns involving locations were common in classical genetics. As DNA, RNA, and protein sequences were actually sequenced, giving us the character strings we are so familiar with today, the computational tools from computer science could be brought to bear on research problems in target identification and validation. Biological sequence data's textual representation made regular expressions (described in the next section) a natural choice for representing concise, precisely defined recurring sequence elements. Although incredibly useful, regular expressions have proved inadequate for more expansive or variant structures, such as protein motifs or secondary structure patterns. The challenge of representing these expressions drove the development of more expressive pattern representation approaches uniquely tailored to the nuances of biological (sequence) data.

2.3 PATTERN REPRESENTATION

In computer science, textual patterns are often represented by regular expressions.

- A regular expression (sometimes abbreviated *regex*) is a way for a computer user or programmer to express how a computer program should look for a specified pattern in text and then what the program is to do when each pattern match is found (www.whatis.com).
- A regular expression (abbreviated as *regexp, regex,* or *regxp*) is a string that describes or matches a set of strings, according to certain syntax rules. Regular expressions are used by many text editors and utilities to search and manipulate bodies of text based on certain patterns (www.wikipedia.org).

The seeds of regular expressions were planted in the early 1940s by two neurophysiologists who developed neuron-level models of the nervous system. These models were formally described by Stephen Kleene in an algebra he called regular sets. He devised a simple notation called regular expressions. This was followed by a rich theoretical math study in the 1950s and 1960s. The first documented use of regular expressions in computer science was by Ken Thompson, whose 1968 article "Regular Expression Search Algorithm" describes a regular expression compiler that produced IBM 7094 object code. This led to Thompson's work on *qed*, an editor that formed the basis for the Unix editor *ed*. *ed* had a command to display file lines that matched a regular expression: "g/Regular Expression/p", read as Global Regular Expression Print, became its own utility as *grep*, then *egrep* (extended *grep*) [3].

As computer science and molecular biology merge in the new field of bioinformatics, the use of pattern expression syntax like regular expressions has been introduced to the life scientists. They see it in databases like PROSITE. They also see it in the Internet search engines like Google.

2.4 DATABASES

In many cases pattern databases will be sufficient for your needs. This is true if the problem you are studying either has been studied before or is closely related to one that has. We have divided the pattern databases into protein and nucleotide.

2.4.1 Protein Patterns

Protein-pattern databases cover both motifs or functional domains and secondary structure. We have included the entire InterPro family of databases, as well as BLOCKS, CDD, and a description of patterns found in Swiss-Prot sequences. DSSP, ISSD, PSSD, and CATH are covered in the secondary-structure section.

2.4.1.1 Motif/Domain

Functionally related proteins typically share sequence features essential to the structural elements underlying their shared biological role. These structures, and their critical sequence features, can be relatively compact, as in the case of a posttranslational modification site. The latter might consist of a target residue embedded within set of contiguous residues "recognized" by a modifying enzyme. Alternatively, functional structures may be quite elaborate, as in the case of an independently folding ligand-binding domain. The latter might be formed from several noncontiguous sequence segments, each containing residues essential to the structural association. Conserved sequence features thus exhibit a range of complexity paralleling that of the structural and functional elements they specify. In turn, a range of pattern description approaches have been developed to capture these diverse "signatures" of functionally related protein groups [4].

2.4.1.1.1 InterPro

Considering the diversity of conserved sequence features, it is apparent that a single ideal pattern representation approach does not exist. Particular techniques each have strengths and weaknesses. To provide a unified interface to a range of often complementary pattern databases (and associated representation approaches), the InterPro resource [5,6] was developed. Signatures describing a particular protein family or domain are grouped into unique InterPro entries. From the latter, one can access specific signature representations found in member databases, together with information on protein sequences known to match the signatures.

2.4.1.1.2 UniProt

According to Bairoch et al. [7],

> UniProt is a central repository of protein sequence and function created by joining the information contained in Swiss-Prot, TrEMBL, and PIR. UniProt is comprised of three components, each optimized for different uses. The UniProt Knowledgebase (UniProt) is the central access point for extensive curated protein information, including function, classification, and cross-reference. The UniProt Non-redundant Reference (UniRef) databases combine closely related sequences into a single record to speed searches. The UniProt Archive (UniParc) is a comprehensive repository, reflecting the history of all protein sequences.

Each InterPro entry includes a "match table" listing UniProt protein sequences matching the entry's signature(s).

2.4.1.2 Single Motif Patterns

The representation of a single motif pattern is similar to a regular expression—simple and precise. The weaknesses are that these patterns are rigid and prone to missing diverged pattern exemplars, as might be found, for example, in highly diverged members of a protein family. In addition, very compact patterns might be highly nonspecific, that is, prone to random matches. PROSITE is the best example of a pattern database that contains single motif patterns.

2.4.1.2.1 PROSITE

PROSITE [8,9] is database of sequence patterns representing biologically significant sites in proteins; patterns are largely represented using a regular expression-based language, with more complex patterns additionally described using sequence profiles. A few (four) PROSITE signature entries (three posttranslational modification sites and a nuclear targeting sequence) are described as free-text rules.

2.4.1.3 Multiple Motif Patterns

One good example of a multiple motif pattern is a fingerprint, a defined sequence of patterns or identifiable features. Using multiple, nonoverlapping, conserved motifs to collectively describe a signature is more flexible than using a single motif pattern. In addition, multiple motif patterns are capable of describing highly diverged exemplars that are missed by single motif descriptors, that is, regular expressions. PRINTS is a good example of a multiple motif pattern database.

2.4.1.3.1 PRINTS

"PRINTS [10] is a compendium of protein fingerprints. A fingerprint is a group of conserved motifs used to characterize a protein family. … Usually the motifs do not overlap, but are separated along a sequence, though they may be contiguous in 3D-space. Fingerprints can encode protein folds and functionalities more flexibly and powerfully than can single motifs: the database thus provides a useful adjunct to PROSITE" (www.ebi.ac.uk/interpro/user_manual.html, appendix D).

2.4.1.4 Profile (HMM) Patterns

A profile hidden Markov model (HMM) is a statistical model of a multiple alignment of sequences drawn from a putative protein family. It captures position-specific information about the relative degree of conservation of different columns in an alignment and the relative likelihood of particular residues occurring in specific positions. The strengths of profile patterns include their rich statistical description of information in nonconserved regions (gaps in a multiple sequence alignment) as well as information in highly conserved segments (aligned blocks in a multiple sequence alignment). The reader should be aware that one weakness of this approach is that the quality of the model (pattern representation) is tied substantially to the quality and functional understanding of the

TABLE 2.1
Protein Motif/Domain Databases

InterPro	http://www.ebi.ac.uk/interpro/
UniProt	http://www.ebi.ac.uk/uniprot/
PROSITE	http://www.expasy.org/prosite/
PRINTS	http://www.bioinf.man.ac.uk/dbbrowser/PRINTS/
Pfam	http://www.sanger.ac.uk/Software/Pfam/
SMART	http://smart.embl-heidelberg.de/
TIGRFAMS	http://www.tigr.org/TIGRFAMs/
PIR SuperFamily	http://pir.georgetown.edu/iproclass/
SUPERFAMILY	http://supfam.mrc-lmb.cam.ac.uk/SUPERFAMILY/
ProDom	http://protein.toulouse.inra.fr/prodom/current/html/home.php
BLOCKS	http://blocks.fhcrc.org/
CDD	http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml
	http://www.ncbi.nlm.nih.gov/COG/
	http://www.ncbi.nlm.nih.gov/Web/Newsltr/FallWinter03/kogs.html.
Swiss-Prot features	http://www.expasy.org/sprot/userman.html#FT_line

sequences used to derive it. Pfam, SMART, TIGRFAMS, PIR SuperFamily, and SUPERFAMILY are examples of profile or HMM databases. Table 2.1 summarizes the protein motif/domain databases discussed in this section.

2.4.1.4.1 Pfam

Pfam [11] is a manually curated database of protein families. Each family in Pfam is represented by both multiple alignments and a profile HMM. In particular, a seed alignment is constructed using representative members of the family. The latter is used to construct a profile HMM using the HMMER2 software package. Finally, the HMM is used to detect additional family members, which are then aligned to obtain a full family alignment. With release 10.0, Pfam families cover 75% of the protein sequences in Swiss-Prot and TrEMBL. For protein sequences that do not match any Pfam family, Pfam-B families are generated using ProDom (detailed next).

2.4.1.4.2 SMART

According to the SMART Web site,

> SMART (Simple Modular Architecture Research Tool) [12–14] is a Web-based resource used for the annotation of protein domains and the analysis of domain architectures, with particular emphasis on mobile eukaryotic domains. Extensive annotation for each domain family is available, providing information relating to function, subcellular localization, phyletic distribution and tertiary structure. The January 2002 release has added more than 200 hand-curated domain models. This brings the total to over 600 domain families that are widely represented among nuclear, signalling and extracellular proteins. Annotation now includes links to the Online Mendelian Inheritance in Man (OMIM) database in cases where a human disease is associated with one or more mutations in a particular domain. (http://smart.embl-heidelberg.de/help/smart_about.shtml)

2.4.1.4.3 TIGRFAMS

TIGRFAMS [15] is an annotated collection of protein families, each described using curated multiple sequence alignments and HMMs.

2.4.1.4.4 PIR SuperFamily (PIRSF)

According to the InterPro User Manual,

> PIR SuperFamily (PIRSF) [16–18] is a classification system based on evolutionary relationship of whole proteins. Members of a superfamily are monophyletic (evolved from a common evolutionary ancestor) and homeomorphic (homologous over the full-length sequence and sharing a common domain architecture). A protein may be assigned to one and only one superfamily. Curated superfamilies contain functional information, domain information, bibliography, and cross-references to other databases, as well as full-length and domain HMMs, multiple sequence alignments, and phylogenetic tree of seed members. PIR SuperFamily can be used for functional annotation of protein sequences. (http://www.ebi.ac.uk/interpro/user_manual.html, appendix H)

2.4.1.4.5 SUPERFAMILY

SUPERFAMILY [19] is a collection of profile HMMs aiming to represent all proteins of known structure. Each model corresponds to a domain described in the SCOP structural classification database and aims to describe the entire SCOP superfamily associated with the domain.

2.4.1.5 Other

2.4.1.5.1 ProDom

ProDom [20,21] is a collection of protein domain families automatically derived from the Swiss-Prot and TrEMBL databases using a novel approach based on recursive PSI-BLAST searches.

2.4.1.6 Non-Interpro Pattern Repositories

2.4.1.6.1 BLOCKS

BLOCKS [22,23] is an automatically generated protein family database closely related to PRINTS; like the latter, it represents patterns characterizing family membership as sets of multiply aligned, ungapped sequence segments (BLOCKS).

2.4.1.6.2 CDD

CDD [24,25] is a database of conserved protein domains associated with particular biological functions, together with tools for identifying such domains in query sequences. This now includes COGs and KOG. According to the National Center for Biotechnology Information (NCBI),

> Clusters of Orthologous Groups of proteins (COGs) were delineated by comparing protein sequences encoded in complete genomes, representing major phylogenetic lineages. Each COG consists of individual proteins or groups of paralogs from at least

three lineages and thus corresponds to an ancient conserved domain. (www.ncbi.nlm.nih.gov/COG/)

2.4.1.6.3 Swiss-Prot features

The feature table (FT) lines of a Swiss-Prot [26,27] entry provide a precise means for annotating sites of interest along a protein sequence. In many instances, these are examples of elements described by sequence patterns (e.g., structural/functional domains, posttranslational modification sites, etc.). For example, the following line describes the extent of a zinc-finger domain in a particular protein sequence, with the numbers representing start and end amino acid residue positions [28].

```
FT    ZN_FING       319      343          GATA-type.
```

2.4.1.7 Protein Secondary Structure

Reliable secondary structures can enhance the prediction of higher order protein structure, and to a limited extent, secondary-structure motifs can even suggest specific fold structures. Sometimes these secondary structures provide insight into function. Definition of Secondary Structure of Proteins (DSSP), Integrated Sequence-Structure Database (ISSD), Protein Secondary Structure Database (PSSD), and CATH are covered in this section (see Table 2.2).

2.4.1.7.1 DSSP

"The DSSP database is a database of secondary structure assignments (and much more) for all protein entries in the Protein Data Bank (PDB)" [29]. "The DSSP program defines secondary structure, geometrical features and solvent exposure of proteins, given atomic coordinates in Protein Data Bank format. The program does *not predict* protein structure" [29].

2.4.1.7.2 ISSD

"The ISSD consists of records, each one containing a coding sequence of gene (sequence of codons) aligned with the amino acid sequence and structural data for polypeptide chain(s) of the corresponding protein" [30]. This organization was originally developed to facilitate analyses of the relation of synonymous codon usage to protein secondary structure. Although the database might seem relatively small, it should be noted that "only non-redundant, non-homologous proteins with high-resolution structures available are included. Also, mutant proteins are avoided and

TABLE 2.2
Protein Secondary Structure Databases

DSSP	http://www.cmbi.kun.nl/gv/dssp/
	http://www.cmbi.kun.nl/gv/dssp/descrip.html
ISSD	http://www.protein.bio.msu.su/issd/
PSSD	http://ibc.ut.ac.ir/pssd/about.html
CATH	http://www.cathdb.info/

an attempt is made to match the source organism, tissue, and/or cell type as precisely as possible for both gene and protein structure data, thus increasing the biological meaning of the database information content" [30].

2.4.1.7.3 PSSD

PSSD is a database that incorporates sequences of secondary-structure elements for all proteins with three-dimensional structures defined by experimental methods (such as NMR-Spectroscopy or X-Ray Crystallography) and for which structural data exist in the Brookhaven protein databank.

2.4.1.7.4 CATH

"The CATH database is a hierarchical domain classification of protein structures in the Brookhaven protein databank." Proteins are clustered at four major levels: Class, Architecture, Topology, and Homologous Superfamily. Class is assigned automatically for the vast majority of protein structures, based on their secondary-structure content. The architecture level captures the overall shape of the domain structure, based on the orientation of secondary-structure elements. "The topology level clusters structures according to their topological connections and numbers of secondary structures. The homologous superfamilies cluster proteins with highly similar structures and functions. The assignments of structures to topology families and homologous superfamilies are made by sequence and structure comparisons" [31–33].

2.4.2 Nucleotide Patterns

The DNA Data Bank of Japan (DDBJ)/European Molecular Biology Laboratory (EMBL)/GenBank database triumvirate contains millions of annotated nucleotide sequences. Many of the annotations (features) are described by patterns. Restriction enzyme cleavage sites (REBASE), repetitive DNA sequences (RepBase Update), and transcription factors and their associated binding sites (TRANSFAC) are also described by patterns. Databases that did not make our short list include the Short Tandem Repeat Database; UTRdb and UTRSite (untranslated regions); databases for Short Interspersed Repetitive Elements and Long Interspersed Repetitive Elements; and JASPAR, a collection of transcription factor DNA-binding preferences. Table 2.3 summarizes the nucleotide pattern databases discussed in this section.

2.4.2.1 DDBJ/EMBL/GenBank Feature Table (FT)

GenBank, EMBL, and DDBJ form the International Nucleotide Sequence Database Collaboration. The partnership databases are the richest source of publicly available annotated nucleotide sequences. The FT describes the features and syntax [28]. Although not all features involve patterns, many do. Examples include

- polyA_signal—"recognition region necessary for endonuclease cleavage of an RNA transcript that is followed by polyadenylation;consensus= AATAAA" [34]
- repeat_region—"region of genome containing repeating units"

- TATA_signal—"TATA box; Goldberg-Hogness box; a conserved AT-rich septamer found about 25 bp before the start point of each eukaryotic RNA polymerase II transcript unit which may be involved in positioning the enzyme for correct initiation; consensus=TATA(A or T)A(A or T)" [35,36]

The syntax used for locations is described in section 3.5 of the FT. The most frequently used location types are

- A single base (e.g., 467)
- A site between two indicated bases (e.g., 123^124 or 145^177)
- A single base chosen from within a specified range of bases (e.g., [102.110])
- A continuous range of bases (e.g., 340..565)

The three databases are freely available.

2.4.2.2 REBASE

Restriction enzymes are used to cleave both DNA strands at specific sites, described by patterns. For example, BamHI matches "GGATCC" and cleaves after the first G. MslI's pattern "CAYNNNNRTG" contains the unknown N (any nucleotide) and the ambiguous R (A or G) and Y (C or T). The Restriction Enzyme database (REBASE) contains information about restriction enzymes and related proteins. It includes "recognition and cleavage sites, isoschizomers, commercial availability, methylation sensitivity, crystal and sequence data, DNA methyltransferases, homing endonucleases, nicking enzymes, specificity subunits and control proteins" [37]. Recent additions are the analytic predictions of DNA methyltransferases and restriction enzymes from genomic sequences. Entry references, both published and unpublished, are included. REBASE files are freely available by FTP.

TABLE 2.3
Nucleotide Pattern Databases

DDBJ	http://www.ddbj.nig.ac.jp/
EMBL	http://www.ebi.ac.uk/embl/
GenBank	http://www.ncbi.nlm.nih.gov/Genbank/GenbankOverview.html
DDBJ/EMBL/ GenBank Feature Table	http://www.ebi.ac.uk/embl/Documentation/FT_definitions/feature_table.html
REBASE	http://rebase.neb.com/rebase/rebase.html http://rebase.neb.com/rebase/rebhelp.html ftp://ftp.neb.com/
Repbase Update	http://www.girinst.org/Repbase_Update.html
TRANSFAC	http://www.gene-regulation.com/pub/databases.html#transfac http://www.gene-regulation.com/pub/databases/transfac/doc/toc.html ftp://transfac.gbf.de/
Transcription Factor Classification	http://www.gene-regulation.com/pub/databases/transfac/cl.html

2.4.2.3 Repbase Update

This database contains sequence exemplars of repetitive DNA from different eukaryotic species. Most entries are consensus sequences of large families and subfamilies of repeats, with smaller families represented by sequence examples. The entries include annotations and references and are released in EMBL format. Repbase Update [38–40] is used by both CENSOR and RepeatMasker, which masks out these common repeats to speed up other analyses. Repbase Update is free to academic users. Commercial users need a license.

2.4.2.4 TRANSFAC

Transcription factors help regulate the transcription of protein-encoding genes by RNA polymerase. TRANSFAC [41] is a database of transcription factors and their associated binding sites, with special emphasis on pathologically relevant sequence mutations. An example entry HS$ALBU_05, taken from the European Molecular Biology Open Software Suite (EMBOSS) documentation for tfextract, has the pattern TCTAGTTAATAATCTACAAT. TRANSFAC is free to academic users. Commercial users need a license.

2.5 STANDARDS

Within the computer science community, regular expressions may be considered a standard [3]. The Open Group has an online standard on this topic (table 2.4). They are working to unify various approaches, including Unix and Perl 5 regular expressions. On the life sciences side, the controlled vocabularies provided by DDBJ/EMBL/GenBank and Swiss-Prot features are *de facto* standards.

TABLE 2.4
Standards in Pattern Representation

The Open Group	http://www.opengroup.org/onlinepubs/007908799/xbd/re.html
Perl regular expressions	http://perldoc.perl.org/perlre.html

2.6 TOOLS

If the pattern databases do not contain what you need, you can turn to many fine software tools that will attempt to discover novel patterns. In the following sections we cover gene finding, protein, and nucleotide patterns, including secondary-structure prediction. We also describe MEME/MAST and HMMER, as well as the pattern matching capabilities of EMBOSS and GCG. Finally, we include some suggestions for those interested in writing their own pattern-finding programs.

2.6.1 Gene Finding

With the advent of sequenced genomes, identifying the genes encoded in the genome has become a major research effort. The programs described next are currently the

TABLE 2.5
Gene-Finding Tools

NCRNASCAN	http://www.genetics.wustl.edu/eddy/software/#ncrnascan
GeneWise	http://www.ebi.ac.uk/Wise2/
	http://www.ebi.ac.uk/Wise2/doc_wise2.html
GFScan	http://rulai.cshl.edu/
Genscan	http://genes.mit.edu/GENSCAN.html
	http://genes.mit.edu/Limitations.html
GeneComber	http://bioinformatics.ubc.ca/genecomber/index.php
GeneMark	http://opal.biology.gatech.edu/GeneMark/
	http://opal.biology.gatech.edu/GeneMark/eukhmm.cgi
FGENES, FGENESH	http://www.softberry.com/berry.phtml?topic=fgenes&group=programs &subgroup=gfind
HMMGene	http://www.cbs.dtu.dk/services/HMMgene/

most popular. Many researchers employ more than one program, investigating the differences between the results to better understand their genes of interest. Table 2.5 summarizes the gene finding tools discussed in this section.

Baxevanis and Ouellette [42] describe three major categories of gene-finding strategies: content based (using bulk properties like codon usage, repeats, and sequence composition), site based (using absence or presence of features like donor and acceptor splice sites, transcription factor binding sites, start and stop/termination codons), and comparative (homology based).

Gene finding is difficult and, as may be expected, not all approaches give the desired results. This has given rise to papers describing methods that do not work. The NCRNASCAN paper by Rivas and Eddy [43] is a good example. Contrary to the authors' expectations, real RNAs do not generally have any more secondary-structure content than random sequence [43].

2.6.1.1 GeneWise

GeneWise [44] allows a nucleic acid sequence to be aligned with a sequence or sequence profile (HMM) associated with a potentially homologous protein or protein family. The protein/protein family information is used to infer a putative intron–exon structure in the nucleic acid sequence. The core of the model is a state model of matches, insertions, and deletions, similar to those used in HMMER and Smith–Waterman algorithms. Two key additions are made to the core. The first addresses frameshifts. The second is a five-region model for introns. The five regions are

- Fixed length, 5' splice site consensus region
- Central part of the intron that constitutes the major part of the intron
- A region of C/T bias upstream of the 3' splice site (polypyrimidine tract)
- An optional region joining the polypyrimidine tract and the 3' splice site
- Fixed length, 3' splice site consensus region

Special care is taken with the overlaps of the splice site consensuses and the coding sequence region. See the Concepts and Conventions section of the online documentation for details of the models and algorithms involved. GeneWise is available online.

2.6.1.2 GFScan

GFScan [45] is a tool for associating genomic DNA sequences with gene families. The family-specific sequence motifs used for matching query sequences are derived from protein motifs in PROSITE and the genomic structure of known members of the family.

2.6.1.3 Genscan

Genscan [46–48] uses an organism-specific probabilistic model of gene structure and composition to predict a "most likely" gene structure (intron, exon, regulatory element, etc.) for a given analyzed sequence. The program was designed primarily for vertebrate genomic sequences using a test set biased toward "short genes with relatively simple exon/intron structure." See the Limitations Web page (http://genes.mit.edu/Limitations.html) for details. The authors suggest masking repeats first with RepeatMasker or CENSOR. Genscan is available online. A BioPerl-based parser is available as part of GeneComber.

2.6.1.4 GeneMark

GeneMark [49,50] is a gene-prediction program that operates by generating a "protein coding potential" distribution over the length of an analyzed genomic DNA sequence; the latter is derived by assessing a "sliding sequence window" with probabilistic (Markov) models of coding and noncoding regions. Genes are defined mainly as open reading frames. The 5' boundary has a range of uncertainty approximately the size of the sliding window, about 100 nucleotides. GeneMark is available online. The Web site also discusses and provides the companion program GeneMark.hmm, an HMM approach that leverages gene structure and uses GeneMark internally. GeneMark.hmm was designed to improve GeneMark's ability to find exact gene boundaries.

2.6.1.5 FGENES, FGENESH

FGENES [51–53] is a gene-prediction program that combines individual gene element (e.g., intron, exon, regulatory element) prediction methods with a dynamic programming approach for finding the optimal combination of these elements along an analyzed sequence. Where FGENES is pattern based, FGENESH uses an HMM approach. Both FGENES and FGENESH are available online. There are several related programs from the same authors:

- FGENES-M: Pattern-based human multiple variants of gene-structure prediction
- FGENESH_GC: HMM-based human gene prediction that allows donor splice site GC donor splice site structure

- BESTORF: Finding potential coding fragment EST/mRNA
- FEX: Finding potential 5' internal and 3' coding exons
- SPL: Search for potential splice sites
- SPLM: Search for human potential splice sites using weight matrices
- RNASPL: Search for exon–exon junction positions in cDNA

2.6.1.6 HMMGene

HMMGene [54] uses a probabilistic (HMM) model of gene structure to predict genes in anonymous DNA. Whole genes are predicted, ensuring that predicted exons are correctly spliced. The program can be used to predict splice sites and start and stop codons. Known features can be used as constraints (e.g., an EST can be "locked" as a noncoding region). HMMGene will then find the best gene structure given the constraints. Because the program uses a probabilistic model of the gene structure, all predictions are accompanied by probabilities, indicating prediction confidence. Suboptimal predictions can also be reported. HMMGene is available online. A BioPerl-based parser is available as part of GeneComber.

2.6.2 Protein Patterns

The tools covered in this section can be used to predict structural or functional motifs and secondary structure. Predicting target structure is critical in advancing candidate compounds (new chemical entities) in the drug-development pipeline.

2.6.2.1 Structural/Functional Motif Prediction

Motifs describing structure and function can be predicted using a variety of techniques. Tools in this section cover the spectrum from using rigid body motion to fold matching to iterative BLAST alignments. We include DomainFinder, ProFunc, EMOTIF, PSI-BLAST, and Maude in this section. Maude is a little out of the mainstream, using a symbolic language to enable simulations, but we think it is worth a look. Table 2.6 summarizes the structural and functional motif prediction tools discussed in this section.

TABLE 2.6
Structural/Functional Motif Prediction Tools

DomainFinder	http://dirac.cnrs-orleans.fr/DomainFinder/
ProFunc	http://www.ebi.ac.uk/thornton-srv/databases/ProFunc/
	http://www.ebi.ac.uk/thornton-srv/databases/profunc/index.html
EMOTIF	http://dlb4.stanford.edu/emotif/
	http://dlb4.stanford.edu/emotif/emotif-maker.html
	http://dlb4.stanford.edu/emotif/emotif-scan.html
	http://dlb4.stanford.edu/emotif/emotif-search.html
PSI-BLAST	http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/psi1.html
Maude	http://maude.cs.uiuc.edu/

2.6.2.1.1 DomainFinder (Dynamical Domains in Proteins)

DomainFinder [55,56] is a program for determining and characterizing "dynamical domains" in proteins. The latter are regions that essentially can move like a rigid body with respect to the broader protein structure. Because the dynamical behavior is implicated in protein function, identification of dynamical domains can facilitate functional inference. DomainFinder is written in Python.

2.6.2.1.2 ProFunc

According to the Web site,

> The aim of the ProFunc [57,58] server is to help identify the likely biochemical function of a protein from its three-dimensional structure. It uses a series of methods, including fold matching, residue conservation, surface cleft analysis, and functional 3D templates, to identify both the protein's likely active site and possible homologues in the PDB. (http://www.ebi.ac.uk/thornton-srv/databases/profunc/index.html)

ProFunc is available online.

2.6.2.1.3 EMOTIF

The EMOTIF [59,60] database is a collection of highly sensitive and specific protein sequence motifs associated with conserved biochemical properties and biological functions. These motifs were derived from protein sequence alignments in the BLOCKS+ and PRINTS databases using the EMOTIF-MAKER program. The EMOTIF-SEARCH program allows the user to identify motifs from the database in a protein query sequence. The EMOTIF-SCAN program retrieves protein sequences containing an EMOTIF specified by a regular expression syntax.

2.6.2.1.4 PSI-BLAST

PSI-BLAST [61,62] iteratively constructs a sequence profile from the highest scoring hits in a series of BLAST searches; the progressively refined profile captures a sequence pattern that can potentially enhance the sensitivity of BLAST similarity searches.

2.6.2.1.5 MEME/MAST/META-MEME

MEME, MAST, and META-MEME are important enough to be covered separately in a later section.

2.6.2.1.6 Maude

Maude [63] is a symbolic language that has been applied to developing qualitative models of signaling pathways and other protein interaction networks. Individual proteins are abstracted as sets of functional domains that specifically participate in protein–protein interactions. Maude can represent both serial and parallel interactions in terms of mediating domains, allowing simulation of biological signaling networks at a manageable yet precise and rigorous level of abstraction. Such simulations can be used to assess and generate hypotheses that can be tested in the laboratory.

2.6.2.2 Secondary-Structure Prediction

Secondary-structure prediction methods have improved substantially in recent years. State-of-the-art approaches utilizing evolutionary constraints derived from high-quality multiple alignments can correctly predict over 70% of residues into one of three states—helix, strand, and other [64]. More specialized methods (e.g., for predicting membrane spanning helices and their topology in integral membrane proteins) can do even better. These results are especially promising, as reliable secondary-structure prediction can enhance the prediction of higher-order protein structure. Indeed, several 3D structure prediction methods use secondary-structure predictions as a starting point. To a limited extent, secondary-structure motifs can even suggest specific fold structures. In addition, secondary-structure predictions can sometimes provide insight into function. For example, active sites of enzymes are typically formed from amino acids positioned in loops. Thus, "identically conserved residues at multiple alignment sites predicted to be in loop regions (i.e., not predicted as helix or strand) could be functional and together elucidate the function of the protein or protein family under scrutiny" [65]. Table 2.7 summarizes the secondary structure prediction tools discussed in this section.

2.6.2.2.1 JPRED

JPRED [66,67] is a neural network-based program for predicting protein secondary structure; sequence residues are assigned to one of three secondary-structure elements (alpha helix, beta sheet, or random coil).

2.6.2.2.2 PSI-PRED

PSI-PRED [68,69] is a neural network-based program for predicting protein secondary structure; query sequences are used as input to the PSI-BLAST program, whose (sequence profile) output forms the actual input to PSI-PRED.

2.6.2.2.3 MEMSAT

MEMSAT [70] is a program for predicting the secondary structure and topology (helical orientation) of integral membrane proteins.

2.6.2.2.4 TMHMM

TMHMM [71,72] is an HMM-based program for predicting the secondary structure and topology of membrane-spanning proteins.

TABLE 2.7
Secondary-Structure Prediction Tools

JPRED	http://www.compbio.dundee.ac.uk/~www-jpred/
PSI-PRED	http://bioinf.cs.ucl.ac.uk/psipred/
MEMSAT	http://saier-144-37.ucsd.edu/memsat.html
TMHMM	http://www.cbs.dtu.dk/services/TMHMM/

TABLE 2.8
Repeat Masking Tools

RepeatMasker	http://www.repeatmasker.org/
	http://www.repeatmasker.org/faq.html
	http://www.geospiza.com/tools/repeatmasker.htm
Censor	http://www.girinst.org/Censor_Server.html

2.6.3 Nucleotide Patterns

The tools covered in this section are used for detecting common repeats, predicting splice sites, and calculating primers for use in polymerase chain reactions (PCR). Table 2.8 summarizes the repeat masking tools discussed in this section.

2.6.3.1 RepeatMasker

RepeatMasker [73–75] is a program for identifying and "masking" frequently occurring nucleic acid sequence elements (tandem repeats, low complexity DNA sequences, etc.). These regions can confound analyses involving similarity search programs. Masked regions can be represented by Ns or by lowercase letters. BLAST programs can use the "-U" option to ignore lowercase bases. RepeatMasker uses Repbase Update as the source for many repeats. A commercial version of RepeatMasker is available from Geospiza (see also CENSOR, by the same group that curates Repbase Update). CENSOR is available online.

2.6.3.2 Splice-Site Prediction

Most eukaryotic genes consist of short coding sequences (exons) and relatively long noncoding sequences (introns). RNA splicing is the process that excises the introns. This process must be precise, as a single nucleotide shift would change the reading frame and result in a different amino acid sequence. Stryer [76] has a cogent description of splice sites. Almost all eukaryotic splice sites, or junctions, have a common pattern: introns begin with GU and end with AG. The 5' consensus sequence in vertebrates is AGGUAAGU and the 3' consensus sequence is ten pyramidines (U or C), followed by any base, and then the AG as just mentioned. Introns also have a branch site 20 to 50 bases upstream of the 3' splice site. The yeast branch site sequence is UACUAAC. The branch site sequence varies in mammals.

The following programs (table 2.9) are worth noting [77]:

- GeneSplicer [78] detects splice sites in the genomic DNA of *Plasmodium falciparum* (malaria), *Arabidopsis thaliana*, human, *Drosophila*, and rice.
- NetGene2 [79,80] is a neural network predictor of splice sites in human, *C. elegans* and *A. thaliana* DNA.
- SpliceView [81] uses a "classification approach (a set of consensuses)."

TABLE 2.9
Splice-Site Prediction Tools

GeneSplicer	http://www.tigr.org/tdb/GeneSplicer/index.shtml
NetGene2	http://www.cbs.dtu.dk/services/NetGene2/
SpliceView	http://l25.itba.mi.cnr.it/~webgene/wwwspliceview.html
SplicePredictor	http://bioinformatics.iastate.edu/cgi-bin/sp.cgi
SPL, SPLM, RNASPL	http://www.softberry.com/berry.phtml?topic=index&group=programs &subgroup=gfind

- SplicePredictor [82–84] implements Bayesian models.
- SPL [51–53] is a search for potential splice sites.
- SPLM [51–53] is a search for human potential splice sites using weight matrices.
- RNASPL [51–53] is a search for exon–exon junction positions in cDNA.

2.6.3.3 Primer Design

Mullis's PCR [76,85–87] allows specified subsequences of DNA to be amplified. This technique has become essential in laboratories since its invention in the mid 1980s. Software programs assist by enabling researchers to design the primers that identify, by hybridization, the desired subsequences. These programs (table 2.10) take into account the three steps in a PCR cycle: strand separation, hybridization of primers, and DNA synthesis. In addition to modeling these steps by using temperature, salt concentration, and durations, primer design programs can filter out poor primers. Typical problems include primers that are insufficiently specific and primers that hybridize to themselves or to their companion primers.

2.6.3.3.1 Primer3

Primer3 [88] is the most popular primer design program. It picks primers for PCR reactions, according to user-specified conditions that are important when attempting to choose the optimal pair of primers for a reaction. These primers include primer size, PCR product size, GC content, oligonucleotide melting temperature, concentrations of various solutions in PCR reactions, primer bending and folding, primer-dimer possibilities, and positional constraints within the source sequence. The program can check existing primers and can design hybridization probes. Primer3 is available online. In addition, EMBOSS's eprimer3 is an interface to Primer3.

TABLE 2.10
Primer Design Tools

Primer3	http://frodo.wi.mit.edu/cgi-bin/primer3/primer3_www.cgi
	http://frodo.wi.mit.edu/primer3/README.primer3_0_9_test
GeneFisher	http://bibiserv.techfak.uni-bielefeld.de/genefisher/
	http://bibiserv.techfak.uni-bielefeld.de/genefisher/help/wwwgfdoc.html
	http://bibiserv.techfak.uni-bielefeld.de/reputer/

2.6.3.3.2 GeneFisher

GeneFisher [89] automates the design of PCR primers used to identify putative homologs of a known gene in DNA derived from closely related organisms. It comes with a built-in alignment tool, allowing GeneFisher to process aligned or unaligned sequences. Alignments can be visually inspected, enabling the user to reject the alignment or accept it and continue to the primer-design step. This step identifies two palindromic positions in the input sequence that are suitable priming sites for one primer, meaning that a single oligonucleotide acts as forward and reverse primer. Possible primer candidates are calculated using REPuter. REPuter identifies exact or approximate (using a Hamming distance model) palindromic repetitive regions on the input sequence and reports suitable PCR priming site positions. GeneFisher is available online.

2.6.4 EMBOSS

EMBOSS [28,90] is an Open Source software package developed to address largely sequence analysis needs. Release 2.9.0 has more than 180 programs. Many deal with patterns. The following programs are examples:

- cpgreport reports CpG rich regions.
- dreg/preg provides a regular expression search of a nucleotide/protein sequence.
- einverted finds DNA inverted repeats.
- eprimer3 picks PCR primers and hybridization oligonucleotides.
- etandem looks for tandem repeats in a nucleotide sequence.
- fuzznuc/fuzzpro provides a nucleic acid/protein pattern search.
- garnier predicts protein secondary structure.
- getorf finds and extracts open reading frames.
- palindrome looks for inverted repeats in a nucleotide sequence.
- restrict finds restriction enzyme cleavage sites.
- sigcleave predicts signal peptide cleavage sites.
- tfextract extracts data from TRANSFAC.

2.6.5 GCG Wisconsin Package

GCG originated in John Devereux's laboratory at the University of Wisconsin–Madison. After having several commercial homes, GCG is now sold by Accelrys. Version 10.3 contains approximately 150 programs. Examples of those programs that deal with patterns include the following:

- FindPatterns identifies sequences that contain short patterns like GAATTC or YRYRYRYR; ambiguities and mismatches are allowed.
- Frames shows open reading frames for the six translation frames of a DNA sequence.
- HmmerPfam compares one or more sequences to a database of profile HMMs, such as the Pfam library.

- HmmerSearch uses a profile HMM as a query to search a sequence database.
- Map maps a DNA sequence, displaying restriction enzyme cut points and protein translations.
- MFold predicts secondary structures for a nucleotide sequence.
- MotifSearch uses a set of profiles (representing similarities within a family of sequences) to search a sequence database.
- Prime selects oligonucleotide primers for a template DNA sequence.
- PrimePair evaluates individual primers to determine their compatibility for use as PCR primer pairs.
- ProfileSearch uses a profile (representing a group of aligned sequences) to search a sequence database.
- SPScan scans protein sequences for the presence of secretory signal peptides.
- StemLoop finds stems (inverted repeats) within a sequence.
- Terminator searches for prokaryotic factor-independent RNA polymerase terminators.

2.6.6 MEME/MAST/META-MEME

The suite of tools described in this section enables the discovery of motifs in groups of related protein or DNA sequences (MEME), the use of these motifs to search a sequence database (MAST), and the integration of these motifs into an HMM (META-MEME).

2.6.6.1 MEME

MEME [91] uses statistical modeling (expectation maximization) techniques to construct conserved sequence motifs from a collection of, presumably, related protein or nucleic acid sequences. According to the Web site,

> MEME represents motifs as position-dependent letter-probability matrices which describe the probability of each possible letter at each position in the pattern. Individual MEME motifs do not contain gaps. Patterns with variable-length gaps are split by MEME into two or more separate motifs. MEME takes as input a group of DNA or protein sequences (the training set) and outputs as many motifs as requested. MEME uses statistical modeling techniques to automatically choose the best width, number of occurrences, and description for each motif. (http://meme.sdsc.edu/meme/intro.html)

2.6.6.2 MAST

MAST [92] uses a motif description (e.g., one generated by MEME) to search a sequence database for sequences matching the motif.

2.6.6.3 META-MEME

META-MEME [93–95] can be used to integrate motifs discovered by MEME, as well as information derived from sets of related sequences, into a single motif-based HMM. The latter can be used to search sequence databases for homologs.

2.6.7 HMMER

HMMER [96] is a freely distributable collection of software for protein-sequence analysis using profile HMMs. A profile HMM [97] is a statistical model of a multiple alignment of sequences drawn from a putative protein family. It captures position-specific information about the relative degree of conservation of different columns in an alignment and the relative likelihood of particular residues occurring in specific positions. Profile HMMs can thus capture the essential features of a structural or functional domain.

They have the following applications:

- Deriving a (profile HMM) model from a sound multiple alignment of known protein family members for use in more effectively searching databases for other, more distantly related family members.
- Enabling the automated annotation of protein domain structure: databases like PFAM and SMART contain HMMER models and curated alignments of known domains. These models can be used to specify a putative domain structure for novel protein query sequences.
- Automated construction and maintenance of large, multiple alignment databases: HMMER can be used derive a model from a well-curated seed alignment. The latter can be used to search databases for more distantly related members of the relevant protein family. These members can then be used to produce a full alignment by aligning all detectable homologues against the seed alignment. This process can be used to automatically organize large numbers of sequences into groups with a probable evolutionary relationship.

Table 2.11 summarizes the program suites discussed in this section.

TABLE 2.11
Program Suites Containing Pattern-Matching Tools

EMBOSS	http://emboss.sourceforge.net/
GCG Wisconsin Package	http://www.accelrys.com/products/gcg_wisconsin_package/index.html
MEME	http://meme.sdsc.edu/meme/website/meme.html
MAST	http://meme.sdsc.edu/meme/website/mast.html
META-MEME	http://metameme.sdsc.edu/
HMMER	http://hmmer.wustl.edu/

2.6.8 Write Your Own

Writing software programs that perform pattern matching can be relatively straightforward for computer programmers. Languages (table 2.12) like Perl [98], Java [99], and Python [100] include regular expression matching. In addition, the BioPerl, BioJava, and BioPython packages include some basic pattern matching specific to bioinformatics. These packages can also be used to execute external programs. For example, EMBOSS programs can be run using BioPerl.

TABLE 2.12
Programming Languages and Libraries

Java	http://java.sun.com/
BioJava	http://www.biojava.org/
Perl	http://www.perl.org/
	http://www.perl.com/
BioPerl	http://www.bioperl.org/
Python	http://www.python.org/
BioPython	http://www.biopython.org/

The following example shows how BioPerl can be used to read protein sequences from a FASTA file and find signal peptide cleavage sites. The file format is inferred from the file extension .fa.

```
use strict;
use warnings;

use Bio::SeqIO;
use Bio::Tools::Sigcleave;

my $seqIterator = Bio::SeqIO->new("-file" =>
"<proteins.fa");

while (my $sequence = $seqIterator->next_seq())
{
   my $sigCleave = Bio::Tools::Sigcleave->new(
                        -seq       => $sequence,
                        -threshold => 3.5,
                        -matrix    => "eucaryotic");

   my $formattedOutput = $sigCleave->pretty_print();
   print("$formattedOutput\n");
}
```

2.7 FUTURE DIRECTIONS

We hope that this chapter has offered a structured overview of existing pattern-based analysis tools and data repositories. Patterns incisively capture the elements most fundamental to biological function. In doing so, they help to manage the considerable extent and complexity of biological sequence data. Pattern-driven approaches are already focusing and enriching discovery efforts. Their further development will undoubtedly present new and exciting applications across the ambitious scope of life sciences research. Consideration of these emerging areas could easily span another chapter. To offer an illustrative glimpse, we present two examples.

2.7.1 Function Prediction in BioPatents

Research and development efforts in the pharmaceutical and biotech industries depend critically on patent protection for commercially valuable biological molecules. In this context, patent laws broadly require clear disclosure of molecular function. Deriving this essential functional information is far from trivial [101]. Basic sequence homology studies are already a first step in these critical research efforts. Pattern-based approaches focusing on fundamental structural and functional motifs can valuably focus expensive and lengthy laboratory efforts. The power of these approaches will invariably increase with expansion and improvement of data repositories and associated analysis tools. The U.S. Patent and Trademark Office already welcomes these sorts of *in silico* studies as valuable adjunct evidence in support of a molecule's functional specification.

2.7.2 Cell Penetrating Peptides

Cell-penetrating peptides (CPPs) [102] are a broad class of molecules that share a capacity to translocate cell membranes and gain access to the cell interior. CPPs show great promise as highly efficient, nontoxic delivery vehicles for various molecules including peptides, oligonucleotides, and proteins. As such, they could enable precise experimental modification of gene and protein activity in living cells, affording novel and incisive avenues for probing biological processes. With further development, CPPs could even facilitate the currently limited therapeutic use of larger molecules such as proteins and oligonucleotides. The fundamental membrane translocation mechanisms associated with CPPs are thought to be diverse and are generally not well understood. Pattern-based approaches may prove valuable in identifying classes of translocation-specific features and motifs. These approaches could focus further studies and facilitate the design of more effective CPPs.

ACKNOWLEDGMENTS

We thank our customers and coworkers for their questions and requests that have given us the opportunity to explore many of the databases and tools described here. We also thank Darryl León for his careful review of this chapter.

REFERENCES

1. Alexander, C. 1979. *The timeless way of building*. New York: Oxford Univ. Press.
2. Gamma, E., R. Helm, R. Johnson, and J. Vlissides. 1994. *Design patterns: Elements of reusable object-oriented software*. New York: Addison Wesley.
3. Friedl, J. 2002. *Mastering regular expressions*. 2nd ed. Sebastopol, CA: O'Reilly.
4. Attwood, T. K. 2000. The role of pattern databases in sequence analysis. *Brief Bioinform* 1(February):45–59.
5. Mulder N. J., R. Apweiler, T. K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bradley, et al. 2005. InterPro, progress and status in 2005. *Nucleic Acids Res* 33, Database Issue:D201–5.

6. Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Bucher, P., Cerutti, L., Copley, R., Courcelle, E., et al. 2005. InterPro User Manual. Available at http://www.ebi.ac.uk/interpro/user_manual.html.
7. Bairoch, A., R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, et al. 2005. The Universal Protein Resource (UniProt). *Nucleic Acids Res* 33, Database Issue:D154–9.
8. Sigrist C. J. A., L. Cerutti, N. Hulo, A. Gattiker, L. Falquet, M. Pagni, A. Bairoch, and P. Bucher. 2002. PROSITE: A documented database using patterns and profiles as motif descriptors. *Briefings in Bioinformatics* 3:265–74.
9. Hulo, N., C. J. Sigrist, V. Le Saux, P. S. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. De Castro, P. Bucher, and A. Bairoch. 2004. Recent improvements to the PROSITE database. *Nucleic Acids Res* 32, Database Issue:D134–7.
10. Attwood, T. K., P. Bradley, D. R. Flower, A. Gaulton, N. Maudling, A. L. Mitchell, G. Moulton, et al. 2003. PRINTS and its automatic supplement, prePRINTS. *Nucleic Acids Res* 31:400–2.
11. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, et al. 2004. The Pfam protein families database. *Nucleic Acids Res*, Database Issue 32:D138–41.
12. Letunic, I., R. R. Copley, S. Schmidt, F. D. Ciccarelli, T. Doerks, J. Schultz, C. P. Ponting, and P. Bork. 2004. SMART 4.0: Towards genomic data integration. *Nucleic Acids Res* 32:D142–4.
13. Letunic, I., L. Goodstadt, N. J. Dickens, T. Doerks, J. Schultz, R. Mott, F. Ciccarelli, R. R. Copley, C. P. Ponting, and P. Bork. 2002. Recent improvements to the SMART domain-based sequence annotation resource. *Nucleic Acids Res* 30:242–4.
14. Schultz, J., R. R. Copley, T. Doerks, C. P. Ponting, and P. Bork. 2000. SMART: A web-based tool for the study of genetically mobile domains. *Nucleic Acids Res* 28:231–4.
15. Haft, D. H., J. D. Selengut, and O. White. 2003. The TIGRFAMs database of protein families. *Nucleic Acids Res* 31:371–3.
16. Wu, C. H., L. L. Yeh, H. Huang, L. Arminski, J. Castro-Alvear, Y. Chen, Z. Hu, et al. 2003. The protein information resource. *Nucleic Acids Res* 31:345–7.
17. Wu, C. H., H. Huang, A. Nikolskaya, Z. Hu, L. S. Yeh, and W. C. Barker. 2004. The iProClass integrated database for protein functional analysis. *Comput Biol Chem* 28:87–96.
18. Wu, C. H., A. Nikolskaya, H. Huang, L. S. Yeh, D. A. Natale, C. R. Vinayaka, Z. Z. Hu, et al. 2004. PIRSF: Family classification system at the Protein Information Resource. *Nucleic Acids Res* 32:D112–4.
19. Gough, J., K. Karplus, R. Hughey, and C. Chothia. 2001. Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *J Mol Biol* 313:903–19.
20. Servant, F., C. Bru, S. Carrère, E. Courcelle, J. Gouzy, D. Peyruc, and D. Kahn. 2002. ProDom: Automated clustering of homologous domains. *Brief Bioinform* 3:246–51.
21. Corpet, F., F. Servant, J. Gouzy, and D. Kahn. 2000. ProDom and ProDom-CG: Tools for protein domain analysis and whole genome comparisons. *Nucleic Acids Res* 28:267–9.
22. Henikoff, J. G., E. A. Greene, S. Pietrokovski, and S. Henikoff. 2000. Increased coverage of protein families with the blocks database servers. *Nucleic Acids Res* 28:228–30.
23. Henikoff, S., J. G. Henikoff, and S. Pietrokovski. 1999. Blocks+: A non-redundant database of protein alignment blocks derived from multiple compilations. *Bioinform* 15:471–9.

24. Marchler-Bauer, A., J. B. Anderson, P. F. Cherukuri, C. DeWeese-Scott, L. Y. Geer, M. Gwadz, S. He, et al. 2005. CDD: A Conserved Domain Database for protein classification. *Nucleic Acids Res* 33, Database Issue:D192–6.
25. Marchler-Bauer A., and S. H. Bryant. 2004. CD-Search: Protein domain annotations on the fly. *Nucleic Acids Res* 32, Web Server Issue:W327–31.
26. Bairoch, A., B. Boeckmann, S. Ferro, and E. Gasteiger. 2004. Swiss-Prot: Juggling between evolution and stability. *Brief Bioinform* 5:39–55.
27. Boeckmann, B., A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, et al. 2003. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res* 31:365–70.
28. Markel, S., and D. León. 2003. *Sequence analysis in a nutshell: A guide to common tools and databases*. Sebastopol, CA: O'Reilly.
29. Kabsch, W. and C. Sander. 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* 22 (December):2577–637.
30. Adzhubei, I. A., and A. A. Adzhubei. 1999. ISSD version 2.0: Taxonomic range extended. *Nucleic Acids Res* 27:268–71.
31. Pearl, F., A. Todd, I. Sillitoe, M. Dibley, O. Redfern, T. Lewis, C. Bennett, 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, Database Issue:D247–51.
32. Pearl, F. M. G., D. Lee, J. E. Bray, I. Sillitoe, A. E. Todd, A. P. Harrison, J. M. Thornton, and C. A. Orengo. 2000. Assigning genomic sequences to CATH. *Nucleic Acids Res* 28:277–82.
33. Orengo, C. A., A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton. 1997. CATH—A hierarchic classification of protein domain structures. *Structure* 5:1093–108.
34. Proudfoot, N., and G. G. Brownlee. 1976. *Nature* 263:211–14.
35. Efstratiadis, A., J. W. Posakony, T. Maniatis, R. M. Lawn, C. O'Connell, R. A. Spritz, J. K. DeRiel, B. G. Forget, S. M. Weissman, J. L. Slightom, et al. 1980. The structure and evolution of the human β-globin gene family. *Cell* 21:653–68.
36. Corden, J., B. Wasylyk, A. Buchwalder, P. Sassone-Corsi, C. Kedinger, and P. Chambon. 1980. Promoter sequences of eukaryotic protein-encoding genes. *Science* 209:1406–14.
37. Roberts, R. J., T. Vincze, J. Posfai, and D. Macelis. *Nucleic Acids Res* 33:D230–2.
38. Jurka, J., V. V. Kapitonov, A. Pavlicek, P. Klonowski, O. Kohany, and J. Walichiewicz. 2005. Repbase update, a database of eukaryotic repetitive elements. *Cytogenet Genome Res* 110:462–7.
39. Jurka, J. 2000. Repbase update: A database and an electronic journal of repetitive elements. *Trends Genet* 9:418–20.
40. Jurka, J. 1998. Repeats in genomic DNA: Mining and meaning. *Curr Opin Struct Biol* 8:333–7.
41. Wingender, E., X. Chen, E. Fricke, R. Geffers, R. Hehl, I. Liebich, M. Krull, et al. 2001. The TRANSFAC system on gene expression regulation. *Nucleic Acids Res* 29:281–3.
42. Baxevanis, A. D., and B. F. F. Ouellette. 2001. *Bioinformatics: A practical guide to the analysis of genes and proteins*. 2nd ed. p. 235. New York: Wiley-Interscience.
43. Rivas, E., and S. R. Eddy. 2000. Secondary structure alone is generally not statistically significant for the detection of noncoding RNAs. *Bioinformatics* 16:583–605.

44. Birney, E., M. Clamp, and R. Durbin. 2004. GeneWise and genomewise. *Genome Res* 14:988–95.
45. Xuan, Z. Y., W. R. McCombie, and M. Q. Zhang. 2002. GFScan: A gene family search tool at genomic DNA sequence level. *Genome Res* 12:1142–9.
46. Burge, C., and S. Karlin. 1997. Prediction of complete gene structures in human genomic DNA. *J Mol Biol* 268:78–94.
47. Burge, C. B. 1998. Modeling dependencies in pre-mRNA splicing signals. In *Computational methods in molecular biology*, ed. S. Salzberg, D. Searls, and S. Kasif, 127–63. Amsterdam: Elsevier Science.
48. Burge, C. B., and S. Karlin. 1998. Finding the genes in genomic DNA. *Curr Opin Struct Biol* 8:346–54.
49. Borodovsky, M., and J. McIninch. 1993. GeneMark: Parallel gene recognition for both DNA strands. *Compu Chem* 17 (19):123–33.
50. Lukashin, A., and M. Borodovsky. 1998. GeneMark.hmm: New solutions for gene finding. *Nucleic Acids Res* 26:1107–15.
51. Salamov, A., and V. Solovyev. 2000. Ab initio gene finding in Drosophila genomic DNA. *Genome Res* 10:516–22.
52. Solovyev, V. V., A. A. Salamov, and C. B. Lawrence. 1995. Identification of human gene structure using linear discriminant functions and dynamic programming. In *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology,* eds. C. Rawling, D. Clark, R. Altman, L. Hunter, T. Lengauer, and S. Wodak, 367–75. Cambridge, England: AAAI Press.
53. Solovyev, V., and C. B. Lawrence. 1993. Prediction of human gene structure using dynamic programming and oligonucleotide composition. Abstract. *Abstracts of the 4th Annual Keck Symposium,* 47.
54. Krogh, A. 1997. Two methods for improving performance of an HMM and their application for gene finding. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology,* ed. T. Gaasterland, et al., 179–86. Menlo Park, CA: AAAI Press.
55. Hinsen, K., A. Thomas, and M. J. Field. 1999. Analysis of domain motions in large proteins. *Proteins* 34:369–82.
56. Hinsen, K. 1998. Analysis of domain motions by approximate normal mode calculations. *Proteins* 33:417–29.
57. Laskowski, R. A., J. D. Watson, and J. M. Thornton. 2005. ProFunc: A server for predicting protein function from 3D structure. *Nucleic Acids Res* 33:W89–93.
58. Laskowski, R. A., J. D. Watson, and J. M. Thornton. 2005. Protein function prediction using local 3D templates. *J Mol Biol* 351:614–26.
59. Huang, J., and D. Brutlag. 2001. The emotif database. *Nucleic Acids Res* 29:202–4.
60. Neville-Manning, C., T. Wu, and D. Brutlag. 1998. Highly specific protein sequence motifs for genome analysis. *Proc Natl Acad Sci USA* 95:5865–71.
61. Altschul, S. F., T. L. Madden, A. A. Schaffer, J. Zhang, Z. Anang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLST: A new generation of protein database search programs. *Nucleic Acids Res* 25:3389–402.
62. Korf, I., M. Yandell, and J. Bedell. 2003. *BLAST.* Sebastopol, CA: O'Reilly.
63. Sriram, M. G. 2003. Modelling protein functional domains in signal transduction using Maude. *Briefings in Bioinformatics*. 4(3):236–45.
64. Rost, B. 2003. Rising accuracy of protein secondary structure prediction. In *Protein structure determination, analysis, and modeling for drug discovery,* ed. D. Chasman, 207–49. New York: Dekker.

65. Heringa, J. 2000. Predicting secondary structure from protein sequences. In *Bioinformatics: Sequence, structure and databanks,* ed. D. Higgins and W. Taylor, 113–41. New York: Oxford Univ. Press.
66. Cuff, J. A., M. E. Clamp, A. S. Siddiqui, M. Finlay, and G. J. Barton. 1998. Jpred: A consensus secondary structure prediction server. *Bioinformatics* 14:892–3.
67. Cuff, J. A., and G. J. Barton. 1999. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *PROTEINS: Structure, Function and Genetics* 34:508–19.
68. McGuffin, L. J., K. Bryson, and D. T. Jones. 2000. The PSIPRED protein structure prediction server. *Bioinformatics* 16:404–5.
69. Jones, D. T. 1999. Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol* 292:195–202.
70. Jones, D. T., W. R. Taylor, and J. M. Thornton. 1994. A model recognition approach to the prediction of all-helical membrane protein structure and topology. *Biochemistry* 33:3038–49.
71. Sonnhammer, E. L. L., G. von Heijne, and A. Krogh. 1998. A hidden Markov model for predicting transmembrane helices in protein sequences. *Proceedings of the Sixth International Conference on Intelligent Systems for Molecular Biology,* 175–182.
72. A. Krogh, B. Larsson, G. von Heijne, and E. L. L. Sonnhammer. 2001. Predicting transmembrane protein topology with a hidden Markov model: Application to complete genomes. *J Mol Biol* 305:567–80.
73. Smit, A. F. A, and P. Green. RepeatMasker. http://repeatmasker.org.
74. Lander, E. S., L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, et al. 2001. Initial sequencing and analysis of the human genome. *Nature* 409:860–921.
75. Smit, A. F. A. 1999. Interspersed repeats and other mementos of transposable elements in the mammalian genomes. *Curr Opin Genet Devel* 9:657–63.
76. Stryer, L, J. M. Berg, and J. L. Tymoczko. 2002. *Biochemistry.* 5th ed. New York: W. H. Freeman.
77. Mathé, C., M.-F. Sagot, T. Schiex, and P. Rouzé. 2002. Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res* 30:4103–17.
78. Pertea, M., X. Lin, and S. L. Salzberg. 2001. GeneSplicer: A new computational method for splice site prediction. *Nucleic Acids Res* 29:1185–90.
79. Hebsgaard, S. M., P. G. Korning, N. Tolstrup, J. Engelbrecht, P. Rouze, and S. Brunak. 1996. Splice site prediction in Arabidopsis thaliana DNA by combining local and global sequence information. *Nucleic Acids Res* 24:3439–52.
80. Brunak, S., J. Engelbrecht, and S. Knudsen. 1991. Prediction of human mRNA donor and acceptor sites from the DNA sequence. *J Mol Biol* 220:49–65.
81. Rogozin, I. B., and L. Milanesi. 1997. Analysis of donor splice signals in different organisms. *J Mol Evol* 45:50–9.
82. Brendel, V., L. Xing, and W. Zhu. 2004. Gene structure prediction from consensus spliced alignment of multiple ESTs matching the same genomic locus. *Bioinformatics* 20:1157–69.
83. Kleffe, J., K. Hermann, W. Vahrson, B. Wittig, and V. Brendel. 1996. Logitlinear models for the prediction of splice sites in plant pre-mRNA sequences. *Nucleic Acids Res* 24:4709–18.
84. Brendel, V., J. Kleffe, J. C. Carle-Urioste, and V. Walbot. 1998. Prediction of splice sites in plant pre-mRNA from sequence properties. *J Mol Biol* 276:85–104.
85. Mullis, K. 1990. The unusual origin of the polymerase chain reaction. *Scientific American* April:56–65.

86. Watson, J. D., M. Gilman, J. Witkowski, and M. Zoller. 1992. *Recombinant DNA*. 2nd ed., 79–98. New York: W. H. Freeman.
87. Rychlik, W. 1993. Selection of primers for polymerase chain reaction. In *Methods in molecular biology, Vol. 15: PCR protocols: Current methods and applications*, ed. B. A. White, 31–40. Totowa, NJ: Humana.
88. Rozen, S., and H. J. Skaletsky. 2000. Primer3 on the WWW for general users and for biologist programmers. In *Bioinformatics methods and protocols: Methods in molecular biology,* ed. S. Krawetz and S. Misener, 365–86. Totowa, NJ: Humana.
89. Giegerich, R., F. Meyer, and C. Schleiermacher. GeneFisher–1996. Software support for the detection of postulated genes. In *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology* (ISMB-96), 68–77. Menlo Park, CA: AAAI Press.
90. Rice, P., I. Longden, and A. Bleasby. 2000. EMBOSS: The European Molecular Biology Open Software Suite. *Trends in Genetics* 16:276–7.
91. Bailey, T. L., and C. Elkan. 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 28–36. Mello Park, CA: AAAI Press.
92. Bailey, T. L., and M. Gribskov. 1998. Combining evidence using p-values: Application to sequence homology searches. *Bioinformatics* 14:48–54.
93. Grundy, W. N., T. L. Bailey, C. P. Elkan, and M. E. Baker. 1997. Meta-MEME: Motif-based hidden Markov models of biological sequences. *Comput App Biosci* 13:397–406.
94. Grundy, W. N. 1998. A bayesian approach to motif-based protein modeling. Ph.D. diss., Univ. of California, San Diego.
95. Grundy, W. N., T. L. Bailey, and C. P. Elkan. 1996. ParaMEME: A parallel implementation and a web interface for a DNA and protein motif discovery tool (with corrections). *Comput App Biosci* 12:303–10.
96. Eddy, S. HMMER user guide. Available at: ftp://ftp.genetics.wustl.edu/pub/eddy/hmmer/ CURRENT/Userguide.pdf.
97. Durbin, R., S. Eddy, A. Krogh, and G. Mitchison. 1998. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge: Cambridge Univ. Press.
98. Wall, L., T. Christiansen, and J. Orwant. 2000. *Programming Perl*. 3rd ed. Sebastopol, CA: O'Reilly.
99. Flanagan, D. 2005. *Java in a nutshell*. 5th ed. Sebastopol, CA: O'Reilly.
100. Lutz, M. 2001. *Programming Python*. 2nd ed. Sebastopol, CA: O'Reilly.
101. Harrison, R. 2003. *Function prediction in BioPatents BIOSILICO*, 1:12–13.
102. Järver, P., and Ü. Langel. 2004. The use of cell-penetrating peptides as a tool for gene regulation. *Drug Discov Today* 9(9):395–402.

3 Tools for Computational Protein Annotation and Function Assignment

Jaume M. Canaves
University of California, San Diego

CONTENTS

3.1 INTRODUCTION TO FUNCTIONAL ANNOTATION

Comprehensive protein annotation and function assignment are the first steps in target selection and validation for drug discovery. Current advances in genome sequencing and high-throughput structural genomics have resulted in an explosive growth in the number of protein sequences and structures without an assigned function. Approximately one-third of protein-coding genes in newly sequenced prokaryotic genomes, and even larger numbers in eukaryotic genomes, lack functional assignment. There are extreme cases like the genome of *Plasmodium falciparum*, where the function of approximately 60% of predicted proteins is unknown [1]. This situation is not limited to protein sequences but recently has expanded to include protein structures: Up to 60% of protein structures deposited in the Protein Data Bank (PDB) [2] by some structural genomics centers do not have any function assignment. That vast and constantly growing repository of sequences and structures is a rich potential source for the identification of new drug-discovery targets. Protein function has multiple definitions. To a cell biologist, function might refer to the network of interactions in which the protein participates or to the location to a certain cellular compartment. To a biochemist, function refers to the metabolic process in which a protein is involved or to the reaction catalyzed by an enzyme. Developmental biologists or physiologists might include temporal patterns of expression or tissue specificity in their definition of protein function. From a drug-discovery point of view, function assignment means elucidation of biochemical function, although additional levels of annotation can be used as qualifiers to evaluate the prospective usefulness of a potential drug-discovery target. This level of function assignment is usually called the *molecular function*. Biochemical and/or molecular function can be deduced in many cases from any combination of sequence, structure, and contextual information. In some cases, further levels of protein function such as cellular location, interacting partners, participation in regulatory networks or metabolic pathways, and so forth, are possible. Function assignment can be achieved experimentally in the laboratory or computationally, and generally there is a strong feedback between the experimental and *in silico* research components in drug-discovery efforts. Computational findings can assist laboratory biologists and chemists to direct experimental design, and subsequent experimental findings can suggest new courses of action for computational biologists. For the computational biologist, the terms *function*

annotation and *function assignment* are somewhat interchangeable and blurred. Preferably, the use of the term *function assignment* should be limited to designate the attribution of an enzymatic function or gene ontology. For the individual pieces of evidence that point to a certain biological function, it is more appropriate to use the term *function annotation*. Frequently, the complexity of function assignment is beyond a single sentence. For instance, tubulin, a component of microtubules and a target for anticancer drugs like Taxol or *Vinca* alkaloids [3], is not only a structural protein but also a GTP-hydrolizing enzyme. Its structural role is not limited to being part of the cytoskeleton but includes roles in intracellular protein and organelle traffic, protein and organelle scaffolding, formation of the mitotic spindle during cell division, or as a component of motile systems. Primary sequence determines protein structure, and in turn protein structure determines protein function. Function is the only element of this first paradigm, central to protein function assignment that cannot be addressed computationally. Therefore, the inference of molecular function from sequence or structure is one of the ultimate goals for postgenomic bioinformatics. The second paradigm in the field of protein function assignment is that similar sequences or similar structures have similar function, hence function assignment can be performed by transferring the annotation of a protein experimentally characterized to the protein being annotated (fig. 3.1). The second paradigm is not an absolute truth. Enzyme Commission numbers classify structurally similar enzymes as functionally dissimilar and vice versa, and very divergent functions are possible in proteins with high levels of sequence conservation. Structural databases such as SCOP [4] or CATH [5] group functionally dissimilar proteins into structurally similar groups. Thus, one should carefully evaluate transference of function based on

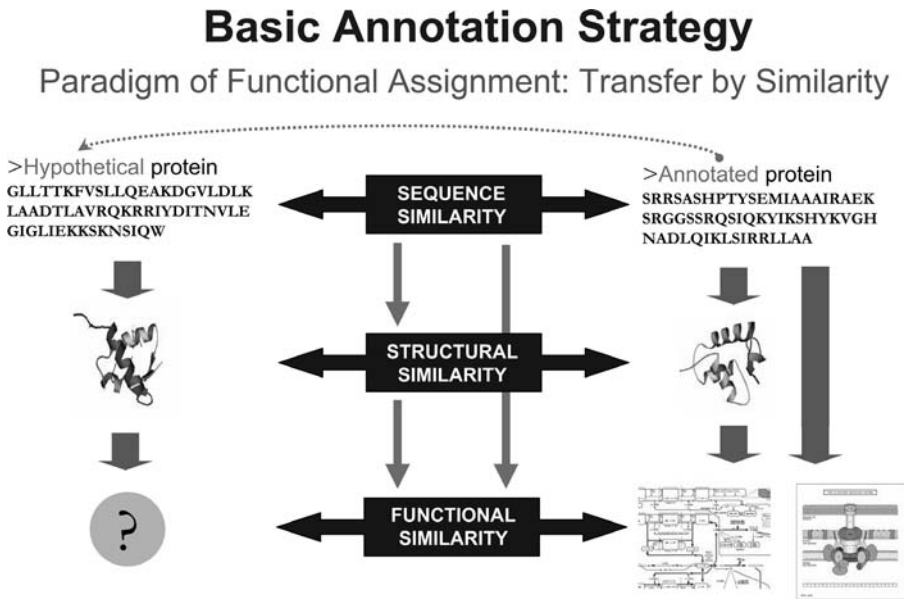


FIGURE 3.1 The paradigm of sequence similarity-based functional assignment is the transfer of annotation between similar proteins.

sequence or structural similarity, always taking into account that the computational results are simply models that require experimental confirmation. Target protein annotation is not limited to *de novo* function assignment. There is a sizeable degree of contamination in public databases due to function assignments being erroneously transferred to newly annotated proteins. For mission critical tasks, every annotation should be deemed suspect. Consequently, correction of assigned function, known as *reannotation*, plays a supplementary yet crucial role in computational function annotation and assignment. In addition, protein annotation can be used to add value to existing assignments. Proteins with known functions can be re-examined to discover new functions that could lead to novel approaches to modulate an already validated drugable target. For instance, through *in silico* methods it might be possible to identify new ligand-binding or protein–protein interaction sites in a target protein for which pharmacological value is already established. The challenges of high-throughput automated function assignment in genome annotation projects and the challenges faced by annotators in target validation projects are different. The main goal of genome-scale annotation is to provide function assignments for all proteins in a genome in an acceptable timescale. There is a trade-off between the depth and quality of the annotations and the time devoted to that process. Conversely, annotators evaluating the possible value of a target for drug discovery have accuracy as their first priority. Accuracy demands human curation and examination of the problem from many possible angles to minimize the chances of erroneous assignment. Genome-scale annotation has a certain built-in margin of error, which means that, from a drug discovery point of view, every public annotation is technically questionable. Often, the confirmation of the function assignment is a fairly routine process, but in other cases that confirmation requires considerable effort. Despite the need of manual curation for mission critical targets, high-throughput automated protein annotation methods are still extremely useful for target drug discovery as target preprocessing and prioritization tools. In that role, high-throughput automatic methods can reduce the number of potential targets from thousands or tens of thousands to a manageable number that can in turn be re-evaluated and validated by expert human curators.

3.2 SEQUENCE-BASED FUNCTION ASSIGNMENT

A usual starting point for a function assignment project is a large-scale survey in search of suitable targets potentially susceptible to pharmacological intervention. In other scenarios, the protein identified as a possible target might be the result of biochemical or yeast two-hybrid experiments, proteomics analysis, microarray data (DNA, RNA, protein, chemical, or antibody), and so on. In every case, the starting point of the annotation project is the primary sequence of a potential target protein. The annotator normally faces two possible scenarios: either the protein lacks any function assignment (proteins usually designated as “hypothetical proteins”) or the protein has an assigned but experimentally unconfirmed function assignment. In either case, if there is no compelling and irrefutable evidence supporting a certain function, annotators need to verify the correctness of the function assignment before committing to bench studies. Even if the annotation originates from a manually

curated database, the existing annotation should be verified prior to using the target protein for critical uses.

3.2.1 Assigning Function by Direct Sequence Similarity

The basic approach to function assignment is to search for functionally annotated sequences similar to the query protein and then transfer their function to the query protein. Sequence similarity between two proteins is assessed through the alignment of their primary sequences. Similarity should not be confused with homology. Homology implies sequence divergence from a common ancestor, whereas analogy indicates the acquisition of common features from evolutionarily unrelated ancestors through convergent evolution [6]. Both homology and analogy can result in sequence similarity. The primary goal in annotation is to detect sequence homologies, because homologous proteins share a common ancestor and a common structural fold, often resulting in a common function. If two proteins are very similar, although not necessarily homologous, it can be expected that their three-dimensional (3D) structures will also be alike, and therefore their functions will also be related. Accordingly, annotation transfer according to similarity can be performed from homologous and analogous sequences, although these concepts are not interchangeable. Homology search methods are based on statistical principles. Therefore, if the similarities observed are very unlikely to occur by chance, it is generally assumed that both proteins are related through evolution.

Pairwise sequence alignment using one of the multiple flavors of Basic Local Alignment Search Tool (BLAST) [7] or FastA [8] is the most common method used for similarity searches. BLAST and FastA are both simplifications of the Smith–Waterman algorithm. BLAST is faster than FastA or the original Smith–Waterman, but it is less sensitive. For some time, FastA was the most widely used search method, but it has now been superseded by improved variants of BLAST. BLAST similarity searches are generally performed against comprehensive databases like the National Center for Biotechnology Information (NCBI) RefSeq database [9], Swiss-Prot and TrEMBL [10], Protein Information Resource (PIR) [11], the sequences in PDB [2], or organism-specific databases (e.g. human, mouse, or *Drosophila* genomes). Although the actual biological meaning of a match between two proteins is not guaranteed, the statistical significance of the observed match can be evaluated, specifically determining the likelihood of finding an alignment between two protein sequences given the size and composition of the searched database. The most frequently used parameter to evaluate the significance of an alignment is the expectation value, or E-value, which is the number of distinct alignments with scores equivalent to or better than the raw alignment score that are expected to occur in a database search by chance. Thus, E-values depend on database size. A lower E-value indicates a more significant alignment.

If significant similarities between uncharacterized and annotated sequences are found, the transfer of annotation is straightforward at high identity levels (> 40%). Below that sequence-identity level, the establishment of firm sequence-function relationships can start to present some problems. The signal starts to become blurred at 20 to 30% identity, the so-called *Twilight Zone* [12,13]. If identity levels are even lower, the region is called the *Midnight Zone* (fig. 3.2). For identity levels over 30%,

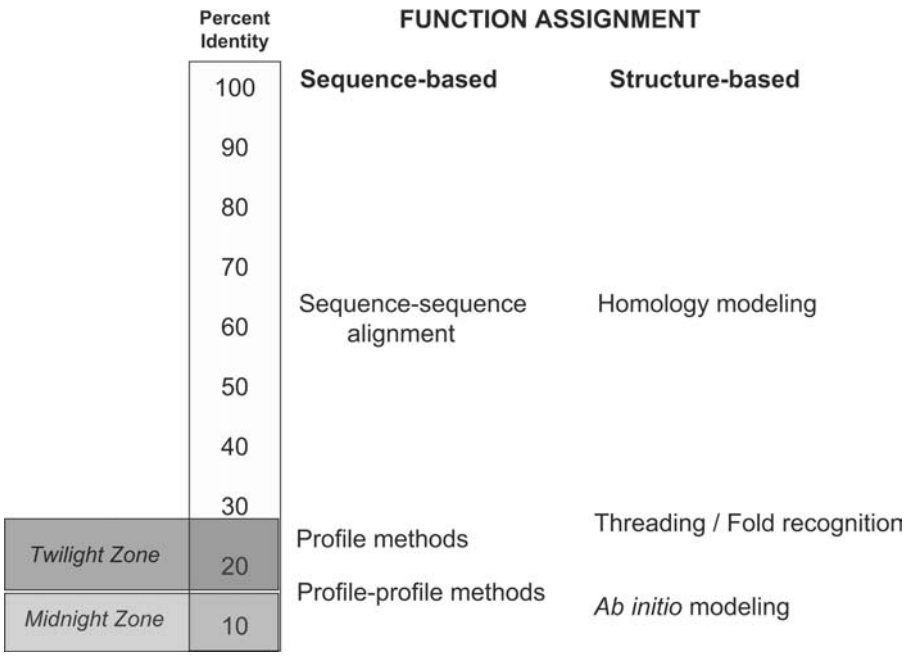


FIGURE 3.2 Techniques used for sequence- and structure-based function assignment and their relationship to percentage of sequence identity between the query protein and observed matches.

enzyme class can be predicted with at least 90% accuracy using sequence-similarity methods, both in single and multidomain proteins. On the other hand, when identity levels are below 30%, functional variation is significant, and structural or other additional data are often necessary to confirm function assignments based exclusively on sequence similarity [14]. Regular BLAST can be used to detect distant homologs close to the Twilight Zone with the appropriate choice of substitution matrix. The default matrix used by NCBI-BLAST 2.0 is BLOSUM62 [15], derived from sequences sharing 62% sequence identity. The BLOSUM45 matrix, derived from more distant sequences (45% identity), will detect more distant sequences in long and weak alignments. Matrix selection depends not only on the degree of divergence of the sequences that we want to detect but also on the length of the query sequence. BLOSUM62 is adequate for lengths over 85 residues, but for shorter lengths other matrices are recommended (e.g., BLOSUM80 for query lengths between 50 and 85 residues, PAM70 for lengths between 35 and 50 residues, and the PAM30 matrix for lengths below 35 residues [16]).

3.2.2 Detection of Distant Similarities with Profile Methods

In a best-case scenario, if a match is almost identical to the query sequence, the functional annotation can be transferred directly from the matching sequence to the

query sequence with a high confidence level. If no close homolog can be identified, the next stage in function annotation is to search for more distant homologs with the hope of detecting significant similarities to a protein with a defined function. A number of matching sequences might be detected by BLAST, but if they fall in the Twilight Zone, the appropriate approach to evaluate the significance of those matches is to use methods that have been specifically designed to detect distant sequence homologies. Even if this effort is not successful in detecting a functionally characterized protein for annotation transfer, the use of these methods might increase the number of similar proteins available. Those sequences could be then used for subsequent computational analyses aimed at ascertaining the role of the query protein.

One of the limitations of pairwise sequence alignments is that the amount of information contained in only two sequences is very limited. A way to address this problem is to collect a larger number of sequences and compare a model built from those sequences against the target database in order to detect distant homologies. Those models, called *profiles*, are scoring tables derived from multiple sequence alignments that define the conservation for each position, the residues present in any given position, and which positions are susceptible to contain insertions. It has been shown that methods using multiple sequences (known as *profile methods*) are vastly superior to methods that rely on single query sequences [17], and the efficiency of these methods is surpassed by approaches that use profiles for both the query and target sequences (known as *profile–profile methods*) [18,19]. Position-Specific-Iterative BLAST (PSI-BLAST) [20] is by far the most used and fastest of the profile methods. PSI-BLAST is an iterative algorithm that uses a position-specific scoring matrix (PSSM) reflecting the distribution of amino acids along the query sequence. In each successive PSI-BLAST iteration, an improved PSSM is produced and used for a new search against the target database. Using this iterative procedure, PSI-BLAST is a very effective tool, capable of uncovering many distant protein relationships that would be missed by regular BLAST. The accuracy of PSI-BLAST can be increased by jumpstarting it with a high-quality multiple sequence alignment computed outside PSI-BLAST. The increase in sensitivity with respect to BLAST comes at a cost, as PSI-BLAST requires a higher level of user expertise both in operation and data interpretation. For instance, the user must select sequences to be included in the next PSSM during the search of very distant homologies. The evaluation of the matches returned by PSI-BLAST also might be laborious, because many of them might have identity levels to the query sequence that situate them well into the Twilight Zone.

HMMER [21] and SAM [22,23] are two popular profile tools based on Hidden Markov Models (HMMs). The most important factor affecting the performance of these methods is the quality of the multiple sequence alignments used as input. It has been observed that, when used with default parameters, SAM consistently produces better models than HMMER. On the other hand, HMMER is much faster than SAM when searching large databases (at least 2,000 sequences) [24]. Summarizing, in general, SAM (the current version is SAM-T02) is better than HMMER at detecting distant homologies, and HMMER is better than PSI-BLAST. On the other hand, PSI-BLAST is faster than HMMER, and HMMER is in turn faster than SAM (fig. 3.3). Accordingly, a good strategy to detecting distant homologies using

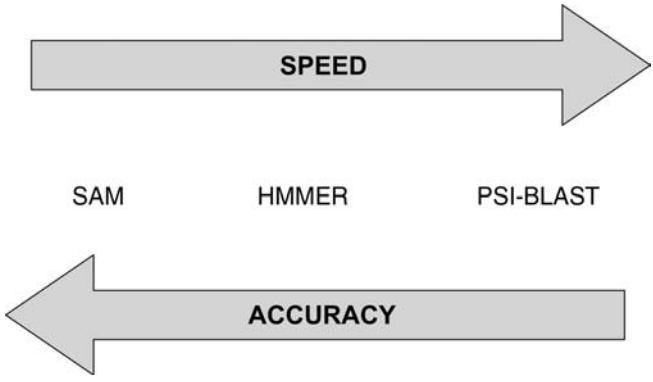


FIGURE 3.3 Comparison of the speeds and accuracies of PSI-BLAST, HMMER, and SAM, three popular profile methods used in the identification of distant sequence homologies.

profile methods is to use PSI-BLAST first; if PSI-BLAST is not successful, then apply HMMER and/or SAM. All the tools mentioned (BLAST, PSI-BLAST, FastA, HMMER, and SAM) can be accessed through public servers or downloaded and installed locally.

To increase detection sensitivity beyond the limits of profile methods, it is also possible to encode the sequences of the searched database into profiles. These procedures are known as profile–profile methods, and at low-identity levels they clearly outperform direct pairwise comparison and profile methods [25] (fig. 3.2). The recognition sensitivity and alignment accuracy obtained through the application of profile–profile methods can be as much as 30% higher than profile methods [26]. For distantly related proteins, structure is more conserved than the underlying sequences. Therefore, relationships between those sequences are only apparent at the structural level. Such relationships can be detected using sensitive profile–profile methods that compare profiles generated from the query sequence against profiles generated from PDB [2] or structural domain databases such as SCOP [4]. FFAS03 [27,28] and ORFeus [29] are two examples of profile–profile methods. FFAS03 compares sequence profiles with each other, although the profiles are generated using a method different than that utilized by PSI-BLAST. The databases searched by FFAS03 are PDB, SCOP, Pfam-A, and COG. In ORFeus, the sequence profiles are generated as in FFAS, but secondary-structure prediction information produced by PSIPRED [30] is added to the scoring function. Both FFAS03 and ORFeus are available to the public through the Structure Prediction Meta Server, but no local executables are available. COMPASS [31,32] is another profile–profile method based on PSI-BLAST that shares important similarities with FFAS. In this case, no Web version is available, but the program can be downloaded for local installation.

Recently, important advances have been made in the application of HMMs to profile–profile methods. The profile–profile approach used by the program COACH [33] involves the alignment of two multiple sequence alignments by constructing an HMM from the first alignment and aligning the second multiple sequence alignment to the HMM. There is no Web server version of COACH, but Linux and Windows

versions of the program are available for download. Benchmarking of COACH indicates that the program compares favorably to COMPASS. HHpred [34] is the most recent addition to the ever-growing collection of tools for remote homology detection. HHpred is a profile–profile method that compares two profile HMMs and utilizes a user-supplied sequence or a multiple sequence alignment as input. The program can be accessed through the Web server or installed locally. When benchmarked against COMPASS, HHpred outperformed it in both accuracy and speed.

The searches through most of Web-based profile–profile methods are limited to profiles corresponding to public databases like Pfam. Both the homology search programs and the databases construct their respective profiles using alignments generated by methods that can be far from optimal. These limitations can be avoided by using profile-sequence methods or profile–profile methods as a means to collect target leads, sometimes even too remote to be statistically significant. Then optimized multiple sequence alignments for the query sequence and target lead can be produced using the latest methods. Finally, locally installed state of the art profile–profile methods like COMPASS, COACH, or HHMpred can be used to compare profiles generated from the optimized alignments and assess the significance of the leads previously gathered.

3.2.3 Multiple Sequence Alignment

The alignment of multiple sequences plays a fundamental role in function assignment and annotation. Accurate alignments that include as many confident sequence homologs as possible are crucial to maximize the quality of the profiles used in profile-sequence or profile–profile sequences [35], which in turn augment the sensitivity of these methods. Multiple sequence alignments can validate the similarities observed in pairwise alignments and reveal conserved features such as domain organization, catalytic residues, or residues important for protein–protein interactions. Thus, multiple sequence alignments provide valuable insights into protein function. Multiple sequence alignments are also used for secondary- and tertiary-structure prediction, homology modeling, characterization of single nucleotide polymorphisms and alternatively spliced variants, or for phylogenetic analysis.

Multiple sequence alignment methods attempt to produce results that are mathematically and statistically optimal. Still, there are numerous factors that can introduce bias and skew the final results. Redundant data sets, where multiple instances of a protein or very close homologs are overrepresented, are one major cause of bias. Some multiple sequence alignment programs can weigh those overrepresented sequences favorably compared to more divergent ones, resulting in the misalignment of the latter. An extreme example would be if there were 30 identical sequences and a shorter sequence sharing a low level of identity with the others. In this case, the mathematical cost of opening gaps and extending gaps in 30 sequences could be smaller than the cost of completely misaligning a single sequence. Clustering programs can be used to reduce dataset redundancy. These programs group similar proteins in clusters, and then a single representative for each cluster can be selected for alignment. The program CD-HIT [36] is very effective in fast-clustering large-sequence datasets. The program processes a database containing a large number of

protein sequences at a user-selected identity level and returns a nonredundant sequence database containing a single representative per cluster plus a file describing the content of each cluster. Other programs that can be used to reduce redundancy through clustering are blastclust (included in the NCBI-BLAST package) [7], DIVCLUS [37], GeneRAGE [38], or TribeMCL [39]. DIVCLUS and GeneRAGE are especially useful when clustering multidomain proteins. The Decrease Redundancy tool at Expasy offers a convenient alternative to stand-alone programs as long the number of sequences to cluster is not too high.

3.2.3.1 Multiple Sequence Alignment Methods

The most commonly used multiple sequence alignment methods are based on the progressive-alignment approach [40,41]. ClustalW [42] is a widely used implementation of this strategy and currently one of the most popular automated multiple sequence alignment tools. ClustalW is a three-step algorithm. First, the program generates pairwise alignments of all pairs of sequences to determine sequence similarity. Second, it defines an order to incorporate sequences to the multiple sequence alignment based on an approximate phylogenetic tree built using the scores from the first step. Finally, the multiple sequence alignment is built progressively based on the order determined in the second step. An alternative to the progressive-alignment approach is the simultaneous alignment of all the sequences, implemented in DCA [43]. Both DCA and ClustalW rely on global alignments, but when proteins share similarity restricted to a domain or motif, it is best to consider methods that rely on local alignments, such as DIALIGN 2. DIALIGN 2 [44] is a segment-based method that builds the multiple alignments by assembling a collection of high-scoring segments through a progressive sequence-independent approach. In general, DIALIGN 2 is slower than the other programs mentioned, but this problem has been partially addressed through the release of a parallelized version [45]. This program often performs very well when there is a clear block of ungapped alignment shared by multiple sequences in different locations. A possible example would be the alignment of several proteins sharing a common domain located at the carboxy terminus in a few sequences, at the amino terminus in some others, and in some cases in the middle of the sequence. POA [46] is a progressive algorithm that compares favorably to ClustalW. POA employs partially ordered graphs instead of generalized profiles to represent aligned sequences. For small alignments (up to 50 sequences) POA is generally as fast, or faster, than ClustalW, and for larger numbers of sequences POA is significantly faster than ClustalW.

T-Coffee [47] is one of the most accurate among current multiple sequence alignment methods and represents a significant improvement over ClustalW. T-Coffee is a progressive method in which the pairwise alignments are preprocessed and used to build a library of information that is subsequently used to guide the progressive alignment. One of the most interesting features of T-Coffee is its ability to integrate information from heterogeneous sources (e.g., alignments produced by other programs or protein structure data) to generate the final alignment. On the other hand, T-Coffee is even slower than ClustalW. Katoh et al. [48] indicated that T-Coffee is unable to align more than 100 sequences of typical length on an average

desktop computer. In contrast, ClustalW can align several hundred sequences under the same conditions, but the practical limit for ClustalW would be approximately 1,000 sequences.

New and innovative multiple sequence alignment methods have recently been developed. MAFFT [48] implements two novel techniques. First, it identifies homologous regions rapidly using the fast Fourier transform (FFT), and second, it uses a simplified scoring system that reduces computation time while increasing the accuracy of the alignments. There are several alignment strategies implemented within MAFFT, both progressive and iterative. Using the BAliBASE benchmark [49], the MAFFT-NS2 progressive method is more accurate than ClustalW and 8 to 10 times faster. Using the same benchmark, the MAFFT-NS-i iterative method is more accurate than DIALIGN 2 and as accurate as T-Coffee but 12 times faster than the first and 8 times faster than the second. For sequence numbers over 60, MAFFT is over 100 times faster than T-Coffee. MUSCLE [50] is a new progressive method that uses kmer counting for fast speed distance estimation and a new profile function to perform the alignment, followed by refinement using tree-dependent restricted partitioning. MUSCLE can achieve accuracies higher than T-Coffee, ClustalW, or even MAFFT at speeds that compare very favorably against these other methods. In fact, MUSCLE is currently the fastest algorithm available: it can align 5,000 sequences, each 350 residues long, in seven minutes in an average desktop computer, a task that would require about a year for ClustalW to complete.

Here we have provided a collection of methods that vary in their speeds, accuracies, and overall performances, but this should not be interpreted as an endorsement of the fastest or most accurate method while the rest are discarded. Each method has its strengths and weaknesses that can make them more or less suitable to approach a given biocomputational problem. Although some methods are faster or more accurate than others on average, there are no general rules that can predict if a program will succeed or fail while trying to perform a complex alignment, and this is especially true for alignments in the Twilight Zone or alignments with multidomain proteins. In those cases, the best possible approach is to use several methods and then evaluate the results globally or integrate additional information (e.g., experimental structural data, biochemical data, or secondary-structure predictions) into the alignment to validate it.

3.2.3.2 Integration of Multiple Sequence Alignments and Structural Data

The accuracy of multiple sequence alignments can be greatly improved by the inclusion of experimental or computational structural data or by the integration of several multiple sequence alignments obtained via different methodological approaches. T-Coffee is remarkably well suited for this task, because it allows the combination of multiple, pairwise, global, or local alignments from different tools into a single model. In addition, it estimates the consistency level of each position in the new integrated alignment with respect to the original alignments. AltAVisT [51] is another possible solution to integrating several multiple sequence alignments. Whereas T-Coffee integrates multiple alignments in a single model, AltAVisT can

compare two multiple sequence alignments, highlighting the local agreement between them and identifying those regions that can be considered to be most reliable. Recently, a novel method based on T-Coffee called 3DCoffee [52,53] has been made available to the scientific community. 3DCoffee allows the integration of conformational information from one or more proteins structures with sequence data to generate an improved multiple sequence alignment. The structure–structure pairs are aligned with the program SAP [54], whereas the sequence-structure pairs are aligned with the threading program Fugue [55]. The resulting collection of pairwise alignments is then combined in a multiple sequence alignment using the T-Coffee algorithm.

3.2.3.3 Analysis of Multiple Sequence Alignment Data

The usefulness of multiple sequence alignments is not restricted to their use as starting points for profile generation in profile-methods (both PSSMs and HMMs) and other computational techniques. Multiple sequence alignments can be analyzed and functional information obtained from them either from the alignment *per se* (e.g., domain organization or local conservation) or by combination with data from other sources (e.g., contextual, 3D structure, secondary-structure prediction, etc.). The analysis of covariations in the amino acids at different alignment positions can provide information about the structural and functional role of those residues. The basic assumption in this analysis is that substitutions in functionally interacting residues are constrained, so when a residue is mutated, the interacting residue will undergo a compensatory mutation to preserve the interaction. The phenomenon would manifest in multiple alignments as correlations between substitutions at pairs of aligned positions. The Web-based CRASP program [56] and the stand-alone PCOAT [57] are two examples of tools capable of detecting and analyzing the occurrence of those coordinated substitutions in multiple sequence alignments and provide valuable structural insights in the absence of an experimental 3D protein structure. The simultaneous exploration of multiple sequence alignments in conjunction with protein structure in a phylogenetic context can be performed via the Evolutionary Trace method. This approach uses phylogenetic information deduced from a multiple sequence alignment to rank the residues according to their conservation and then maps those residues onto a representative 3D structure. Clusters of residues can identify functional sites in catalytic active sites, protein–protein interaction surfaces, and so on [58,59]. Several implementations of the Evolutionary Trace method are available through servers such as TraceSuite II and ConSurf [60], in addition to the stand-alone JEvTrace program [61].

Another important aspect in the analysis of multiple sequence alignments is quality assessment and error correction. The integration of alignments generated using multiple methods or integration with structural information can prevent some of these errors, but even the best programs can reach only about 86% accuracy in alignment tests using the BAliBASE benchmark. Some of these errors can be addressed by postprocessing the alignments with a dedicated program like RASCAL [62].

3.2.3.4 Visualization and Edition of Multiple Sequence Alignments

Multiple sequence alignments contain a very high density of information. The extraction of that information in an understandable, concise, and visually appealing format can be achieved through a variety of methods, which can be divided into two distinct categories. Some methods use the sequence alignment as a scaffold to represent features such as residue conservation or conservation of physicochemical properties, whereas other methods process the information contained in a large multiple sequence alignment and display it in a concise manner (domain/motif organization, entropy plots, sequence logos, or phylogenetic trees).

Multiple sequence alignment viewers and editors provide convenient means to manipulate and visualize the information contained in the alignment. Each position in a multiple sequence alignment represents a conserved functional position, that is, different amino acids located in the same position in the alignment are located in the same position in the protein structure and share a common function. It is possible to distinguish patterns in conservation in multiple sequence alignments using coloring or shading based on strict identity or relative conservation. Alternatively, amino acid grouping and coloring can correspond to various physicochemical properties such as volume, charge, aromaticity, hydrophobicity, flexibility, or tendency to adopt a certain conformation. These tasks can be simplified considerably with the use of one of a number of viewers and alignment editors that are available as Web services or as stand-alone programs for different computational platforms. Although automated methods simplify the task of aligning a large number of sequences, they are by no means perfect. In some cases there might be obvious errors in the alignments and they might require manual curation with the help of an editor program. Some of the available choices of editors and viewers are BioEdit, GeneDoc [63], SeaView [64], Cinema [65], Belvu, DCSE [66], AMAS [67], Pfaat [68], POAVIZ [69], Jalview [70], or WAVis [71]. Multiple sequence alignments can be visualized simultaneously with 3D structure data, and the structures can then guide the curation of the alignments using programs such as STRAP [72], ViTO [73], or ModView [74].

Sequence logos [75] are graphical representations that greatly simplify the display of information contained in multiple sequence alignments, particularly when the number of sequences is substantial. In a logo, each position in the alignment is represented by a stack of letters where the height of each stack is proportional to the residue conservation, and the height of each single-letter amino acid code within the stack indicates the relative frequency of the residue in that position (fig. 3.4). WebLogo [76] is a recent and very comprehensive Web-based implementation of the sequence logo method capable of generating highly customized logos suitable for publication. The conservation information contained in multiple sequence alignments can also be visualized through column graphics showing the variability in each position along the alignment. This type of representation can be generated by programs like Entropy Calculator [77] and WebVar [78].

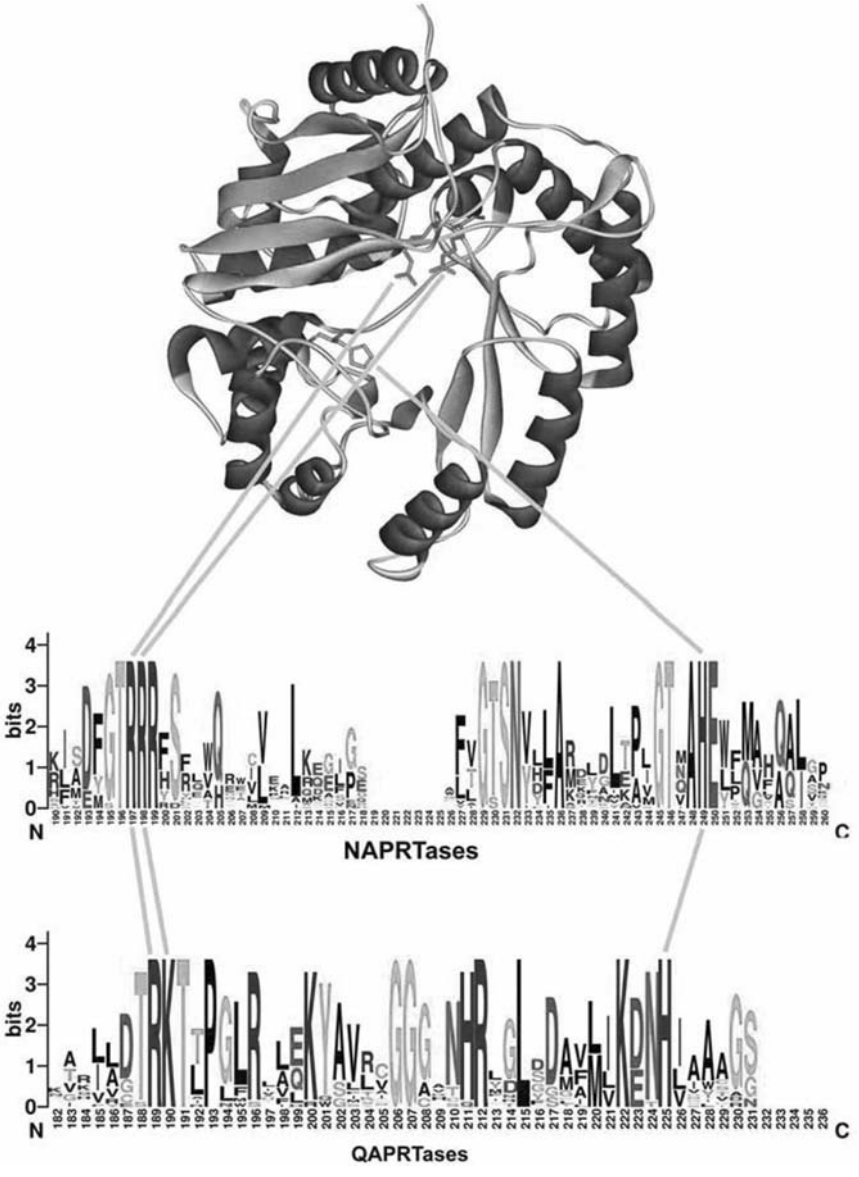


FIGURE 3.4 (See color insert following page 306) Integration of sequence and structure information. Two large multiple sequence alignments including residues catalytically important for two protein families have been condensed into two sequence logos. The logos describe the quinolinic acid phosphoribosyltransferase (QAPRTase) and nicotinic acid phosphoribodyltransferase (NAPRTase) families. Critical residue conservation has been mapped between the logos, and those residues have been subsequently mapped from the logos to the three-dimensional structure of yeast NAPRTase.

3.2.4 Functional Domain Identification

Proteins are modular entities in which each module or domain can have a distinct functional role. There are many definitions of protein domain, although in general a protein domain can be described as a region within a protein that exhibits a well-defined set of characteristics (regarding sequence, structure, or function) and constitutes an independent folding unit. Single-domain organization is prevalent in prokaryotic and archaeal proteins. In contrast, eukaryotic proteins tend to contain several domains, and this modularity can be confusing to many sequence similarity methods because of possible domain rearrangements. Previously, I discussed the process of function assignment by annotation transfer between a pair of homologous proteins. Annotation transfer can also be achieved by matching the query protein to a cluster of proteins sharing a common domain structure (protein family). Family and domain databases identify conserved sequence blocks from multiple sequence alignments and encode the information content into profiles. Searches against family and domain databases are more efficient than searches against sequence databases because of the reduction in redundancy resulting from the inclusion of a number of protein sequences into every single profile.

Because domains can be considered independent structural and functional units, each domain can be analyzed independently once it has been determined that the query protein contains more than one domain. The identification of functional domains can be performed directly by matching the entire query sequence or a portion of it to a profile from a domain database. Alternatively, the existence of functional domains can be evaluated through indirect inference. For instance, if the query protein contains a well-characterized domain that matches a database profile and the rest of the sequence is not covered by any known domain, that uncovered region (provided it has a reasonable length) can be assumed to contain an additional domain. For cases in which there are no matches to domains or protein families in databases, the existence of multiple domains in the protein of interest can still be inferred through other methods. For example, the connectors between domains tend to be disordered or flexible linkers. Accordingly, predictions of disorder or composition bias, linker predictions, or secondary-structure predictions can be used to infer the spatial location of uncharacterized domains.

3.2.4.1 Direct Domain Assignment through Search in Domain/Family Databases

A comprehensive description of the databases and methods for domain, family, and pattern identification is available in chapter 2. Therefore, in this chapter, the discussion of the application of family and domain information to function assignment will be limited to Pfam, the virtual standard in protein domain/family classification, and to InterPro and CDD, two resources that integrate multiple domain databases. In addition, I discuss tools that can be used to scan those databases, namely HMMER,

IprScan, and RPS-BLAST, and how they can be optimized to maximize the detection of distant homologies.

Pfam [79], an abbreviation for Protein Families, is a library of profiles corresponding to protein families and conserved functional domains. Pfam can be accessed through a number of Web servers (see http://pfam.wustl.edu for Pfam access and a list of mirrors) via the Pfam::Alyzer Java interface or installed locally. Pfam is composed of two parts: Pfam-A is a manually curated database of protein domains and families, whereas Pfam-B is a database of domains generated by automatic clustering of the Swiss-Prot sequences not covered by Pfam-A using the program Domainer [80]. For each protein family in Pfam-A, a curated multiple sequence alignment has been built and an HMM has been generated from this seed alignment. Pfam domains and families contain curated annotations describing their function, the variability of function within a certain family, and structure links when available. Therefore, those annotations are a valuable asset in deciding to which degree it is possible to transfer the annotation between matching domain(s) and the query sequence.

Pfam searches are performed using HMMER to scan a query protein for the occurrence of Pfam domains. HMMER can search for either complete domains or fragments. The fragment mode is less sensitive in general, but it is useful to detect distant domain similarities especially when there are insertions. The cutoffs used by HMMER can be of two types: gathering thresholds or E-values. Gathering thresholds are the default, and they are very reliable limits set manually by the curators of the Pfam database to avoid false positives. Accordingly, they are useful for automated annotation. For functional discovery in the Twilight Zone, E-values are preferred. More distant homologies can be detected using this cutoff method but at the cost of requiring manual evaluation of the results due to the higher chance of occurrence of false positives, overlapping domains, and so on. The Pfam Web servers allow users to select their cutoff method of choice and specify threshold E-values. Therefore, the Pfam Web servers can be used to explore very distant homologies and nonstatistically significant matches.

The access to Pfam through most of the Web sites is limited to searching one sequence at a time. Until recently the only option for large searches against Pfam was using a local installation or the batch search service of the Pfam server at the Sanger Institute. A recent option for batch searches is the SledgeHMMER server [81], which as an added bonus utilizes an optimized version of the *hmmsearch* algorithm that is several times faster than the *hmmpfam* program included in the HMMER 2.3.2 package used by the official Pfam mirrors. SledgeHMMER can be downloaded and installed locally.

The search of a sequence against a library of profiles can be also accomplished using the program RPS-BLAST (Reverse PSI-BLAST), part of the NCBI-BLAST package [7]. Whereas PSI-BLAST searches a profile against a database of sequences, RPS-BLAST searches a query sequence against a database of profile family models, hence the name Reverse PSI-BLAST. PSSMs generated by PSI-BLAST can be converted to the models used by RPS-BLAST using the programs *makemat* and *copymat*, also part of the NCBI-BLAST package. The NCBI Conserved Domain

Database (CDD) [82], which consists of a collection of PSSMs derived from the Pfam, SMART [83], and COG [84] databases, uses RPS-BLAST as its search tool.

Pfam and SMART have been integrated with PROSITE [85], ProDom [86], PRINTS [87], UniProt [88], TIGRFAMs [89], PIR-SuperFamily [90], and SUPERFAMILY [91] to form InterPro [92], which has become a *de facto* one-stop shop for domain- and protein family-related analysis. InterPro can be searched with InterProScan [93], a tool that combines the different recognition methods used by each specific database into a single integrated resource (FingerPRINTScan for PRINT; ScanRegExp and ProfileScan for PROSITE; BlastProDom for ProDom; HMMSearch for SuperFamily; and HMMPfam for Pfam, SMART, TIGRFAMs, and PIRSuperFamily). The cutoffs used by each individual search method included in InterProScan are very conservative (e.g., gathering thresholds for HMMPfam), so exploration of very distant homologies is not possible using the InterPro Web server. On the other hand, the InterPro database and InterProScan can be downloaded and installed locally. The cutoffs are defined in configuration files for each database and application used, and they can be edited by the user, transforming InterPro into a powerful tool capable of exploring remote homologies across multiple domain databases.

The functional assignment of a query protein can also be accomplished by using tools specifically designed to assign a protein to a certain functional group (e.g., enzymes). The SVM-ProtEnzyme [94,95] and ArchaeaFun [96] enzyme predictors are good examples of this specific class of function annotation tools.

3.2.4.2 Domain Assignment through Indirect Evidence

The location of functional domains can be also inferred indirectly through a variety of computational approaches. Domain boundaries can be predicted from primary sequence using a number of available tools like DOMAINATION [97], DGS [98], SnapDRAGON [99], CHOPnet [100], DOMpro, DomPred and DomSSEA [101], PASS [102], ScoobyDo [103], Domain predictor @ Biozon.org, and DomCut [104]. Some of these domain-prediction programs use secondary-structure predictions to identify possible interdomain linkers. In general, any protein secondary-structure prediction method (e.g., PSIPRED [105]) can be used as a crude tool to predict the location of interdomain regions, as in many cases those linkers are extended random-coil segments. The algorithms used to predict domain boundaries are extremely diverse, ranging from the amazingly simple DGS, which uses sequence length as its only input and relies on statistical domain distributions, to elaborated methods such as SnapDRAGON, which uses *ab initio* structural modeling.

The likelihood of multiple domains in a protein can also be estimated from the distribution of compositionally biased or predicted disordered regions, which tend to function as interdomain linkers. This type of subsequences can be identified using tools such as SEG [106], CAST [107], DisEMBL [108], PONDR [109], Disopred 2 [110], GlobPlot [111], DISPro, CARD [112], and NORsp [113]. Armadillo, DLP2 [114], or DomCut can also be used to predict the linker regions of multidomain proteins from their primary sequence.

In addition, domain organization can be inferred from the presence of transmembrane helices. Transmembrane helices are compositionally biased sequences,

so in the case of proteins with a single transmembrane helix, the helix functions as a low-complexity region separating domains located in opposite sides of a biological membrane. In the case of proteins with multiple transmembrane helices, such as receptors, channels, and transporters, the pore-forming transmembrane helices constitute a domain, whereas the extramembrane segments might contain other functional domains (e.g., oligomerization domains, gating particles, ligand binding domains, etc.). TMHMM [115] has been considered the best-performing transmembrane prediction program [116] for some time, although the recently released Phobius [117] appears to be even more accurate than TMHMM.

Domain linkers are also likely to be regions with high sequence flexibility. The relative flexibility of each position in a protein sequence can be estimated using the ProtScale tool at Expasy with an average flexibility index amino acid scale. Multiple sequence alignments are an extra tool that can be used to predict the location of interdomain linkers, because those regions are usually less mutationally constrained than functional domains, resulting in regions of the alignments that show poor region conservation and a high level of entropy.

3.2.5 Function Assignments Based on Contextual Information

Context-based methods in function annotation are recent additions to the collection of methods that the computational biologist can use to assign function to uncharacterized proteins. These methods, complementary to homology-based function predictions, generally yield less information than similarity-based approaches. Still, they are appealing because some of them take full advantage of genomic information (e.g., conservation of genomic context, phylogenetic profiles, etc.), becoming true genomic methods. Whereas homology-based function assignment relies on annotation transfer, the basic principle or paradigm underlying contextual methods is *guilt by association* [118]. Under this principle, proteins that colocalize chromosomally through protein–protein interactions, clustered expression profiles, or phylogenetically are considered to be functionally linked. One of the caveats of the use of contextual information is that whereas similarity-based methods usually yield biochemical functions, contextual methods tend to provide broader biological predictions (e.g., metabolic pathway; subcellular location; or interacting proteins, ligands, or cellular components). Studies performed on the *Saccharomyces cerevesiae* genome estimated that up to 30% of the contextual function predictions were false. Conversely, when predictions were based on consensus built from two or three of the contextual methods used, the rate of false positives decreased to 15% [119], suggesting that in some cases these techniques might compete successfully with annotation approaches based on homology.

3.2.5.1 Gene Fusions: The Rosetta Stone Method

The principle underlying the use of gene fusions to predict protein function is that if two proteins A and B are present in one organism, they are likely to interact if their homologs are expressed as a single AB protein in another organism [119,120].

These fused proteins are called Rosetta Stones. The practical application of Rosetta Stones to the assignment of function to hypothetical proteins is as follows: if a query hypothetical protein is homologous to an uncharacterized domain fused to a characterized domain, the function of the characterized domain can be used to infer the function of the unknown domain (fig. 3.5). The identification of a protein as a Rosetta Stone provides functional information and evidence of its essentiality. The comparison of *Drosophila* with other organisms shows in an overwhelming majority of cases the Rosetta components participate in the core metabolism (i.e., intermediary and basic information transfer metabolism) [121]. In some respects, the gene fusion analysis is similar to the use of gene clusters for inferring functional links from genomic context.

Among other resources, gene fusion data can be retrieved from The Search Tool for Recurring Instances of Neighboring Genes (STRING) [122], Allfuse [123], Phydbac [124], Prolinks [125], FusionDB [126], or InterWeaver [127]. Allfuse contains exclusively gene fusion data and does not allow homology searches using a query sequence. On the other hand, Allfuse allows search by query ORF name or by annotation and performs other types of searches. Searches with a query sequence can be performed in STRING, FusionDB, or InterWeaver. The advanced search interface in FusionDB has extended functionalities that include the same types of searches that can be performed by Allfuse. If there is a positive result, FusionDB returns sequence alignments between the query protein and Rosetta Stones, phylogenetic trees for the component moieties of the fused protein, and structural links if available.

STRING is a comprehensive contextual annotation system not restricted to gene fusion data. STRING also includes conserved gene neighborhood analysis, phylogenetic profiles, coexpression, experimental data evidence, information retrieved from other databases, and text mining. Results are displayed in (a) a summarized view, (b) detailed views for each method used to infer function annotation, and (c) a graphical network view showing the predicted functional associations. InterWeaver

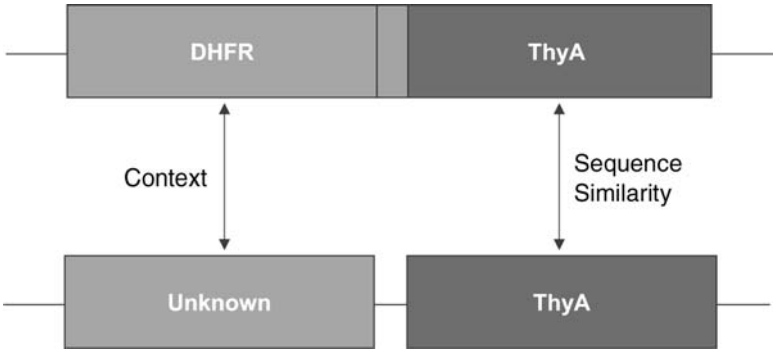


FIGURE 3.5 The use of gene fusions (Rosetta Stones) for contextual annotation of hypothetical proteins. The example shows that if an unknown protein and a thymidylate synthase (ThyA) are present in a number of organisms, it is very likely that the unknown protein is a dihydrofolate reductase (DHFR), if its length, sequence, or other physicochemical characteristics are compatible with those expected in a DHFR.

shares some of the capabilities of STRING, namely the integration of gene fusion, protein–protein interaction, and biomedical literature text mining.

3.2.5.2 Domain Co-occurrence

In many cases, the detection of a domain in a query protein using homology methods can suggest a range of functions too diverse to be of practical use for function assignment. In contrast, the contextual information provided by the presence of additional domains, known as domain architecture, can present additional constraints that can refine a protein's predicted functional role. For example, the presence of a cAMP-binding domain or a DNA-binding domain are not enough to assign a defined functional role to a protein, but the presence of both domains in the same protein point toward a function as a transcription regulator. As previously shown, Pfam, InterPro, or other domain/family assignment tools can be tailored to allow the detection of distant homologies. If several predicted domains are detected in the query sequence but the matches have low confidence levels, the relevance of these results can be supported if the matches are concordant with domain architectures previously observed in other proteins.

Domain information can be used to perform searches using the domain architecture query tools featured by Pfam and SMART. For instance, a search for "cNMP_binding and CRP" would identify all the proteins containing a cyclic-nucleotide binding domain and a catabolite regulator protein (CRP) domain. CDD does not include a domain architecture query capability, but it does contain CDART [128], a similarity search tool and architecture viewer. CDART can display the domain organization of the query protein as well as the domain architectures of other proteins containing those domains. InterPro also lacks the architecture search capabilities of Pfam and SMART, but InterPro entries contain a very useful "Architectures" view. In addition, InterPro features a Domain Architecture (IDA) concise view that facilitates domain composition analysis of proteins and contains links to alternative and more detailed architecture visualizations (e.g., http://www2.ebi.ac.uk/interpro/ISpy?ipr=IPR000595&mode=IDA).

3.2.5.3 Genomic Context: Gene Neighborhoods, Gene Clusters, and Operons

Gene context analysis is an effective tool for function assignment, and its power is continuously increasing with the growth in the number of sequenced genomes. In correlated gene context analysis, a functional link may be inferred between two proteins if their respective genes are found to be neighbors in several genomes. The presence of a gene as part of an operon (i.e., a group of genes arranged as a single transcriptional unit) provides functional information, because proteins originating from the same operon are usually part of the same metabolic pathway or they are involved in coordinated cellular activities in response to a common stimulus. Normally, these groups of genes are contiguous, and they are transcribed in the same direction. Although conservation of operon architecture tends to be poor [129], genes linked through an operon in an organism have been observed in close proximity in

other organisms, indicating the existence of a stabilizing selective pressure that tends to keep genes functionally linked nearby and prevents the disruption of those gene clusters. Multiple lines of evidence suggest that the vast majority of gene clusters that are conserved in bacterial and archaeal genomes are operons [130].

For practical uses, if the gene encoding the potential target is part of a gene cluster conserved across multiple species, it can be assumed that the proteins encoded by the genes in the cluster are functionally linked. If some members of the cluster are functionally annotated, this annotation might be used to propose a role for the target protein. In some cases, a gene cluster might contain only hypothetical proteins, in which case each of those would be subject to the same annotation efforts devoted to the initial target.

Even if a protein of interest is not in a conserved gene cluster in its parent organism, genomic context analysis can still be applied. If homologs in other organisms are part of conserved gene clusters and the members of the cluster have homologous proteins in the source organism of the target protein, a context-based function assignment is still feasible. Gene context-based functional prediction is considerably more difficult in eukaryotes than in prokaryotes because of the apparent lack of clustering of functionally linked genes. Yet there are exceptions to this rule. Several operons have been identified in *C. elegans* [131], and there are even cases in which functional gene context has been preserved in archaea, bacteria, and multiple eukaryotes, including humans [132]. Gene neighborhoods and gene order can be assessed using tools such as ERGO [133], STRING [122], or SNAPper [134]. ERGO is capable of performing two types of genomic context analysis. The first method, based on the computation of "pairs of close bidirectional best hits" [135], predicts operons that have been preserved between pairs of genomes. The discriminating criteria to consider that two genes are part of an operon is that they are not more than 300 base pairs apart and that they are situated on the same DNA strand in each of the analyzed genomes. In the second analysis method, orthologs within a 2 to 20 kb segment area common between genomes, independently from their chromosomal orientation, are mapped through the Pinned Regions tool. This tool highlights the proteins that tend to cluster together within a certain genomic region, although the nature of that clustering is less precise than the functional linkage observed in operons. Thus, the annotations from co-occurrence of clustered genes in pinned regions are less confident than annotations derived from genomic context in conserved operons.

STRING is not limited to genomic context analysis. Instead, it combines several contextual annotation methods, and their outputs can be displayed individually or combined in a concise and visually appealing network representation. STRING allows the exploration of distant functional relationships by two different methods. First, the user can select the level of confidence of the predictions, from low confidence (15%) to highest confidence (90%). Second, the user can select the depth of the interaction network explored by STRING. The default is a single level network. If higher depths are selected (up to 5), after the best neighbors of the query gene have been identified, STRING searches in turn for their respective neighbors and then it continues iterating until the user-selected limit is reached or the search has converged (i.e., when an iteration detects no new neighbors).

SNAPper is similar to STRING, but it does not require that related genes form conserved gene strings. Instead, they only need to be in the vicinity of each other, which is comparable to the pinned region analysis performed by ERGO. Pinned analysis (as well as analysis of gene fusions, metabolic reconstruction, etc.) can also be performed by SEED. Other available online resources to access genome context information are the "Exhaustive Search for Gene Clusters in Two Genomes" option in KEGG [136], the Swiss-Prot Genome Proximity Viewer option for each Swiss-Prot entry (e.g., http://us.expasy.org/cgi-bin/genomeview.pl?bn=THEMA&GN=TM0449#ORF), organism-specific resources (e.g., Flybase, Wormbase, Ensembl-Human, or Ensembl-Mouse), the Gene Neighbors database and tool collection [137], the NCBI Map Viewer, the "Search Genes and Operons" tool in PRODORIC [138], GeConT [139], or Prolinks [125].

3.2.5.4 Phylogenomic Profiles

Phylogenomic profiling relies on the correlated evolution between interacting proteins. The evolution of a protein pair is correlated when the proteins share a common pattern of protein presences and absences over a set of complete genomes. Interacting proteins tend to share similar evolutionary histories because the preservation of interactions and biochemical functions requires the coordination of evolutionary changes. Therefore, proteins that share a similar phylogenomic profile are expected to be functionally linked [140,141] and can be clustered based on the similarity of their respective phylogenomic profiles. If an uncharacterized protein is included in a cluster that contains one or more functionally defined proteins, a functional linkage can be established.

Functional links can be inferred from similar matching phylogenomic profiles and from complementary phylogenomic profiles. Complementary phylogenomic analysis looks for protein pairs in which one of the proteins is present in a genome and absent in the other, and vice versa. These protein pairs often correspond to proteins that perform the same function. The case of the Thymidylate Synthase Complementing Proteins (TSCPs), a family of alternative thymidylate synthases, is remarkable and shows the power of this method. Thymidylate synthase is an essential enzyme; therefore, when its absence was documented in a number of prokaryotic genomes, it was a perplexing anomaly. The functional role of TSCPs was predicted based on (a) a phylogenomic profile that, with few exceptions, complemented that of thymidylate synthases [142] and (b) literature data indicating that a TSCP could promote the growth of a thymidine auxotrophic strain of *Dictyostelium* in the absence of thymidine [143]. Subsequently, it was experimentally confirmed that TSCPs function as thymidylate synthases [144] and are structurally unrelated to typical thymidylate synthases [145,146].

The use of phylogenomic profiles showing the patterns of gene distribution among particular lineages of organisms with completed genomes is still limited due to the high noise levels. Phylogenomic profiles still can be very useful to support weak predictions generated by other methods or in combination with experimental data. The power of this method is increasing rapidly in conjunction with the growing number of sequenced genomes.

Phylogenomic profile analysis can be carried out using tools such as STRING [122], Phydbac [147], ADVICE [148], PLEX, MATRIX [149], or Prolinks [125]. Both STRING and Phydbac can be queried using a single sequence as input, making their use straightforward when these methods are applied to a hypothetical protein. ADVICE and MATRIX require a pair of sequences to compare their phylogenomic profiles, making them more useful to validate functional linkage hypothesis generated from other contextual data sources. PLEX can use as input either a sequence or a phylogenomic profile, whereas Prolinks uses exclusively a protein identifier as input.

3.2.5.5 Metabolic Reconstruction

The process of predicting the entire set of metabolic reactions in an organism is known as metabolic reconstruction. Metabolic reconstructions are performed by deducting the core metabolic functionality for an organism through the integration of primary sequence with biochemical, pathway, physiological, or gene organization data. The reconstruction of the metabolic networks normally results in "gaps" in the pathways that can be filled with hypothetical proteins, forming the basis for function assignment [150]. The assignment of functionally uncharacterized genes to functions in a metabolic network can be accomplished by similarity, contextual methods, or both. Once a protein with unknown function has been placed into a specific metabolic pathway or network, it is possible to infer a functional role. The information that can be gleaned from metabolic networks is not limited to enzymatic activities. In some cases, it is possible to propose a regulatory role, propose a subcellular location, or even predict expression patterns.

Resources such as KEGG and MetaCyc [151] provide basic background information about biochemical pathways. Metabolic pathway analysis and metabolic reconstruction can be performed using PathBLAST [152], SEED, PUMA2, ERGO, metaSHARK, PathFinder, PRIAM [153], or BioSilico [154]. Visualization of pathway information can be performed by a number of commercial products as well as the open source Cytoscape [155].

3.2.5.6 Protein–Protein Interactions

The study of protein–protein interactions is a powerful approach to gain insight into protein function, because proteins rarely function alone in cells. Instead, they tend to interact with other proteins and often are part of complexes or networks that are constitutive elements of signal transduction pathways. Experimental protein–protein interaction data from genome wide yeast two-hybrid screens, coimmunoprecipitation followed by mass spectrometry, GFP labeling plus fluorescence resonance energy transfer, or protein arrays is increasingly available through public databases.

Unfortunately, analyses of multiple independent protein–protein interaction data sets corresponding to the same organism have reported high rates of false-positive interactions. Therefore, protein–protein interaction data should be used with caution and only as corroborating evidence to support other sources of functional evidence. This is especially true when interactions are predicted by extrapolations based on sequence similarity between query proteins and experimentally described interacting pairs.

The majority of protein–protein interaction analysis tools rely on information obtained from repositories such as BIND [156], DIP [157], MINT [158], and GRID [159]. Interaction data from public databases can be accessed directly to retrieve experimental information. Alternatively, protein-interaction data can be accessed via predictive tools such as STRING and InterWeaver. Identifying protein–protein interactions can provide clues beyond a functional role or subcellular location. It is possible to predict a protein's metabolic importance based on the density of edges connecting nodes, because it has been shown that nodes with a high density of connections are more likely to contain essential genes [160].

3.2.5.7 Microarray Expression Profiles

The emergence of public databases containing expression data derived from DNA microarrays allows alternative approaches to explore protein function. Microarray-based functional annotation relies on the clustering of proteins according to their expression behavior under a certain range of experimental conditions (developmental stage, mutations, exposure to different environmental agents, effect of drugs, etc.). Subsequently, function is assigned via guilt by association based on the premise that clustered genes must be involved in similar biological functions [161,162,163]. Therefore, if a search of an uncharacterized protein against a microarray database reveals a match to a specific functional cluster, one can assume that either the function of the query protein is the same as the proteins in the annotated cluster or it is functionally related to them.

Although more prone to error, indirect approaches can be used. For instance, if there is no microarray expression data for the organism or target of interest, it is still possible to apply sequence similarity methods to search for homologous proteins in a microarray database. When sequence similarity does not return any matches in the microarray databases, other hybrid transitive strategies can be applied. For example, a protein functionally linked to the target of interest could be identified through a Rosetta Stone approach, and subsequently that new protein could be used to search microarray data. However, the lack of consistency among microarray datasets negates most of the potential of these methodologies (see [164] for a review about this issue). Currently, microarray data cannot be used reliably as a stand-alone tool for function annotation in the absence of corroborating experimental or computational evidence, yet it is expected that improvements in the replicability of this technology will strengthen and validate its usefulness for function assignment.

Several public databases serve microarray data and include tools for microarray analysis, including the Stanford Microarray Database [165], the RIKEN Expression Array Database [166], and ArrayExpress at the European Bioinformatics Institute [167].

3.2.5.8 Other Sources of Contextual Information for Protein Annotation

Literally hundreds of bioinformatics tools can be applied to obtain additional supporting data for functional inference. Among others, those tools can be used to predict posttranslational modifications (e.g., phosphorylation, myristoylation, sulfation,

sumoylation, or glycosylation), intracellular sorting and subcellular location (e.g., mitochondrial, plastidial, peroxisomal, nuclear, or periplasmic), structural features (e.g., secondary structure, transmembrane alpha-helices, transmembrane beta-barrels, coiled-coils, disordered regions, or amphipatic helices) or primary structure characteristics (e.g., molecular weight, isoelectric point, or global descriptors based on amino acid scales such as average hydrophobicity, polarity, or flexibility).

The CMS Molecular Biology Resource [168] at the San Diego Supercomputer Center contains an excellent compendium of Web-accessible bioinformatics tools. The ExPASy (Expert Protein Analysis System] proteomics server of the Swiss Institute of Bioinformatics [169] and the EBI Toolbox are two other popular tool collections. Another useful resource is Herbert Mayer's Bioinformatics World Web site, which contains a comprehensive collection of bioinformatics tools and software, including evaluations, comments, and tutorials.

The fast pace of the development of technical advances in this field of research requires frequent evaluation and acquisition of new computational tools and databases. To put the current state of bioinformatic tools and database development into perspective, the 2004 update of the Molecular Biology Database Collection, published every year in the journal *Nucleic Acids Research,* contains background information and links for 548 databases [170]. In addition, each year *Nucleic Acids Research* has a special issue dedicated to Web-based tools. The 2004 issue contained articles about 137 Web-accessible tools, which is a small fraction of the tools available considering that in the journal *Bioinformatics* 30 to 40 new bioinformatics tools and databases are reported in each issue. Other literature sources for computational tools applicable to functional annotation are *BMC Bioinformatics, Genome Research, Journal of Bioinformatics and Computational Biology, BioTechniques, Journal of Computational Biology, Protein Science,* and *Genome Biology.*

3.3 FROM SEQUENCE TO STRUCTURE: HOMOLOGY AND *AB INITIO* STRUCTURE MODELS

When no function information can be obtained from primary sequence and no 3D structure is available to attempt structure-based function annotation, a valid alternative strategy is the construction of a structural model followed by the application of structural analysis methods. Depending on the quality of the structural templates, gauged by the degree of identity between query protein and template structure, homology modeling, fold recognition/threading, or *ab initio* structure prediction methods can be applied (fig. 3.2).

Homology modeling techniques generally can build a structural model for a target protein if the sequence identity to a high-resolution structural template is greater than 30%. In this case, the key to success and the limiting factor in the accuracy of the model is the correct structural alignment of the query sequence to the template. The process of building a homology model is conceptually simple. First, an alignment is performed between the sequence of the target protein and that of the structural template (a protein structure that has been determined by experimental methods). Next, the residues in the template structure are replaced with the

residues from the target protein. Finally, the conformation of the side chains in the model needs to be optimized. In some cases, complete chains must be built in the case of insertions (loops).

Some common choices of software for homology modeling are Modeller [171], WhatIf [172], or Jackal. Modeller is available as a stand-alone program, as a Web-accessible tool (ModWeb), or integrated in the commercial product Accelrys Discovery Studio Modeling. Homology modeling can also be performed through a number of automated servers, including Swiss-Model [173], LOOPP [174], 3D-Jigsaw [175], HOMER, or CPHmodels [176].

If identity levels between the target protein and the sequences of structures in PDB fall into the Twilight Zone, suitable templates can still be identified and aligned to our target by using threading or fold recognition algorithms. These approaches are equivalent to the distant detection profile–profile methods for the search on distant homologies previously discussed in section 3.2. In many cases the same tools used for distant homology searches are used for fold recognition simply by switching from searching a sequence database like Pfam to a structural database such as CATH, SCOP, or PDB. When these methods are used, the target protein fold is predicted by "threading" the target sequence through a library of 3D folds to try to find a match. This is accomplished by using a scoring function that assesses the fit of the target sequence to a certain fold. Multiple threading methods exist; some of those tools can be accessed through individual servers (e.g., FFAS03, SAMT-02, or 3D-PSSM) or multiple threading methods can be accessed simultaneously through metaservers such as Bioinfo.PL [177] or Genesilico [178]. *Ab initio* protein structure prediction methods use physical principles and do not rely on homology modeling, threading, or secondary structure predictions. The methods are appropriate when no significant structural templates can be detected, as the only required input is the primary sequence of the protein of interest. Essentially, *ab initio* prediction algorithms take the polypeptide chain and calculate a folded 3D structure with minimal potential energy. Several different methods for *ab initio* protein structure prediction exist, and they have improved dramatically in recent years. The best results have been obtained with Rosetta [179]. Rosetta is fully capable of modeling protein 3D structures in the absence of detectable sequence similarity to a previously determined structure, using exclusively the primary sequence of a target protein as initial input [180]. The predictions from Rosetta are limited by the size of the protein, up to approximately 150 residues. Despite this limitation, *ab initio* structure prediction can be a valuable method, because a significant fraction of hypothetical proteins are within the acceptable sequence length range.

Once protein structures have been predicted, they can be compared against the PDB to detect structural similarities and infer possible functions. Subsequently, these predictions might be integrated with contextual sources of functional associations to improve the reliability of the predicted function assignments [181,182]. The Rosetta program is Web accessible through the Robetta [183] and HMMSTR/Rosetta servers [184], or it can be installed locally.

3.4 STRUCTURE-BASED FUNCTIONAL ANNOTATION

Structure-based functional annotation often succeeds when sequence-based methods fail, because in many cases evolution retains the folding pattern long after sequence homology becomes undetectable. Accordingly, structural comparisons can reveal functional similarities that would be impossible to detect from sequence alone. The starting point of a structure-based function assignment project can be a 3D structure built using comparative modeling, a model derived from fold recognition and threading, an *ab initio* model, or the protein structure of a protein lacking function assignment.

This last case was indeed very rare until recently. The advent of structural genomics projects has resulted in an unprecedented number of protein structures deposited in the PDB identified as *hypothetical proteins*. The number of structures labeled as hypothetical protein or "unknown function" ranges from 30 to 60% of the total number of depositions depending on the structural genomics center, and this number is increasing exponentially. Currently, the number of structures of hypothetical proteins deposited in the PDB doubles every six months.

The structures of these hypothetical proteins have been deposited in the PDB without any function assignment due to the lack of sequence similarity to functionally characterized proteins at the moment when they were initially annotated, thus making them targets for annotation. This task is facilitated by the explosive growth in both sequence and structure data, which provides a continuous flow of information that can be used for function assignment. As in the case of sequence-based annotation, the assignment of function when 3D similarities have been detected between two structures is derived from annotation transfer.

The large number of structures from genomics centers might become an ideal initial target pool from which to identify target proteins suitable for drug discovery. The computational effort required to assign function to structures or protein sequences is comparable. As a bonus, once the potential pharmacological value of the target structure has been established, the structure can be used immediately for virtual ligand screening and other bioinformatics or cheminformatics methods. The basic procedures involved in annotation of protein structures as well as common tools used for the task are outlined next.

3.4.1 Structural Database Searches

Typically, the first step in structural annotation is to search the query structure against other structures deposited in the PDB using tools such as DALI [185], VAST, SSM [186], CE [187], DEJAVU [188], or MATRAS [189]. If the structure contains multiple structural domains, it is advisable to split the structure into separate coordinate files and submit each domain separately for structural comparison. Structural domains can be identified using tools that rely on geometric criteria and protein dynamics, such as Protein Domain Parser [190], DomainParser [191], or DomainFinder [192]. Alternatively, domain composition can be evaluated by comparison to

databases of structural domains such as CATH [5] or SCOP [4]. The GRATH server [193] can search a query structure against CATH, and SSM or MATRAS can be used to query a structure against SCOP. The geometry-based partitioning methods are especially useful to analyze the domain organization of novel protein structures that contain new folds or fold variants not present in CATH or SCOP.

3.4.2 Structural Alignments

When possible structural matches have been identified through searches, the next step is to confirm their significance by structural alignment. Superimposed structures allow comparing functionally relevant features, conserved residues required for catalysis, residues critical for ligand binding or protein–protein interactions, and so forth. Three different approaches to structural alignment exist: rigid, flexible, and nontopological alignments. In the first and most common case, when similarities reported by database searches are high, both structures can be simply superimposed as rigid entities. Rigid structural alignments can be generated with CE, MATRAS, SSAP [194], SuperPose [195], C-alpha Match [196], or ProFit. In some cases, the database searches return hits with low confidence levels, which can be due in part to structural rearrangements. Consequently, partitioning the structure into domains and realigning them separately with the structure hit using a rigid approach can reveal a better structural match. Alternatively, the structure superimposition can be improved by using flexible alignment. For instance, when trying to compare two kinases, one in a closed conformation and the other in an open conformation, rigid alignment tools will align a domain very well and misalign the other. Aligning each domain separately would result in a better alignment for each domain, and a flexible alignment would align well both domains by introducing a hinge between the large and small domains of one of the aligned kinase structures. Flexible structural alignments can be built using FATCAT [197] or FlexProt [198]. Finally, it is possible that two proteins share a similar fold, but the connections between secondary structure elements are different. In this scenario the use of nontopological alignment methods, such as SARF2 [199] or MASS [200], can reveal interesting and unexpected structural similarities. As with protein sequences, the structural alignments can be pairwise or include multiple entities. Structural multiple sequence alignments can be produced by programs such as CE-ME [201], MASS, ProFit, or MATRAS.

3.4.3 Use of Structural Descriptors

When structural alignments do not reveal structural similarities that allow annotation transfer, other approaches can be used to obtain information about the function of the target protein. The analysis of the conservation of 3D patterns of functionally relevant residues and evolutionary trace analysis (described in section 2.3.3) are examples of these methodologies. Structural patterns consist of coordinate files in PDB format containing the spatial positions of functionally important residues without considering their positions on the primary or secondary structure. In fact, these patterns can correspond to functional sites present in proteins with completely different folds. The program PINTS (Patterns In Non-homologous Tertiary Structures)

[202] can compare a protein structure against a database of patterns or a structural pattern extracted from the query structure against a database of protein structures. This last type of search can also be performed using the program SPASM [203]. PASS [204], a fast cavity-detection program for the identification and visualization of possible protein-binding sites, is one of the tools that can be used to extract structural motifs that can be used subsequently by PINTS or SPASM.

Surface features are not strictly dependent on sequence or fold conservation. Experimentally determined surface features from a certain structure can be used to search and detect matching sites on the surfaces of unrelated proteins without function assignment. This can be accomplished using methods such as SiteEngine [205]. Other tools that can be used for the analysis of protein surfaces are Surface, GRASS [206], and SURFNET [207].

3.5 FINAL REMARKS AND FUTURE DIRECTIONS

With thousands of genomes completed or in progress (table 3.1) and almost 39 million entries in GenBank (Release 144, October 2004), functional annotation and function assignment are increasingly urgent and demanding tasks. Among the new protein sequences, proteins that lack function annotation (described as hypothetical, uncharacterized, or unknown) are particularly challenging. Currently, the Entrez protein database contains over 1 million hypothetical proteins.

Functions have been experimentally determined for a small number of all the proteins for which sequences are known. These sequences play a crucial role, because they are the basis for all computational function assignments. Therefore, increases in high-quality computational functional annotation are inevitably dependent on increases in the volume of empirical data gathered using experimental methods, fundamentally from biochemistry, biophysics, and cell biology. The sheer size of the task of annotating all the hypothetical protein sequences will possibly require a focused effort reminiscent of the structural genomics initiatives as well as constant feedback between experimental and annotation efforts. Hypothetical proteins targeted

TABLE 3.1
Current Status of Genomic Projects

Genome Source	Completed	Ongoing
Eukaryal	31	451
Bacterial	178	496
Archaeal	20	27
Viruses	1680	n.a.
Organelles	679	n.a.
Phages	237	n.a.
Viroids	36	n.a.

Source: From Genomes On-line Database (http://www.genomesonline.org/) and Entrez Genome (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Genome).

for experimental characterization will need to be selected according to prioritization schemas that will maximize the functional space that can be covered with limited experimental resources.

Structural genomics projects were expected to provide clear evidence regarding the functions of hypothetical proteins and allow their functional annotation with minimal or no experimental poststructural analysis. It has been observed that the functions of hypothetical proteins can seldom be inferred exclusively from their structures. In fact, in a majority of cases structures provide only corroborating evidence for inconclusive annotations derived from sequence similarity searches. These observations argue for the need for increased efforts to characterize hypothetical proteins biochemically.

The shortage of proteins with experimentally determined function prompts the need for methods capable of performing similarity searches deep into the Twilight Zone, which is one of the reasons for the subsequent increase in dubious or incorrect function assignments, particularly in automatically annotated databases. When new sequences are annotated based on homology to those incorrectly annotated entries, the errors propagate, leading to database contamination. Database contamination can be avoided or at least minimized by comparing the sequences being annotated to the subset of experimentally characterized proteins (i.e., using primary instead of derived data) or using conservative cutoffs (e.g., gathering thresholds in Pfam). The author hopes that the spread of database contamination will be contained by a combination of improved computational techniques and increased availability of reliable experimental data. Until that goal is achieved, for mission critical uses, tainted annotations due to database contamination can be avoided through the implementation of in-house functional annotation projects. In-house functional annotation is also favored because many public annotations are obsolete; they were generated when an organism's genome was sequenced and deposited and never updated. Even if a function cannot be assigned to a protein today, the exponential grow of experimental and computational data, and the development of new bioinformatics tools, can reverse that situation rapidly. Accordingly, the computational assignment of protein function is an open-ended task.

Even in cases when there is a significant match between the sequence or structure of interest and a similar protein or structure, it still might be unclear how much information can be transferred from the match to the query protein. One possibility is that sequences with significant sequence similarities can have different functions, or even proteins with similar structures can have substantially different functions. Multiple studies have noted problems when function assignments are performed by the incorrect application of similarity-based annotation transfer [208–210]. Furthermore, the databases do not usually explain the origin of the annotations, the confidence of the assignment, or even if the annotations are experimental or predicted. Despite these problems, the majority of functional annotations in public databases are the result of homology transfer.

The proliferation of online bioinformatics tools facilitates the computation and retrieval of data, which sometimes can be incorrect. The default parameters used to execute some programs either locally or through Web servers can be inappropriate,

numerical calculations can be mathematically correct but biologically meaningless, databases can be contaminated, and so on. The only way to limit these problems is to possess a good understanding of the algorithms underlying the tools used and knowledge of the capabilities and limitations of those tools.

When using online tools, it is important to be able to verify how recent the databases used for the searches are to avoid using obsolete tools. Commercial products for target annotation and high-throughput automated annotation systems tend to integrate well-known algorithms, which in many cases are far from being state of the art. In most cases the data flows and parameters used by the integrated tools are hidden from the final user, limiting the usability of these black boxes for nonroutine annotation projects. Bioinformatics workflows such as Incogen's VIBE, Taverna, Pegasys, or SciTegic's Pipeline Pilot provide viable alternatives with the best of both worlds: A visually appealing graphical user interface, pipeline automation, and control of the parameters fed to the tools integrated in the workflow (fig. 3.6).

Highly sophisticated computational methods are invaluable assets in the search for the biological function of uncharacterized proteins. For many proteins computational sequence analysis can suggest at best a general biochemical function. Still, this information can be used to design experimentally testable hypotheses aimed at identifying

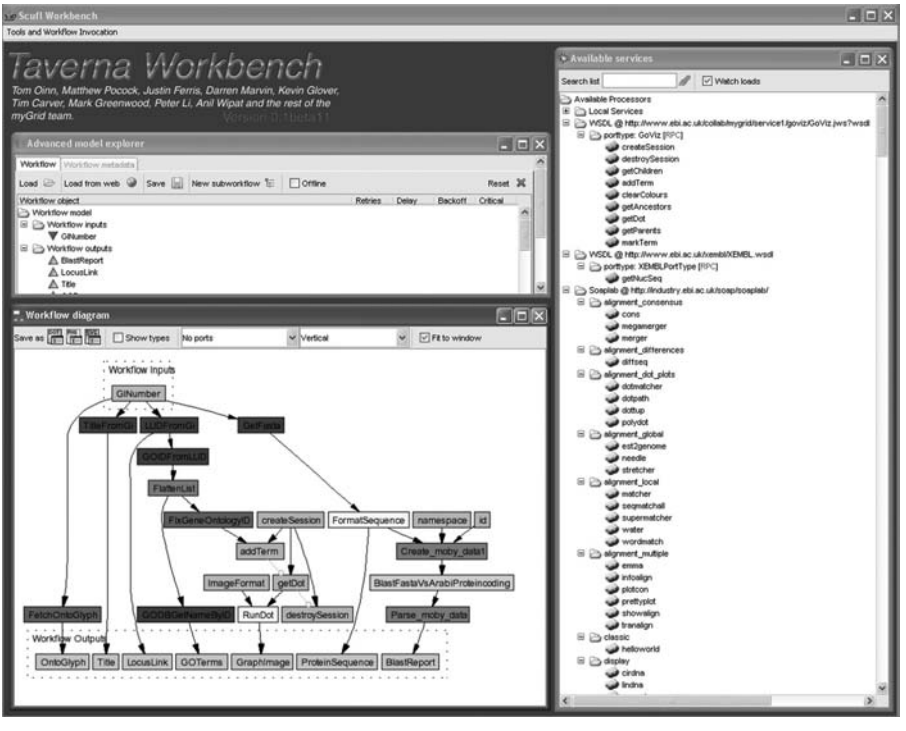


FIGURE 3.6 (See color insert) Graphical user interface of Taverna Workbench, a tool for the composition and execution of bioinformatics workflows. The workflows are written in a new language called Scufl (Simple conceptual unified flow language).

the exact function of a given gene. As the list of the hypothetical proteins continues growing exponentially, interdisciplinary studies combining experimental and computational approaches will help to identify the functional roles of these fascinating proteins. In turn, this identification will provide valuable new insights into protein function and will facilitate the identification of novel pharmacological targets.

ACKNOWLEDGMENTS

I thank the members of the Bioinformatics Core of the Joint Center for Structural Genomics for their intellectual and material support and helpful discussions. I also thank to Anna Canaves for her assistance in manuscript preparation, critical reading, and contributions to graphics design.

REFERENCES

1. Gardner, M. J., N. Hall, E. Fung, O. White, M. Berriman, R. W. Hyman, J. M. Carlton, et al. 2002. Genome sequence of the human malaria parasite Plasmodium falciparum. *Nature* 419:498–511.
2. Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. 2000. The Protein Data Bank. *Nucleic Acids Res* 28:235–42.
3. Hadfield, J. A., S. Ducki, N. Hirst, and A. T. McGown. 2003. Tubulin and microtubules as targets for anticancer drugs. *Prog Cell Cycle Res* 5:309–25.
4. Andreeva, A., D. Howorth, S. E. Brenner, T. J. P. Hubbard, C. Chothia, and A. G. Murzin. 2004. SCOP database in 2004: Refinements integrate structure and sequence family data. *Nucleic Acids Res* 32:D226–9.
5. Orengo, C. A., F. M. Pearl, and J. M. Thornton. 2003. The CATH domain structure database. *Methods Biochem Anal* 44:249–71.
6. Reeck, G. R., C. de Haen, D. C. Teller, R. F. Doolittle, W. M. Fitch, R. E. Dickerson, P. Chambon, A. D. McLachlan, E. Margoliash, and T. H. Jukes. 1997. "Homology" in proteins and nucleic acids: a terminology muddle and a way out of it. *Cell* 50:667.
7. Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. 1990. Basic local alignment search tool. *J Mol Biol* 215:403–10.
8. Lipman, D., and W. Pearson. 1985. Rapid and sensitive protein similarity searches. *Science* 227:1435–41.
9. Pruitt, K. D., T. Tatusova, and D. R. Maglott. 2003. NCBI Reference Sequence project: Update and current status. *Nucleic Acids Res* 31:34–7.
10. Boeckmann, B., A. Bairoch, R. Apweiler, M.-C. Blatter, A. Streicher, E. Gasteiger, M. J. Martin, et al. 2003. The SWISS-PROT protein knowledgebase and its supplements TrEMBL in 2003. *Nucleic Acids Res* 21:365–70.
11. Wu, C. H., A. Nikolskaya, H. Huang, L. L. Yeh, D. A. Natale, C. R. Vinayaka, Z. Hu, et al. 2004. PIRSF: Family classification system at the Protein Information Resource. *Nucleic Acids Res* 32:D112–4.
12. Doolittle, R. F. 1986. *Of URFs and ORFs: A primer on how to analyze derived amino acid sequences*. Mill Valley, CA: Univ. Science Books.
13. Rost, B. 1999. Twilight zone of protein sequence alignments. *Protein Eng* 12:85–94.
14. Todd, A. E., C. A. Orengo, and J. M. Thornton. 2001. Evolution of function in protein superfamilies, from a structural perspective. *J Mol Biol* 307:1113–43.

15. Henikoff, S., and J. G. Henikoff. 1962. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci USA* 89:10915–9.
16. Schwartz, R. M., and M. O. Dayhoff. 1978. Matrices for detecting distant relationships. In *Atlas of protein sequence and structure*. Vol. 5, Suppl. 3:353–8. Washington, DC: National Biomedical Research Foundation.
17. Lindalhl, E., and A. Elofsson. 2000. Identification of related proteins on family, superfamily and fold level. *J Mol Biol* 295:613–25.
18 Wallner, B., H. Fang, T. Ohlson, J. Frey-Skott, and A. Elofsson. 2004. Using evolutionary information for the query and target improves fold recognition. *Proteins* 54:342–50.
19. Olhson, T., B. Wallner, and A. Elofsson. 2004. Profile-profile methods provide improved fold-recognition: A study of different profile-profile alignment methods. *Proteins* 57:188–97.
20. Altschul, S. F., and E. V. Koonin. 1998. Iterated profile searches with PSI-BLAST—A tool for discovery in protein databases. *Trends Biochem Sci* 23:444–7.
21. Eddy, S. R. 1998. Profile hidden Markov models. *Bioinformatics* 14:755–63.
22. Karplus, K., C. Barrett, and R. Hughey. 1998. Hidden Markov Models for detecting remote protein homologies. *Bioinformatics* 14:846–56.
23. Park, J., K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard, and C. Chothia. 1998. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J Mol Biol* 284:1201–10.
24. Madera, M., and J. Gough. 2002. A comparison of profile hidden Markov model procedures for remote homology detection. *Nucleic Acids Res* 30:4312–28.
25. Sauder, J. M., W. Arthur, and R. L. Dunbrack. 2000. Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins* 40:6–22.
26. Panchenko, A. R. 2003. Finding weak similarities between proteins by sequence profile comparison. *Nucleic Acids Res* 31:683–9.
27. Rychlewski, L. , L. Jaroszewski, W. Li, and A. Godzik. 2000. Comparison of sequence profiles: Strategies for structural predictions using sequence information. *Protein Sci* 9:232–41.
28. Jaroszewski, L., L. Rychlewski, and A. Godzik. 2000. Improving the quality of twilight-zone alignments. *Protein Sci* 9:1487–96.
29. Ginalski, K., J. Pas, L. S. Wyrwicz, M. von Grotthuss, J. M. Bujnicki, and L. Rychlewski. ORFeus: Detection of distant homology using sequence profiles and predicted secondary structure. *Nucleic Acids Res* 31:3804–7.
30. Jones, D. T. 1999. Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol* 292:195–202.
31. Sadreyev, R., and N. Grishin. 2003. COMPASS: A tool for comparison of multiple protein alignments with assessment of statistical significance. *J Mol Biol* 326:317–26.
32. Sadreyed, R. L., D. Baker, and N. V. Grishin. 2003. Profile-profile comparison by COMPASS predicts intricate homologies between protein families. *Protein Sci* 12:2262–72.
33. Edgar, R. C., and K. Sjolander. 2004. COACH: Profile-profile alignment of protein families using hidden Markov models. *Bioinformatics* 20:1309–18.
34. Soeding, J., and A. N. Lupas. 2004. Homology search by HMM-HMM comparison detects more than three times as many remote homologs as PSIBLAST or HMMER. Abstract No. 41. 3DSIG Structural Bioinformatics Meeting at ISMB.
35. Sadreyev, R. I., and N. V. Grishin. 2004. Quality of alignment comparison by COMPASS improves with inclusion of diverse confident homologs. *Bioinformatics* 20:818–28.

36. Li, W., L. Jaroszewski, and A. Godzik. 2002. Tolerating some redundancy significantly speeds up clustering of large protein databases. *Bioinformatics* 18:77–82.
37. Park, J., and S. A. Teichmann. 1998. DIVCLUS: An automatic method in the GEANFAMMER package that finds homologous domains in single- and multi-domain proteins. *Bioinformatics* 14:144–50.
38. Enright, A. J., and C. A. Ouzounis. 2000. GeneRAGE: A robust algorithm for sequence clustering and domain detection. *Bioinformatics* 16:451–7.
39. Enright, A. J., S. Van Dongen, and C. A. Ouzounis. 2002. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res* 30:157–84.
40. Feng, D. F., and R. F. Doolittle. 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J Mol Evol* 25:351–60.
41. Taylor, W. R. 1988. A flexible method to align large numbers of biological sequences. *J Mol Evol* 28:161–9.
42. Chenna, R., H. Sugawara, T. Koike, R. Lopez, T. J. Gibson, D. G. Higgins, and J. D. Thompson. 2003. Multiple sequence alignment with the Clustal series of programs. *Nucleic Acids Res* 31:3497–500.
43. Stoye, J., V. Moulton, and A. W. Dress. 1997. DCA: An efficient implementation of the divide-and-conquer approach to simultaneous multiple sequence alignment. *Comput Appl Biosci* 13:625–6.
44. Morgenstern, B. 1999. Dialign2: Improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics* 15:211–8.
45. Schmollinger, M., K. Nieselt, M. Kaufmann, and B. Morgenstein. 2004. DIALIGN P: Fast pair-wise and multiple sequence alignment using parallel processors. *BMC Bioinformatics* 5:128.
46. Lee. C., C. Grasso, and M. Sharlow. 2002. Multiple sequence alignment using partial order graphs. *Bioinformatics* 18:452–64.
47. Notredame, C., D. G. Higgins, J. Heringa, and J. T-Coffee. 2000. A novel method for fast and accurate multiple sequence alignment. *J Mol Biol* 302:205–17.
48. Katoh, K., K. Misawa, K. Kuma, and T. Miyata. 2002. MAFFt: A novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res* 30:3059–66.
49. Thompson, J. D., F. Plewniak, and O. Poch. 1999. BAliBASE: A benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics* 15:87–8.
50. Edgar, R. C. 2004. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res* 32:1792–7.
51. Morgenstern, B., S. Goel, A. Sczyrba, and A. Dress. 2003. AltAVisT: Comparing alternative multiple sequence alignments. *Bioinformatics* 19:425–6.
52. O'Sullivan, O., K. Suhre, C. Abergel, D. G. Higgins, and C. Notredame. 2004. 3DCoffee: Combining protein sequences and structures within multiple sequence alignments. *J Mol Biol* 340:385–95.
53. Poirot, O., K. Suhre, C. Aberge, E. O'Toole, and C. Notredame. 2004. 3DCoffee@igs: A web server for combining sequences and structures into a multiple sequence alignment. *Nucleic Acids Res* 32:W37–40.
54. Taylor, W. R., and C. A. Orengo. 1989. Protein structure alignment. *J Mol Biol* 208:1–22.
55. Shi, J., T. L. Blundell, and K. Mizuguchi. 2001. FUGUE: Sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J Mol Biol* 310:243–57.

56. Afonnikov, D. A., and N. A. Kolchanov. 2004. CRASP: A program for analysis of coordinated substitutions in multiple alignments of protein sequences. *Nucleic Acids Res* 32:W64–8.
57. Qi, Y., and N. V. Grishin. 2004. PCOAT: Positional correlation analysis using multiple methods. *Bioinformatics*. Dec 12;20(18):3697–9. Epub 2004 Jul 22.
58. Madabushi, S., H. Yao, M. Marsh, D. M. Kristensen, A. Philippi, M. E. Sowa, and O. Lichtarge. 2002. Structural clusters of evolutionary trace residues are statistically significant and common in proteins. *J Mol Biol* 316:139–54.
59. Lichtarge, O., H. R. Bourne, and F. E. Cohen. 1996. An evolutionary trace method defines binding surfaces common to protein families. *J Mol Biol* 257:342–58.
60. Glaser, F., T. Pupko, I. Paz, R. E. Bell, D. Bechor-Shental, E. Martz, and N. Ben-Tal. 2003. ConSurf: Identification of functional regions in proteins by surface-mapping of phylogenetic information. *Bioinformatics* 16:163–4.
61. Joachimiak, M., and F. E. Cohen. 2002. JEvTrace: Refinement and variations of the evolutionary trace in JAVA. *Genome Biol* 3:research0077.
62. Thompson, J. D., J. C. Thierry, and O. Poch. 2003. RASCAL: Rapid scanning and correction of multiple sequence alignments. *Bioinformatics* 19:1155–61.
63. Nicholas, K. B., H. B. Nicholas, and D. W. Deerfield. 1997. GeneDoc: Analysis and visualization of genetic variation. *EMBNet News* 4:14.
64. Galtier, N., M. Gouy, and C. Gautier. 1996. SeaView and Phylo_win, two graphic tools for sequence alignment and molecular phylogeny. *Comput Appl Biosci* 12:543–8.
65. Sinnott, J., S. Pettifer, and T. Attwood. 2004. An introduction to the CINEMA5 sequence alignment editor. *EMBNet News* 10 (3):3–9.
66. De Rijk, P., and R. Wachter. 1993. DCSE: An interactive tool for sequence alignment and secondary structure search. *Comput Appl Biosci* 9:735–40.
67. Livingstone, C. D., and G. J. Barton. 1993. Protein sequence alignments: A strategy for the hierarchical analysis of residue conservation. *Comput Appl Biosci* 9:745–56.
68. Johnson, J. M., K. Mason, C. Moallemi, H. Xi, S. Somaroo, and E. S. Huang. Protein family annotation in a multiple alignment viewer. *Bioinformatics* 19:544–5.
69. Grasso, C., M. Quist, K. Ke, and C. Lee. 2003. POAVIZ: A partial order multiple sequence alignment visualizer. *Bioinformatics* 19:1446–8.
70. Clamp, M., J. Cuff, S. M. Searle, and G. J. Barton. 2004. The Jalview Java alignment editor. *Bioinformatics* 20:426–7.
71. Zika, R., J. Paces, A. Pavlicek, and V. Paces. 2004. WAViS server for handling, visualization and presentation of multiple alignments of nucleotide or amino acids sequences. *Nucleic Acids Res* 32:W48–9.
72. Gille, C., S. Lorenzen, E. Michalsky, and C. Frömmel. 2003. KISS for STRAP: User extensions for a protein alignment editor. *Bioinformatics* 12:2489–90.
73. Catherinot, V., and G. Labesse. 2004. ViTO, a tool for refinement of protein sequence-structure alignments. *Bioinformatics*. Dec 12;20(18):3694–6. Epub 2004 Jul 22.
74. Ilyin, V. A., U. Pieper, A. C. Stuart, M. A. Marti-Renom, L. McMahan, and A. Sali. 2003. ModView, visualization of multiple protein sequences and structures. *Bioinformatics* 19:165–6.
75. Schneider, T. D., and R. M. Stephens. 1990. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Res* 18:6097–100.
76. Crooks, G. E., G. Hon, J. M. Chandonia, and S. E. Brenner. 2004. WebLogo: A sequence logo generator. *Genome Res* 14:1188–90.
77. Ramazzotti, M., D. Degl'Innocenti, G. Manao, and G. Ramponi. 2004. Entropy calculator: Getting the best from your multiple sequence alignments. *Ital J Biochem* 53:16–22.

78. Mignone, F., D. S. Horner, and G. Pesole. 2004. WebVar: A resource for the rapid estimation of relative site variability from multiple sequence alignments. *Bioinformatics* 20:1331–3.
79. Bateman, A., L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, et al. 2004. The Pfam Protein Families Database. *Nucleic Acids Res* 32:D13841.
80. Sonnhammer, E. L. L., and D. Kahn. 1994. Modular arrangement of proteins as inferred from analysis of homology. *Protein Sci* 3:482–92.
81. Chukkapalli, G., C. Guda, and S. Subramaniam. 2004. SledgeHMMER: A web server for batch searching the Pfam database. *Nucleic Acids Res* 32:W542–4.
82. Marchler-Bauer, A., A. R. Panchenko, B. A. Shoemaker, P. A. Thiessen, L. Y. Geer, and S. H. Bryant. 2002. CDD: A database of conserved domain alignments with links to domain three-dimensional structure. *Nucleic Acids Res* 30:2813.
83. Letunic, I., R. R. Copley, S. Schmidt, F. D. Ciccarelli, T. Doerks, J. Schultz, C. P. Ponting, and P. Bork. 2004. SMART 40: Towards genomic data integration. *Nucleic Acids Res* 32:142–4.
84. Tatusov, R. I., N. D. Fedorova, J. D. Jackson, A. R. Jacobs, B. Kiryutin, E. V. Koonin, D. M. Krylov, et al. 2003. The COG database: An updated version includes eukaryotes. *BMC Bioinformatics* 4:41.
85. Hulo, N., C. J. Sigrist, V. Le Saux, P. S. Langendijk-Genevaux, L. Bordoli, A. Gattiker, E. De Castro, P. Bucher, and A. Bairoch. 2004. Recent improvements to the PROSITE database. *Nucleic Acids Res* 32:134–7.
86. Servant, F., C. Bru, S. Carrere, E. Courcelle, J. Gouzy, D. Peyruc, and D. Kahn. ProDom: Automated clustering of homologous domains. *Brief Bioinformatics* 3:246–51.
87. Attwood, T. K., P. Bradley, D. R. Flower, A. Gaulton, N. Maudling, A. L. Mitchell, G. Moulton, et al. 2003. PRINTS and its automatic supplement, prePRINTS. *Nucleic Acids Res* 31:400–2.
88. Apweiler, R., A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, et al. 2004. UniProt: The Universal Protein knowledgebase. *Nucleic Acids Res* 32:D115–9.
89. Haft, D. H., J. D. Selengut, and O. White. 2003. The TIGRFAMs database of protein families. *Nucleic Acids Res* 31:371–3.
90. Wu, C. H., A. Nikolskaya, H. Huang, L. S. Yeh, D. A. Natale, C. R. Vinayaka, Z. Z. Hu, et al. 2004. PIRSF: Family classification system at the Protein Information Resource. *Nucleic Acids Res* 32:D112–4.
91. Madera, M., C. Vogel, S. K. Kummerfeld, C. Chothia, and J. Gough. 2004. The SUPERFAMILY database in 2004: Additions and improvements. *Nucleic Acids Res* 32:D235–9.
92. Mulder, N. J., R. Apweiler, T. K. Attwood, A. Bairoch, D. Barrell, A. Bateman, D. Binns, et al. 2003. The InterPro Database, 2003 brings increased coverage and new features. *Nucleic Acids Res* 31:315–8.
93. Zdobnov, E. M., and R. Apweiler. 2001. InterProScan—An integration platform for the signature-recognition methods in InterPro. *Bioinformatics* 17:847–8.
94. Cai, C. Z., L. Y. Han, Z. L. Ji, X. Chen, and Y. Z. Chen. 2003. SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Res* 31:3692–7.
95. Cai, C. Z., L. Y. Han, Z. L. Ji, and Y. Z. Chen. 2004. Enzyme family classification by support vector machines. *Proteins* 55:66–76.
96. Jensen, L. J., M. Skovgaard, and S. Brunak. 2002. Prediction of novel archaeal enzymes from sequence-derived features. *Protein Sci* 11:2894–8.

97. Goerge, R. A., and J. Heringa. 2002. Protein domain identification and improved sequence similarity searching using PSI-BLAST. *Proteins* 48:672–81.
98. Wheelan, S. J., A. Marchler-Bauer, and S. H. Bryant. 2000. Domain size distributions can predict domain boundaries. *Bioinformatics* 16:613–8.
99. George, R. A., and J. Heringa. 2002. SnapDRAGON: A method to delineated protein structural domains from sequence data. *J Mol Biol* 316:839–51.
100. Liu, J., and B. Rost. 2004. Sequence-based prediction of protein domains. *Nucleic Acids Res* 32:3522–30.
101. Marsden, R. L., L. J. McGuffin, and D. T. Jones. 2002. Rapid protein domain assignment from amino acid sequence using predicted secondary structure. *Protein Sci* 11:2814–24.
102. Kuroda, Y., K. Tani, Y. Matsuo, and S. Yokoyama. 2000. Automated search of natively folded protein fragments for high-throughput structure determination in structural genomics. *Protein Sci* 9:2313–21.
103. George, R. A., K. Lin, and J. Heringa. 2005. Scooby-domain prediction of globular domains in protein sequence. *Nucleic Acids Res* 33:W160–3.
104. Suyama, M., and O. Ohara. 2003. DomCut: Prediction of inter-domain linker regions in amino acid sequences. *Bioinformatics* 19:673–4.
105. McGuffin, L. J., K. Bryson, and D. T. Jones. 2000. The PSIPRED protein structure prediction server. *Bioinformatics* 16:404–5.
106. Wootton, J. C. 1994. Non-globular domain in protein sequences: automated segmentation using complexity measures. *Comput Chem* 18:269–85.
107. Promponas, V. J., A. J. Enright, S. Tsoka, D. P. Kreil, C. Leroy, S. Hamodrakas, C. Sander, and C. A. Ouzounis. 2000. CAST: An iterative algorithm for the complexity analysis of sequence tracts. *Bioinformatics* 16:915–22.
108. Linding, R., L. J. Jensen, F. Diella, P. Bork, T. J. Gibson, and R. B. Russell. 2003. Protein disorder prediction: Implications for structural genomics. *Structure* 11:1453–9.
109. Romero, P., Z. Obradovic, X. Li, E. C. Garner, C. J. Brown, and A. K. Dunker. 2001. Sequence complexity of disordered protein. *Proteins* 42:38–48.
110. Ward, J. J., J. S. Sodhi, L. J. McGuffin, B. F. Buxton, and D. T. Jones. 2004. Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *J Mol Biol* 337:635–45.
111. Linding, R., R. B. Russell, V. Neduva, and T. J. Gibson. 2003. GlobPlot: Exploring protein sequences for globularity and disorder. *Nucleic Acids Res* 31:3701–8.
112. Shin, S. W., and S. M. Kim. 2005. A new algorithm for detecting low-complexity regions in protein sequences. *Bioinformatics*. Jan 15;21(2):160–70. Epub 2004 Aug 27.
113. Liu, J., and B. Rost. 2003. NORSp: Predictions of long regions without regular secondary structure. *Nucleic Acids Res* 31:3833–5.
114. Miyazaki, S., Y. Kuroda, and S. Yokoyama. 2002. Characterization and prediction of linker sequences of multi-domain proteins by a neural network. *J Struct Funct Genom* 2:37–51.
115. Sonnhammer, E. L., G. von Heijne, and A. Krogh. 1998. A hidden Markov model for predicting transmembrane helices in protein sequences. *Proc Intl Conf Intell Sys Mol Biol* 6:175–82.
116. Moller, S., M. D. Croning, and R. Apweiler. 2002. Evaluation of methods for the prediction of membrane spanning regions. *Bioinformatics* 7:646–53.
117. Kall, I., A. Krogh, and E. L. L. Sonnhammer. 2004. A combined transmembrane topology and signal peptide prediction method. *J Mol Biol* 338:1027–36.

118. Aravind, L. 2000. Guilt by association: Contextual information in genome analysis. *Genome Res* 10:1074–77.
119. Marcotte, E. M., M. Pellegrini, M. J. Thompson, T. O. Yeates, and D. Eisenberg. 1999. A combined algorithm for genome-wide prediction of protein function. *Nature* 402:83–6.
120. Enright, A. J., I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis. 1999. Protein interaction maps for complete genomes based on gene fusion events. *Nature* 402:86–90.
121. Veitia, R. A. 2002. Rosetta Stone proteins: Chance and necessity? *Genome Biol* 3:interactions1001.1–1001.3.
122. von Mering, C., M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snel. 2003. STRING: A database of predicted functional associations between proteins. *Nucleic Acids Res* 31:258–61.
123. Enright, A. J., and C. A. Ouzounis. 2001. Functional associations of proteins in entire genomes by means of exhaustive detection of gene fusions. *Genome Biol* 2:341–7.
124. Enault, F., K. Suhre, O. Poirot, C. Abergel, and J. M. Claverie. Phydbac2: Improved inference of gene function using interactive phylogenomic profiling and chromosomal location analysis. *Nucleic Acids Res* 32:W336–9.
125. Bowers, P. M., M. Pellegrini, M. J. Thompson, J. Fierro, T. O. Yeates, and D. Eisenberg. Prolinks: A database of protein functional linkages derived from coevolution. *Genome Biol* 5:R35.
126. Suhre, K., and J. M. Claverie. FusionDB: A database for in-depth analysis of prokaryotic gene fusion events. *Nucleic Acids Res* 32:273–6.
127. Zhang, Z., and S. K. Ng. 2004. InterWeaver: Interaction reports for discovering potential protein interaction partners with online evidence. *Nucleic Acids Res* 32:W73–5.
128. Geer, L. Y., M. Domrachev, D. J. Lipman, and S. H. Bryant. 2002. CDART: Protein homology by domain architecture. *Genome Res* 12:1619–23.
129. Dandekar, T., B. Snel, M. Huynen, and P. Bork. 1998. Conservation of gene order: A fingerprint of proteins that physically interact. *Trends Biochem Sci* 23:324–8.
130. Wolf, Y. I., I. B. Rogozin, A. S. Kondrashov, and E. V. Koonin. 2001. Genome alignment, evolution of prokaryotic genome organization, and prediction of gene function using genomic context. *Genome Res* 11:356–72.
131. Page, A. 1999. A highly conserved nematode protein folding operon in *Caenorhabditis elegans* and *Caenorhabditis briggsae*. *Gene* 230:267–75.
132. Canaves, J. M. 2004. Predicted role for the archease protein family based on structural and sequence analysis of TM1083 and MTH1598, two proteins structurally characterized through structural genomics efforts. *Proteins* 56:19–27.
133. Overbeek, R., N. Larsen, T. Walunas, M. D'Souza, G. Pusch, E. Selkov, K. Liolios, et al. 2003. The ERGO™ genome analysis and discovery system. *Nucleic Acids Res* 31:164–71.
134. Kolesov, G., H. W. Mewes, and D. Frishman. SNAPping up functionally related genes based on context information: A colinearity-free approach. *J Mol Biol* 311:639–56.
135. Overbeek, R., M. Fonstein, M. D'Souza, G. D. Pusch, and N. Maltsev. 1999. Use of contiguity on the chromosome to predict functional coupling. *In Silico Biol* 1:93–108.
136. Kanehisa, M., and S. Goto. 2000. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res* 28:27–30.
137. Date, S. V., and E. M. Marcotte. 2003. Discovery of uncharacterized cellular systems by genome-wide analysis of functional linkages. *Nat Biotechnol* 21:1055–62.

138. Munch, R., K. Hiller, H. Barg, D. Heldt, S. Linz, E. Windenger, and D. Jahn. PRODORIC: Prokaryotic database of gene regulation. *Nucleic Acids Res* 31:266–9.
139. Ciria, R., C. Abreu-Goodger, E. Morett, and E. Merino. 2004. GeConT: Gene context analysis. *Bioinformatics* 20:2307–8.
140. Pellegrini, M., E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. 1999. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proc Natl Acad Sci USA* 96:4285–8.
141. Marcotte, E. M., I. Xenarios, A. M. van der Bliek, and D. Eisenberg. Localizing proteins in the cell from their phylogenetic profiles. *Proc Natl Acad Sci USA* 97:12115–20.
142. Galperin, M. Y., and E. V. Koonin. 2000. Who's your neighbor? New computational approaches for functional genomics. *Nat Biotechnol* 18:609–13.
143. Dynes, J. L., and R. A. Firtel. 1989. Molecular complementation of a genetic marker in *Dyctiostelium* using a genomic DNA library. *Proc Natl Acad Sci USA* 86:7966–70.
144. Myllykallio, H., G. Lipowski, D. Leduc, J. Filee, P. Forterre, and U. Lieb. 2002. An alternative flavin-dependent mechanism for thymidylate synthesis. *Science* 297:105–7.
145. Mathews, I. I., A. M. Deacon, J. M. Canaves, D. McMullan, S. A. Lesley, S. Agarwalla, and P. Kuhn. Functional analysis of substrate and cofactor complex structures of a thymidylate synthase-complementing protein. *Structure* 11:677–90.
146. Kuhn, P., S. A. Lesley, I. I. Mathews, J. M. Canaves, L. S. Brinen, X. Dai, A. M. Deacon, et al. 2002. Crystal structure of thy1, a thymidylate synthase complementing protein from *Thermotoga maritima* at 2.25 A resolution. *Proteins* 49:142–5.
147. Enault, F., K. Suhre, O. Poirot, C. Abergel, and J. M. Claverie. Phydbac2: improved inference of gene function using interactive phylogenomic profiling and chromosomal location analysis. *Nucleic Acids Res* 32:W336–9.
148. Tan, S. H., Z. Zhang, and S. K. Ng. 2004. ADVICE: Automated detection and validation of interaction by co-evolution. *Nucleic Acids Res* 32:W69–72.
149. Ramani, A. K., and E. M. Marcotte. 2003. Exploiting the co-evolution of interacting proteins to discover interaction specificity. *J Mol Biol* 327:273–84.
150. Karp, P. D., C. Ouzounis, and S. Paley. 1996. HinCyc: A knowledge base of the complete genome and metabolic pathways of *H. influenzae*. *Proc Int Conf Intell Syst Mol Biol* 4:116–24.
151. Krieger, C. J., P. Zhang, L. A. Mueller, A. Wang, S. Paley, M. Arnaud, J. Pick, S. Y. Rhee, and P. D. Karp. 2004. MetaCyc: A multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Res* 32:D438–42.
152. Kelley, B. P., B. Yuan, F. Lewitter, R. Sharan, B. R. Stockwell, and T. Ideker. 2004. PathBLAST: A tool for alignment of protein interaction networks. *Nucleic Acids Res* 32:W83–8.
153. Claudel-Renard, C., C. Chevalet, T. Faraut, and D. Kahn. 2003. Enzyme-specific profiles for genome annotation: PRIAM. *Nucleic Acids Res* 31:6633–9.
154. Hou, B. K., J. S. Kim, J. H. Jun, D. Y. Lee, Y. W. Kim, S. Chae, M. Roh., Y. H. In, and S. Y. Lee. 2004. BioSilico: An integrated metabolic database system. *Bioinformatics*. Nov 22;20(17):3270–2. Epub 2004 Jun 16.
155. Shannon, P., A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. 2003. Cytoscape: A software environment for integrated models of biomolecular networks. *Genome Res* l3:2498–504.
156. Bader, G. D., I. Donaldson, C. Wolting, B. F. Ouellette, T. Pawson, C. W. Hogue. 2001. BIND—The Biomolecular Interaction Network Database. *Nucleic Acids Res* 29:242–5.

157. Xenarios, I., L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, and D. Eisenberg. 2002. DIP, the Database of Interacting Proteins: A research tool for studying cellular networks of protein interactions. *Nucleic Acids Res* 30:303–5.
158. Zanzoni, A., L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni. 2002. MINT: A Molecular INTeraction database. *FEBS Lett* 513:135–40.
159. Bretkreutz, B. J., C. Stark, and M. Tyers. 2003. The GRID: The General Repository for Interaction Datasets. *Genome Biol* 4:R23.
160. Jeong, H., S. P. Mason, A. L. Barabasi, and Z. N. Oltvai. 2001. Lethality and centrality in protein networks. *Nature* 411:41–2.
161. Elsen, M. B., P. T. Spellman, P. O. Brown, and D. Botstein. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA* 95:14863–8.
162. Tamayo, P., D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrosky, E. S. Lander, and T. R. Golub. 1999. Interpreting patterns of gene expression with self-organizing maps: Methods and applications to hematopoietic differentiation. *Proc Natl Acad Sci USA* 96:2907–12.
163. Lockhart, D. J., and E. A. Winzeler. 2000. Genomics, gene expression and DNA arrays. *Nature* 405:827–36.
164. Marshall, E. 2004. Getting the noise out of gene arrays. *Science* 306:630–1.
165. Gollub, J., C. A. Ball, G. Binkley, J. Demeter, D. B. Finkelstein, J. M. Hebert, T. Hernandez-Boussard, et al. 2003. The Stanford Microarray Database: Data access and quality assessment tools. *Nucleic Acids Res* 31:94–6.
166. Bono, H., T. Kasukawa, Y. Hayashizaki, and Y. Okazaki. 2002. READ: RIKEN Expression Array Database. *Nucleic Acids Res* 30:211–3.
167. Brazma, A., H. Parkinson, U. Sarkans, M. Shojatalab, J. Vilo, N. Abeygunawardena, E. Holloway, et al. 2003. ArrayExpress—A public repository for microarray gene expression data at the EBI. *Nucleic Acids Res* 31:68–71.
168. Zhao, T., L. Wild, E. Milik, and C. Smith. 2000. CMS Molecular Biology Resource: Web portal to data analysis and databases. Abstract 907. ASBMB/ASPET Annual Meeting, Boston, MA. *FASEB Journal*. 15(5): A879.
169. Gasteiger, E., A. Gattiker, C. Hoogland, I. Ivanyi, R. D. Appel, and A. Bairoch. 2003. ExPASy: The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res* 31:3784–8.
170. Galperin, M. Y. 2004. The molecular biology database collection: 2004 update. *Nucleic Acids Res* 32:D3–22.
171. Marti-Renom, M. A., A. Stuart, A. Fiser, R. Sanchez, F. Melo, and A. Sali. 2000. Comparative protein structure modeling of genes and genomes. *Annu Rev Biophys Biomol Struct* 29:291–325.
172. Rodriguez, R., G. Chinea, N. Lopez, T. Pons, and G. Vriend. 1998. Homology modeling, model and software evaluation: Three related resources. *Comp Appl Biol Sci* 14:523–8.
173. Schwede, T., J. Kopp, N. Guex, and M. C. Peitsch. 2003. SWISS-MODEL: An automated protein homology-modeling server. *Nucleic Acids Res* 31:3381–5.
174. Teodorescu, O., T. Galor, J. Pillardy, and R. Elber. 2004. Enriching the sequence substitution matrix by structural information. *Proteins* 54:41–8.
175. Bates, P. A., L. A. Kelley, R. M. MacCallum, and M. J. Sternberg. 2001. Enhancement of protein modeling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. *Proteins* S5:39–46.

176. Lund, O., M. Nielsen, C. Lundegaard, and P. Worning. 2002. CPHmodels 2.0: X3M a computer program to extract 3D models. Abstract A102. CASP5 Conference, Asilomar, CA.
177. Bujnicki, J. M., A. Elofsson, D. Fischer, and L. Rychlewski. 2001. Structure prediction meta server. *Bioinformatics* 17:750–1.
178. Kurowski, M. A., and J. M. Bujnicki. 2003. GeneSilico protein structure prediction meta-server. *Nucleic Acids Res* 31:3305–7.
179. Bonneau, R., J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C. E. Strauss, and D. Baker. 2001. Rosetta in CASP4: Progress in *ab initio* protein structure prediction. *Proteins* S5:119–26.
180. Rohl, C. A., C. E. Strauss, K. M. Misura, and D. Baker. 2004. Protein structure prediction using Rosetta. *Methods Enzymol* 383:66–93.
181. Bonneau, R., N. S. Baliga, E. W. Deutsch, P. Shannon, and L. Hood. 2004. Comprehensive *de novo* structure prediction in a systems-biology context for the archaea *Halobacterium sp*. NRC-1. *Genome Biol* 5:R52.
182. Bonneau, R., J. Tsai, I. Ruczinski, and D. Baker. 2001. Functional inferences from blind *ab initio* protein structure predictions. *J Struct Biol* 134:186–90.
183. Kim, D. E., D. Chivian, and D. Baker. 2004. Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Res* 32:W526–31.
184. Bystroiff, C., and Y. Shao. 2002. Fully automated *ab initio* protein structure prediction using I-SITES, HMMSTR and ROSETTA. *Bioinformatics* 18:S54–61.
185. Holm, L., and C. Sander. 1993. Protein structure comparison by alignment of distance matrices. *J Mol Biol* 233:123–38.
186. Krissinel, E., and K. Henrick. Protein structure comparison in 3D based on secondary structure matching (SSM) followed by C-alpha alignment, scored by a new structural similarity function. In *Proceedings of the 5th International Conference on Molecular Structural Biology*, Vienna, September 3–7 (2003) ed. A. J. Kungl and P. J. Kungl, p. 88.
187. Shindyalov, I. N., and P. E. Bourne. 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng* 11:739–47.
188. Kleywegt, G. J., and T. A. Jones. 1997. Detecting folding motifs and similarities in protein structures. *Methods Enzymol* 277:525–45.
189. Kawabata, T. 2003. MATRAS: A program for protein 3D structure comparison. *Nucleic Acids Res* 31:3367–9.
190. Alexandrov, N., and I. Shindyalov. 2003. PDP: Protein domain parser. *Bioinformatics* 19:429–30.
191. Guo, J. T., D. Xu, D. Kim, and Y. Xu. 2003. Improving the performance of DomainParser for structural domain partition using neural network. *Nucleic Acids Res* 31:944–52.
192. Hinsen, K., A. Thomas, and M. J. Field. 1999. Analysis of domain motions in large proteins. *Proteins* 34:369–82.
193. Harrison, A., F. Pearl, I. Sillitoe, T. Slidel, R. Mott, J. Thornton, and C. Orengo. Recognizing the fold of a protein structure. *Bioinformatics* 19:1748–59.
194. Orengo, C. A., and W. R. Taylor. 1996. SSAP: Sequential structure alignment program for protein structure comparison. *Methods Enzymol* 266:617–35.
195. Maiti, R., G. H. Van Domselaar, H. Zhang, and D. S. Wishart. 2004. SuperPose: A simple server for sophisticated structural superposition. *Nucleic Acids Res* 32:W590–4.
196. Bachar, O., D. Fischer, R. Nussinov, and H. J. Wolfson. 1993. A computer vision based technique for 3-D sequence independent structural comparison of proteins. *Protein Eng* 6:279–88.

197. Ye, Y., and A. Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics* 19:246–55.
198. Shatsky, M., H. J. Wolfson, and R. Nussinov. 2002. Flexible protein alignment and hinge detection. *Proteins* 48:242–56.
199. Alexandrov, N. N., and D. Fischer. 1996. Analysis of topological and nontopological structural similarities in the PDB: New examples with old structures. *Proteins* 25:354–65.
200. Dror, O., H. Benyamini, R. Nussinov, and H. Wolfson. 2003. MASS: Multiple structural alignment by secondary structures. *Bioinformatics* 19, Suppl. no. 1:95–104.
201. Guda, C., S. Lu, E. D. Scheeff, P. E. Bourne, and I. N. Shindyalov. 2004. CE-MC: A multiple protein structure alignment server. *Nucleic Acids Res* 32:W100–3.
202. Stark, A., S. Sunyaev, and R. B. Russell. 2003. A model for statistical significance of local similarities in structure. *J Mol Biol* 326:1307–16.
203. Kleywegt, G. J. 1999. Recognition of spatial motifs in protein structures. *J Mol Biol* 285:1887–97.
204. Brady, G. P., and P. F. W. Stouten. 2000. Fast prediction and visualization of protein binding pockets with PASS. *J Comput Aided Mol Des* 14:383–401.
205. Shulman-Peleg, A., R. Nussinov, and H. J. Wolfson. 2004. Recognition of functional sites in protein structures. *J Mol Biol* 339:607–33.
206. Nayal, M., B. C. Hitz, and B. Honig. 1999. GRASS: A server for the graphical representation and analysis of structures. *Protein Sci* 8:676–9.
207. Laskowski, R. A. 1995. SURFNET: A program for visualizing molecular surfaces, cavities and intermolecular interactions. *J Mol Graph* 13:323–30.
208. Devos, D., and A. Valencia. 2001. Intrinsic errors in genome annotation. *Trends Genet* 17:429–31.
209. Devos, D., and A. Valencia. 2000. Practical limits of function prediction. *Proteins* 41:98–107.
210. Galperin, M. Y., and E. Koonin. 1998. Sources of systematic error in functional annotation of genomes: Domain rearrangement, non-orthologous gene displacement and operon disruption. *In Silico Biol* 1:55–67.

LINKS TO TOOLS MENTIONED IN THE TEXT

3.1 Introduction to Functional Annotation
n/a

3.2 Sequence-Based Function Assignment
n/a

3.2.1 Assigning Function by Direct Sequence Similarity
NCBI-BLAST (http://www.ncbi.nlm.nih.gov/BLAST/)
FASTA (http://fasta.bioch.virginia.edu/) (ftp://ftp.virginia.edu/pub/fasta)

3.2.2 Detection of Distant Similarities with Profile Methods
PSI-BLAST (http://www.ncbi.nlm.nih.gov/BLAST/)
HMMER (http://bio.ifom-firc.it/HMMSEARCH/)
(http://hmmer.wustl.edu/)
SAM (http://www.cse.ucsc.edu/research/compbio/HMM-apps/T02-query.html)
FFAS03 (http://ffas.ljcrf.edu)
Structure Prediction Meta Server (http://bioinfo.pl/meta)

COMPASS (ftp://iole.swmed.edu/pub/compass/)
COACH (http://www.drive5.com/lobster/)
HHpred (http://protevo.eb.tuebingen.mpg.de/toolkit/index.php?view=hhpred)

3.2.3 Multiple Sequence Alignment

Clustering Programs

CD-HIT (http://bioinformatics.ljcrf.edu/cd-hi/)
Blastclust (http://www.ncbi.nlm.nih.gov/BLAST/)
DIVCLUS (http://www.mrc-lmb.cam.ac.uk/genomes/divclus_home.html)
GeneRAGE (http://www.ebi.ac.uk/research/cgg/services/rage/)
TribeMCL (http://www.ebi.ac.uk/research/cgg/tribe/)
Decrease Redundancy Tool (http://au.expasy.org/tools/redundancy/)

3.2.3.1 Multiple Sequence Alignment Methods

Multiple Sequence Alignment Programs

ClustalW (http://www.ebi.ac.uk/clustalw/)
DCA (http://bibiserv.techfak.uni-bielefeld.de/dca/)
DIALIGN2 (http://bibiserv.techfak.uni-bielefeld.de/dialign/)
POA (http://www.bioinformatics.ucla.edu/poa/)
T-Coffee (http://www.ch.embnet.org/software/TCoffee.html)
MAFFT (http://www.biophys.kyoto-u.ac.jp/~katoh/programs/align/mafft/)
WebMAFFT (http://www.biophys.kyoto-u.ac.jp/webmafft/)
MUSCLE (http://www.drive5.com/muscle)

3.2.3.2 Integration of Multiple Sequence Alignments and Structural Data

T-Coffee (http://www.ch.embnet.org/software/TCoffee.html)
AltAVisT (http://bibiserv.techfak.uni-bielefeld.de/altavist/)
3D-Coffee (http://igs-server.cnrs-mrs.fr/Tcoffee)

3.2.3.3 Analysis of Multiple Sequence Alignment Data

CRASP (http://wwwmgs2.bionet.nsc.ru/mgs/programs/crasp/)
PCOAT (ftp://iole.swmed.edu/pub/PCOAT/)
TraceSuite II (http://www-cryst.bioc.cam.ac.uk/~jiye/evoltrace/evoltrace.html)
Consurf (http://consurf.tau.ac.il/)
JEVTrace (http://www.cmpharm.ucsf.edu/~marcinj/JEvTrace/)
RASCAL (ftp://ftp-igbmc.u-strasbg.fr/pub/RASCAL)

3.2.3.4 Visualization and Edition of Multiple Sequence Alignments

Sequence Alignment Editors/Viewers

Amas (http://barton.ebi.ac.uk/barton/servers/amas_server.html)
BioEdit (http://www.mbio.ncsu.edu/BioEdit/bioedit.html)
Genedoc (http://www.psc.edu/biomed/genedoc/)
SeaView (http://pbil.univ-lyon1.fr/software/seaview.html)
Cinema 5 (http://aig.cs.man.ac.uk/research/utopia/cinema/cinema.php)
Belvu (http://www.cgb.ki.se/cgb/groups/sonnhammer/Belvu.html)
Jalview (http://www.jalview.org/)
Pfaat (http://www.pfizerdtc.com)

POAVIZ (http://www.bioinformatics.ucla.edu/poa/POA_Online/Visualize.cgi)
WAVis (http://wavis.img.cas.cz/)
DCSE (http://rrna.uia.ac.be/dcse/index.html#whatisDCSE)

Sequence Alignment Editors/Viewers + Structure

STRAP (http://www.charite.de/bioinf/strap/)
ViTO (http://bioserv.cbs.cnrs.fr/VITO/DOC/index.html)
ModView (http://mozart.bio.neu.edu/%7Eilyin/mv-neu/modview-neu.html)

Processing and Visualization

WebLogo (http://weblogo.berkeley.edu/)
Entropy Calculator (http://www.unifi.it/unifi/scibio/bioinfo/ent_man.html)
WebVar (http://www.pesolelab.it/Tools/WebVar.html)

3.2.4 Functional Domain Identification

n/a

3.2.4.1 Direct Domain Assignment through Search in Domain/Family Databases

Databases

Pfam @ WUSTL (http://pfam.wustl.edu)
Pfam @ Sanger (http://www.sanger.ac.uk)
NCBI-CDD (http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml)
SMART (http://smart.embl-heidelberg.de/)
COG (http://www.ncbi.nlm.nih.gov/COG/)
PROSITE (http://au.expasy.org/prosite/)
ProDom (http://protein.toulouse.inra.fr/prodom.html)
PRINTS (http://bioinf.man.ac.uk/dbbrowser/PRINTS/)
UniProt (http://www.uniprot.org/)
TIGRFAMs (http://www.tigr.org/TIGRFAMs)
PIR-Superfamily (http://pir.georgetown.edu/pirsf/)
SUPERFAMILY (http://supfam.mrc-lmb.cam.ac.uk/SUPERFAMILY/)
InterPro (http://www.ebi.ac.uk/interpro/)

Batch Searches

Pfam Batch Searches @ Sanger (http://www.sanger.ac.uk/Software/Pfam/search.shtml)
SledgeHMMER (http://SledgeHmmer.sdsc.edu)

Search Tools

InterProScan (ftp://ftp.ebi.ac.uk/pub/databases/interpro/iprscan/)
RPS-BLAST (ftp://ftp.ncbi.nlm.nih.gov/blast/)
Pfam::Alyzer (http://pfam.cgb.ki.se/pfamalyzer/)

Enzyme Classification

SVM-ProtEnzyme (http://jing.cz3.nus.edu.sg/cgi-bin/svmprot.cgi)
ArchaeaFun (http://www.cbs.dtu.dk/services/ArchaeaFun/)

3.2.4.2 Domain Assignment through Indirect Evidence

Domain Predictors

DGS (http://www.ncbi.nlm.nih.gov/Structure/dgs/DGSWeb.cgi)
DOMpro (http://www.ics.uci.edu/~baldig/dompro.html)

DomPred—DomSSEA (http://bioinf.cs.ucl.ac.uk/dompred/)
PASS (http://www.bio.gsc.riken.go.jp/PASS/pass_query.htm)
ScoobyDo (http://ibivu.cs.vu.nl/programs/scoobywww/)
Domain Predictor @ Biozon.org (http://biozon.org/tools/domains/)
DomCut (http://www.bork.embl-heidelberg.de/~suyama/domcut/)

Secondary-Structure Prediction

PSIPRED (http://bioinf.cs.ucl.ac.uk/psipred/)

Disorder Predictors

SEG (http://fasta.bioch.virginia.edu/fasta_www/pseg.htm)
CAST (http://maine.ebi.ac.uk:8000/services/cast/)
DisEMBL (http://dis.embl.de/)
PONDR (http://www.pondr.com)
Disopred 2 (http://bioinf.cs.ucl.ac.uk/disopred/)
GlobPlot (http://globplot.embl.de/)
DISPro (http://www.ics.uci.edu/~baldig/dispro.html)
CARD (http://bioinfo.knu.ac.kr/research/CARD/)
NORsp (http://cubic.bioc.columbia.edu/services/NORSp)

Linker Predictors

Armadillo (http://armadillo.blueprint.org/)
DLP2 (http://www.bio.gsc.riken.go.jp/cgi-bin/DLP/dlp2.cgi)
DomCut (http://www.bork.embl-heidelberg.de/~suyama/domcut/)

Transmembrane Helix Predictors

TMHMM (http://www.cbs.dtu.dk/services/TMHMM/)
Phobius (http://phobius.cgb.ki.se/)

Other

ProtScale (http://www.expasy.org/tools/protscale.html)

3.2.5 Function Assignments Based on Contextual Information
n/a

3.2.5.1 Gene Fusions: The Rosetta Stone Method
STRING (http://string.embl.de/)
Allfuse (http://maine.ebi.ac.uk:8000/services/allfuse/)
FusionDB (http://igs-server.cnrs-mrs.fr/FusionDB/)
InterWeaver (http://interweaver.i2r.a-star.edu.sg/)
Phydbac (http://igs-server.cnrs-mrs.fr/phydbac/)

3.2.5.2 Domain Co-occurrence
Pfam Domain Query Tool (http://www.sanger.ac.uk/cgi-bin/Pfam/dql.pl)
SMART Domain Query Tool (http://smart.embl-heidelberg.de/)
CDART (http://www.ncbi.nlm.nih.gov/Structure/lexington/lexington.cgi?cmd=rps)

3.2.5.3 Genomic Context: Gene Neighborhoods, Gene Clusters, and Operons
ERGO (http://ergo.integratedgenomics.com/ERGO)
STRING (http://string.embl.de/)
SNAPper (http://pedant.gsf.de/snapper)
SEED (http://theseed.uchicago.edu/FIG/)

KEGG Gene Cluster Search (http://www.genome.jp/kegg-bin/mk_genome_cmp_html)
Gene Neighbors Database (http://bioinformatics.icmb.utexas.edu/operons/index.html)
NCBI Map Viewer (http://www.ncbi.nlm.nih.gov/mapview/static/MVstart.html)
PRODORIC (http://prodoric.tu-bs.de)
GeConT (http://www.ibt.unam.mx/biocomputo/gecont.html)
Prolinks (http://dip.doe-mbi.ucla.edu/pronav)

3.2.5.4 Phylogenomic Profiles
STRING (http://string.embl.de/)
Prolinks (http://dip.doe-mbi.ucla.edu/pronav)
Phydbac (http://igs-server.cnrs-mrs.fr/phydbac/)
ADVICE (http://advice.i2r.a-star.edu.sg/)
PLEX (http://bioinformatics.icmb.utexas.edu/plex/plex.html)
MATRIX (http://orion.icmb.utexas.edu/matrix/)

3.2.5.5 Metabolic Reconstruction
KEGG (http://www.genome.jp/kegg/)
MetaCyc (http://metacyc.org/)
PathBLAST (http://www.pathblast.org/bioc/pathblast/blastpathway.jsp)
SEED (http://theseed.uchicago.edu/FIG/)
PUMA2 (http://compbio.mcs.anl.gov/puma2/cgi-bin/index.cgi)
ERGO (http://ergo.integratedgenomics.com/ERGO/)
metaSHARK (http://bioinformatics.leeds.ac.uk/shark/)
PathFinder (http://bibiserv.techfak.uni-bielefeld.de/pathfinder/)
PRIAM (http://bioinfo.genopole-toulouse.prd.fr/priam/)
BioSilico (http://biosilico.kaist.ac.kr)
Cytoscape (http://www.cytoscape.org)

3.2.5.6 Protein–Protein Interactions
BIND (http://bind.ca/)
DIP (http://dip.doe-mbi.ucla.edu/)
MINT (http://160.80.34.4/mint/)
GRID (http://biodata.mshri.on.ca/grid/servlet/Index)
STRING (http://string.embl.de/)
InterWeaver (http://interweaver.i2r.a-star.edu.sg/)

3.2.5.7 Microarray Expression Profiles
Stanford Microarray Database (http://genome-www5.stanford.edu/)
RIKEN Expression Array Database (http://read.gsc.riken.go.jp/)
EBI ArrayExpress (http://www.ebi.ac.uk/arrayexpress/)

3.2.5.8 Other Sources of Contextual Information for Protein Annotation
CMS Molecular Biology Resource (http://restools.sdsc.edu/)
ExPASy Proteomics Server (http://www.expasy.org/tools/)
EBI Toolbox (http://www.ebi.ac.uk/Tools/)

Herbert Mayer's Bioinformatics World (http://homepage.univie.ac.at/herbert.mayer/)
NAR 2004 Molecular Biology Database Collection (http://nar.oupjournals.org/cgi/content/full/32/suppl_1/D3/DC1)
NAR 2004 Tools Issue (http://nar.oupjournals.org/content/vol32/suppl_2/index.dtl)

3.3 From Sequence to Structure: Homology and *Ab Initio* Structure Models
Homology Modeling
Modeller (http://salilab.org/modeller/)
WhatIf (http://www.cmbi.kun.nl/whatif/)
Jackal (http://trantor.bioc.columbia.edu/programs/jackal/index.html)
ModWeb (http://alto.compbio.ucsf.edu/modweb-cgi/main.cgi)
Swiss-Model (http://swissmodel.expasy.org//SWISS-MODEL.html)
LOOP (http://ser-loopp.tc.cornell.edu/cbsu/loopp.htm)
3D-Jigsaw (http://www.bmm.icnet.uk/servers/3djigsaw/)
HOMER (http://protein.cribi.unipd.it/ssea/)
CPHmodels (http://www.cbs.dtu.dk/services/CPHmodels/)
Fold Recognition Metaservers
Bioinfo.PL (http://bioinfo.pl/meta/)
Genesilico (http://genesilico.pl/meta/)
Ab Initio Modeling
Rosetta (http://depts.washington.edu/ventures/UW_Technology/Express_Licenses/Rosetta/)
Robetta (http://robetta.bakerlab.org/)
HMMSTR/Rosetta (http://www.bioinfo.rpi.edu/~bystrc/hmmstr/server.php)

3.4 Structure-Based Functional Annotation
n/a

3.4.1 Structural Database Searches
Database Searches
DALI (http://www.ebi.ac.uk/dali/)
VAST (http://www.ncbi.nlm.nih.gov/Structure/VAST/vast.shtml)
SSM (http://www.ebi.ac.uk/msd-srv/ssm/)
CE (http://cl.sdsc.edu/ce.html)
DEJAVU (http://portray.bmc.uu.se/cgi-bin/dejavu/scripts/dejavu.pl)
MATRAS (http://biunit.aist-nara.ac.jp/matras/)
Domain Partitioning
PDP (http://123d.ncifcrf.gov/pdps.html)
DomainParser (http://compbio.ornl.gov/structure/domainparser/)
DomainFinder (http://dirac.cnrs-orleans.fr/DomainFinder/)
Structural Domain Databases and Tools
CATH (http://www.biochem.ucl.ac.uk/bsm/cath/)
SCOP (http://scop.mrc-lmb.cam.ac.uk/scop/)
GRATH (http://www.biochem.ucl.ac.uk/cgi-bin/cath/Grath.pl)

3.4.2 Structural Alignments

Rigid

CE (http://cl.sdsc.edu/ce.html)
MATRAS (http://biunit.aist-nara.ac.jp/matras/)
SSAP (http://www.biochem.ucl.ac.uk/cgi-bin/cath/GetSsapRasmol.pl)
SuperPose (http://wishart.biology.ualberta.ca/SuperPose/)
C-alpha Match (http://bioinfo3d.cs.tau.ac.il/c_alpha_match/)
ProFit (http://www.bioinf.org.uk/software/profit)

Flexible

FATCAT (http://fatcat.burnham.org)
FlexProt (http://bioinfo3d.cs.tau.ac.il/FlexProt/)

Nontopological

SARF2 (http://123d.ncifcrf.gov/sarf2.html)
MASS (http://bioinfo3d.cs.tau.ac.il/MASS/server.html)

Multiple Structural Alignment

CE-ME (http://cemc.sdsc.edu/)
MASS (http://bioinfo3d.cs.tau.ac.il/MASS/server.html)
ProFit (http://www.bioinf.org.uk/software/profit)
MATRAS (http://biunit.aist-nara.ac.jp/matras/)

3.4.3 Use of Structural Descriptors

PINTS (http://www.russell.embl.de/pints/)
SPASM (http://portray.bmc.uu.se/cgi-bin/spasm/scripts/spasm.pl)
PASS (http://www.ccl.net/cca/software/UNIX/pass/overview.shtml)
SiteEngine (http://bioinfo3d.cs.tau.ac.il/SiteEngine/)
Surface (http://cbm.bio.uniroma2.it/surface/)
GRASS (http://honiglab.cpmc.columbia.edu/)
SURFNET (http://www.biochem.ucl.ac.uk/~roman/surfnet/surfnet.html)

3.5 Final Remarks and Future Directions

Entrez Protein Database (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Protein)
Genomes On-line Database (GOLD) (http://www.genomesonline.org/)
Entrez Genome Database (http://www.ncbi.nlm.nih.gov//entrez/query.fcgi?db=Genome)

Bioinformatics Workflows

VIBE (http://www.incogen.com/VIBE)
Taverna (http://taverna.sourceforge.net/)
Pegasys (http://www.bioinformatics.ubc.ca/pegasys/)
SciTegic Pipeline Pilot (http://www.scitegic.com/products_services/pipeline_pilot.htm)

4 The Impact of Genetic VariationonDrugDiscovery and Development

Michael R. Barnes
GlaxoSmithKline

CONTENTS

4.1 SECTION 1

4.1.1 Introduction

Genomic technologies have become increasingly more integrated within the pharmaceutical research and development process to accelerate identification of disease-validated targets and consequently novel chemical entities (NCEs), which act on these targets. Historically, more than 90% of NCEs entering development have not reached the clinic [1]. Failure of these compounds can be multifaceted; they may show insufficient efficacy, often as a result of inadequate target validation, or they may have unacceptable toxicity profiles in animal studies or initial testing in humans. Even after clinical trials involving hundreds of patients, there is still a risk of the emergence of unexpected toxicity in a subset of the population due to rare or population-specific adverse events. Genetic variation can be an underlying factor in all these issues of failure in the drug-discovery process; pharmaceutical companies are now beginning to recognize this and invest their efforts accordingly.

These multiple sources of failure in the drug-discovery process can be minimized by integrating genetics into this process. This integration is already yielding new disease-validated targets for the drug-discovery process. Pharmacogenetics (PGx) is another practical application of this integration and involves the study of the impact of genetic variation on differential response to drugs. Chapter 13 deals with this subject in some detail, so for the purposes of this chapter, I only allude to practical applications of PGx; the reader is directed to some reviews of the impact of genetics in this field, including those by Lindpaintner [2] and Roses [3].

In the context of the discovery of new targets and the development of new drugs, knowledge of genetic variation can inform on many of the functional parameters and critical regions of a gene, protein, or regulatory region. Study genetic variation, and a picture of the driving force of gene evolution emerges. For example, Majewski and Ott [4] surveyed single nucleotide polymorphism (SNP) frequency across exons and introns in the human genome. They found that SNP density declined steadily in the region of exon–intron boundaries and not simply at splice sites. This observation provides compelling evidence that gene-splicing control elements (SCEs), which control gene regulation and splicing, may occur more frequently near intron–exon boundaries. Specifically the evidence suggests that SCE elements are likely to extend (with decreasing frequency) as far as 125 bp into the exon and up to 20 bp into intron sequences.

This example provides an insight into a specific function of a gene that might not have been easily determined by direct study methods. The same principle applies to the use of genetics in target identification and validation, helping to elucidate the function of genes and pathways by studying their function and dysfunction in normal and diseased states. In this chapter I examine the data and some of the underlying technologies that are enabling the integration of genetics into the drug-discovery process. The chapter reviews some of the principle forms of genetic variation and the key databases from which it can be accessed and manipulated. Finally, some of the methods for analysis of the functional impact of genetic variation are reviewed.

4.1.2 Human Genetic Variation in a Drug-Discovery Context

Figure 4.1 illustrates how genetic variation can impact the drug-discovery and development process. The figure shows some of the common activities that are being incorporated into the standard discovery and development pipeline for new drugs. This includes, among other things, an input at the start of the pipeline from targets that have been identified by means of their variation in human disease, screening patient populations for genetic variants in targets that might alter drug efficacy or safety, and the integration of pharmacogenetic studies into the clinical trial process during drug development. The latter activity appears more like a possible future regulatory requirement, as the Food and Drug Administration (FDA) becomes increasingly PGx focused.

The FDA conducted a survey of recent Investigational New Drug and New Drug Applications to identify the extent to which PGx was used in clinical studies [5]. The survey found more than 15 applications in which PGx tests were reported, with

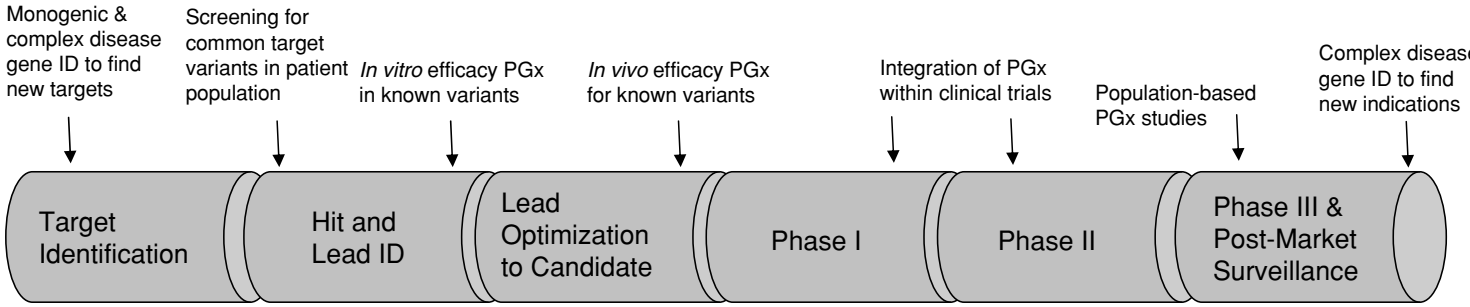


FIGURE 4.1 The impact of genetic variation on the drug-discovery and development process.

all but one test related to pharmacogenetic variability in Cytochrome P450 (CYP) enzymes. The reported use of this information was (a) to define patient subgroups in early-phase pharmacokinetic (PK) studies and/or late-phase efficacy trials using sparse sample analysis and population PK methodologies to assess the significance of geno-/phenotype as a covariate, (b) to provide a *post hoc* explanation for the variability in drug exposure (and response) in some subjects and/or patients (e.g., outliers where plasma levels were too high or low) as a basis to exclude such subjects from an analysis, (c) to measure the impact of CYP enzyme polymorphism on plasma drug clearance in subject subgroups defined by PGx, and (d) to use the results of PGx as an entrance or exclusion criteria for subjects in drug interaction studies. Despite the interest of the FDA, pharmaceutical companies are still moving forward cautiously in integrating PGx in their drug-development programs. In the case of the 15 applications just mentioned, differences thought to be related to PGx subgroups were not used as a basis for any specific dosing recommendations in the product labels.

4.1.3 Forms and Mechanisms of Genetic Variation

Most genetic variation in humans arises from two types of mutational events. The simplest type of variant results from a single nucleotide mutation that substitutes one base for another. This mutation event accounts for the most common variant in the genome, the SNP. Most other forms of human variation result from the insertion or deletion of a section of DNA. Insertion/deletion (INDEL) events also occur frequently throughout the genome. INDELs are particularly prone to occur in repetitive DNA sequences, where repeated nucleotide patterns, so-called variable number tandem repeat polymorphisms (VNTRs), expand or contract as a result of INDEL events. VNTRs are subdivided on the basis of the size of the repeating unit; minisatellites are composed of repeat units ranging from ten to several hundred base pairs. Simple tandem repeats (STRs or microsatellites) are composed of 2 to 6 bp repeat units. All of these variation forms occur in genes and gene regions and account for most of the genetic heterogeneity that underlies human phenotypic variation. Of course, this genetic heterogeneity argues for a dynamic appraisal of the impact of variation in the human genome, moving beyond a monolithic focus on the SNP (the current approach) to encompass the full array of variations, which may have an impact on disease and drug response.

4.1.4 How Much Variation?

The quantity of genetic variation in the human genome is something that until recently has been the realm of many conflicting estimates. Empirical studies quickly identified that, on average, comparison of chromosomes between any two individuals will generally reveal common SNPs (> 20% minor allele frequency) at 0.3 to 1 kb average intervals, which scales up to 5 to 10 million SNPs across the genome [6]. The availability of a complete human genome has allowed increasingly accurate estimates of the number of potentially polymorphic mini- and microsatellites, as VNTRs over a certain number of repeats can be reliably predicted to be polymorphic.

Breen et al. [7] completed an *in silico* survey of potentially polymorphic STRs in the human genome and identified over 100,000 potentially polymorphic microsatellites. Other forms of variation such as small insertion deletions are more difficult to quantify, although they are likely to fall somewhere between the numbers of SNPs and VNTRs.

4.1.5 Single Nucleotide Variation: SNPs and Mutations

Terminology for variation at a single nucleotide position is defined by allele frequency. In the strictest sense, a single base change, occurring in a population at a frequency of less than 1%, is termed an SNP. When a single base change occurs at less than 1%, it is considered to be a mutation. However, this definition is often disregarded; instead, single nucleotide "mutations" occurring at less than 1% in general populations might more appropriately be termed low-frequency SNPs. The term *mutation* is often used to describe a variant identified in diseased individuals or arising somatically in tissues, with a demonstrated role in the disease phenotype. Mutation databases and polymorphism databases have generally been delineated by this definition. The high level of interest in SNP data has led to the development of an excellent centralized SNP database, dbSNP [8], which is reviewed next. Mendelian mutation databases are still lagging behind SNPs in terms of data integration and visualization on the human genome, but these data should not be overlooked, as they can obviously provide a great deal of information about the biology of a gene.

4.1.6 Functional Impact of SNPs and Mutations

The potential functional impact of an SNP is defined by its location in the genome. SNPs may alter gene function by changing recognition sequences in gene regulatory elements, they may alter gene transcript secondary structure and stability, and most obviously they may alter the coding sequence of a gene. Such SNPs within the coding sequence of a gene are termed *synonymous,* where the amino acid codon remains unchanged by the SNP substitution; *nonsynonymous,* where the amino acid codon is altered to code for an alternative amino acid; or *nonsense,* where an amino acid codon is altered to a stop codon.

4.1.7 Candidate SNPs: When Is an SNP Not an SNP?

There is one overwhelming caveat that needs to be considered when dealing with SNP data: most of the SNPs in public databases are "candidate" SNPs of unknown frequency that have been seen at least once in at least one individual. The simple fact is that many SNPs do not exist at a detectable frequency in general populations. More than 60% of the SNPs in dbSNP were detected by statistical methods for identification of candidate SNPs by comparison of DNA sequence traces from different individuals. Marth et al. [9] investigated the reliability of these candidate SNPs in some depth, completing two pilot studies to determine how well candidate SNPs would progress to working assays in three common populations. In both studies, they found that between 52 and 54% of the characterized SNPs turn out to be common SNPs (> 10%) for each population. Significantly, between 30 and 34%

of the characterized SNPs were not detected in each population. These results suggest that if a candidate SNP is selected for study in a common population, there is a 66 to 70% chance that the SNPs will have detectable minor allele frequency (1%–5%) and a 50% chance that the SNPs are common in that population (> 10%). Put another way, approximately 17% of candidate SNPs will have no detectable variation in common populations. These "monomorphic" SNP candidates are likely to represent "private" SNPs, which exist in the individual screened but not appreciably in populations. This finding probably reflects the massive increase in population size and admixture over the past 500 years [10]. Beyond validation of the SNP, the last hurdle is assay design. Many SNPs are located in repetitive or AT-rich regions, which makes assay design difficult; this can account for a further 10 to 30% fallout, depending on the assay technology.

Any SNP-based study needs to take these levels of attrition between SNP and working assay into account. There is only one solution to this problem: to determine the frequency of the 10 million or so public SNPs in common ethnic groups. This is now widely recognized in the SNP research community, and there have been several large-scale SNP frequency determination projects that have provided frequencies for a little less than 10% of these SNPs.

There is one other significant source of SNP validation—the simple observation of an SNP on independent occasions from different individuals. The massive scale of SNP discovery naturally has resulted in the repeated identification of SNPs across different individuals and populations. This determination usually indicates that an SNP is likely to be widely spread in populations and often of higher frequency. These so-called 2-hit SNPs have been identified in dbSNP and provide preliminary validation for approximately 45% of the SNPs in the database. This allows the user to specify 2-hit validation as a minimal requirement in a query of the database. As an aside, the problem of SNP validation is particularly pertinent to the study of nonsynonymous SNPs, as many nonsynonymous SNPs, particularly those that are nonconservative in nature, tend to be "single-hit" SNPs with no validation information. Attempts to validate these SNPs tend to be prone to failure.

4.1.8 VNTR Polymorphisms

VNTRs also have potential to impact the function of genes and regulatory regions. The polymorphic nature of a VNTR is thought to depend on a range of factors: the number of repeats, their sequence content, their chromosomal location, the mismatch repair capability of the cell, the developmental stage of the cell (mitotic or meiotic), and/or the sex of the transmitting parent [11]. Much evidence exists to demonstrate that tandem repeats exert a functional effect on genes; thus, VNTRs in themselves can be candidates for disease or pharmacogenetic susceptibility alleles. The best characterized of these are the triplet repeat expansion diseases. Insertion of triplet repeats is strongly favored over deletion of repeats, so pathological triplet repeat expansions manifest through successive generations with worsening symptoms known as *anticipation*. Most triplet repeat expansions have been identified in monogenic diseases and may occur in almost any genic region. Over five triplet repeat classes have been described so far, causing a range of diseases including Fragile X,

Myotonic Dystrophy, Friedreich's ataxia, several Spinocerebellar ataxias, and Huntington's disease [12]. Spinocerebellar ataxia 10 (SCA10) is notably caused by the largest tandem repeat seen in the human genome [13]. In general populations the SCA10 locus is a 10-22mer ATTCT repeat in intron 9 of the SCA10 gene; in SCA10 patients, the repeat expands to more than 4,500 repeat units, which makes the disease allele up to 22.5 kb larger than the normal allele.

Tandem repeats have also been associated with complex diseases; for example, different alleles of a 14mer VNTR in the insulin gene promoter region have been associated with different levels of insulin secretion. Different alleles of this VNTR have been robustly linked with type I diabetes [14], and in obese individuals they have been associated with the development of type II diabetes [15]. There are also examples of associations with differences in drug response. For example, a tandem repeated GGGCGG polymorphism within the promoter of the 5-Lipoxygenase gene (ALOX5), the first enzyme in the leukotriene biosynthetic pathway, has been shown to play an important role in response to leukotriene modifier therapy [16]. Most individuals carry five repeats of the GGGCGG motif; however, 5 to 35% of Caucasian individuals carry at least one allele with three or two repeats at this locus. Reduction in copies of this repeat removes the consensus binding motif for the transcription factor, SP1, which is also GGGCGG. An alteration in the number of repeats has been shown to decrease the efficiency of ALOX5 transcription. As asthma patients harboring reduced copies of this repeat have diminished ALOX5 gene transcription, their asthma is less dependent on leukotriene formation, and as a result they are less sensitive to the antiasthmatic effects of leukotriene inhibitors, which are one of the mainstays of asthma treatment.

Potentially polymorphic novel VNTRs can be identified directly from genomic sequence using tools such as Tandem repeat finder (TRF) or Tandyman [17] (http://www.stdgen.lanl.gov/tandyman/docs/index.html). A complete analysis of microsatellites in the human genome sequence using both TRF and Tandyman has been presented in the UCSC human genome browser in the "simple repeats" and "microsatellites" tracks [7].

4.1.9 Insertion/Deletion Polymorphisms

Although tandem repeat polymorphisms are a major form of variation in genomes, they may also mediate other forms of variation by predisposing DNA to localized rearrangements between homologous repeats. Such rearrangements give rise to INDEL polymorphisms ranging in size from several base pairs to a megabase or more; these appear to be quite common in most genomes studied so far, probably reflecting their association with common VNTRs. INDELs of all sizes have been associated with an increasing range of genetic diseases. For example, Cambien et al. [18] found association between coronary heart disease and a 287 bp INDEL polymorphism situated in intron 16 of the Angiotensin converting enzyme (ACE). This INDEL, known as the ACE/ID polymorphism, accounts for 50% of the interindividual variability of plasma ACE concentration. Although INDEL polymorphisms are likely to be widely distributed throughout the genome, relatively few have been characterized, and there is no central database collating this form of polymorphism. The Marshfield Clinic

(http://research.marshfieldclinic.org/genetics/indels/) maintains the most comprehensive source of short insertion deletion polymorphisms (SIDPs). Over 200,000 are maintained in a form that can be searched by chromosome location. Other databases such as dbSNP and HGVBASE also capture SIDPs to some extent. Larger INDELs are generally overlooked in databases unless associated with a specific gene or study, in which case they appear in OMIM and other similar sources.

4.1.10 Genetics and the Search for Disease Alleles

The search for genetic alleles that predispose to specific disease or pharmacogenetic response phenotypes calls for study of genetic variation at increasing levels of detail. In the first instance, markers need to be identified at a sufficient density to build marker maps, which can detect disease alleles by linkage or association with marker alleles that are assayed across the genome. This approach relies on the premise that common genetic markers will be in linkage disequilibrium (LD) with rarer disease alleles (see Borecki and Suarez [19] for a review of linkage and association methods). Once this linkage or association is detected, a denser framework of markers is needed to refine the signal to a smaller region. In the case of association of marker and disease alleles, marker density needs to be increased to a level at which most genetic diversity in a population is captured. This action may call for the construction of very dense marker maps down to a resolution of 5 to 10 kb. Ultimately, once LD is established between a marker and a phenotype, it may be necessary to identify all genetic variation across the refined locus, which also shows some LD with the associated marker, prospectively allowing the identification of the disease allele. Each stage of linkage or association analysis involves a progression of bioinformatics tools, each with their own caveats in use as the requirements for detail of biological interpretation increase [20]. However, the availability of a complete human genome has simplified this process considerably as genetic data are now fully integrated on the genome sequence framework. This makes some powerful tools available for detecting, organizing, and analyzing genome scan data.

4.1.11 The Genome as a Framework for Data Integration of Genetic Variation Data

In terms of understanding of the biology of variants and the genes that they impact, exact base pair localization of each variant in the genome allows comprehensive comparison across multiple data domains in relation to genes and regulatory regions. The possibilities for data integration are immense, making it possible to make complex queries of SNP data, using databases like dbSNP—for example, to identify all nonsynonymous SNPs within a specific ethnic group or above a certain allele frequency. By incorporating data from genome viewers such as Ensembl, it is possible to add layers of information. For example, it is possible to identify all SNPs that fall within Human/Mouse conserved regions (outside of exons), which is indicative of evolutionarily conserved function. With the availability of the HapMap, which describes the genetic structure of the genome in several major populations (see the next section), it is even possible to identify SNPs that show coinheritance, which could allow for analysis of correlated mutations within genes and possibly between genes.

4.2 SECTION 2

4.2.1 Human Genetic Variation Databases and Web Resources

The vast range of human genetic variation cannot currently be derived from a single database (although it can mostly be viewed in a genome browser). At best, to gain a comprehensive view of variation in a specific gene or locus, data need to be gathered from several databases, or worse still the data may not be readily available in a database at all, in which case other bioinformatic analysis approaches may be necessary to identify potential variants directly in the genome sequence (e.g., VNTRs). Having described the main forms of human variation, table 4.1 introduces the key databases for mining and visualizing this information. This list is by no means comprehensive; however, it does provide the most comprehensive resources in this field. The reader is encouraged to search the Web using a general search engine (e.g., Google), as many other specialist resources are available.

4.2.2 Mutation Databases: An Avenue into Human Phenotype

The polymorphism data stored in dbSNP are valuable biological information that help to define the natural range of variation in genes and the genome, however, most of the polymorphisms can be assumed to be functionally neutral. By contrast, mutation data are usually functionally and phenotypically well defined and have obvious implications for the understanding of gene and pathways underlying disease. Mutant data can be derived from *in vitro* and *in vivo* experimental sources, such as site-directed mutagenesis and knockouts in model organisms (some example databases are listed in table 4.1), or it can be derived from natural sources in human populations. The study of naturally occurring mutations in humans has been very important in understanding human disease pathology, particularly the relationships between genotype and phenotype and between DNA and protein structure and function. These types of studies can obviously support the drug discovery and development process. A large number of Mendelian disease mutations have been identified over the past 20 years. These discoveries have helped to define many key biological mechanisms, including gene regulatory motifs and protein–protein interactions. Many highly specialized locus specific databases (LSDBs) have been established to exhaustively collate mutation data around a single gene; this information in itself can be highly informative about the functional parameters of a gene. These LSDBs are comprehensively indexed at the Human Genome Variation Society (HGVS) Mutation Waystation site (table 4.1). In this chapter I could not hope to cover all these databases; instead I focus on Online Mendelian Inheritance in Man (OMIM), the most accessible and widely used Mendelian mutation database.

4.2.3 OMIM

OMIM is an online catalog of human genes and their associated mutations, based on the long-running catalog Mendelian Inheritance in Man (MIM), started in 1967

TABLE 4.1
Genetic Variation Focused Databases and Tools on the Web

Mutation Databases	
OMIM	http://www.ncbi.nlm.nih.gov/Omim/
HGMD	http://www.hgmd.org
HGVS Mutation Waystation	http://www.centralmutations.org/
Central Databases (SNPs and mutations)	
HGVbase	http://hgvbase.cgb.ki.se/
dbSNP	http://www.ncbi.nlm.nih.gov/SNP/
Site-Directed Mutagenesis and Model Organism Databases	
GPCRDB snakelike plots	http://www.gpcr.org/7tm/seq/snakes.html
Protein Mutation Database	http://pmd.ddbj.nig.ac.jp/
TBASE (Mouse KOs and Mutations)	http://tbase.jax.org/
Model organism mutation dbs	http://www.humgen.nl/orgspecdatabases.html
Gene-Orientated SNP and Mutation Visualization	
SNPper	http://snpper.chip.org/
GeneSNPs	http://www.genome.utah.edu/genesnps/
CGAP-GAI SNP database	htpp://lpgws.nci.nih.gov/
SNP500	http://snp500cancer.nci.nih.gov/
Entrez Gene	http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene
Genome-Orientated for SNP and Mutation Visualization	
Ensembl	http://www.ensembl.org
UCSC Human Genome Browser	http://genome.ucsc.edu/index.html
NCBI Map Viewer	http://www.ncbi.nlm.nih.gov/mapview/map_search.cgi/
SNP Data Consortiums and Data Standards	
The SNP consortium (TSC)	http://snp.cshl.org/
JSNP	http://snp.ims.u-tokyo.ac.jp/
HapMap	http://www.hapmap.org/
OMG SNP RFP	http://lsr.omg.org/home.html
XML interface to SNPper for SNP-related data	http://www.bioconductor.org/repository/devel/package/html/RSNPper.html

Note: SNP = single nucleotide polymorphism.

by Victor McKusick at Johns Hopkins [21]. OMIM is an excellent resource for a quick background biology on genes and diseases; it includes information on the most common and clinically significant mutations and polymorphisms in genes. Despite the name, OMIM also covers complex diseases to varying degrees of detail. In January 2002, the database contained over 13,285 entries (including entries on 9,837 gene loci and 982 phenotypes). OMIM is curated by a highly dedicated but small group of curators, but the limits of a manual curation process mean that entries may not be current and are not always comprehensive. With this caveat aside, OMIM is a very valuable database, which usually presents an accurate digest of the literature (it would be difficult to do this automatically). A major bonus of OMIM is that it is very well integrated with the NCBI database family, which makes movement from a disease to a gene to a locus and vice versa fairly effortless. OMIM is now integrated as a Distributed Annotation System track in the Ensembl human genome viewer,

which makes OMIM an even more highly recommended resource for mutation-related data.

4.2.4 SNP Databases

The deluge of SNP data generated over the past few years can be traced primarily to two major overlapping sources: the SNP consortium (TSC) [6] and members of the human genome sequencing consortium, particularly the Sanger Institute and Washington University. The predominance of SNP data from this small number of closely related sources has facilitated the development of a central SNP database—dbSNP at the National Center for Biotechnology Information (NCBI) [8]. Other valuable databases have developed using dbSNP data as a reference; these tools and databases bring focus to specific subsets of SNP data (e.g., gene-oriented SNPs) while enabling further data integration around dbSNP.

4.2.4.1 The dbSNP Database

The NCBI established the dbSNP database in September 1998 as a central repository for both SNPs and short INDEL polymorphisms. In June 2004 (Build 121) dbSNP contained 19.9 million SNPs. These SNPs collapse into a nonredundant set of 9.9 million SNPs, known as Reference SNPs (RefSNPs). Further information exists on a subset of the 9.9 million RefSNPs: 4.5 million are validated, at least in as much as they have been observed more than once, which is a fairly reliable indicator that the SNPs are likely to be real and of relatively high frequency. There are 840,038 that have a know frequency in at least one population. These quantities of SNPs give a very high level of coverage across the genome, with most known exons now within 1 to 2 kb of at least one SNP in the dbSNP database.

4.2.4.2 The RefSNP Dataset

The dbSNP nonredundant RefSNP dataset is produced by clustering SNPs at identical genomic positions and creating a single representative SNP (designated by an "rs" ID). The sequence used in the RefSNP record is derived from the SNP cluster member with the longest flanking sequence. The RefSNP record collates all information from each member of the cluster (e.g., frequency and subject information). The availability of the RefSNP dataset considerably streamlines the process of integrating SNPs with other data sources. External resources almost exclusively use the RefSNP dataset. This makes the RefSNP ID the universal SNP ID in the SNP research community. RefSNPs have also become an integral part of the NCBI data infrastructure, so that the user can effortlessly browse to dbSNP from diverse NCBI resources, including Entrez Gene, Mapview, OMIM, PubMed, and Genbank.

4.2.4.3 Searching dbSNP

There is a bewildering range of approaches for searching dbSNP. The database can be searched directly by SNP accession number, submitter, detection method, population studied, publication, or a sequence-based BLAST search. The Entrez SNP

interface to dbSNP has a complex search form that allows the user to formulate flexible freeform queries of the dbSNP database (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Snp&cmd=Limits). This flexibility allows the user to select SNPs that meet multiple criteria; for example, it is possible to search for all validated nonsynonymous SNPs in gene-coding regions on chromosome 1 that have known allele frequencies in European populations (fig. 4.2). This is a very powerful interface, but it can be somewhat confounding to use. In tests I found that it is necessary

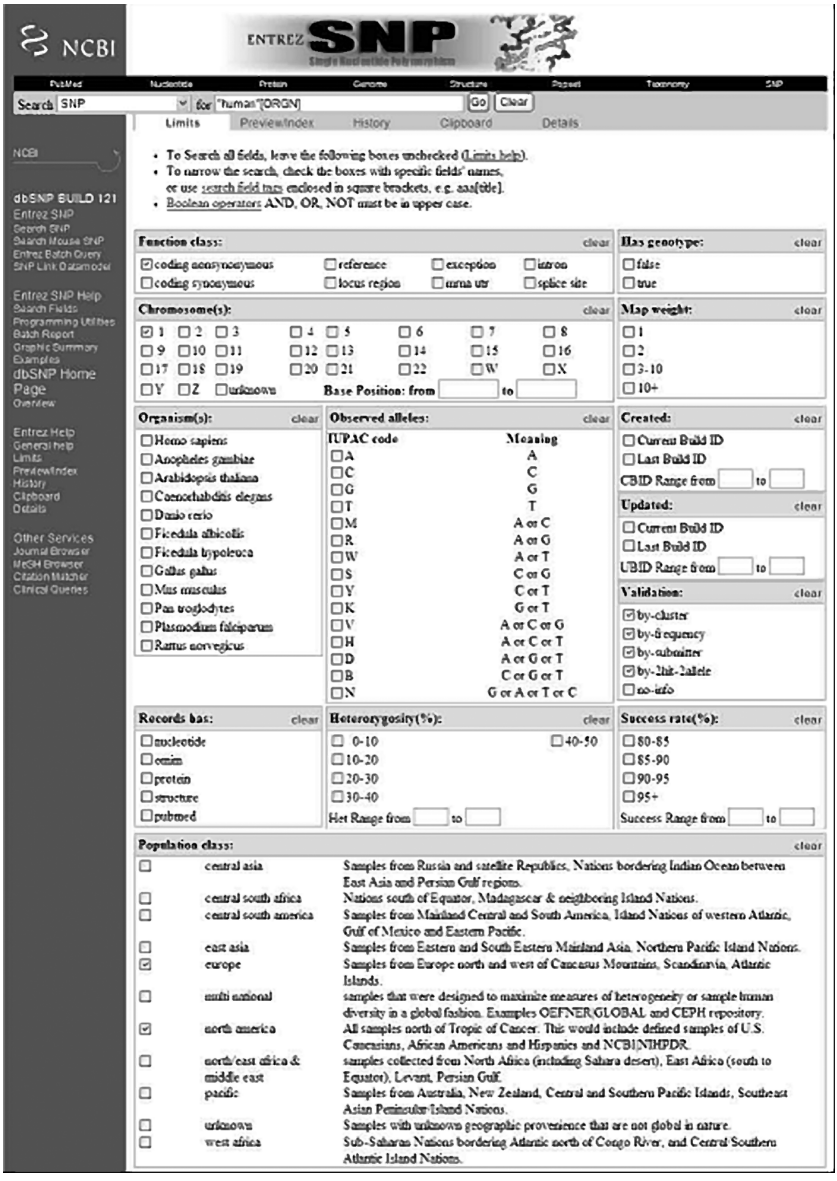


FIGURE 4.2 The Entrez SNP search interface to dbSNP.

TABLE 4.2
Breakdown of Single Nucleotide Polymorphisms by Functional Category in dbSNP (Build 121)

Functional Category	Total Number	Total Validated	Total with Frequency Data
Intronic	2,648,392	1,334,942	197,438
UTR	550,057	267,511	45,980
Promoter region	379,176	181,686	27,031
Coding, nonsynonymous	45,896	17,750	6,975
Coding, synonymous	35,782	18,086	7,956
Splice site	738	134	37

Note: UTR = untranslated region.

to enter a very broad search term in the search box before selecting limits on the search. So, for example, enter ("human"[ORGN]) in the search box and then specify the limits on the query. Using this query facility is easy, and it is possible to build up a view of the quantities of SNPs in genes and gene regions (table 4.2).

There are many other tools that use the dbSNP dataset, for example, Entrez Gene, GeneSNPs, SNPper, and the human genome browsers (table 4.1). These tools can offer powerful alternative interfaces for searching dbSNP, but be aware that third-party (non-NCBI) tools and software may not use the latest version of dbSNP (this is a common problem) so it is important to check which build of the database is being used. Different tools may also use filtering or repeat masking protocols, which can lead to the exclusion of SNPs with poor-quality or short-flanking sequence, or SNPs in repeat regions. If it is important to identify *all* SNPs in a given gene or region, then it is worth consulting several different tools and comparing the results. Some of the best tools for visualizing SNPs across gene and genomic regions are discussed later in this chapter.

4.2.4.4 Human Genome Variation Database

The Human Genome Variation database (HGVbase), previously known as HGBASE ([20]; http://hgvbase.cgb.ki.se/), was initially created in 1998, with a remit to capture all intragenic (promoter to end of transcription) sequence polymorphism. In November 2001, the HGBASE project adopted the new name HGVbase [23]. This modification reflected a change in the scope of the database as it took on a HUGO-endorsed role as a central repository for mutation collection efforts undertaken in collaboration with the HGVS. In 2004 a decision was made to develop HGVbase into a Phenotype/Genotype database. Data exchange with other databases such as dbSNP is maintained, although further submissions are not being accepted while fund-raising and database redesign are ongoing.

There is no doubt that dbSNP has assumed the *de facto* position of the primary central SNP database. To accommodate this, HGVbase has repeatedly sought to

assume a complementary position, with a broader remit covering all single nucleotide variation—both SNPs and Mutations. HGVbase has taken a distinct approach to dbSNP by seeking to summarize all known SNPs as a semivalidated, nonredundant set of records. HGVbase is seeking to address some of the problems associated with candidate SNPs and so, in contrast to the automated approach of dbSNP, HGVbase is highly curated. The curators are aiming to provide a more extensively validated SNP dataset by filtering out SNPs in repeat and low-complexity regions and by identifying SNPs for which a genotyping assay can successfully be designed. The HGVbase curators have also identified SNPs and mutation data from the literature, particularly from older publications before database submission was the norm. HGVbase currently contains 2.85 million nonredundant human polymorphisms and mutations (Release 15.0, July 23, 2003). This currently represents a little over 25% of the data available in dbSNP, so obviously one should not rely on HGVbase as a comprehensive source of SNP data. It remains to be seen how the database will develop in the next year as its focus shifts toward Phenotype and Genotype data.

4.2.4.5 Evolution of SNP-Based Research and Technologies

The perceived value of SNP data to pharmaceutical companies and government research agencies has been demonstrated very early on in the "genomic revolution" that has accompanied the sequencing of the human genome. Indeed the two efforts of genome sequencing and polymorphism discovery have progressed hand in hand, being largely complementary. Sequencing of the genome has involved sequencing a pool of individuals, so naturally polymorphism data, generated from comparisons of these individual sequences, have been a useful byproduct of the sequencing process. However, as the genomic technologies that have allowed such high-throughput sequencing have developed, so the same research groups and funding agencies have become involved in specific polymorphism discovery projects. The earliest and largest of these projects was the SNP Consortium (TSC), closely followed by an equivalent Japanese-funded project (Japanese Single Nucleotide Polymorphisms, or JSNP); most recently, many members of both JSNP and TSC came together to form the HapMap consortium, which sought to investigate the genetic structure of these polymorphisms in common populations.

4.2.4.6 The SNP Consortium (TSC)

The SNP Consortium (TSC) was established in 1999 as a Wellcome Trust–driven collaborative venture funded by more than 10 pharmaceutical companies to produce a public resource of human SNPs. The initial goal, to discover 300,000 SNPs in two years, was impressively exceeded by the TSC, as more than 1.4 million SNPs were released into the public domain by the end of 2001 [24]. Data generated by the TSC were submitted to dbSNP and can also be viewed on the consortium's Web site (http://snp.cshl.org). The SNP consortium was the first major public SNP discovery effort. At this stage, SNP data were relatively scarce, and so to enable effective genetic-mapping approaches the TSC adopted a shotgun sequencing approach to randomly identify SNPs across the entire genome rather than gene-specific regions. The TSC Web site contains no additional SNPs beyond those submitted to dbSNP.

Generally its focus is now applied to serve the genetics research community by providing population and genetic map centric information.

4.2.4.7 JSNP—A Database of Japanese Single Nucleotide Polymorphisms

The JSNP project was started in April 2000 as a two-year collaboration between several Japanese government and academic groups [25]. Its mission was to identify up to 150,000 SNPs in Japanese populations, which were not well represented in the TSC dataset. In contrast to TSC, which took a random shotgun approach across the genome, JSNP focused on gene regions for SNP discovery. This gene focus was very deliberate to try to identify relationships between polymorphisms and common diseases or drug adverse reactions, making this a pharmaceutically relevant dataset. By the end of the project in 2002, 190,562 genetic variations had been discovered and entered into dbSNP. The data are also available, along with details of the project, at the JSNP Web site (http://snp.ims.u-tokyo.ac.jp/).

4.2.5 The HapMap

Following the success of the TSC, JSNP, and of course the human genome consortium, the HapMap emerged as the next logical step in the understanding of these data, effectively bridging the human genome and SNP datasets. The HapMap was born out of some of the technical limitations that hinder the hunt for common disease genes. Despite improvements in technology, initial efforts at genome wide mapping of complex diseases using SNPs have fallen foul of the sheer volume of SNPs required to detect association. Put simply, the human genome contains about 10 million common SNPs (a frequency of > 1%), so finding how these patterns of SNPs differ between diseased individuals and healthy controls is impractical in terms of both cost and DNA resources. The goal of the International HapMap Project is to develop a haplotype map of the human genome, the HapMap, which captures the common patterns of human SNP variation so that a subset of SNPs can be genotyped to capture genomic variation as a whole. To explain the principle behind the HapMap, SNPs that are close together tend to be inherited together. In this context *haplotype* is the term used to describe a set of physically associated SNP alleles in a region of a chromosome. Most chromosome regions are thought to have a limited range of common haplotypes (with a frequency of at least 5%); this accounts for most of the intraindividual variation in a population. Study of haplotype patterns has revealed that a chromosome region may contain many SNPs, but just a few SNPs, known as *Tags*, can provide most of the information on the pattern of genetic variation in the region. Take the following haplotype example.

```
        SNP      1  2  3  4  5  6
Haplotype 1 ..T..g..a..G..c..t..
Haplotype 2 ..T..g..t..C..a..t..
Haplotype 3 ..G..c..t..C..c..a..
```

Three common haplotypes are shown. The two uppercase, underlined SNPs (SNP 1 and 4) are sufficient to identify or tag each of the three haplotypes. For example,

if a chromosome has alleles T and G at these two tag SNPs, then haplotype 1 is likely to be observed. At the simplest level, this allows two SNPs to be genotyped to capture information about six SNPs. On a genome-wide scale, this adds up to considerable economies in genotyping and analysis. In association studies, observation of a haplotype that cosegregates with a disease can reduce an associated region down to 10 to 20 kb (the average size of a haplotype block in European populations). Such a small region usually equates to one or two genes, which makes for a much simpler process of elimination.

The HapMap project aims to describe common haplotypes in three human populations [26]: 90 Utah residents of northern European origin, 90 Yorubans from Nigeria, and 90 Asians (45 Japanese and 45 Chinese). The HapMap will include only relatively common SNPs, found with a frequency of at least 5% in all three populations. The HapMap originally aimed to build the map using 600,000 SNPs, spaced roughly evenly at 5 kb intervals. However, this target is likely to increase, as recent studies by members of the consortium [27] suggest that 600,000 SNPs might not be sufficient. Several studies have now confirmed that Yoruban haplotypes are shorter than those of Europeans and Asians because, like many African populations, Yorubans have evolved over a longer period of time and are more genetically diverse. Evidence so far suggests that the average length of Yoruban haplotypes might be as little as 2 kb to 5 kb. This means that many more SNPs will be needed to effectively analyze African populations.

In terms of drug discovery and development, this also has a direct implication on the way drugs are used across different ethnic groups. There are already many observed differences in drug response between different ethnic groups [28]. These findings suggest that it may not always be possible to extrapolate genetic findings—for example, PGx response alleles, to some key populations with considerable unmet medical needs, such as the African American population. In many cases these populations will need to be studied separately to find population-specific alleles. The HapMap consortium's solution to these problems is to increase the SNP density of the HapMap, and work to accomplish this is ongoing.

4.2.6 Defining Standards for SNP Data

The unprecedented volume of SNP and genotype data that has been generated by the TSC, JSNP, the HapMap, and the human genome sequencing project in general has led to some concerns about the data standards that are applied to this information. The Object Management Group (OMG), a little like the TSC, is another not-for-profit consortium that produces and maintains computer industry specifications to ensure complete interoperability between enterprise applications. The Life Sciences Research group members within the OMG, which represents pharmaceutical companies, academics, and technology vendors, are working together to improve communication and interoperability between SNP databases by setting agreed standards for SNP data storage and exchange. Details of these standards are available at the OMG Web site (http://lsr.omg.org/home.html). As SNP data become increasingly integrated within the fabric of genomic research, these standards are likely to play

a critical role in maintaining efficient communication between life sciences software and databases.

4.3 SECTION 3

4.3.1 Tools for Visualization of Genetic Variation: The Genomic Context

The human genome is the ultimate framework for organization of genetic-variation data, and so genome viewers are also one of the best tools for searching and visualizing polymorphisms. The three main human genome viewers—EnsEMBL, the UCSC Human Genome Browser (UCSC), and the NCBI MapViewer (table 4.1)—all maintain consistently high-quality SNP annotation on the human genome, although none currently consistently annotate mutation data. Most of the information in these viewers overlap, but each contains some different information and interpretation, and so it usually pays to consult at least two viewers, if only for a second opinion. Consultation between viewers is easy as all three now use the same whole genome contig—known as "the golden path"—and so they link directly between viewers to the same golden path coordinates.

User-defined queries with these tools can be based on many variables, DNA accessions, gene symbols, cytobands, or golden path coordinates. This places SNPs and other variants into their full genomic context, giving detailed information on nearby genes, transcripts, and promoters. Ensembl and UCSC both show conservation between human and mouse genomes; UCSC also includes a nice graph of genome sequence conservation between human, chimp, mouse, rat, and chicken (fig. 4.3). This illustration may be particularly useful for identification of SNPs in potentially functional regions, as genome conservation is generally restricted to genes (including undetected genes) and regulatory regions [29]. A major strength of both

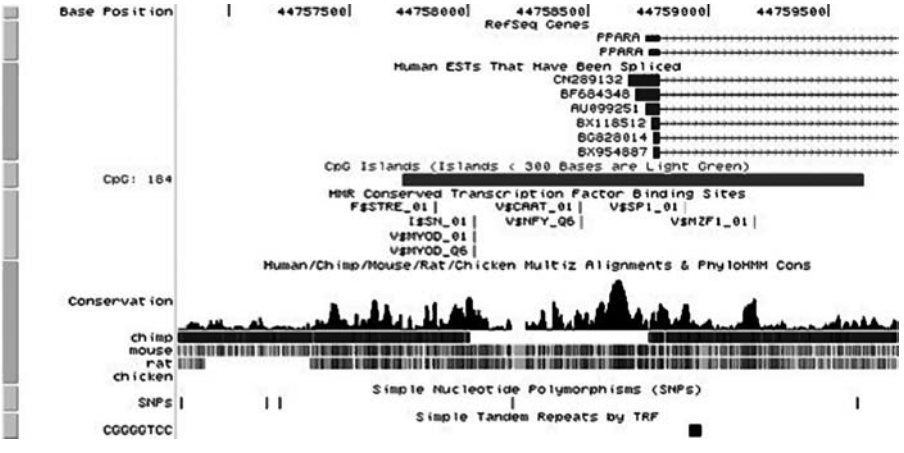


FIGURE 4.3 SNP-to-gene visualization in the UCSC human genome browser. A detailed view of the PPARA gene allows the user to assess the functional context and genomic conservation of the region surrounding each SNP.

the UCSC and Ensembl viewers is their ability to export a range of data across a user-defined locus or a gene, making both viewers among the most flexible interfaces to genetic data. Ensembl is providing perhaps the most flexible tool, EnsMart [30]. EnsMart is a very comprehensive data-mining tool to extract data from the Ensembl database. It provides approximately the same set of data as dbSNP but allows integration with any other genome-mapped feature.

4.3.2 Tools for Visualization of Genetic Variation: The Gene Centric Context

For the purposes of drug-target discovery, validation, and PGx, SNP information is generally of most interest when located in genes or gene regions, where implicitly each SNP can be evaluated for potential impact on gene function or regulation. Many tools are available to identify and analyze such SNPs and most are based on the dbSNP dataset (but it is important to check the version of the data), but most have different approaches to the presentation of data (see table 4.1 for a list of these tools). Choice of tool may be a matter of personal preference, so it is worth taking a look at a few. Some of these tools are maintained by small groups, so sometimes the tools may not be using comprehensive or current datasets, which is a drawback. New tools are constantly appearing in this area, so it is worth running a Web search to look for novel tools; for example, one can enter "SNP AND gene AND database" as search terms to retrieve most new tools in this field.

4.3.3 Entrez Gene and dbSNP Geneview

The NCBI Entrez Gene database is a reliable tool for gene-orientated searching of dbSNP. It can be queried by gene name or symbol; query results show a graphic view of the gene. A click on this graphic will display a report detailing all RefSNP records mapped across the gene in the context of the mRNA and translated sequence. Almost all NCBI tools integrate directly with dbSNP, which also has a "geneview" report. The dbSNP geneview summary details all SNPs across the entire gene locus, including upstream regions, exons, introns, and downstream regions. Nonsynonymous SNPs are identified, and the amino acid change is recorded; analysis also accommodates splice variants. NCBI tools have the advantage of a robust support infrastructure, so they are probably one of the most comprehensive and reliable data sources for gene-orientated SNP information.

Although Entrez Gene and dbSNP benefit from the reliability bestowed by the infrastructure and resources available at the NCBI, several other tools present gene-focused SNP data with much more user-friendly interfaces. There are many tools that fit into this category, some of which are listed in table 4.1, but for the purposes of this chapter a handful of the more outstanding tools are reviewed.

4.3.4 SNPper

SNPper is a Web-based tool developed by the Children's Hospital Informatics Program, Boston [31]. The SNPper tool maps dbSNP refSNPs to known genes, allowing SNP searching by name (e.g., using the dbSNP rs name) or by the golden path

position on the chromosome. Alternatively, you can first find one or more genes you are interested in and find all the SNPs that map across the gene locus, including flanking regions, exons, and introns. SNPper produces an effective gene report that displays SNP positions, alleles, and the genomic sequence surrounding the SNP. It also presents useful text reports that mark up SNPs across the entire genomic sequence of the gene and another report that marks up all the amino-acid-altering SNPs on the protein. The program also has a nice tool for comparing the properties of amino acids, which is valuable for evaluating the possible impact of amino acid substitutions. One of the great strengths of SNPper lies in its data export and manipulation features (an XML export/query tool is even available; see table 4.1). At the SNP report level, SNPs can be sent directly to automatic primer design through a Primer3 interface. At a whole-gene level or even at a locus level, SNP sets can be defined and refined and e-mailed to the user in an Excel spreadsheet. SNPper currently contains information on around 5.8 million unambiguously mapped refSNPs (dbSNP build 118). At the time this chapter was written, dbSNP build 122 was available, which contained 9.8 million refSNPs, so the comprehensiveness of the results returned by SNPper may be an issue.

4.3.5 GeneSNP

The GeneSNPs Web site (http://www.genome.utah.edu/genesnps/) integrates gene, sequence, and SNP data into a carefully annotated subset of human genes of high interest to researchers participating in the Environmental Genome Project. It provides extensive visualization and data export features, including a way of displaying SNPs within the genomic sequence of the gene to which they belong, similar to the one available in SNPper. Its main limitation is that it only contains a relatively small number of genes (584 genes in August 2004), although most of these genes are relevant to drug discovery and development and particularly drug metabolism. Each gene is viewable in a graphical SNPCard, which contains information on the annotated gene model, representative RNA and DNA sequences, and SNPs. Where available, the sequence extends 10 kb, both 5 and 3, of the expressed region of the gene. The location and potential functional impact of each SNP is classified based on the location (coding nonsynonymous, untranslated region [UTR], etc.). SNP allele frequency is clearly indicated on a graph below the gene, making it easy to identify high-frequency SNPs across the gene.

4.3.6 Cancer Genome Annotation Project: Genetic Annotation Initiative

The Cancer Genome Annotation Project (CGAP) genetic annotation initiative (GAI) database (htpp://lpgws.nci.nih.gov/) is a valuable resource that identifies candidate SNPs by *in silico* prediction from alignments of expressed sequence tags (ESTs) [32]. The database was established specifically to mine SNPs from ESTs generated by CGAP's Tumor Gene Index project [33], which is generating more than 10,000 ESTs per week from over 200 tumor cDNA libraries. Candidate SNPs in ESTs can easily be viewed with the CGAP–GAI Web interface in a graphical Java assembly.

SNPs in ESTs are identified by an automated SNP calling algorithm, mining EST data with greater than 10 reads from the same transcribed region yielded predicted SNPs with an 82% confirmation rate [32]. All SNPs that meet the stringent calling criteria are submitted to dbSNP. It is also worthwhile searching CGAP directly if you are interested in a specific gene. The threshold for automated SNP detection is set very high, so many potential SNPs are deliberately excluded by the highly conservative automatic detection process, but these candidate SNPs can be identified quite easily by eye, simply by looking for single base conflicts where sequence is otherwise high quality. The JAVA view of trace data helps to support the base call of a potential SNP in an EST, although laboratory investigation is the only completely reliable SNP confirmation. Intriguingly this resource could potentially contain some somatic mutations from tumor ESTs; these would probably be discarded by the automatic detection algorithm that requires some degree of redundancy to call the SNP.

4.3.7 SNP500Cancer

The SNP500Cancer database (http://snp500cancer.nci.nih.gov/home.cfm) is another CGAP resource that is specifically designed as a resource for applying genetic approaches to understanding the etiology of different cancers as well as related phenotypes. The SNP500Cancer project has resequenced 102 reference samples from four ethnically diverse groups from the Coriell Biorepository (Camden, NJ) in an effort to validate known SNPs of potential importance to molecular epidemiology of cancer. Selection of SNPs for validation was based on review of the literature and input from investigators in the field. SNPs within or closely situated to candidate genes, implicated in one or more cancers, have been targeted. Hence, there is a heavy weighting toward nonsynonymous SNPs. The SNP500 group welcome suggestions for new SNPs to be included in the project, particularly SNPs with known or suspected functional impact.

4.3.8 Comparison of Consistency Across SNP Tools and Databases

Consistency across SNP tools and databases is a real issue that needs to be considered. Savas et al. [34] completed a comparison of most of the tools reviewed previously in this chapter. They searched 88 DNA repair genes for SNPs using dbSNP, HGVbase, CGAP-GAI, SNP500, and GeneSNP. They noticed several problems concerning the specificity and accuracy of SNP positional and functional annotations. They managed to compile 1,000 SNP entries from the 88 genes using the five Web-based SNP resources. Of interest, they found 150 nonsynonymous SNPs (nsSNPs) throughout these genes, four of which were unique to the CGAP–GAI database. Most of the nsSNPs were found in dbSNP (n = 128, 85.3%), GeneSNP (n = 105, 70.0%), and HGVbase (n = 89, 59.3%). Total numbers of nsSNPs retrieved from CGAP–GAI and SNP500 were obviously lower; however, neither tool claims to contain a comprehensive SNP dataset. This clearly illustrates the problem with reliance on a single tool, if a comprehensive view of variation is required.

4.4 SECTION 4

4.4.1 Determining the Impact of a Polymorphism on Gene and Target Function

Genetic variation can impact almost any biological process, hence the scope of analysis required to evaluate the impact of variation is immense. Much of the precedent in the area of functional analysis of variation has focused on the most obvious variation—nonsynonymous changes in genes. Alterations in amino acid sequences have been identified in a great number of diseases, particularly those that show Mendelian inheritance. These identified alterations may reflect the severity of many Mendelian phenotypes, but in the case of complex disease and drug response this is probably not due to an increased likelihood that coding variation changes function but rather a bias in analysis that focuses in functional terms on the "low-hanging fruit"—coding variation. Coding variants may impact protein folding, active sites, protein–protein interactions, protein solubility, or stability. But the effects of DNA polymorphism are by no means restricted to coding regions. Variants in regulatory regions may alter the consensus of transcription factor binding sites or promoter elements; variants in the UTR of mRNA may alter mRNA stability; variants in the introns and silent variants in exons may alter splicing efficiency. Many of these noncoding changes may have an almost imperceivable impact on phenotype, but this may well reflect the nature of complex disease and drug response, where subtle alterations can nonetheless lead to serious phenotypic effects in combination with other factors, such as lifestyle, environment, or simply the passage of time.

Approaches for evaluating the potential functional effects of genetic variation are almost limitless, but there are only a few tools designed specifically for this task. Instead almost any bioinformatics tool that makes a prediction based on a DNA, RNA, or protein sequence can be commandeered to analyze polymorphisms, simply by analyzing both alleles of a variant and looking for an alteration in predicted outcome by the tool (many such tools are listed in table 4.3). Polymorphisms can also be evaluated at a more fundamental level by looking at physical considerations of the properties of genes and proteins, or they can be evaluated in the context of a variant within a family of homologous or orthologous genes or proteins.

4.4.2 Principles of Predictive Functional Analysis of Polymorphisms

The complex arrangements that regulate gene transcription, translation, and protein function are all potential mechanisms through which disease could act, and so analysis of potential disease alleles needs to evaluate almost every eventuality. Figure 4.4 illustrates the logical decision-making process that needs to be applied to the analysis of polymorphisms and mutations. The tools and approaches for the analysis of variation are completely dependent on the location of the variant within a gene or regulatory region. Many of these questions can be answered quickly using genomic viewers such as Ensembl or the UCSC human genome browser. Placing a polymorphism in full genomic context is useful for evaluating variants in terms of

TABLE 4.3
Tools for Functional Analysis of Variation in Genes and Proteins

Transcriptional Start Site and Promoter Prediction	
Promoser	http://biowulf.bu.edu/zlab/PromoSer/
First Exon Finder	http://rulai.cshl.org/tools/FirstEF/
Promoter 2.0	http://www.cbs.dtu.dk/services/Promoter/
Transcription Factor Binding Site Prediction	
ConSite	http://mordor.cgb.ki.se/cgi-bin/CONSITE/consite
TFSEARCH	http://www.cbrc.jp/research/db/TFSEARCH.html
Other DNA and mRNA Regulatory Elements	
UTRdb	http://bighost.area.ba.cnr.it/BIG/UTRHome/
ESE finder	http://exon.cshl.org/ESE/
Rescue ESE	http://genes.mit.edu/burgelab/rescue-ese/
Detection of Novel Regulatory Elements and Comparative Genome Analysis	
PipMaker	http://bio.cse.psu.edu/pipmaker/
TRES	http://bioportal.bic.nus.edu.sg/tres/
Regulatory Vista	http://gsd.lbl.gov/vista/rvista/submit.shtml
Integrated Platforms for Gene, Promoter, and Splice Site Prediction	
Webgene	http://www.itba.mi.cnr.it/webgene/
NNPP, SPLICE, Genie	http://www.fruitfly.org/seq_tools/
Protein Secondary-Structure Prediction	
TMPRED	http://www.ch.embnet.org/software/TMPRED_form.html
TMHMM	http://www.cbs.dtu.dk/services/TMHMM/
PREDICTPROTEIN	http://www.embl-heidelberg.de/predictprotein/
GPCRdb 7TM snake plots	http://www.gpcr.org/7tm/seq/snakes.html
Protein 3D Structure Analysis	
DeepView/Swiss PDB viewer	http://www.expasy.org/spdbv/
Cn3D	http://www.ncbi.nih.gov/Structure/CN3D/cn3d.shtml
Identification of Protein Functional Motifs	
INTERPRO	http://www.ebi.ac.uk/interpro/scan.html
PROSITE	http://www.ebi.ac.uk/searches/prosite.html
SIGNALP, NetPhos, NetOGlyc & NetNGlyc (Signal peptide, phosphorylation and glycosylation site analysis)	http://www.cbs.dtu.dk/services/
Swiss-Prot (Functional annotation)	http://www.expasy.ch/cgi-bin/sprot-search-ful
Amino Acid Properties	
Properties of amino acids	http://www.russell.embl-heidelberg.de/aas/
SIFT	http://blocks.fhcrc.org/sift/SIFT.html
Specific Tools for SNP Functional Analysis	
pupaSNP	http://pupasnp.bioinfo.cnio.es
FastSNP	http://fastsnp.ibms.sinica.edu.tw/fastSNP/index.htm
PolyPhen	http://www.bork.embl-heidelberg.de/PolyPhen/

location within or near genes (exonic, coding, UTR, intronic, promoter region) and other functionally significant features, such as CpG islands, repeat regions, or recombination hotspots. Once approximate localization is achieved, specific questions need to be asked to place the polymorphism in a specific genic or intergenic region. This

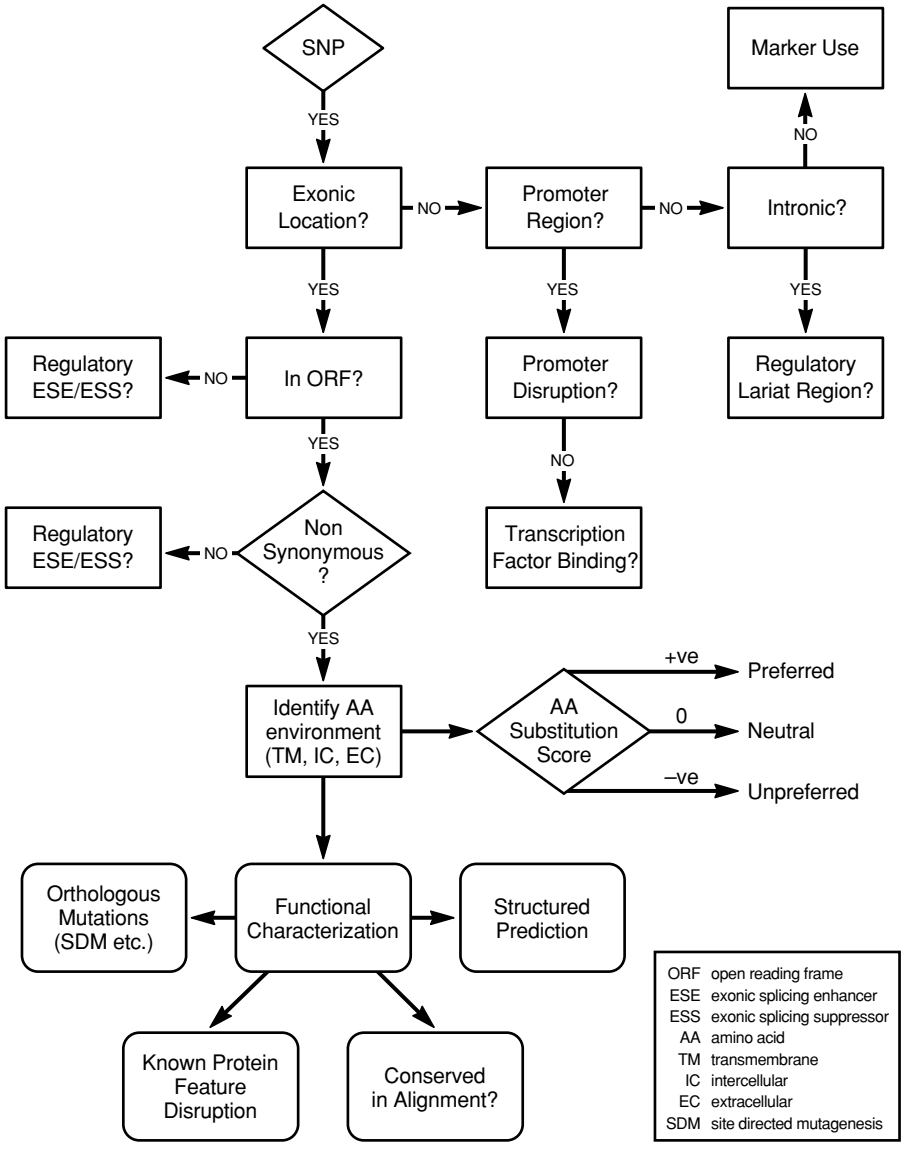


FIGURE 4.4 A decision tree for polymorphism analysis.

questioning will help to narrow down the potential range of functional effects attributable to a variant, which in turn will help identify the appropriate lab follow-up approach to evaluate function.

4.4.3 A Decision Tree for Polymorphism Analysis

The first step in the decision tree for polymorphism analysis (fig. 4.4) is a simple question: is the polymorphism located in an exon? Answering this accurately may

not always be simple or even possible with exclusively *in silico* resources. Delineation of genes is the key step in all subsequent analyses; once the exact location of a gene is known, all other functional elements fall into place based on their location in and around genes. The art of delineating genes to identify the true boundaries of exons may seem superfluous in the "postgenome" era, but we still know very little about the full diversity of genes, and the vast majority of first exons in particular are still incompletely characterized. Gene prediction and gene cloning generally has focused on the open reading frame—the protein coding sequence (ORF/CDS) of genes. For the most part UTR sequences have been neglected in the rush to find an open reading frame (ORF) and a protein. In the case of polymorphism analysis, these sequences should not be overlooked as the extreme 5' and 3' limits of UTR sequence delineate the true boundaries of genes. This delineation of gene boundaries is illustrated in a canonical gene model in figure 4.5. As the model shows, most of the known regulatory elements in genes are localized to specific regions based on the location of the exons. So, for example, the promoter region is generally located in a 1 to 2 kb region immediately upstream of the 5' UTR and splice regulatory elements flank intron–exon boundaries. Many of these regulatory regions were first identified in Mendelian disorders, and now some are also being identified in complex disease phenotypes.

4.4.4 The Anatomy of Promoter Regions and Regulatory Elements

Prediction of eukaryotic promoters from genomic sequence remains one of the most challenging tasks for bioinformatics. The biggest problem is over prediction. Current methods on average will predict promoter elements at 1 kb intervals across a given genomic sequence, in stark contrast to the estimated average 40 to 50 kb distance of functional promoters in the human genome [35]. Although it is possible that some of these predicted promoters may be expressed cryptically, the vast majority of predictions are likely to be false positives. To avoid these false predictions it is essential to provide promoter prediction tools with the appropriate sequence region, that is the region immediately upstream of the gene transcriptional start site (TSS). It is important to define the TSS accurately. It is insufficient to simply take the sequence upstream from the start codon as 5' UTR can often span additional 5' exons in higher eukaryotes [35]. Uwe Ohler [36], a member of the Drosophila genome project, put this succinctly: "Without a clear idea of the TSS location we may well be looking for a needle in the wrong haystack" (p. 539). If the TSS is identified, then the majority of RNA polymerase promoter elements are likely to be located within 150 bp of this site. However, this is not always the case. Some may be more distant, so it may be important to analyze 2 kb or more upstream, particularly when the TSS is not well defined. Promoser is a good tool for identifying TSS sites (Table 4.3, [37]); these are identified computationally by considering alignments of a large number of partial and full-length mRNA and EST sequences to genomic DNA, with provision for alternative promoters.

Once a potential TSS has been identified, there are many tools that can be applied to identify promoter elements and transcription factor binding sites. The UCSC and

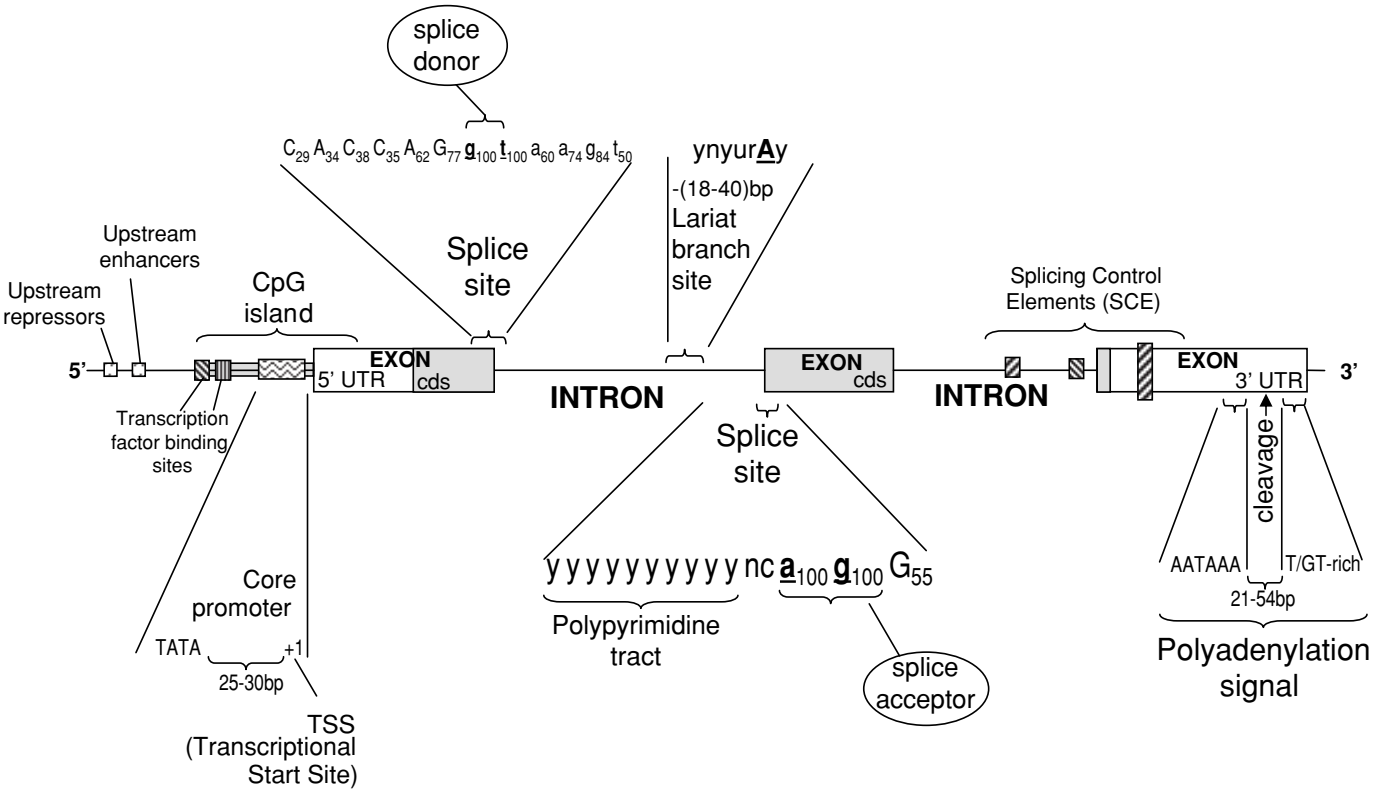


FIGURE 4.5 The anatomy of a gene. This figure illustrates some of the key regulatory regions that control the transcription, splicing, and posttranscriptional processing of genes and transcripts. Polymorphisms in these regions should be investigated for functional effects.

Ensembl genome browsers are the single most valuable resources for the analysis of promoters and regulatory elements. Specifically, Ensembl annotates putative promoter regions using the Eponine tool. The UCSC browser annotates transcription factor binding sites that fall within human/mouse/rat conserved regions (see fig. 4.3). This is a valuable confirmation of potential functional conservation; a binding site is considered to be conserved across the alignment if its score meets the threshold score for that binding site in all three species. These tools are very useful for rapid evaluation of the location of variants in relation to these features, although these data need to be used with caution, as whole-genome analyses may overpredict or overlook evidence for alternative gene models. Characterization of gene promoters and regulatory regions is not only valuable for functional analysis of polymorphisms but can also provide important information about the regulatory cues that govern the expression of a gene, which may be valuable for pathway expansion to assist in the elucidation of the function of disease associated genes and, in the case of drug discovery, expanding a pathway to find a tractable target.

4.4.5 Gene Splicing

Alternative splicing is also an important mechanism for regulation of gene expression, which can expand the coding capacity of a single gene to allow production of different protein isoforms with different functions. Analysis of the human genome gives an interesting insight into this form of gene regulation. Despite initial estimates of a human gene complement of more than 100,000 genes, direct analysis of the sequence suggests that humans may only have 25,000 to 30,000 genes, which is only a two- or threefold gene increase over invertebrates [38]. Indeed, extrapolation of results from an analysis of alternatively spliced transcripts from chromosomes 22 and 19 have led to estimates that at least 59% of human genes are alternatively spliced [39]. This highlights the significance of splicing as an alternative means to express the full phenotypic complexity of vertebrates without the burden of a very large number of genes.

4.4.6 Splicing Mechanisms, Human Disease, and Functional Analysis

Regulation of splicing is mediated by the spliceosome, a complex network of small nuclear ribonucleoprotein (snRNP) complexes and members of the serine/arginine-rich (SR) protein family. In a nutshell, splicing of pre-mRNA involves precise removal of introns to form mature mRNA with an intact ORF. *Correct* splicing requires exon recognition with accurate cleavage and rejoining at the exon boundaries designated by the invariant intronic GT and AG dinucleotides, respectively known as the splice donor and splice acceptor sites (fig. 4.5). Other more variable consensus motifs have been identified in adjacent locations to the donor and acceptor sites, including a weak exonic "CACCAG" consensus flanking the splice donor site, an intronic polypyrimidine (Y: C or T) rich tract flanking the splice acceptor site, and a weakly conserved intronic "YNYURAY" consensus 18 to 40 bp from the acceptor site, which acts as a branch site for lariat formation (fig. 4.5). Other regulatory motifs

are known to be involved in splicing, including exonic splicing enhancers (ESE) and intronic splicing enhancers (ISE), which both promote exon recognition and exonic and intronic splicing silencers (ESS and ISS, respectively), which have an opposite action, inhibiting the recognition of exons. DNA recognition motifs for splicing enhancers and silencers are generally quite degenerate. The degeneracy of these consensus recognition motifs points to fairly promiscuous binding by SR proteins. These interactions can also explain the use of alternative and inefficient splice sites, which may be influenced by competitive binding of SR proteins and hnRNP determined by the relative ratio of hnRNP to SR proteins in the nucleus. A natural stimulus that influences the ratio of these proteins is genotoxic stress, which can lead to the often observed phenomenon of differential splicing in tumors and other disease states [40].

4.4.7 Functional Analysis of Polymorphisms in Putative Splicing Elements

If taken individually, there are many sequences within the human genome that match the consensus motifs for splice sites, but most of them are not used. To function, splice sites need appropriately arranged positive (ESEs and ISEs) and negative (ESSs and ISSs) *cis*-acting sequence elements. These arrangements of regulatory elements can be both activated and deactivated by genetic variants. Polymorphism at the invariant splice acceptor (AG) and donor (GT) sites are generally associated with severe diseases and so are likely to be correspondingly rare (e.g., there are only 134 validated SNPs in splice sites in dbSNP; see table 4.2). But as described earlier, recognition motifs for some of the elements that make up the larger splice site consensus sequence are variable, so splice site prediction from undefined genomic sequence is still imprecise at the best of times. Bioinformatics tools can fare better when applied to known genes with known intron/exon boundaries; this information can be used to carry out reasonably accurate evaluations of the impact of polymorphisms in putative splice regions. There are several tools that will predict the location of splice sites in genomic sequence, and all match and score the query sequence against a probability matrix built from known splice sites (see table 4.3). These tools can be used to evaluate the effect of splice region polymorphisms on the strength of splice site prediction by alternatively running wild-type and mutant alleles. As with any other bioinformatics prediction tool, it is always worth running predictions on other available tools to look for a consensus between different prediction methods.

Splice site prediction tools will generally predict the functional impact of a polymorphism within close vicinity of a splice donor or acceptor site, although they will not predict the functional effect of polymorphisms in other elements, such as lariat branch sites. Definition of consensus motifs for these elements (fig. 4.5) makes it possible to assess the potential functional impact of polymorphisms in these gene regions by simply inspecting the location of a polymorphism in relation to the consensus motif. As with all functional predictions, laboratory investigation is required to confirm the hypothesis.

Other *cis*-regulatory elements, such as ESE, ESS, ISE, and ISS sites, are still relatively poorly defined and may be found in almost any location within exons and

introns (hence, overprediction is a problem). There are currently two tools available to predict the locations of these regulatory elements—ESEfinder and Rescue ESE (table 4.3). Another possible approach for *in silico* analysis of such elements is to use comparative genome data to look for evolutionarily conserved regions, particularly between distant species (e.g., comparison of human/Fugu (fish) genomes). Although there may be some value in these approaches, confirmation of *cis*-regulatory elements needs to be achieved by laboratory methods. (See D'Souza and Schellenberg [41] for a description of such methods.)

4.4.8 Functional Analysis on Nonsynonymous Coding Polymorphisms

Returning to the decision tree for polymorphism analysis (fig. 4.4), the consequences of an amino acid substitution are first and foremost defined by the environment in which the amino acid exists. Different cellular locations can have very different chemical environments, which all have different effects on the properties of amino acids. The cellular location of proteins can be divided at the simplest level between intracellular, extracellular, and transmembrane environments. The latter location is the most complex, as amino acids in transmembrane proteins can be exposed to all three cellular environments, depending on the topology of the protein and the location of the particular amino acid. Environments will also differ in extracellular and intracellular proteins, depending on the location of the residue within the protein. Amino acid residues may be buried in a protein core or exposed on the protein surface. Once the environment of an amino acid has been defined, different matrices are available to evaluate and score amino acid changes. A Web site on the properties of amino acids (table 4.3) provides four amino acid substitution matrices based on the environmental context of an amino acid. These matrices can be used to evaluate amino acid changes in extracellular, intracellular, and transmembrane proteins; where the location of the protein is unknown, a matrix for "all proteins" is also available. Preferred (conservative) substitutions have positive scores, neutral substitutions have a zero score, and unpreferred (nonconservative) substitutions are scored negatively. These matrices are an application of "inverse genetics," constructed by observing the propensity for exchange of one amino acid for another based on comparison of large sets of related proteins. Defining the environment of an amino acid by looking at existing protein annotation, or better still a known tertiary structure, may be relatively straightforward if the protein is known. Beyond the cellular environment of a variant there are many other important characteristics of an amino acid that need to be evaluated. These include the context of an amino acid within known protein features and the conservation of the amino acid position in an alignment of related proteins. Alignment of mutated amino acid sequences with vertebrate and invertebrate orthologues and homologues in a protein family will indicate whether the residue is highly conserved throughout the gene family. Beyond evolutionary clues, there are many different sources of protein annotation and tools to evaluate the impact of substitutions in known and predicted protein features; some of the best are listed in table 4.3. The overriding principle of such an analysis approach is to get to know a protein: first seek known annotation, then seek to annotate where

annotation does not exist, and finally look at the impact of the variant in relation to all that you now know.

4.4.9 Integrated Tools for Functional Analysis of Genetic Variation

4.4.9.1 PupaSNP and FastSNP

Two tools have recently been developed that offer to make the process of polymorphism analysis a little easier. PupaSNP and FastSNP are both integrated platform applications that will analyze all known (in the case of PupaSNP, also user submitted) polymorphisms in a given gene or list of genes. This feature obviously offers great benefits to the user in terms of speed and convenience, and as a first-pass analysis these tools both do a fine job; however, it is worth taking some time to fully explore all avenues of analysis to add to the output of these tools.

PupaSNP [42] (table 4.3) is for high-throughput analysis of SNPs with potential phenotypic effect. The tool takes a list of genes as an input and retrieves SNPs from evolutionarily conserved regions that could impact gene regulation and protein function. PupaSNP is quite comprehensive and uses a range of tools to investigate the impact of SNPs on splice boundaries, exonic splicing enhancers, transcription factor binding sites (TFBS), and changes in amino acids. It also provides additional functional information from gene ontology (a descriptive hierarchy of gene function), OMIM, and model organisms. Fast SNP is also worth a mention (table 4.3). It provides a complete platform for SNP analysis, although the number of analyses performed are slightly reduced, focusing on TFBS and ESE prediction and amino acid substitution analysis.

4.5 CONCLUSIONS

The last few years have revolutionized our knowledge of polymorphism and mutation in the human genome. SNP discovery efforts and processing of genome-sequencing data have yielded several million base positions and several hundred thousand VNTRs that are likely to be polymorphic in the genome. This information is complemented by a more select collection of mutation data painstakingly accumulated over many years of disease gene hunting and mutation analysis. The sheer scale of these data offers tremendous opportunities for drug discovery. We are now entering a new phase in drug discovery and development where experiments are being designed to capture the full genetic diversity of populations. This era may herald a revolution in drug discovery, allowing rapid progression from disease gene to efficacious drug. Alternatively it may simply identify further downstream bottlenecks in the progression to validated drug targets. An understanding of mutation and polymorphism may be an important aid in this process—with mutations representing the extreme boundaries beyond which genes seriously dysfunction and polymorphisms perhaps representing the functional range within which genes can operate. Our knowledge of the breadth and variety of human genetic variation can only increase our understanding of the mechanisms of disease, and more important, it

may help us to define better targets for intervention and ultimately safer and more effective medicines.

REFERENCES

1. Drews, J. 1996. Genomic sciences and the medicine of tomorrow. *Nature Biotechnol* 14:1516–8.
2. Lindpaintner, K. 2003. Pharmacogenetics and the future of medical practice. *J Mol Med* 81:141–53.
3. Roses, A. D. 2002. Genome-based pharmacogenetics and the pharmaceutical industry. *Nat Rev Drug Discov* 1:541–9
4. Majewski, J., and J. Ott. 2002. Distribution and characterization of regulatory elements in the human genome. *Genome Res* 12:1827–36.
5. Lesko, L. J., and J. Woodcock. 2002. Pharmacogenomic-guided drug development: Regulatory perspective. *Pharmacogenomics J* 2:20–4.
6. Altshuler, D., V. J. Pollara, C. R. Cowles, W. J. Van Etten, J. Baldwin, L. Linton, and E. S. Lander. 2000. A SNP map of the human genome generated by reduced representation shotgun sequencing. *Nature* 407:513–6.
7. Breen, G., R. Viknaraja, D. Collier, D. Sinclair, and M. R. Barnes. 2004. Distributions of polymorphic microsatellites in mammalian and other genomes. Manuscript in preparation.
8. Sherry, S. T., M. H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin. 2001. dbSNP: The NCBI database of genetic variation. *Nucleic Acids Res* 29:308–11.
9. Marth, G., R. Yeh, M. Minton, R. Donaldson, Q. Li, S. Duan, R. Davenport, R. D. Miller, and P. Y. Kwok. 2001. Single-nucleotide polymorphisms in the public domain: How useful are they? *Nat Genet* 27:371–2.
10. Miller, R. D., and P. Y. Kwok. 2001. The birth and death of human single-nucleotide polymorphisms: New experimental evidence and implications for human history and medicine. *Hum Mol Genet* 10:2195–8.
11. Debrauwere, H., C. G. Gendrel, S. Lechat, and M. Dutreix. 1997. Differences and similarities between various tandem repeat sequences: Minisatellites and microsatellites. *Biochimie* 79:577–86.
12. Usdin, K., and E. Grabczyk. 2000. DNA repeat expansions and human disease. *Cell Mol Life Sci* 57:914–31.
13. Matsuura, T., T. Yamagata, D. L. Burgess, A. Rasmussen, R. P. Grewal, K. Watase, M. Khajavi, et al. 2000. Large expansion of the ATTCT pentanucleotide repeat in spinocerebellar ataxia type 10. *Nat Genet* 26:191–4.
14. Lucassen, A. M., C. Julier, J. P. Beressi, C. Boitard, P. Froguel, M. Lathrop, and J. I. Bell. 1993. Susceptibility to insulin dependent diabetes mellitus maps to a 4.1 kb segment of DNA spanning the insulin gene and associated VNTR. *Nat Genet* 4:305–10.
15. Le Stunff, C., D. Fallin, N. J. Schork, and P. Bougneres. 2000. The insulin gene VNTR is associated with fasting insulin levels and development of juvenile obesity. *Nat Genet* 26:444–6.
16. Silverman, E. K. H. In, C. Yandava, and J. M. Drazen. 1998. Pharmacogenetics of the 5-lipoxygenase pathway in asthma. *Clin Exp Allergy* 28(Suppl. 5):164–70.
17. Benson, G. 1999. Tandem repeats finder: A program to analyze DNA sequences. *Nucleic Acids Res* 27:573–80.

18. Cambien, F., O. Poirier, L. Lecerf, A. Evans, J. P. Cambou, D. Arveiler, G. Luc, et al. (1992). Deletion polymorphism in the gene for angiotensin-converting enzyme is a potent risk factor for myocardial infarction. *Nature* 359:641–4.
19. Borecki, I. B., and B. K. Suarez. 2001. Linkage and association: Basic concepts. In *Genetic dissection of complex traits,* ed. D. C. Rao and M. A. Province, 45–66. San Diego: Academic Press.
20. Gray, I. C. 2003. From linkage peak to culprit gene: Following up linkage analysis of complex phenotypes with population-based association studies. In *Bioinformatics for geneticists,* ed. M. R. Barnes and I. C. Gray, 165–78. Chichester, UK: Wiley.
21. Hamosh, A., A. F. Scott, J. Amberger, D. Valle, and V. A. McKusick. 2000. Online Mendelian Inheritance in Man (OMIM). *Hum Mutat* 15:57–61.
22. Brookes, A. J., H. Lehväslaiho, M. Siegfried, J. G. Boehm, Y. P. Yuan, C. M. Sarkar, P. Bork, and F. Ortigao. 2000. HGBASE: A database of SNPs and other variations in and around human genes. *Nucleic Acids Res* 28:356–60.
23. Fredman, D., M. Siegfried, Y. P. Yuan, P. Bork, H. Lehvaslaiho, A. J. Brookes. 2002. HGVbase: A human sequence variation database emphasizing data quality and a broad spectrum of data sources. *Nucleic Acids Res* 30:387–91.
24. Thorisson, G. A., and L. D. Stein. 2003. The SNP Consortium Website: Past, present and future. *Nucleic Acids Res* 31:124–7.
25. Haga, H., R. Yamada, Y. Ohnishi, Y. Nakamura, and T. Tanaka. 2002. Gene-based SNP discovery as part of the Japanese Millennium Genome Project: Identification of 190,562 genetic variations in the human genome. *J Hum Genet* 47:605–10.
26. International HapMap Consortium. 2004. Integrating ethics and science in the International HapMap Project. *Nat Rev Genet* 5:467–75.
27. Gabriel, S. B., S. F. Schaffner, H. Nguyen, J. M. Moore, J. Roy, B. Blumenstiel, J. Higgins, et al. 2002. The structure of haplotype blocks in the human genome. *Science* 296:2225–9.
28. McCarthy, L. C., K. J. Davies, and D. A. Campbell. 2002. Pharmacogenetics in diverse ethnic populations—Implications for drug discovery and development. *Pharmacogenomics* 3:493–506.
29. Aparicio, S., A. Morrison, A. Gould, J. Gilthorpe, C. Chaudhuri, P. Rigby, R. Krumlauf, and S. Brenner. 1995. Detecting conserved regulatory elements with the model genome of the Japanese puffer fish, Fugu rubripes. *Proc Natl Acad Sci USA* 92:1684–8.
30. Kasprzyk, A., D. Keefe, D. Smedley, D. London, W. Spooner, C. Melsopp, M. Hammond, P. Rocca-Serra, T. Cox, and E. Birney. 2004. EnsMart: A generic system for fast and flexible access to biological data. *Genome Res* 14:160–9.
31. Riva, A. A., and I. S. Kohane. 2001. A web-based tool to retrieve human genome polymorphisms from public databases. *Proc AMIA Symp* 558–62.
32. Riggins, G. J., and R. L. Strausberg. 2001. Genome and genetic resources from the Cancer Genome Anatomy Project. *Hum Mol Genet* 10:663–7.
33. Strausberg, R. L., K. H. Buetow, M. R. Emmert-Buck, and R. D. Klausner. 2000. The cancer genome anatomy project: Building an annotated gene index. *Trends Genet* 16:103–6.
34. Savas, S., D. Y. Kim, M. F. Ahmad, M. Shariff, and H. Ozcelik. 2004. Identifying functional genetic variants in DNA repair pathway using protein conservation analysis. *Cancer Epidemiol Biomarkers Prev* 13:801–7.
35. Reese, M. G., D. Kulp, H. Tammana, and D. Haussler. Genie—Gene finding in Drosophila melanogaster. *Genome Res* 10:529–38.

36. Ohler, U. 2000. Promoter prediction on a genomic scale: The Adh experience. *Genome Res* 10:539–42.
37. Halees. A. S., D. Leyfer, and Z. Weng. 2003. PromoSer: A large-scale mammalian promoter and transcription start site identification service. *Nucleic Acids Res* 31:3554–9.
38. Aparicio, S. A. 2000. How to count ... human genes. *Nat Genet* 25:129–30.
39. Lander, E. S., L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, et al. 2001. Initial sequencing and analysis of the human genome. *Nature* 409:860–921.
40. Hastings, M. L., and A. R. Krainer. 2001. Pre-mRNA splicing in the new millennium. *Curr Opin Cell Biol* 13:302–9.
41. D'Souza, I., and G. D. Schellenberg. 2000. Determinants of 4-repeat tau expression. Coordination between enhancing and inhibitory splicing sequences for exon 10 inclusion. *J Biol Chem* 275:17700–9.
42. Conde, L., J. M. Vaquerizas, J. Santoyo, F. Al-Shahrour, S. Ruiz-Llorente, M. Robledo, and J. Dopazo. 2004. PupaSNP Finder: A Web tool for finding SNPs with putative effect at transcriptional level. *Nucleic Acids Res* 32:W242–8.

5 Mining of Gene-Expression Data

Aedin Culhane
Dana-Farber Cancer Institute

Alvis Brazma
European Bioinformatics Institute

CONTENTS

5.1 INTRODUCTION

Since microarrays were first described over 10 years ago [1], they have evolved into a standard, though still relatively expensive, experimental technology that has had a profound impact in molecular biology. Microarrays exploit preferential binding of DNA or RNA to their complementary single-stranded sequences. A microarray chip consists of thousands of single-stranded DNA molecules attached in fixed locations onto a solid surface. The microarray chip is incubated with a biological extract (mRNA or cDNA) that is labeled with a fluorescent dye. Thus a single microarray experiment produces thousands of data points, each of which is a measure of the quantity of fluorescent label bound at a feature on the microarray chip. These fluorescent intensities are indirect estimates from which the concentrations of mRNA

molecules are inferred. Tens of thousands of features can be fixed to a single microarray chip, thus providing an opportunity to quantify thousands of gene transcripts, indeed the entire human transcriptome in a single experiment. Knowing the presence or absence of transcripts for all the genes in the genome at a particular moment, and their changes relative to some reference state, is extremely valuable information, and these detailed transcriptomic portraits provide hitherto unimaginable insights into the regulation of biological processes in the cell.

The number of applications of microarrays has increased dramatically. In addition to assessing mRNA abundance, microarrays have also been applied to the quantification of DNA copy number, DNA sequence variations, and protein-binding sites in a genome. Technological advancement means that higher numbers of features can be spotted on microarrays. This has lead to the development of genomic tiling arrays, SNP arrays, microRNA arrays, and all-exon arrays for whole-genome analysis [2,3]. This means that one now has the power to examine protein-coding RNA, alternative splice variants, and nonprotein-coding RNA. The impact of these technologies has been considerable, and a major part of this is due to the efforts of the Microarray Gene Expression Data Society, which had the foresight first to propose and develop standards for microarray data [4] and second to encourage publication of raw data in public data repositories [5]. These repositories, ArrayExpress [6], Gene Expression Omnibus [7], and Stanford Microarray Database [8], now contain thousands of experiments, and the availability of these data is rapidly accelerating the pace of scientific discovery. These repositories contain an impressive and rapidly expanding collection of gene expression data of numerous cell types, in numerous experimental conditions in several species (fig. 5.1), which form an excellent resource for *in silico* drug-target discovery, drug design, and the assessment of drug toxicity. Although there is increasing interest in the application of proteomic, metabolomic, and other new technologies to drug-target identification, gene expression microarrays remain a powerful technology (box 5.1). Microarrays produce an indirect measure of expression, and processing of these data into biologically meaningful units is a nontrivial task. Complex processing normalization and analysis are required to obtain even the most basic information from these data. Numerous bioinformatics, statistical, and machine learning methods have been applied to microarray data. We describe some of these, highlighting the most popular approaches, and describe a typical analyses workflow.

5.2 PREPROCESSING OF MICROARRAY DATA

The goal of microarray data preprocessing is to transform fluorescent signal values into biologically meaningful measurements. Unfortunately, the relationship between the fluorescence intensity and the abundance of a given RNA molecule is not straightforward. Therefore, to correctly estimate the true gene expression level, we need first to assess experimental noise or variance due to random and systematic error [9]. Random errors are statistical fluctuations in the measured data due to precision limitations. These cannot be avoided, but taking the mean of replicates can reduce the effect of these. By contrast, systematic errors are reproducible inaccuracies that are consistently in the same direction (e.g., dye bias between different Cy3 and Cy5 fluorescent dyes, or scanner sensitivity). Systematic errors are often due to

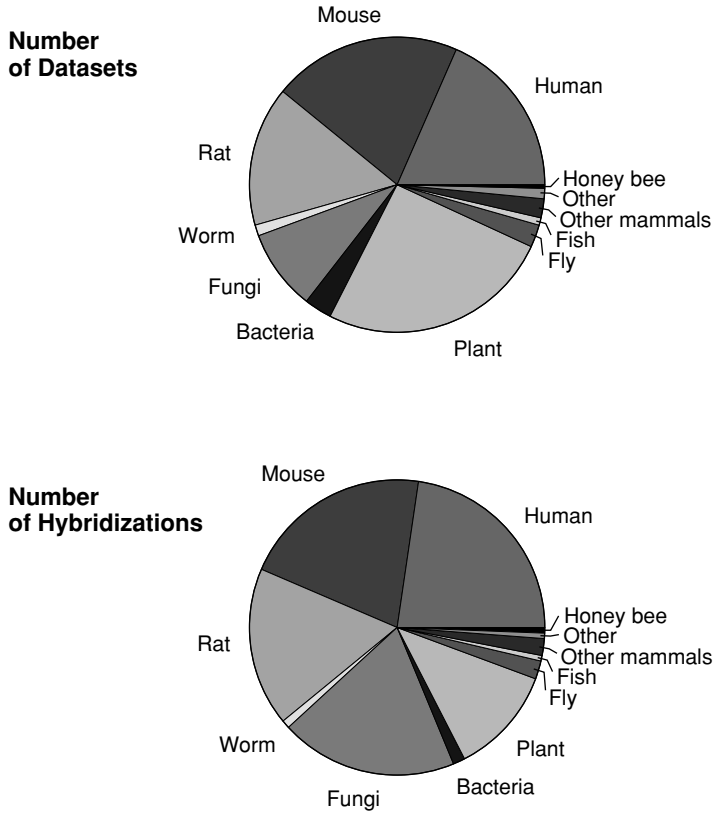


FIGURE 5.1 The organisms represented in 588 microarray experiments (Pie Chart A) or ~17,000 hybridizations (Pie Chart B) in ArrayExpress are shown (H. Parkinson, ArrayExpress Curator, personal communication, April 11, 2005). "Other mammals" includes cat and several primates, and "Other" includes *Methanococcus jannaschii, Plasmodium falciparum, Schistosoma mansoni, Trypanosoma cruzi*. For further information about ArrayExpress, see the database statistics that are available online at http://www.ebi.ac.uk/arrayexpress/Help/stats.

a problem, which persists throughout the entire experiment. Removing systematic noise is particularly critical in the case of microarray data, when one realizes that there are frequently many thousands of variables (genes) and only a few tens of samples. Therefore it is essential that microarray data are preprocessed appropriately to increase the signal-to-noise ratio, prior to data mining of microarray data.

Normalization refers to methods of removing systematic error within a dataset. Several methods have been proposed, and these have been reviewed extensively [9–13]. We provide a brief overview of the most commonly used approaches. The most simple and earliest normalization methods made the assumption that the number of genes up and down regulated would be roughly constant in all samples. These studies simply scaled all arrays so the total sum of intensities was equal on all arrays. Although more refined methods are now used, the assumption that the sum of total gene intensities on arrays is constant remains a central feature of most methods. Most

BOX 5.1
Advantages of Microarrays

Rapid Coverage

Genome-wide association studies of complex genetic diseases can be performed using microarrays. The technology can accommodate high densities of features per slide, making it is possible to study the whole genome in one quick study. Whole genome chips are now widely available from many companies including Affymetrix (Human Genome U133 Plus 2.0 array). By comparison, only a few thousand proteins can be analysed on one 2D gel.

High Resolution

Because the whole genome is on the chip, technically there should be no false negatives. By contrast with proteomics and metabolomics, the number of false negatives is unknown.

Informative

The quantity and quality of genome annotation is improving continuously. The gene and protein sequence, gene structure, genome locus, and often functional, biological action and pathway information is available for each feature (probe set) on a microarray.

Efficient

Comparative genomics is easily accessible. The complete genome of several organisms (human, mouse, and rat) are available on microarray chips; thus, the progression from model organism to clinic is facilitated.

Reproducible

The reliability of the technology has improved enormously in the past two years. Some labs have reported a correlation coefficient of .9 or greater between technical replicates of commercial arrays.

Scalable

Although microarrays are still expensive, the cost of microarrays is no longer prohibitive even for small laboratories. High-throughput microarray screens are financially and technically possible—for example, the study of the gene expression of whole populations or large numbers of people with and without a disease to find potential drug targets.

Standardized Data

Data standards are now in force, and data repositories for microarray data exist.

Computational Tools

Microarrays have a computational head start. Statisticians and computational scientists have developed, applied, and tested numerous methods for microarray analysis. More important, recent publications and meetings such as the Critical Assessment of Microarray Data Analysis [75] have begun to benchmark, critically compare, and assess methods. As a result, methods for microarray analysis

are evolving, and consensuses on the "best" methods for the preprocessing, normalization, and analysis of microarray data are emerging. See table 5.1 for a list of software.

Disadvantages of Microarrays

High Noise Levels

The signal-to-noise ratio is obviously much lower than in DNA sequencing and in molecular structure predictions. There are also few established error models.

Dye Effects

There are direct measures of gene expression, but these tend to be limited by expense. They include TaqMan assays, which can measure the expression of several hundred genes directly, massively parallel signature sequencing (MPSS), expressed sequence tag (EST) counts, and serial analysis of gene expression (SAGE). Microarrays measure gene expression indirectly, and dye effects such as the dye bias of Cy3/Cy5 require *lowess* correction.

Sensitive to Good Reporter (Probe) Design

Intensities are sensitive to the GC content and the DNA sequence of the reporter. Although the whole genome is known, it is still possible probes may cross-hybridise to homologous or unrelated genes.

"Guilt by Association" Theory

The focus of a gene chip experiment is frequently to identity gene expression correlated with a covariant. However, it is difficult to determine if differential expression is a causative. Statistical significance may not equate with biological significance and frequently results in the identification of bystander genes. For example, detection of increased expression of metabolism genes in a rapidly dividing cell may not explain cancer-induction, hence thorough experimental design is crucial.

Increased transcript production as measured by gene chips does not always correlate with production of protein. The complexity of posttranslation modification and regulations of protein pathways are not detected using microarray technology.

Inference of Biological Pathways

It is frequently difficult to infer which biological pathways are activated given a list of differentially expressed genes. The biological outcome of a differentially expressed gene is dependent on the simultaneous activation of many more gene products. But our knowledge of these dependences and interconnections between biological pathways is incomplete.

Cross-Platform Meta Analysis

There is a need for more-refined integration methods for gene-expression data across platforms and for integrations of gene-expression data with other data sources such as protein expression and chemoinformatics data to enable the parallel datamining of "omic" data.

normalization methods also assume that relatively few transcripts (less than 50% of genes on a chip) are being regulated [13–16], though it is believed that these assumptions are generally reasonable. In studies where transcription is heavily influenced, such as in methylation or transcription factor knock-out experiments, these assumptions cannot hold. These assumptions may warrant further consideration, because global changes in gene expression have been reported even in yeast stationary phase [17]. In these cases, normalization to a rank invariant gene set [18] or using external spike in controls is recommended [17].

Different microarray technological platforms produce data in different formats [19]; each have platform-specific errors, which must be processed in slightly different manners. It is beyond the scope of this review to explain these in detail, but we briefly describe the processing of the most commonly used microarray platforms: two-channel microarrays and commercial Affymetrix single-channel oligonucleotide arrays.

The first microarrays were pioneered by Pat Brown and colleagues at Stanford University [1]. These microarrays were generated by robotically spotting PCR or cDNA fragments onto glass slides, and differential expression was measured by means of two-color fluorescent hybridization. This means that slides were incubated with two biological samples, each labeled with a different fluorescent dye, and the ratios between these dyes were measured. Two channel microarrays are popular, are produced both commercially and by laboratories in-house, and make up two-thirds of the published microarray data in ArrayExpress [H. Parkinson, ArrayExpress Curator, personal communication, April 11, 2005]. When normalizing two-channel arrays, a dye basis from the red (Cy5) and green (Cy3) channel needs to be corrected, which is usually achieved using a local weighted linear regression (loess or lowess) curve. Furthermore, correction for within and between print tip effects is normally required [9,13]. These methods are available in many software packages, including the Bioconductor package Limma [20] and the desktop software TM4 [21].

In contrast to dual-channel microarray data, which produce ratios data, single-channel microarray produces measure expression in one sample only. Affymetrix is probably the most popular single-channel oligonucleotide microarray platform. Currently Affymetrix data make up approximately 41% of data in GEO [T. Barrett, GEO Curator, personal communication, March 25, 2005] and just under one-third of the data in ArrayExpress [6]. On Affymetrix microarrays, each gene is measured using several pairs of short oligonucleotide probes [19,22]. In each pair, one probe binds to the target gene, while the second probe contains a single-point mutation (mismatch probe). In theory, this second probe will not bind the target gene and should measure background noise. Therefore, processing of Affymetrix data involves the conversion of these probe data to a single gene expression value, and this is normally achieved using a three-step procedure consisting of (a) background correction, (b) normalization, and (c) summarization of probe set values into one expression measure [23]. In some cases a probe-specific background correction (subtracting mismatch value) is included before Step 3. Several methods are commonly used, of which robust multichip average (RMA) or a variant of RMA called gcRMA that accounts for percentage of GC content in the mismatch probes have outperformed other approaches in com-

petitive tests [15,23]. These methods are available within the Bioconductor package [24] and other microarray analysis packages (see table 5.1).

Microarray measurements usually are log transformed. There are at least two reasons why log transformations are advantageous. First, log transformations make ratios of gene expression increase and decrease symmetrically (e.g., a twofold increase or a decrease by one-half are equal to base 2 logarithms of +1 or –1, respectively). The second reason is that increases in gene expression are generally perceived to be exponential rather than linear. A linear change is one that increases by a fixed increment in each period, whereas an exponential increase is a fixed percentage of the previous total. We use the base 2 log transformation, as biologists generally express gene expression as a fold increase. However, the use of a log transformation analysis is limited [25]. Moreover, studies show that microarray data variance is linear (additive) at low levels of expression but is multiplicative at high levels of expression. Thus, other transformations, such as *arsinh* as implemented in variance stabilizing normalization (vsn), may be more appropriate [14]. Visual plots such as histograms or boxplots are useful in data quality assessment [9]. Other useful plots are MvA or RvI plots. These show intensity-dependent effects in two microarray samples. These could be data from two dye channels (Cy3, Cy5) of dual-channel data or a pair of replicates in the case of single-channel data. The plot shows the ranked intensity (I) or average ratio (A) on the horizontal axis and the fold difference (M) or ratio (R) of these on the vertical axis. In addition, self–self plots have proved to be useful in the assessment of the success normalization methods to remove systematic variance. In an ideal self–self plot, the slope of the line should be 1 and there should be minimum deviation from this line.

5.3 STATISTICAL ANALYSIS OF MICROARRAY DATA

A whole spectrum of statistical techniques have been applied to the analysis of DNA microarray data [26–28]. These include clustering analysis (hierarchical, K-means, self-organizing maps), dimension reduction (singular value decomposition, principal component analysis, multidimensional scaling, or correspondence analysis), and supervised classification (support vector machines, artificial neural networks, discriminant methods, or between-group analysis) methods. More recently, a number of Bayesian and other probabilistic approaches have been employed in the analysis of DNA microarray data [11]. Generally, the first phase of microarray data analysis is exploratory data analysis.

5.4 EXPLORATORY ANALYSIS

Exploratory methods are used not to test hypotheses but rather to get an overview of data. Various clustering methods and ordination are excellent tools for exploratory analysis of microarray data. These unsupervised methods do not require external class or group information. Clusters are generated purely based on the intrinsic similarity of the gene or sample expression profiles. No null hypothesis can be rejected, and p values are not generated to test statistical significance. Methods that

TABLE 5.1
A Selection of Free or Open Source Software Packages

Package	Description	URL
RMAExpress	Desktop program to compute gene-expression values for Affymetrix data using Robust Multichip Average (RMA) method	http://stat-www.berkeley.edu/users/bolstad/RMAExpress/RMAExpress.html
dChip	Win program. Computes Li and Wong [18] model based gene values and normalized Affymetrix data using invariant gene set	http://www.dchip.org/
TM4	Extensive suite of desktop packages. Consists of four applications: MADAM (a data manager), SpotFinder (image analysis), MIDAS, and MeV (data analysis) [21]	http://www.tm4.org/
Expression Profiler	An online-based microarray data analysis tool provided by the European Bioinformatics Institute [35]	www.ebi.ac.uk/expressionprofiler
Bioconductor	Statistical package written in R (see Bioconductors Packages section)	http://www.bioconductor.org
Cluster, TREEView	Popular hierarchical clustering analysis programs [30]	http://rana.lbl.gov/ A Java version of cluster is available at http://bonsai.ims.u-tokyo.ac.jp/~mdehoon/software/cluster
GeneCluster, GenePattern	Clustering using K-Nearest Neighbours, SOM, and more	http://www.broad.mit.edu/cancer/software/software.html
SAM	Gene ranking using significance analysis of microarray	http://www-stat.stanford.edu/~tibs/SAM/
ade4	Multivariate analysis in R	http://pbil.univ-lyon1.fr/ADE-4/
GoMiner, MatchMiner	Gene Ontology tools from the genomics and bioinformatics groups at the National Cancer Institute	http://discover.nci.nih.gov/tools.jsp
SOURCE	Extract gene information, particularly useful for analysis of IMAGE ID clones	http://source.stanford.edu
GenMAPP	Useful for pathway analysis	http://www.genmapp.org/
NetAffx	Online tools at Affymetrix Web site	http://www.affymetrix.com/analysis/
BASE	A MIAME-compliant microarray database	http://base.thep.lu.se/

Bioconductor Packages[a]

ArrayMagic	Input and Processing of dual-channel cDNA arrays
Affy	Processing of Affymetrix data. Affy can call expression values using MAS5.0, RMA, gcRMA, or Li and Wong methods. It contains an extensive number of normalization procedures, including vsn
limma	Input and normalization of dual-channel data. Extensive number of functions. Performs linear models, gene selection of both Affymetrix and cDNA arrays
vsn	Normalization of both Affymetrix and cDNA arrays
made4	Made4 calls ade4 for multivariate analysis of microarray data

Selected Commercial Software Packages

GeneChip Operating Software	GCOS is Affymetrix's new software program for basic analysis	http://www.affymetrix.com/products/software/
GeneSpring	Popular microarray analysis suite from Silicon Genetics	http://www.silicongenetics.com
GenePix	Image analysis of spotted arrays	http://www.axon.com
J-Express	Desktop package for microarray analysis [Dysvik and Jonassen 2001]	http://www.molmine.com/index.asp

Useful Web Resources Links

A beginner's guide to microarray from the National Center for Biotechnology Information	http://www.ncbi.nlm.nih.gov/About/primer/microarrays.html
Microarray Gene Expression Data Society (MGED)	http://www.mged.org/
ArrayExpress	http://www.ebi.ac.uk/arrayexpress/
Gene Expression Omnibus	http://www.ncbi.nlm.nih.gov/geo/
Stanford Microarray Database	http://genome-www.stanford.edu/microarray/
Emmanuel Paradis's R for Beginners: Useful to those unfamiliar with R	http://cran.r-project.org/doc/contrib/rdebuts_en.pdf
Resource in basic statistics	http://www.stats-consult.com/tutorials.html
Mark Reimers' guide to microarray analysis	http://discover.nci.nih.gov/microarrayAnalysis/Microarray.Home.jsp

[a] R and Bioconductor contain hundreds of useful statistical and machine learning packages this is only a selection of those. A list of Bioconductor packages is available at http://www.bioconductor.org.

can be used to test a hypothesis are discussed in section 5.5. Unsupervised methods provide a useful view of groups or trends in the data, and it can be useful to check if samples group in clusters associated with a covariant of interest. Of course, unexpected or unknown associations in the data may be discovered. This may be biologically interesting; however, samples frequently cluster by experiment date, labeling kit, technician, or quality of RNA, suggesting that additional normalization and/or samples replicates are required prior to further analysis.

Clustering is a well-established field, and various clustering algorithms have been applied to microarray data. Clustering may impose a hierarchical or flat structure on the data, which may be built using divisive (top-down) or agglomerative (bottom-up) approaches. For instance, *hierarchical agglomerative clustering* is based on iteratively grouping together the objects that are most similar, in a process that starts with all objects in individual clusters, and successively fusing these. *K-means clustering* [29] is the most common method of flat-partition-based clustering. In K-means clustering, the number of expected clusters (K) is set *a priori* either randomly or by the user. For each *a priori* chosen cluster, the algorithm calculates the distance between each object to its gravity center. It excludes all elements that are closer to the gravity center of some other cluster (and includes them in the respective cluster). For each new cluster the new gravity centers are found. The algorithim is iterated until either all elements in all clusters are closer to their respective gravity centers than any other one, or after some predefined number of iteration (e.g., after 10,000 iterations). It is important to remember that the results obtained from a clustering method are dependent on the measure of distance or similarity used. Thus, the understanding of a basic distance or similarity measure is important.

5.4.1 Distance Measures

A fundamental concept in clustering is the understanding of the importance of the metric, which defines the similarity (or distance) between samples. There are numerous metrics, and depending on the metric selected, results of analyses may vary dramatically. The choice of metric normally depends on the question that is being asked of the data.

The most commonly used distance metric is *Euclidean distance*. The Euclidean distance between two points is the everyday distance we measure with a ruler or measuring tape. It can be calculated easily using Pythagoras's theorem and is the square root of the sum of the square distances between the points in each dimension.

Frequently one seeks genes that are coregulated. In this case the timing of change is more important than the magnitude of change. For example, you seek two genes with the same cycle, or genes that change expression at the one time. In this case a correlation measure, such as a Pearson correlation coefficient or a Spearman rank correlation, may be more appropriate. Alternatively, different metrics may be used if you seek genes that have a phase shift between them, such as genes that are switched on or off by a promoter. The choice of metric is discussed in detail by Causton et al. [28].

5.4.2 Interpretation of Hierarchical Clustering Dendrograms and Eisen Heatmaps

Hierarchical clustering is a concept that is familiar to most biologists because it has been used extensively to develop taxonomies. It was one of the first clustering methods applied to microarray gene expression data [30,31] and remains a popular approach. Several similar or distance metrics, together with several linkage methods, can be applied to hierarchical clustering, and each will produce quite different results. In microarray analysis, a correlation coefficient metric with average linkage is most commonly used. We refer the reader to several excellent reviews and tutorials on hierarchical clustering that discuss these in more detail [28,32,33]. In this review, we provide a few tips on the interpretation of the results from these analyses.

The results of hierarchical clustering analysis are usually visualized using a dendrogram, the treelike diagram. A node joins two objects, and the line length between two nodes is proportional to the similarity or dissimilarity between these objects, depending on the distance metric and linkage method used. Thus, given a scale (fig. 5.2), one can read the distance between two nodes. Although one may be tempted, note that in most analyses, the order of samples is not important. A dendrogram can be compared to a baby's mobile hanging from the ceiling. The relationships between the parts of the mobile do not change even if the parts are rotated. Similarly, in a dendrogram, each node can be rotated freely around each internal branch on the tree without affecting the topology of the tree. For instance, even though the branch order is reordered, Tree A is equivalent to Tree B in figure 5.2, and the distance between the objects of the tree remains constant.

Eisen et al. [30] presented the results of clustering (dendrogram), together with a *heatmap* of gene expression values. A heatmap is a representation of normalized gene expression values, where the number of rows in the heatmap are equal the number of genes and the number of columns are equal to the number of samples.

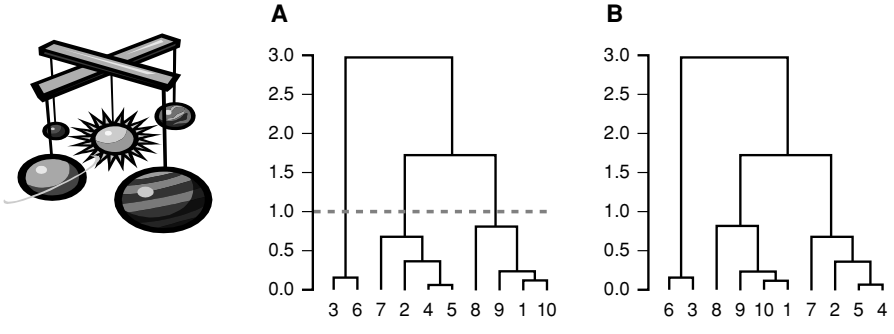


FIGURE 5.2 Trees A and B show the results of a hierarchical cluster analysis of the same data. Tree B has been reordered**.** Dendrogram A is equivalent to Dendrogram B. Rotating an internal branch does not affect the topology of a tree. If Tree A or Tree B were cut at the position of the dashed line (a height of 1.0), each tree would produce three equivalent clusters. That is, the members in each of these three branches are the same.

Each "pixel" in the heatmap is colored, where a color gradient scale is used to represent gene expression intensity. Typically green to red, or more recently blue to yellow, is used to represent increasing gene expression. These plots are effective visualizations and have been widely used in microarray literature. These plots are often referred to as Eisen plots.

To interpret the dendrogram, one must decide how many branches are robust. Generally only the upper branches of a tree are robust. The deep, short branches of a tree are often weak, and sample membership of deep branches may be quite random. The decision of where to cut a tree is critical. For example, if one cuts the Tree A in figure 5.2 at a distance of 1.0, there are three clusters; if one cuts the tree deeper, there are more clusters. In phylogenetic analysis, bootstrap permutation is frequently used to estimate the robustness of branches; however, this has not been adopted widely within the microarray community. Typically dendrograms from hierarchical clustering of microarray data are cut to give clusters that appear tight or that include genes with similar functional annotation. It is easy to see that the interpretation of a dendrogram is subjective.

5.4.2.1 Assumptions and Limitations of Clustering

The limitations and assumptions inherent in hierarchical clustering need to be considered when interpreting a tree. The use of a different metric, linkage, or data normalization method will produce a different tree, and almost always a plausible biological hypothesis can always be forced on it. Thus, when performing hierarchical cluster analysis, it is worth remembering the first tree is not the "one and only" tree. For example, numerous different animal taxonomies have been produced in phylogenetic analyses, but we still await a consensus on the correct model. Second, hierarchical clustering forces a hierarchical topology on data, which may not be appropriate. Biologically a gene (e.g., a kinase) could belong to numerous clusters, but in a dendrogram a gene can belong to one cluster. Equally partitioning experimental samples into discrete hierarchal groups has the disadvantage that it may force arbitrary artificial divisions in a dataset even if it is naturally a continuous gradient (e.g., dose response).

The use of different clustering algorithms or different parameters often produces rather different results on the same data. Biological interpretation of clustering results requires understanding how different clusters relate to each other. It is particularly nontrivial to compare the results of a hierarchical and a flat (e.g., K-means) clustering. To this end, a method for comparing and visualizing relationships between different clustering results, which can be either flat versus flat or flat versus hierarchical, has been recently described [34]. When comparing a flat to a hierarchical clustering, the algorithm cuts different branches in the dendrogram at different levels to optimize the correspondence between the obtained clusters based on graph layout aesthetics or on mutual information. The clusters are displayed using a bipartite graph where the edges are weighted proportionally to the number of common elements in the two clusters and the number of weighted crossings is minimized. The algorithm is available online in the gene expression data analysis tool, Expression Profiler [35].

It is also worthwhile using additional exploratory analysis methods. An ordination method such as principal component analysis (PCA) is a complement to hierarchical clustering.

5.4.3 Ordination: Visualization in a Reduced Dimension

Ordination is a different but complementary approach to clustering, because ordination considers the variability of the whole data matrix, bringing out general gradients or trends in the data, whereas clustering investigates pairwise distances among objects, looking for fine relationships [33]. Factor analysis, PCA, correspondence analysis (COA), and nonmetric multidimensional scaling (MDS) are ordination methods. PCA and COA have been applied to microarray data analysis [36–38]. PCA and COA can be computed using singular value decomposition (SVD), [39], and thus all of these methods are closed related.

The idea behind PCA and ordination is quite intuitive. Figure 5.3 graphically explains the principal of dimension reduction to those unfamiliar with the approach. This example is taken from the extensive literature available with the ADE4 package for multivariate analysis of ecological data [40]. In this example, the morphological measures (length, height, maximum width) and sex (male/female) of 48 painted turtles were recorded [41]. It is clear that there is a linear corelationship between these, and one can see quite easily that they could be represented on one axis (turtle size = x.length + y.height + z.width, where xyz are the weights or loadings that explain importance of each variable in the equation). This is exactly the role of dimension reduction: collinear or correlated variables are represented by a new axis. It can be viewed as a rotation of the existing axes to new positions in the space. In this new rotation, there will be no correlation between axes, the new axes are orthogonal. These new axes are called *principal components* or *eigenvectors* and explain the principal trends in the data. One can easily imagine that within a microarray experiment, many genes are coregulated, and these could be represented by a few principal components that would explain the main patterns of gene expression. For example, in a simple study of control versus treated, one or two principal components might significantly represent the major trends (expressed in control versus expressed in treated). However, if the experiment was a complicated time course, or diagnosis of multiple cancer types, more principal components would be require to explain the main trends in the data.

Each principal component has an associated eigenvalue, which scores the amount of variance or information represented by that component. In PCA and COA, the eigenvalues are ranked from highest to lowest. As a result, the first eigenvalue is the largest, and the first principal component will always describe the strongest trend in these data. The second eigenvalue will be the next largest, and so on. For example, in the turtle data (fig. 5.3), the first principal components represents 97% and thus explains most of the variance in the dataset (fig. 5.3E). One must determine how many eigenvalues or principal components are biologically meaningful. This number will determine the dimensionality of the reduced space. Luckily the problem of choosing the number component to use is simplified by the use of a scree plot (fig. 5.3E). The term *scree* comes from geology, as the plot resembles a mountainside with loose rocks at

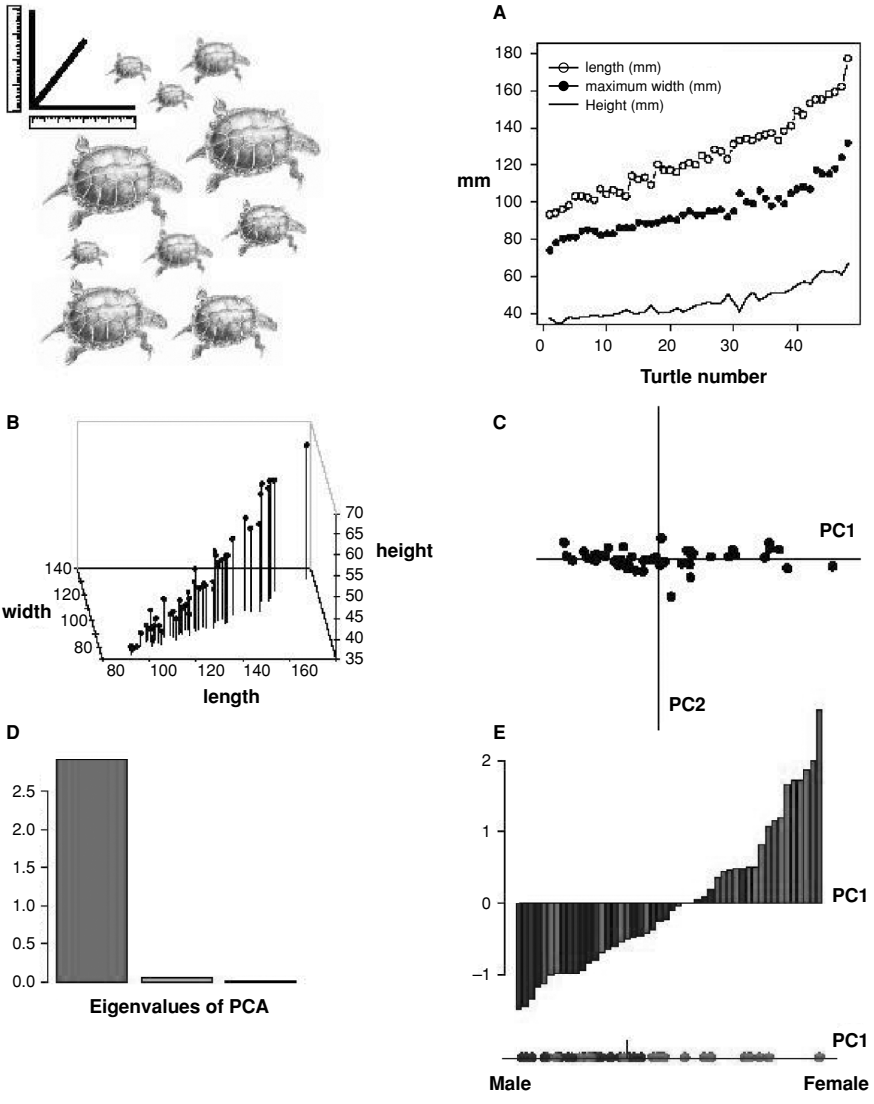


FIGURE 5.3 The concept of dimension reduction, Principal Component Analysis. The length (mm), height (mm), maximum width (mm), and sex (male/female) of 48 painted turtles were recorded [41]. The data were ranked by the product of the length, height, and width. Turtle length, height, and width are shown in Graphs A and B; it is clear that the tree variables could be represented on one axis (turtle size = x.length + y.height + z.width). This is exactly the role of dimension reduction: collinear variables are regressed onto a new axis. PCA produces new axes that represent trends in the data. Graph C shows the first two axes (PC1, PC2) of a PCA. D shows a scree plot of the eigenvalues, which describes the amount of variance (information) represented by each axes. It is clear that the first axis (PC1) is the most important. PC1 accounts for 97% of the variance in the data. E shows the loadings on PC1. There is a clear split between male (blue) and female (magenta) turtles. In fact, further analysis of the second axis (PC2) shows that females tend to be higher and narrower than males of the same length.

the base. One guide to selecting a cutoff is to select the number of eigenvalues before the plot levels off, or reaches the base of the cliff on the scree plot.

Of course, each trend or principal component can be expressed in terms of genes or microarrays. For example, given gene-expression profiles of different tissues such as the gene-expression atlas [42], one might see a principal component showing that immune cells are characterized by high levels of expression of cytokine genes. Equally, one could view the same component from the viewpoint of the genes. Thus, this component would weight cells of immune origin highly but possibly give low weights to liver, kidney, and testis. Therefore PCA analysis results in two plots, one that shows the weighting of genes in array space and a second that shows the weighting of microarrays in gene space. In the microarray community, sometimes these principal components are called *eigengenes* and *eigenarrays,* respectively [39].

It is easy to see that PCA is a useful tool in microarray analysis. These methods date back to Karl Pearson's elegant paper in 1901, in which he posed the problem of finding lines and planes of closest fit to a cloud of points in multidimensional Euclidean space [43]. Given this long history, the terminology may be a challenge to persons new to the field. PCA has been mathematically described several times and can be computed in several ways [33,44], including using eigenanalysis and singular value decomposition. As a result principal components can be known as *eigenvectors* or *singular vectors*. But also principal components are also sometimes referred to as *principal factors, principal axes,* or *latent variables*. A latent variable could be described as a variable that cannot be measured directly but underlies the observed variables. Moreover, as just described, the terms *eigenarrays* and *eigengenes* or *metagenes* are used in microarray analysis [39,45].

To help in understanding the relationships between different ordination methods, it is useful to learn a little about the mathematics behind them. Although we could provide a lengthy computation, we use a nice matrix computation called SVD, which when performed on a matrix results in three new matrices (fig. 5.4). These three new matrices are the singular values (S) and the left (U) and right (V) singular vectors. The singular values are the eigenvalues. The left eigenvectors produce the new loadings and coordinates of variables (genes). The right eigenvectors produce the new loadings and coordinates of cases (microarray samples). Thus each principal component has an eigenvalue, a vector of gene coordinates, and a vector of array coordinates. It is easy to see that the total number of principal components must equal the number of rows or columns in the matrix (which is less). Because the first few components will explain the majority of the variance in the data, the original dimension of the data is transformed into just a few principal components. This is referred to as a *dimension reduction*.

COA, PCA, and many other ordinations can be viewed as matrix decomposition (SVD) following transformation of the data matrix (fig. 5.4). Transformations can include centering with respect to variable means, normalization of variables, square root, and logarithmic transforms. In each case, the transformation modifies the view of the data, and thus different questions are posed. PCA is typically a decomposition of a column mean centered (covariance matrix). That is, the mean of each column (array) is subtracted from each individual gene-expression value before SVD. For more information, see Wall [46], where the mathematical relation between PCA and

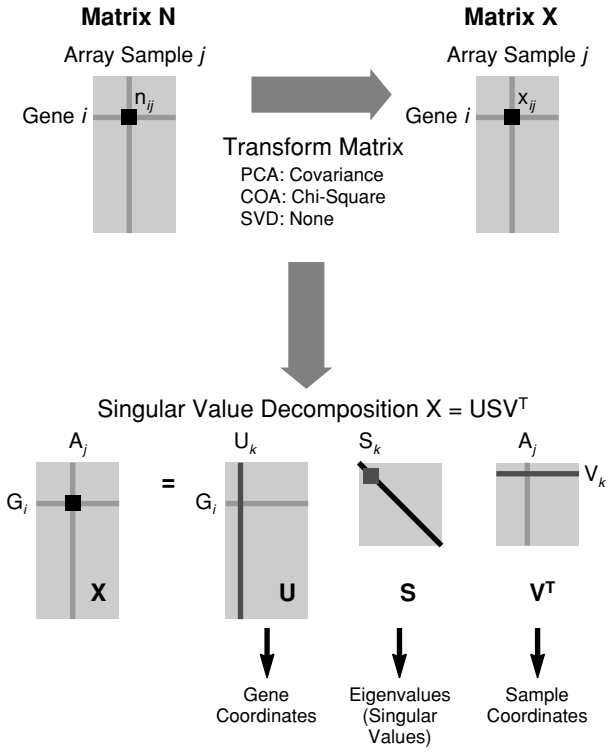


FIGURE 5.4 (See color insert following page 306) Data matrix transformations in advance of SVD. A gene expression matrix N will be transformed to obtain a transform matrix X, which is the one that will be actually displayed in a reduced dimension, achieved using SVD. In this figure, U_k and V_k are eigenvectors and S_k is an eigenvalue. U_k is an eigenassay (or eigenarray) and V_k is an eigengene.

SVD is explained relatively simply. It is worth noting that some preprocessing, such as centering of the data prior to analysis, will impact on results. For example, the first component of a PCA on the yeast sporulation data was uninteresting [37], however structure was revealed when SVD was applied to data that were centered and normalized [47]. A size effect is easily removed by row-centering microarray data prior to PCA. Alternatively a dual scaling approach such as used by COA is effective when analyzing microarray data [36].

5.4.3.1 Interpretation of Plots from PCA or COA

Results of ordination analysis are visualized using two-dimensional plots of the new axes. Although sometimes each axis can be viewed independently, or three axes can be viewed on a three-dimensional plot, typically a two-dimensional scatter plot is most informative. Interpretation of ordination scatter plots is quite intuitive, but we provide a few guidelines. Each PCA and COA principal component (PC) is decorrelated (*orthogonal*), and thus each axis represents an independent or different trend

in the data. In most ordination techniques (but not MDS or independent component analysis), the axes are ranked, thus PC1 accounts for more variability than PC2. Typically the first component (PC1) is represented on the horizontal axis, and the second (PC2) is on the vertical axis. The further a variable or sample (case) is projected from the origin, the greater its importance (or loading). However, the direction of the axes (e.g., positive or negative end) is arbitrary, and the gradient along the axis is more important.

In PCA, components are associated with maximal variance directions. A fundamental assumption of PCA is that the variables are linearly related and variables are measured on the same scale. In the case where variables are measured on different scales, normalized PCA, where values are column mean centered and also divided by the column standard deviation prior to decomposition, must be used. Although most software packages only provide these two PCA options (same scale: centered PCA; different scales: normalized), there are in fact several other options with PCA, and confusion can frequently arise from the use for the same terminology (PCA) for each option. PCA has problems with data with many zeros in them. Interpretation of PCA of microarray data is sometimes difficult, because much of the variance may not be associated with covariates or sample classes of interest. Thus, from a biological point of view, it is worth examining the variance associated with each axis carefully (fig. 5.5).

COA has been successfully applied to microarray data [36,48]. Traditionally COA is a technique for analysis of two-way contingency tables of whole-number positive integers and has been widely used in the analysis of species data in ecological statistics. COA can accommodate species abundance data that are unimodally not linearly distributed, and contain high number of zeros. Up to 50% of species data can be zeros. In COA the data are scaled so that rows and columns are treated equivalently by transforming the data into chi-square values. The sum of the eigenvalues equals the sum of the chi-square value for the data set; each of the axes represent a proportion of the total chi-square for the matrix. The chi-square distance measures the square differences between the observed data points divided by an *expected* or average value. The expected value is the product of the average row and column weight for that data point. This metric tends to equalize the contributions of rare and frequent groups. Thus COA is powerful in analysis of microarray data as it measures the correspondence or strength of association between a variable and a case. As a result, coordinates of genes and samples from a COA are often plotted on the one plot (or a biplot) on which associations between cases and variables are easy to visualize. Samples and genes, which are strongly associated, will lie in a similar direction from the origin.

A biologist's insight, experience, and knowledge of the literature are the most important tools for interpreting these exploratory ordination analyses. Figure 5.5 shows an example of results from a PCA and COA analysis of the same data. As shown, both methods produce different results and ask different questions of the data. Although initial exploration of the data using clustering and ordination approaches is essential, it is worth remembering that exploratory methods do not test any statistical hypothesis. Thus, statistical methods to rank genes are necessary to detect the genes most associated with a covariant of interest, and more sophisticated supervised

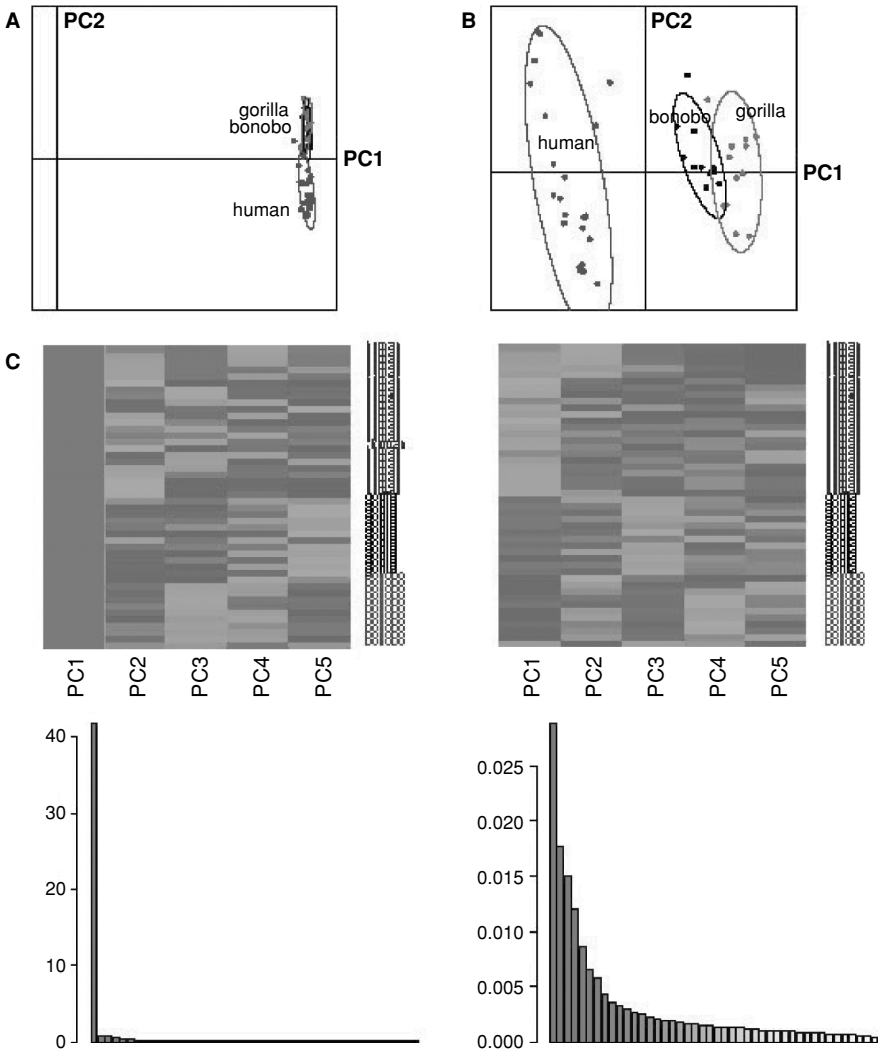


FIGURE 5.5 (See color insert) PCA and COA analysis of gene expression data (12,625 probesets) from fibroblasts isolated from human, gorilla, and bonobo. These figures demonstrate the difference between a PCA and COA, which ask different questions of data. PCA presents the trends in the data with the most variance. A and B show a scatter plot of the first two principal components (PC1, PC2) of a PCA and a COA, respectively. C shows a heatmap of the scores' first five principal components, where red to green is positive-to-negative ends of the axis from the PCA. It is clear that the first component, which represents the 90% of variance in the data, is not associated with samples groupings. In B, the strongest correspondences between genes and samples are analyzed. PC1 represent 19% and PC2 represents 12% of the total chi-square association between samples and gene expression profiles. In COA high chi-squares will be associated with increased gene expression in samples. Thus, if a gene is increased in expression in a many samples, there will be a high chi-square value showing this association. On the scatter plot the positive end of PC1 represents genes that are up-regulated in gorilla and down-regulated in human.

analysis methods are required if we wish to build gene classifiers that predict the class of samples.

5.5 SUPERVISED CLASSIFICATION AND CLASS PREDICTION

Supervised analysis methods use our knowledge about genes or experimental conditions in the analysis. We specify groups in advance. These groups may be sample (patient) information or may be derived from clusters observed in exploratory data analysis. Supervised methods use this prior class information and attempt to construct classifiers based on these predefined classes and corresponding gene-expression profiles. These classifiers form the best discriminators of the groups and can be used to predict the class of future unknown samples. Supervised analysis methods include *linear regression or linear discriminant analysis, support vector machines, artificial neural networks, nearest neighborhood analysis,* and *decision trees* [49,50].

Although almost any supervised or machine learning approach could be applied, in practice supervised analysis is limited by the numbers of samples available. To perform supervised analysis, sufficient samples are required to form both training and test datasets. A classifier is produced using a training dataset. This classifier must be rigorously tested using cross-validation. There are two commonly used cross-validation approaches. The first is a leave-one-out or jackknife approach. To perform this, a sample is removed from the training dataset. The classifier is trained, and the classification of the excluded sample is predicted. This process is repeated until all samples in the dataset have been tested and the percentage of samples that were accurately predicted can be calculated. Generally, leave-one-out approaches overestimate the accuracy of a classifier. Thus, a second approach of cross-validation using a new independent dataset is recommended, particularly in light of the high levels of noise inherent in microarray data, where it would be easy to overfit a classifier to data. There are many reports of insufficient cross-validation, and this has been the subject of much criticism [49]. Other reports have criticized bias in selection of samples in test datasets [51]. Ein-Dor [51] analyzed the van't Veer et al. [52] breast cancer dataset and concluded that the gene signature selected was not unique but was strongly influenced by the subset of patients used for the gene selection.

A second consideration when performing supervised analysis of microarray data is that the number of objects that we want to classify (samples) typically is much smaller than the number of parameters that can be used in classification (genes). This is known as the "curse of dimensionality" and is driving the development of new data analysis methods. This problem is most commonly resolved by selecting subsets of genes in advance or iteratively during training. For example, many methods preselect genes or use PCA to reduce the dimensions of the data prior to supervised analysis. Such gene selection may be cumbersome to produce, possibly involving arbitrary selection criterion, or may miss highly informative combinations of genes. To this end, Culhane et al. [48] described a powerful yet simple supervised method called *between-group analysis* (BGA), a multiple discriminant approach that could be safely used when the number of genes exceeds the number of samples [53].

The basis of BGA is to ordinate the group means rather than the individual samples. For N groups (or classes of samples) we find N-1 eigenvectors or axes that arrange the groups so as to maximize the between-group variances. The individual samples are then plotted along them. Each eigenvector can be used as a discriminator to separate one of the groups from the rest. New samples are then placed on the same axes and can be classified on an axis-by-axis basis or by proximity to the group centroids. BGA was shown to be a fast and simple approach to use yet produced accurate discrimination as judged by the performance on gene-expression test data or by a jackknife leave-one-out cross-validation analysis [48].

Many studies have insufficient sample numbers to perform the supervised approach just described. In this case one may simply rank the genes that are most differentially expressed between classes and subsequently employ experimental approaches to validate these genes.

5.6 TARGET IDENTIFICATION: GENE FEATURE SELECTION

Feature (or gene) selection is complex due to the high dimensionality of the data, and thus the risk of detection of false-positive genes is high. We briefly describe some methods that have and are used in gene ranking and refer the reader to an excellent review by Huber et al. [9].

In early microarray studies, most studies simply used fold-change or the maximum difference between the sample groups means to rank genes. This approach is not recommended, as gene variance is not considered, and generally a high number of false-positive genes are expected with this approach. More recent studies employed a Student's *t*-test, or other statistics that incorporated a measure of the difference in the means relative to the standard deviation.[1] To perform a *t*-test on two groups, one first calculates the *t*-statistic, which is the difference between the means of two groups, relative to the standard error of the difference of the means. The *t*-test *p* value measures the chance or probability of obtaining the observed *t*-statistic or something more extreme given the null hypothesis, which assumes that there is no difference in the two population means. Generally $p < .05$ is interpreted as significant. However, due to the number of variables, this analysis is still prone to false positives. For example, in a dataset of 20,000 genes, a 5% type I (false positive) error rate would equate 1,000 false-positive genes. This is the problem of multiple testing large numbers of genes [54]. The classical solution is to use a Bonferroni's correction, which sets a more stringent *p* value. This is calculated by simply dividing the significance level by the number of tests (20,000). In this case the correction would be 0.05/20,000, and $p < .0000025$ would be required for a gene to be significantly differentially expressed. As you can see, this correction is conservative, and many genes would be excluded unnecessarily.

---

1 Incidentally the Student *t*-test was first described by W. S. Gossett (England, 1876–1936) when working in Guinness's brewery in Dublin. However, because Guinness has restrictions on publication, he published under the pseudonym "Student." So when you next drink a Guinness, you may think of the Student's *t*-test!

Therefore, various approaches for computing adjusted p values have been applied. These include permutation-adjusted p values, such as MaxT [55], which uses a two-sample Welch t-statistic (unequal variances) with step-down resampling procedures. Typically these adjusted p values are computed with an order of 10,000 permutations, which is computationally intensive. Although these methods are effective with large numbers of replicates, unfortunately this approach is not effective when datasets have small numbers of samples per group [56].

Classical approaches are complicated by the level of noise in the data, the low number of experimental replicates, and the high number of genes. These characteristics have led to the development of new methods that use a moderated t-statistic, of which, significance analysis of microarrays (SAM) [57] and limma [20] are popular. In these approaches, the standard error is calculated using a pool of variances of genes; thus, these methods "borrow" information across genes. SAM ranks genes using a moderated t-statistic, and the statistical significance of this score is determined by permutation of the samples, and significance of the score is measured in terms of a false-positive rate (FDR). The lowest FDR at which a gene is called significant is the q value. Typically when we use SAM, we compare each cluster to the remaining set (one class response, unpaired data, 1,000 permutations), and use a q value cut off of 1% or 5% significance. Limma fits a linear model to the data but moderates the gene standard error using an empirical Bayes model to obtain a moderated t-statistic. It also produces p values that are adjusted for multiple testing.

5.7 APPRAISAL OF CANDIDATE GENES

The most difficult part of any microarray analysis is appraising genes from a ranked list and deciding which if any (or if all) should be targeted for experimental follow-up. Often the genes of most biological interest are not those that are most highly differential expressed. In fact, biological important phenomena are often the result of many small changes in expression of many genes. In contrast, statistical analysis is optimized to detect large expression changes in a minimum number of genes.

To this end, many groups have produced software to identify groups of functionally or biologically related genes in these gene lists. For example, FatiGO, an online software program, extracts Gene Ontology (GO) terms that are significantly over- or underrepresented in sets of genes [58]. The GO [59] is a large collaborative project that aims to produce consistent descriptions of all gene products. The top three levels in GO are biological function, cellular location, and biological process. A gene could have one or more molecular functions (e.g., catalytic or binding activity) and will be used in one or more biological processes (e.g., signal transduction or cell growth) and may be associated with one or more cellular components (e.g., nucleus or ribosome). Other gene information, such as Kyoto Encyclopedia of Genes and Genomes, literature resources such as PubGene, and databases on cellular pathways such as BioCarta, may also be useful in interpreting gene lists. In the case of identification of drug targets, one may wish to reduce a gene list to those 3,000 or so genes that are potentially amenable to pharmacological intervention, or "druggable" [60].

5.8 META-ANALYSIS

Microarray experiments are relatively expensive, but meta-analysis of expression datasets, particularly when combined with other relevant datasets, may bring new insights that go beyond the scope of the original studies. For example, although many studies have reported gene signatures of metastatic potential of cancer, the overlap between these gene sets is almost zero, even though these gene sets successfully predicted survival of patients in each case. Methods that combine gene-expression profiles from many studies are likely to provide more robust gene signatures [61,62]. Simple methods, such as *co-inertia analysis*, can be used to compare the global correlation between gene-expression profiles of the same tissues or cell lines obtained in different studies, even if these studies have used arrays with different catalogs and numbers of genes [63]. Comparison of the expression profiles of homologous genes across a range of organisms can help in predicting orthologous genes [64]. More recently, meta-analyses of cancer gene expression datasets have begun to provide a more robust estimate of genes that can be implicated in cancer prognosis and progression [61,65,66]. It is likely that combination of microarray data with proteomics and other information will provide great advances in our understanding of cellular processes.

One of the most interesting resources for meta-analysis is gene-expression and drug-response data on a panel of cell lines at National Cancer Institute (NCI). Since 1989, the NCI has screened more than 100,000 compounds against this series of 60 cell lines. These cell lines represent leukaemia and melanoma as well as lung, colon, central nervous system, ovarian, renal, breast, and prostate cancer. Several studies have examined the gene-expression profiles of these cell lines using Affymetrix oligonucleotide-based [67,68] and spotted cDNA arrays [69]. In each case, microarray studies were performed on untreated cells and reported that gene-expression profiles clustered by cellular phenotype [69]. Given these data, a number of studies have attempted to correlate the gene-expression and drug sensitivities profiles of these cells. Scherf et al. [70] calculated Pearson correlation coefficients to relate expression profiles of 1,415 genes with 118 drugs with know mechanisms of action or with 1,400 compounds that at least four replicates [70]. A number of known gene–drug interactions were observed. For example, there was a highly significant negative correlation between 5' fluorouracil potency and dihydropyrimidine dehydrogenase gene expression, a gene that is rate limiting in uracil and thymidine catabolism. A subsequent study correlated these two datasets with a database of the compound substructural and chemical characteristics [71]. It found subclasses of quinones correlated well with genes expressed in melanomas or leukemias [71]. Of interest, they reported a subclass of benzodithiophenedione containing compounds with electron-donating substituents displayed strong positive correlation with Rab7, a gene highly expressed in melanoma. Indolonaphthoquinone-containing compounds were highly correlated with haematopoietic specific genes [71]. In another analysis, Butte et al. [68] and Staunton et al. [67] compared relationships between Affymetrix gene-expression data and compounds from the same drug-response database. The majority of resulting clusters consisted of gene–gene or drug–drug families; however, one significant gene–drug association between the gene *LCP1* and a thiazolidine carboxylic acid derivative (NSC 624044)

was identified. Staunton et al. [67] used an algorithm derived from the weighted voting approach proposed by Golub et al. [72] to identify a 120-gene classifier, which was enriched in extracellular matrix or cytoskeleton genes, that predicted chemosensitivity to cytochalasin D in 20 cell lines with an accuracy of 80%. Other studies have used the multivariate statistical procedure partial least squares to correlate this gene and drug database [73,74].

Thus, although these cell lines have been the subject of much pharmacogenetic analysis, consensus between these reports is yet to be reached. Analyses of large-scale databases such as these, where the number of variables far exceeds the number of cases, are difficult to model statistically and are prone to false-positive results. It is likely that further bioinformatic and statistic tools will be developed for this emerging field. Whether these findings from cell culture apply to real tumors remains to be seen, as it is likely other factors such as drug absorption, metabolism, and access to the tumor site contribute to determining drug response *in vivo*. However, with growing amounts of microarray, other high-throughput data, and parallel advances in analysis and meta-analysis methods, it is likely that these will open new avenues for drug-target identification and design.

REFERENCES

1. Schena, M., Shalon, D., Davis, R. W., and P. O. Brown. 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270:467–470.
2. Nelson, P.T., Baldwin, D.A., Scearce, L.M., Oberholtzer, J.C., Tobias, J.W., and Mourelatos, Z. 2004. Microarray-based, high-throughput gene expression profiling of microRNAs. *Nat Methods* 1:155–161.
3. Johnson, J.M., Edwards, S., Shoemaker, D., and Schadt, E.E. 2005. Dark matter in the genome: Evidence of widespread transcription detected by microarray tiling experiments. *Trends Genet* 21:93–102.
4. Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C.A., Causton, H.C., et al. 2001. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet* 29:365–371.
5. Ball, C.A., Brazma, A., Causton, H., Chervitz, S., Edgar, R., Hingamp, P., Matese, J.C., Parkinson, H., Quackenbush, J., Ringwald, M., et al. 2004. Submission of microarray data to public repositories. *PLoS Biol* 2:E317.
6. Parkinson, H., Sarkans, U., Shojatalab, M., Abeygunawardena, N., Contrino, S., Coulson, R., Farne, A., Lara, G.G., Holloway, E., Kapushesky, M., et al. 2005. ArrayExpress—a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res* 33:D553–555.
7. Barrett, T., Suzek, T.O., Troup, D.B., Wilhite, S.E., Ngau, W.C., Ledoux, P., Rudnev, D., Lash, A.E., Fujibuchi, W., and Edgar, R. 2005. NCBI GEO: Mining millions of expression profiles—Database and tools. *Nucleic Acids Res* 33:D562–566.
8. Ball, C.A., Awad, I.A., Demeter, J., Gollub, J., Hebert, J.M., Hernandez-Boussard, T., Jin, H., Matese, J.C., Nitzberg, M., Wymore, F., et al. 2005. The Stanford Microarray Database accommodates additional microarray platforms and data formats. *Nucleic Acids Res* 33:D580–582.

9. Huber, W., von Heydebreck, A., and Vingron, M. 2003. Analysis of microarray gene expression data. In *Handbook of Statistical Genetics*, 2nd ed. (eds. D.J. Balding, M. Bishop, and C. Cannings). John Wiley & Sons Ltd., Chichester, UK.
10. Yang, C., Bakshi, B.R., Rathman, J.F., and Blower, P.E., Jr. 2002. Multiscale and Bayesian approaches to data analysis in genomics high-throughput screening. *Curr Opin Drug Discov Devel* 5:428–438.
11. Yang, Y.H., Dudoit, S., Luu, P., Lin, D.M., Peng, V., Ngai, J., and Speed, T.P. 2002. Normalization for cDNA microarray data: A robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res* 30:E15.
12. Quackenbush, J. 2002. Microarray data normalization and transformation. *Nat Genet* 32 Suppl.:496–501.
13. Smyth, G.K., and Speed, T. 2003. Normalization of cDNA microarray data. *Methods* 31:265–273.
14. Huber, W., von Heydebreck, A., Sultmann, H., Poustka, A., and Vingron, M. 2002. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics* 18, Suppl. 1:S96–104.
15. Irizarry, R.A., Hobbs, B., Collin, F., Beazer-Barclay, Y.D., Antonellis, K.J., Scherf, U., and Speed, T.P. 2003. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* 4:249–264.
16. Bolstad, B.M., Irizarry, R.A., Astrand, M., and Speed, T.P. 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 19:185–193.
17. van de Peppel, J., Kemmeren, P., van Bakel, H., Radonjic, M., van Leenen, D., and Holstege, F.C. 2003. Monitoring global messenger RNA changes in externally controlled microarray experiments. *EMBO Rep* 4:387–393.
18. Li, C., and Wong, W.H. 2001. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proc Natl Acad Sci USA* 98:31–36.
19. Hardiman, G. 2004. Microarray platforms—comparisons and contrasts. *Pharmacogenomics* 5:487–502.
20. Smyth, G.K. 2004. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology* 3:Article 3.
21. Saeed, A.I., Sharov, V., White, J., Li, J., Liang, W., Bhagabati, N., Braisted, J., Klapa, M., Currier, T., Thiagarajan, M., et al. 2003. TM4: A free, open-source system for microarray data management and analysis. *Biotechniques* 34:374–378.
22. Lockhart, D.J., Dong, H., Byrne, M.C., Follettie, M.T., Gallo, M.V., Chee, M.S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., et al. 1996. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat Biotechnol* 14:1675–1680.
23. Wu, Z., and Irizarry, R.A. 2004. Preprocessing of oligonucleotide array data. *Nat Biotechnol* 22:656–658.
24. Gentleman, R.C., Carey, V.J., Bates, D.M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., et al. 2004. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biol* 5:R80.
25. Sharov, V., Kwong, K.Y., Frank, B., Chen, E., Hasseman, J., Gaspard, R., Yu, Y., Yang, I., and Quackenbush, J. 2004. The limits of log-ratios. *BMC Biotechnol* 4:3.
26. Slonim, D.K. 2002. From patterns to pathways: Gene expression data analysis comes of age. *Nat Genet* 32, Suppl.:502–508.

27. Quackenbush, J. 2001. Computational analysis of microarray data. *Nat Rev Genet* 2:418–427.
28. Causton, H.C., Quackenbush, J., and Brazma, A. 2003. *Microarray Gene Expression Data Analysis: A Beginner's Guide*. Blackwell Publishing, Oxford, UK, p. 192.
29. Hartigan, J. 1975. *Clustering algorithms*. Wiley, New York.
30. Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA* 95:14863–14868.
31. Wen, X., Fuhrman, S., Michaels, G. S., Carr, D. B., Smith, S., Barker, J. L., Somogyi, R. 1998. Large-scale temporal gene expression mapping of central nervous system development. *Proc Natl Acad Sci USA*. 95(1):334–339.
32. Liu X, K.P. 2003. Mining gene expression data. In *Bioinformatics: Genes, Proteins and Computers*. (ed. O.C. Thornton J, Jones D), pp. 229–237. BIOS Scientic Publishers, Oxford.
33. Legendre, P., and Legendre, L. 1998. *Numerical Ecology*, 2nd English ed. Elsevier, Amsterdam.
34. Torrente, A., Kapushesky, M. Brazma, A. 2005. A new algorithm for comparing and visualizing relationships between hierarchical and flat gene expression data clusterings. (Submitted)
35. Kapushesky, M., Kemmeren, P., Culhane, A.C., Durinck, S., Ihmels, J., Korner, C., Kull, M., Torrente, A., Sarkans, U., Vilo, J., et al. 2004. Expression Profiler: Next generation—an online platform for analysis of microarray data. *Nucleic Acids Res* 32:W465–470.
36. Fellenberg, K., Hauser, N.C., Brors, B., Neutzner, A., Hoheisel, J.D., and Vingron, M. 2001. Correspondence analysis applied to microarray data. *Proc Natl Acad Sci USA* 98:10781–10786.
37. Raychaudhuri, S., Stuart, J.M., and Altman, R.B. 2000. Principal components analysis to summarize microarray experiments: Application to sporulation time series. *Pac Symp Biocomput*:455–466.
38. Crescenzi, M., and Giuliani, A. 2001. The main biological determinants of tumor line taxonomy elucidated by a principal component analysis of microarray data. *FEBS Lett* 507:114–118.
39. Alter, O., Brown, P.O., and Botstein, D. 2000. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci USA* 97:10101–10106.
40. Thioulouse, J., Chessel, D., Dolédec, S., and Olivier, J.M. 1997. ADE-4: A multivariate analysis and graphical display software. *Statistics and Computing* 7:75–83.
41. Jolicoeur, P., and Mosimann, J.E. 1960. Size and Shape Variation in the Painted Turtle: A Principal Component Analysis. *Growth*. 24:339–354.
42. Su, A.I., Cooke, M.P., Ching, K.A., Hakak, Y., Walker, J.R., Wiltshire, T., Orth, A.P., Vega, R.G., Sapinoso, L.M., Moqrich, A., et al. 2002. Large-scale analysis of the human and mouse transcriptomes. *Proc Natl Acad Sci USA* 99:4465–4470.
43. Pearson, K. 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine* 2:559–572.
44. Greenacre, M. 1984. *Theory and Applications of Correspondence Analysis*. Academic Press, London.
45. Huang, E., Ishida, S., Pittman, J., Dressman, H., Bild, A., Kloos, M., D'Amico, M., Pestell, R. G., West, M., and Nevins, J. R. 2003. Gene expression phenotypic models that predict the activity of oncogenic pathways. *Nat Genet* 34:226–230.

46. Wall, M. E., Rechtsteiner, A., and Rocha, L. M. 2003. Singular value decomposition and principal component analysis. In *A Practical Approach to Microarray Data Analysis*. (eds. D. P. Berrar, W. Dubitzky, M. Granzow), pp. 91–109, Norwell, MA: Kluwer.
47. Holter, N.S., Mitra, M., Maritan, A., Cieplak, M., Banavar, J.R., and Fedoroff, N.V. 2000. Fundamental patterns underlying gene expression profiles: Simplicity from complexity. *Proc Natl Acad Sci USA* 97:8409–8414.
48. Culhane, A.C., Perriere, G., Considine, E.C., Cotter, T.G., and Higgins, D.G. 2002. Between-group analysis of microarray data. *Bioinformatics* 18:1600–1608.
49. Simon, R. 2003. Diagnostic and prognostic prediction using gene expression profiles in high-dimensional microarray data. *Br J Cancer* 89:1599–1604.
50. Kuo, W.P., Kim, E.Y., Trimarchi, J., Jenssen, T.K., Vinterbo, S.A., and Ohno-Machado, L. 2004. A primer on gene expression and microarrays for machine learning researchers. *J Biomed Inform* 37:293–303.
51. Ein-Dor, L., Kela, I., Getz, G., Givol, D., and Domany E. 2005. Outcome signature genes in breast cancer: Is there a unique set? *Bioinformatics* 21(2):171–178. Epub 2004 Aug 12.
52. van't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T. et al. 2002. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*. 415(6871):530–536.
53. Dolédec, S., and Chessel, D. 1987. Rhythmes saisonniers et composantes stationelles en milieu aquatique I- Description d'un plan d'observations complet par projection de variables. *Acta Oecologica Oecologica Generalis* 8:403–426.
54. Dudoit, S., Popper Shaffer J., Boldrick J.C. 2003. Multiple Hypothesis Testing in Microarray Experiments. *Statistical Science* 18:71–103.
55. Ge, Y., Dudoit, S., and Speed, T. P. 2003. Resampling-based multiple testing for microarray data hypothesis. *Test* 12(1):1–44.
56. Jeffery, I. B., Higgins, D. G., Culhane, A. C. 2006. Comparison and evaluation of microarray feature selection methods. In preparation.
57. Tusher, V.G., Tibshirani, R., and Chu, G. 2001. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences USA* 98:5116–5121.
58. Al-Shahrour, F., Diaz-Uriarte, R., and Dopazo, J. 2004. FatiGO: A web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics* 20(4):578–80. Epub 2004 Jan 22.
59. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al. 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* 25:25–29.
60. Orth, A.P., Batalov, S., Perrone, M., and Chanda, S.K. 2004. The promise of genomics to identify novel therapeutic targets. *Expert Opin Ther Targets* 8:587–596.
61. Rhodes, D.R., and Chinnaiyan, A.M. 2004. Bioinformatics strategies for translating genome-wide expression analyses into clinically useful cancer markers. *Ann N Y Acad Sci* 1020:32–40.
62. Wang, J., Coombes, K. R., Highsmith, W. E., Keating, M. J., Abruzzo, L. V. 2004. Differences in gene expression between B-cell chronic lymphocytic leukemia and normal B cells: A meta-analysis of three microarray studies. *Bioinformatics* 20(17):3166–3178.
63. Culhane, A.C., Perriere, G., and Higgins, D.G. 2003. Cross-platform comparison and visualisation of gene expression data using co-inertia analysis. *BMC Bioinformatics* 4:59.

64. Grigoryev, D.N., Ma, S.F., Irizarry, R.A., Ye, S.Q., Quackenbush, J., and Garcia, J.G. 2004. Orthologous gene-expression profiling in multi-species models: Search for candidate genes. *Genome Biol* 5:R34.
65. Shen, R., Ghosh, D., and Chinnaiyan, A.M. 2004. Prognostic meta-signature of breast cancer developed by two-stage mixture modeling of microarray data. *BMC Genomics* 5:94.
66. Segal, E., Friedman, N., Koller, D., and Regev, A. 2004. A module map showing conditional activity of expression modules in cancer. *Nat Genet* 36:1090–1098.
67. Staunton, J.E., Slonim, D.K., Coller, H.A., Tamayo, P., Angelo, M.J., Park, J., Scherf, U., Lee, J.K., Reinhold, W.O., Weinstein, J.N., et al. 2001. Chemosensitivity prediction by transcriptional profiling. *Proc Natl Acad Sci USA* 98:10787–10792.
68. Butte, A.J., Tamayo, P., Slonim, D., Golub, T.R., and Kohane, I.S. 2000. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proc Natl Acad Sci USA* 97:12182–12186.
69. Ross, D.T., Scherf, U., Eisen, M.B., Perou, C.M., Rees, C., Spellman, P., Iyer, V., Jeffrey, S.S., Van de Rijn, M., Waltham, M., et al. 2000. Systematic variation in gene expression patterns in human cancer cell lines. *Nat Genet* 24:227–235.
70. Scherf, U., Ross, D.T., Waltham, M., Smith, L.H., Lee, J.K., Tanabe, L., Kohn, K.W., Reinhold, W.C., Myers, T.G., Andrews, D.T., et al. 2000. A gene expression database for the molecular pharmacology of cancer. *Nat Genet* 24:236–244.
71. Blower, P.E., Yang, C., Fligner, M.A., Verducci, J.S., Yu, L., Richman, S., and Weinstein, J.N. 2002. Pharmacogenomic analysis: Correlating molecular substructure classes with microarray gene expression data. *Pharmacogenomics J* 2:259–271.
72. Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., and Caligiuri, M. 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286:531–537.
73. Dasgupta, L., Lin, S. M., Carin, L. 2002. Modeling Pharmacogenomics of the NCI-60 Anticancer Data Set: Utilizing kernel PLS to correlate the Microarray Data to Therapeutic Responses. In *Methods of Microarray Data Analysis II* (ed. S. Lin and K. M. Johnson). Kluwer Academic Publishers.
74. Musumarra, G., Condorelli, D.F., Costa, A.S., and Fichera, M. 2001. A multivariate insight into the in vitro antitumour screen database of the National Cancer Institute: Classification of compounds, similarities among cell lines and the influence of molecular targets. *J Comput Aided Mol Des* 15:219–234.
75. Johnson, K. F., and Lin, S. M. 2001. Critical assessment of microarray data analysis: The 2001 challenge. *Bioinformatics* 17(9):857–858.

Part II

Target Validation

6 Text Mining

Bruce Gomes
AstraZeneca Pharmaceuticals

William Hayes
Biogen-Idec

Raf M. Podowski
Oracle

CONTENTS

6.1 INTRODUCTION

An increasingly prominent topic at conferences concerns text mining and extracting the full value of the literature for drug discovery. There have been a few false starts and misunderstandings about what text mining is and what it can do. As with many new technologies, some of the claims are perhaps excessive, yet we predict that text mining will join many others in drug discovery as a key enabling technology. Figure 6.1 shows the aspects of drug discovery where text mining is highly applicable.

What is text mining? At a basic level, text mining is the process of highlighting a small volume of relevant information from a very large set of possibly interesting documents. There is nothing magical about the process. Text-mining software's "understanding" of the literature is still rather rudimentary. However, although it may make mistakes where a human will see the interpretation as incorrect, it is often right. Unlike tired humans, its actions are consistent. The greatest benefit of software, in analyzing the literature, is that it can grind through amounts of text that no human could ever hope to manage.

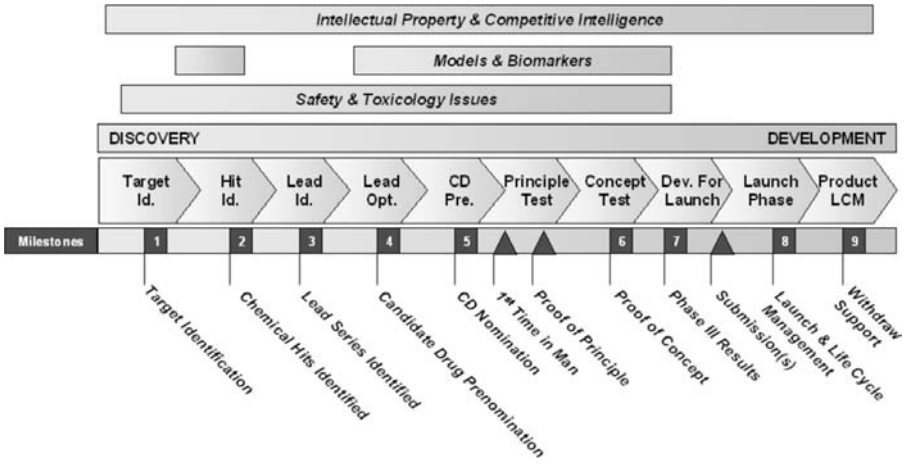


FIGURE 6.1 Drug-discovery process. The pharmaceutical industry usually divides the drug-discovery process into phases that are based on milestones (key indicators of progression). Text mining has specific application to some phases (see discussion in 6.3.3) but also has areas of broad utility that extend across the entirety of the process. The horizontal bars at the top of the figure indicate these broad utility areas.

One cannot expect text mining to produce accurate, final knowledge with no need for human review. However, the current state of the art can reduce the work of the human reader tremendously. The need is indisputable; the number of patents and articles has doubled in the last 10 years, but our methods for dealing with the flood remain unchanged.

Fortunately, text mining is beginning to mature.

- Electronic content is available for mining and is becoming more available as journal publishers recognize text mining for the value it provides in "advertising" their content.
- Computer hardware has progressed to the point where maintaining millions of dynamically indexed documents is relatively inexpensive.
- Specialized biomedical "ontologies" are now available through public and commercial sources and are complete enough to be useful.
- Practical natural language processing (NLP) tools are now available.

Recall and *precision* are two of the main measures of accuracy applied to text-mining results. Recall is the number of correct results found from all possible correct results. Precision is the number of correct results found from the total number of returned results (correct and incorrect). For various applications and corpora (in text-mining terms, a *corpus* is a set of documents), recall and precision usually range between 30 and 80%. One generally has to balance the recall versus the precision, but there are ways to enhance both at the same time. NLP applications provide better precision and recall by adding additional heuristics for extracting information from the text. Statistical text-mining applications can disambiguate (e.g., distinguish the gene symbol CAT as catalase OR chloramphenical transferase) gene names prior to performing gene/disease co-occurrence analyses. Practical accuracy values give a more realistic representation than the theoretical accuracy numbers from controlled studies. Most important facts in the literature are mentioned more than once. If a text-mining application finds only one of three instances of the relationship "Raf phosporylates Mek," the standard measure of recall on the three instances is only 33%, but the fact has been found, and that is the practical success criteria. To take an even more pragmatic approach, one might say the bottom line recall value that matters is the number of facts that one can reasonably extract given the man-hours available for a project. If one has 40,589 MEDLINE abstracts (as of July 20, 2004) to review for Alzheimer's disease (AD)-related protein interactions, this will not be possible to curate manually. Even 30% recall on 40,589 abstracts is going to be much better than 100% recall on the few that one could read manually. The bottom line is that text mining is certainly mature enough to use, but, because precision is still well below 100%, the results currently require human review.

Text mining is a process (see fig. 6.2). Steps in a typical study might be

- Collect sources (abstracts, MEDLINE, full-text journal articles, patents)
- Develop landscape map of important concepts using statistical text mining
- Use NLP to extract facts
- Use semantically typed databases for the resulting knowledge management

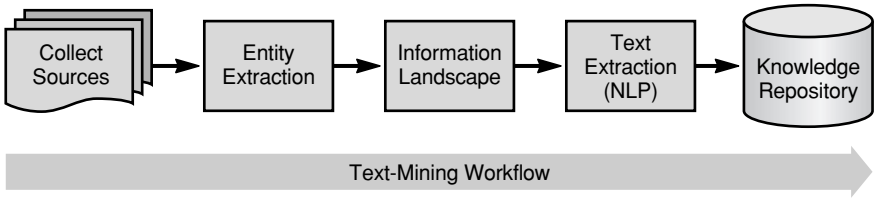


FIGURE 6.2 Text-mining workflow. Typical text-mining workflow involves identification and subsequent collection of document sources, biological/biochemical/medical entity extraction, statistical content analysis, and natural language processing utilizing relevant ontologies, with a final goal of a relevant knowledge repository.

Text-mining tools can be used in many ways; to focus the process, it is important to start with a clearly stated question. One must indicate what types of concepts and relationships between concepts are essential in order to answer a question before starting to extract them from the literature. Various applications fit together in the text-mining application portfolio to go from broad to focused results. As an example, starting with the pathological process of steatosis, one may first develop a *knowledge map* of the pathways, proteins, compounds, and so on, associated with steatosis and how they relate to each other using a network graph and associated tables. Given that map of knowledge, one then determines that the specific chemical compound/protein interactions associated with the steatosis literature should be extracted and incorporated into a knowledge base for subsequent data-mining exercises.

6.2 TECHNICAL ASPECTS OF TEXT MINING

A review of text-mining technologies is presented next. The various types of analyses with background and references to more information are discussed to enhance the reader's understanding of the underlying technology and strengths and weaknesses of each approach.

6.2.1 Keyword Searching and Manual Methods

6.2.1.1 Text Search

Text searching represents the first level of text mining. It is more formally known as information retrieval (IR). The resulting output is usually presented as ordered document lists. Documents are sorted by a keyword-based scoring function. Because documents are treated merely as "bags of words," all context and semantic variation is ignored. Therefore, keyword searches tend to return a high volume of hits with little ability to discriminate nuance, complex connections, or even the relevance of the concepts communicated in the document in which the keyword resides. On the other hand, keyword-based text search is the most popular form of text mining because it is the most familiar. One runs a search and then analyzes the results manually. One can also run multiple searches looking for intersections between literature sets [1].

To improve the relevance of keyword searches, some search engines allow the following:

- Wildcard characters and truncation (e.g., IGF? = IGF, IGF1 or IGF2 while *ase = phosphorylase, kinase, convertase, etc.)
- Boolean operators (OR, AND, or NOT)
- Mechanisms for searching for phrases, usually quoted phrases (e.g., "multiword phrase") or bracketed phrases (e.g., [multi-word phrase])
- Proximity limiters (e.g., (coronary AND disease)/5 finds documents that contain both words within five words of each other)
- Term weighting, fuzzy matches to words, and stemming words are other ways to improve the relevant ordering of the resulting documents

The first major drawback of keyword searching is that the task of sorting the search engine output falls on the investigator, which exploits neither the computer's nor the human's strengths. In addition, keyword searches suffer from polysemy (the same word having different meaning in different contexts), which requires the reader to examine documents for relevance, where a large number may be completely incorrect, and synonymy (multiple words referring to the same concept), which requires the investigator to know (and employ) all possible alternative synonyms to ensure a complete search. Synonymy is a particularly difficult problem in the biology literature, where proteins routinely have many names and abbreviations often shared with common English words.

6.2.1.2 Large-Scale Commercial Curation Efforts

A number of companies have developed massive, manually curated databases. Some of these companies are

- BioBase [2]
- GVK Bio [3]
- Molecular Connections [4]
- Jubilant Biosystems [5]
- Ingenuity [6]

Each of these companies uses large groups of curators to read and manually extract relevant facts from abstracts or journal articles. Databases of extracted data that can be manipulated to discover literature-based, nontrivial relationships about protein–protein interactions, gene-to-disease associations, small molecule to enzyme target information, and so on, are provided to the customer. Some of the curation companies attempt to read every abstract on every named gene, whereas others read a selected number of full-length reviews on particular signaling pathways. Others provide custom packages that provide great depth with a narrow focus or overviews of broad areas. All of these approaches have value depending on the type of information required. One perceived weakness of manual curation methods is that the user does not control the quality and coverage of the derived information. Curation

companies try to counter these concerns by having well-trained staff (often master's and doctoral-level scientists), a multitier fact-checking system, and data entry protocols. A second limitation of this method is that the companies are somewhat inflexible. There is little opportunity to steer the text-mining process if the needs of the user change after the databases are completed.

6.2.2 Literature Resources for Text Mining

Access to the literature is critical for any text-mining effort. The literature sources are scattered across many different databases and Web sites with a variety of access mechanisms and licensing controls. Finding the literature sources required for a task can be a monumental project in itself. It is nearly impossible to be certain that one has identified all applicable repositories or documents. One also needs to download a local copy of the complete text of any literature source or search results (for a specialized corpus) to analyze it effectively.

An extremely important task in the collection of literature for text mining is making sure that the license agreements for use of the literature are appropriate for the use and that copyright law is not being violated. This task is actually quite tricky, and the copyright laws of each country can differ substantially. Copyright clearance centers are available to assist with copyright licensing [7].

6.2.2.1 Abstract Collections

There are a variety of text sources for biomedical text mining (see table 6.1). Abstract collections are the text sources, or corpora, that are easiest to access. One can also collect full-text patent collections from the various resellers or from the Patent Authorities directly (U.S. Patent Office, European Patent Office, World Intellectual Property Office, etc.). Other sources include full-text journal articles, Web documents, and news articles.

The premier source, which has catalyzed this entire area of research, is MEDLINE. Without MEDLINE, the authors posit that very little progress in biomedical

TABLE 6.1
Document Corpora Sources for Biomedical Text Mining

Corpus (as of August 2004)	Records (Million)	Size (Gigabytes)
Medline	14	48
Biosis	7.7	28 (est.)
EMBASE	16	59 (est.)
U.S. patents	5 (est.)	500 (est.)
Biomedical journals	20 (est.)	11,200 (est.)

Note: The sizes of the more common life-science-oriented corpora are listed. The patent corpus includes only pharmaceutically relevant patents, not the entire electronically available patent collection.
est. = estimated sizes.

text mining would have been made to date. MEDLINE includes approximately 14 million abstracts collected from several thousand journals on a daily basis. The processing of the database also includes manually assigned keywords for both the MeSH taxonomy and a substance listing.

Other abstract sources include various conference abstracts available on the Internet, such as the American Association of Cancer Research [8], the premier Biosis [9], and EMBASE [10] abstract collections. There is significant overlap between the MEDLINE, EMBASE, and Biosis abstract collections. MEDLINE is focused more on medical journals, including biology journals that support medical research. EMBASE is also focused on medical research but includes more conference abstracts and European journals. Biosis is more focused on basic life sciences.

6.2.2.2 Patents

The patent corpora worldwide can also be a valuable resource. Some scientific information that is not published elsewhere can be found in the introduction and background presentations of patent applications. It can be difficult to mine patent applications effectively due to the obfuscated manner in which many are written. So far, patents have been found to be most useful for competitive intelligence purposes. Care should be taken to achieve full value from patent collections. It might be more efficacious to divide the Abstracts/Claims section from the remainder of the full text. Micropatents [11], Delphion [12], or other commercial entities are able to provide a single-source access to the major patent collections of the world.

6.2.2.3 Full-Text Journal Access

There are three ways to access full-text journal articles:

- License full-text content from each publisher, which covers downloading, storing, analyzing, and delivering the results to the text-mining customer.
- Download the available open source journal articles.
- Utilize existing standard license agreements with publishers to download full-text journal articles specific to a search request, for personal use.

At this point, there are many different biomedical literature publishers and various resellers of literature content that provide access to their published literature. Few recognize the importance of text mining to increasing the value of their literature holdings. Therefore, it may be quite difficult to negotiate full-text literature access. Because Open Access journals have only recently become available, their scope is limited and cannot be relied on exclusively.

Quosa [13] (see fig. 6.3) is a product that may provide a solution to generating small corpora from the licensed content available to the text-mining researcher. Quosa allows the searching of full-text and abstract collection indices and subsequent downloading of full-text journals associated with the search results. Other tools may also be available for this task. The variety of PDF formats and locations of the full-text content on each publisher's Web site make collection of the content rather

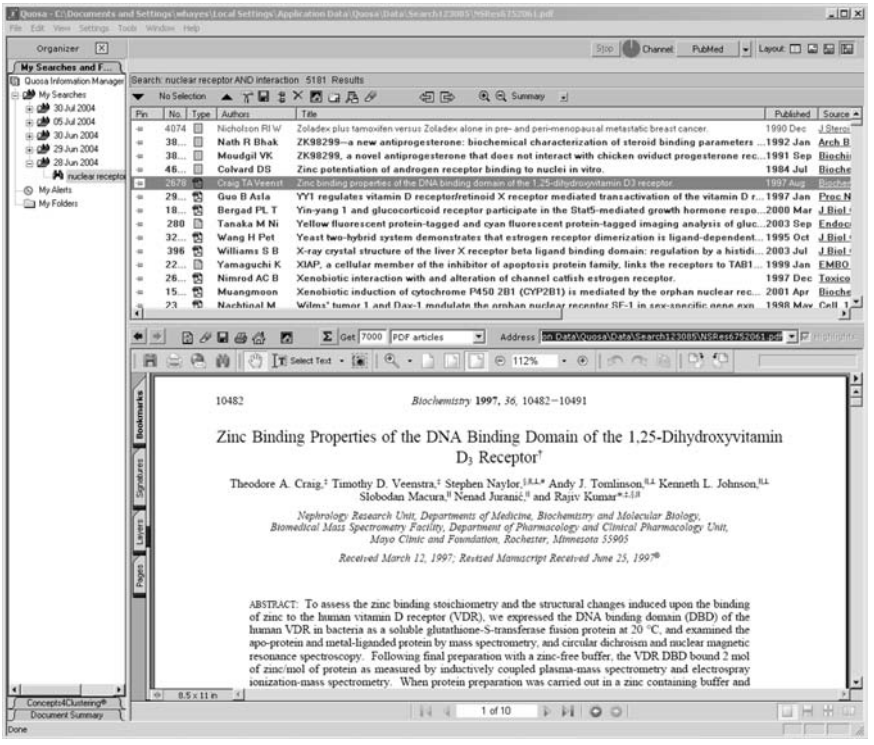


FIGURE 6.3 Document corpora generation. A view of a user interface of the program Quosa, used for searching and downloading licensed, full-text journal documents. Search result with a search/literature folder management window on the left side of the screen, and a title selection window on the upper right with the full-text document viewer in the window on the lower right.

difficult. The authors do not recommend any use of a technology such as this without sufficient legal guidance concerning the licensing and copyright issues surrounding use of the literature.

6.2.3 Ontology

An *ontology* is a machine-readable description of physical and nonphysical things and the relationships that exist between them. Ontology, as a discipline, is a science providing a framework for communication and context. The great value of ontologies is fully realized for text mining in their potential for query generation and result management. Please refer to chapter 20 in this book for a more thorough review of ontologies and their uses.

An ontology structures concepts (e.g., a gene, a physiological process, physical interaction vs. indirect interactions between proteins) in a cyclical graph or network structure. A concept can be related to another concept via a relationship ontology. The "is-a" relation is used to indicate "is-a-child," "is-a-parent," and "is-a-synonym." The "part-of" can relate tissues and organs in the body. Cardiac tissue is part-of the

heart. The ontology can also incorporate relations from a larger relationship ontology where the relations could be, for example, protein–protein interactions (e.g., "interacts-with" → "physically interacts" → "adheres-to" → "binds-to"). Lists, thesauri, controlled vocabularies, and taxonomies are degenerate representations of ontologies. They are often quite useful; for example, a protein ontology that incorporates the "is-a-synonym" relation can be converted to a thesauri form for performing expanded queries for a protein name in a variety of text search and text-mining applications. One can enter "ApoE" into a search form or run a text-mining analysis and have the query or analysis expand to use "ApoE," "apoE," or "apolipoprotein E." If the ontology incorporates the is-a-child or part-of relationship, one can generate a hierarchically structured thesaurus. Then one can run text-mining analyses or text search queries encompassing all protein-tyrosine kinases or nuclear hormone receptors. A great deal of the domain knowledge for a particular area of research can be captured as an ontology (which functions as a knowledge schema). The resulting ontology can provide the domain knowledge required for text-mining applications. For example, proteins, protein interaction modalities, biological effects and processes, pathological processes, pathway names, and so on, can all be incorporated into ontologies for use in biomedical text mining.

An additional use of ontologies, related to text mining, is for information captured in a semantically normalized fashion. Ontologies also provide contextualization of information. For instance, they can directly represent the fact that *Raf* is a protein-tyrosine kinase AND is an enzyme found in humans. They can also be used as the foundation for semantically normalized multirelational databases that can act as knowledge stores.

Unfortunately, the authors have not found the open-source ontologies for text mining to be useful at this time. The Gene Ontology (GO) [14] does not incorporate the is-a-synonym that is necessary for effective use in text mining or text search. GO is a gene *attribute* ontology and performs well as a controlled vocabulary. UMLS [15] has proven to be an excellent source for specialized thesauri development, but it has a limited hierarchical structure.

6.2.4 Text Categorization and Clustering

6.2.4.1 Text Categorization

Automatic text categorization for text mining is usually employed to filter documents into specified subsets. Text categorization is a supervised machine-learning technique—meaning that it requires training data to generate the category models. It can be used to generate large taxonomic structures of documents similar to Yahoo's [16] classification of Web pages. On the whole, the authors believe that most text-mining-oriented text categorization needs to be highly focused and customized to the end user rather than global/company-wide text classifications.

A caveat concerning automatic text categorization is that it is often difficult to know what features of the training documents are being used to categorize the literature. One may attempt to develop a cancer versus noncancer document filter (text-categorization-based filter) and find the filter is actually using the *publisher*

name as the major discriminating feature because all the cancer training documents contained a reference to the publisher. Text categorization, more than any other area of text mining, requires very careful thought and design.

Another difficulty is the updating of category training data. As one's interest in a particular filter changes in time, one may corrupt the initial intent of the training data or expand it beyond its purpose. For example, one starts with a neurodegeneration literature filter that has 10 neurodegeneration documents versus 1,000 non-neurodegeneration documents. Later, if one adds 5 *neuronal development* articles to the neurodegeneration set, one can end up with a filter that returns practically any document about neuronal cells.

The null model (such as the non-neurodegeneration documents) in the previous example is critical to the success of the filter. If the null model is neither appropriately balanced in content nor balanced in number of nulls versus positive results, then the results of the filter will not be correct. Training the filter on WHAT IS and IS NOT a match is critical. The filter must be informed about how many matches are expected in the test documents. If only 1 in 1,000 documents being tested is about neurodegeneration, then the training set bias of the positive and null training sets needs to be similar. Often it is difficult to curate a large enough training dataset to set the bias based on correct ratios of training documents. Many of the text-categorization algorithms have a weighting factor that can be set such that one can correct for the expected ratio of documents.

The null model is often more difficult to create in a balanced manner than the target category. If the entirety of MEDLINE is used to filter documents concerning neurodegeneration, then it is necessary to randomly sample from MEDLINE at least 10,000 documents (assuming a ratio of 1/1000 neurodegeneration documents, which can be estimated by collecting counts from simple keyword searches) to be able to include 10 documents for neurodegeneration. It is also necessary to ensure that the null model training data contain documents about aspects of neuronal research other than neurodegeneration. For better discrimination, one might weight the null model with a higher-than-expected ratio of non-neurodegenerative neural research articles. Testing of the filter is required to understand the behavior of the categorization model.

Several algorithms exist for text categorization. The major ones are Naïve Bayes [17], Support Vector Machine (SVM), and decision trees [18]. SVMs [19–21] are generally considered the most accurate categorization algorithm at this time, but they are difficult to scale to large datasets and multiclass problems. Furthermore, accuracy assessment is much more time consuming than using a Naïve Bayes classifier. Decision trees can be built automatically. Decision trees are somewhat unusual for these machine-learning techniques in that they are fairly transparent for human comprehension. They may need to be manually *pruned* to enhance their effectiveness. Naïve Bayes–based classifiers are very fast and easy to set up for multiclass categories. Naïve Bayes classifiers also perform rapid accuracy assessment using leave-one-out analysis. They have been found to be less accurate than SVM classifiers, however. SVMs also handle imbalanced training sets better than Naïve Bayes classifiers. In general, the decision on which algorithm to use is based on analysis speed and computing power versus accuracy.

There are many classification tools available for use. Some examples are Oracle Text, ReelTwo's CS, and SVMLight. Oracle has several built into their Oracle10g database. SVMLight is available for SVM-based text classification [20]. ReelTwo [22] has a classifier with a user interface to ease development of categorization models. Most of the authors' text-categorization experience is based on Oracle or ReelTwo. A variety of other classification applications are available. Selection of a specific application is dependent on classification goals, user interface, application features, and scalability requirements.

The authors caution that end users will need to be directly involved in developing text-classification filters, but the end users will need significant assistance from experts in text classification.

6.2.4.2 Clustering

Categorization and clustering are both machine-learning methods. They are used in different ways, however. Text categorization is a content classification method, requiring some manual preparation or user curation of a model and therefore some idea of what one is looking for. Document clustering is used for knowledge discovery and provides a hint of the diversity of themes within an otherwise uncharacterized document collection. In particular, clustering is used when exploratory searches result in hundreds or thousands of documents.

A simple keyword-based IR approach greatly limits the exploitation of the knowledge structure contained in returned documents. Clustering provides a major improvement in the grouping and prioritization of a set of documents. Clustering arose from a need to improve IR systems [23,24], identify similar documents [25], and better organize and browse a group of documents [26,27].

Document clustering is a form of unsupervised machine learning [28]. At its simplest and purest level, document clustering requires no prior knowledge or expectations about the contents and provides concept extraction [29] and knowledge navigation. Furthermore, concept extraction can seed an automatic derivation of classifications and serve as a method of "ontology induction" [30].

Traditional clustering methods rely less on semantic analysis and instead utilize multivariate statistical techniques to form clusters of similar objects in a multidimensional space [31]. In every case, the process involves generation of characteristic document vectors. Most frequently, these vectors are based on individual word frequencies in the document. To reduce the significance of frequently occurring words found in a majority of the documents, such as common English words, a number of normalization or weight schemes can be applied: inverse document frequency, probabilistic weights, stoplists (a list of specific words that will be excluded from the analysis), or domain-specific weighted theme lists. At this stage, a method may diverge from purely automatic clustering toward semisupervised, partially categorization-based cluster identification (see the following examples).

Document vectors are used for calculating a similarity (or distance) metric between two documents or a document and a cluster *centroid* (a vector representing the center of a cluster of documents). Similarly, centroid vectors can be used to

obtain a similarity metric between two clusters. The cosine measure is most frequently used to compute these values.

The main two approaches to clustering are hierarchical or partitional [32]. Hierarchical techniques produce a treelike structure of nested document groups. At the root of the hierarchy is a single cluster representing the complete document corpus. The leaves of the tree represent individual documents. There are two basic approaches to generating a hierarchical clustering: agglomerative and divisive. Agglomerative algorithms start with the documents as individual clusters and, at each step, merge the most similar or closest pair of clusters. Merging clusters requires defining and applying a cluster similarity or distance metric. Divisive algorithms start with all documents in one cluster and proceed by splitting existing clusters until only individual documents remain in each leaf cluster. At each step of the algorithm, a decision has to be made as to which cluster to split and how to perform the split. The hierarchical-clustering approach has a number of drawbacks when applied to text documents. In general, it is computationally taxing due to a quadratic time complexity O(n2). Additionally, the very nature of text leads to a frequent finding that, based on a single similarity metric, the topics of neighboring documents can vary significantly [32].

Partitional clustering techniques such as K-means assign documents to a specified number of unnested clusters with no apparent hierarchy. Partitional clustering tries to optimize the distribution of cluster centroids within the multidimensional document space. Assignment of documents to a cluster can be based on a cluster quality measure or relative cluster sizes. Partitional clustering methods do not produce hierarchies of documents, which in general results in linear run times, O(n).

These methods use a quality measure to assess cluster "goodness." An internal quality measure compares sets of clusters without reliance on preexisting knowledge, such as user validation or classification models. A number of external quality measures exist and are often used to rate the quality of a cluster or the performance of a clustering method. Entropy [33] and F-measure are two examples of cluster quality measures.

A hybrid method, bisecting K-means, combines the divisive hierarchical and K-means methods to produce a controlled number of hierarchical document clusters. It has been shown to perform as good as or better than hierarchical methods while retaining the performance of the K-means approach [32]. The process of this method involves bisecting a selected cluster of documents (biggest or poorest quality) into two smaller clusters but optimizing the centroids to obtain new clusters with the best possible quality. An example of an implementation of this type of method is the Oracle Text hierarchical K-means algorithm.

Recently, two new methods have been utilized in improving the document attribute vectors from the traditional term occurrence methods. These techniques attempt to better describe the semantics of the document contents. Clusters can then be generated based on one of the standard methods just described.

One method is Non-Negative Matrix Factorization (NMF), which learns to recognize semantic features of the text [34]. A corpus of documents can be summarized by a matrix of words versus documents. This matrix is sparse, with many zero values. The algorithm extracts a set of semantic features, combinations of which can

be used to characterize any single document in the collection. Weighted representation by semantic features implies that each document is associated with a subset of a larger number of topics contained within the complete document corpus. These topics are more informative than individual word occurrences. Each semantic feature then consists of semantically related words. In practice, every document is represented as a combination of several semantic features. As an added bonus, semantic features are able to differentiate between multiple meanings of the same word. The utility of NMF in clustering documents can be implemented with the Oracle Data Mining package. The Oracle MEDLINE Text Mining demo contains an implementation of this methodology [35].

Another method is Latent Semantic Analysis. This method creates a statistical word-usage model that permits comparisons of semantic similarity between pieces of textual information [36,37]. An improved version is Probabilistic Latent Semantic Analysis (PLSA) [38]. This method explicitly models document topics. The Expectation Maximization [39] algorithm is then used to fit the model given a set of documents. Each document is defined in terms of a combination of topics based on the model-fitted conditional probabilities of word occurrences in each topic class.

NMF and PLSA methods are computationally heavy. A model is often generated based on a subset of the complete corpus (random or representative document sampling). Similarly to categorizations methods, the models in turn can be applied to other documents to generate a characteristic vector. A particular vector will characterize a new document within the themes identified by the pregenerated model, but the presence of any additional themes or topics will be missed.

A number of clustering methods utilize domain-specific knowledge bases, concept ontologies, or supplemental structured data to improve clustering and emulate automatic classification. This information replaces human category creation and training document assignment. There are obvious reasons for such approaches, such as accumulation and improvement of a domain-specific knowledge area or customization of results toward a specific user profile. The danger is that new or emerging themes will be down-weighed, diluted among more common themes, or completely missed. Continuous effort has to be exerted in updating existing and identifying new themes.

Recommind's Mindserver system utilizes PLSA to automatically categorize (as opposed to cluster) documents [40]. Vivismo's Clustering Engine performs document clustering with a heuristic algorithm aimed at identification of well-described clusters [41]. Megaputer's TextAnalyst uses linguistic and neural network technologies to create a treelike, semantic knowledge representation of a set of documents, which in turn can be used for document clustering [42].

6.2.5 Entity Extraction

Entity extraction is the process of tagging "things'" in the text as specific items such as genes, cell lines, people, chemical compounds, and so on. Two specific requirements for entity extraction are unique and critical to biomedical applications of text mining. Gene name disambiguation and chemical compound name tagging require different approaches.

6.2.5.1 Gene Name Disambiguation

Automated disambiguation of gene and protein names can play a significant role in accelerating disease research and drug development. Researchers are hindered by a lack of standard naming conventions for genes and proteins. Near-frivolous choices of gene synonyms result in gene names like IT, midget, or ER, which means researchers must endure long, and sometimes fruitless, searches for literature about genes or proteins.

The absence of an automated approach for resolving ambiguity between gene synonyms is a key problem [43,44]. Further, text analytics in the biomedical domain are dependent on good gene name tagging and disambiguation. NLP in particular is dependent on term disambiguation, which has been called the "great open problem" of natural language lexical analysis [45]. In the biomedical domain, gene and protein name disambiguation is essential for providing quality protein–protein interactions, disease associations, and other complex biomedical analysis. This problem can also have a substantial impact on the efficiency of IR methods, such as biomedical thesauri [46] or molecular pathway identification [47].

Disambiguation tasks fall into two basic categories: determining if a term refers to a gene or gene product (does CAT refer to "catalase" or "the feline" or "computed axial tomography") and identifying the true meaning of a synonymous gene name or abbreviation (does CAT refer to "catalase" or "chloramphenicol transferase"). Both of these problems occur often with keyword-based searches.

Natural language researchers began focusing on automated approaches to term disambiguation in the late 1980s and early 1990s. Yarowsky [48] used statistical models built from entries in Roget's thesaurus to assign sense to ambiguous words in text, using a Bayesian model to weight the importance of words related to the targeted ambiguous term. Gale, Church, and Yarowsky [49] outlined an approach that used the 50 words preceding and following the target term to define a context for that term's sense. In developing a method for general word sense disambiguation using unsupervised learning, Yarowsky [50] took a document classification approach to solving the problem of general term disambiguation. He also showed in this study that generic English language terms often have only one sense per colocation with neighboring words.

Around the year 2000, computational linguists and computational biologists began to look at term disambiguation in the biomedical domain. A number of researchers [46,51] have proposed solutions that involve manually crafted rules to help natural language processing and IR systems correctly process ambiguous synonyms. These rules are often combined with supervised learning methods (in which systems are provided with human-curated training data) and in some cases unsupervised learning methods (also often referred to as "clustering'). Recent work by Yu and Agichtein [52] compared four different approaches to solving the disambiguation problem: manual rules, fully supervised learning, partially supervised learning, and unsupervised. The manual method is then combined with several of the machine-learning approaches to yield a system capable of extracting synonymous genes and proteins from biomedical literature. Liu et al. [44] also explored a partially supervised learning approach based on disambiguation rules defined in the Unified Medical Language System. In the case of both papers, results are promising, but the systems require a pre-existing set of handcrafted corpora, raising questions about scaling up

to a level where a significant portion of human genes and proteins can be covered. Hatzivassiloglou et al. [47] applied machine learning to the problem of gene, protein, and RNA in text, showing that accuracy levels, as defined by F-measure, of nearly 85% can be attained for classifying terms as belonging to the class of gene or protein. Note, however, that the problem they have tackled is simpler than the one reported here, which seeks to identify a specific gene. SureGene [53] is a recent development in large-scale human gene and protein name disambiguation for MEDLINE abstracts. For genes that have 10 or more associated MEDLINE abstracts, the accuracy of the Suregene models were 80% or higher.

6.2.5.2 Chemical Compound Entity Extraction

Chemical compound entity extraction is a different type of entity extraction. One cannot refer to a complete dictionary or thesaurus of chemical compound names, as there is practically an infinite number of compound names. Systematic names, such as the International Union of Pure and Applied Chemistry (IUPAC) nomenclature system [54], are created from a set of rules that determine the name of a chemical compound based on the chemical structure. As an example we show aspirin, which has the common chemical name acetyl salicylic acid and the IUPAC name 2-acetyloxybenzoic acid. The common names, including aspirin, can be found using a thesaurus-based approach with low levels of ambiguity. The IUPAC name is a systematically determined name that must be tagged in the text using a heuristics-based approach (rules based). In many cases, one will have only the IUPAC name, not a known common name, for a compound.

To effectively mine the literature for gene versus active chemical compounds (inhibitors or activators), one must tag all of the chemical compound references in the literature and disambiguate all the gene names. After this is done, both NLP and statistical text-mining applications will be much more accurate in both precision and recall.There are currently three commercial efforts to provide chemical compound entity extraction: IBM [55], MAI Chem in conjunction with CambridgeSoft [56], and ReelTwo's SureChem [57]. At this time, MDL probably has the best system for extracting not only chemical entities but also reactions; however, at this time, it is not known whether the system will be available commercially (personal communication, Helmut Grotz, MDL, 2004).

6.2.6 Statistical Text Analyses

A wide variety of methods have been developed to analyze the literature using statistical methods that go beyond text categorization or text clustering. The simplest of these methods is co-occurrence analysis. The main limitation of these methods is that they do not perform syntactic/semantic parsing to extract relations between entities. A benefit to the statistical analysis approach is that it is not constrained to relations presented in a sentence or an anaphoric reference[1] from a sentence. As an

1 Reference to an antecedent, for example: "The catalase gene…It has the…"—"It" is an anaphor of "The catalase gene."

example, one may impute a relationship between the gene ApoE and AD due to the significant number of times the two co-occur in the same abstract.

The first reference known to the authors of this technique being applied to biology is the work by Stapley and Benoit [58]. The gene–gene co-occurrence network presented by Stapley and Benoit was also visualized using a network Java applet. This was quickly followed by PubGene [59], which focused on gene interactions expanded to MeSH keyword, disease names, and GO terms found co-occurring with the gene names. The focus of PubGene was analysis of the literature associated with gene-expression gene clusters.

Gene-expression research was the first high-throughput technology driving text mining to support analysis efforts [59–62]. The original method of studying one gene at a time made it much easier to become an expert through manual analysis of the literature and generally keep up to date with additional publications. Gene expression requires learning about sets of genes (gene clusters) on a continuing basis. Various mining technologies, including text mining, are employed to understand the biology behind a gene cluster. Statistical text mining is the most useful technology for the literature analysis of sets of genes. The entire set of literature associated with a gene cluster can be analyzed for links between entity classes such as genes, diseases, biological processes, pathological processes, and so on. One can potentially use gene co-occurring terms in the gene-expression clustering algorithm itself to assist in the determination of gene clusters [63].

Statistical text mining can be used simply to show the frequency of a gene name in the comparison of two subsets of literature, such as AD-related literature versus Parkinson's. It can also take the form of tri-occurrences [64], extracting all sentences that contain protein1, protein2, and an interaction verb/noun.

One of the most important uses of statistical text mining is in developing a broad and shallow understanding of a focused subset of the literature. For example, when initially undertaking the study of a new gene, the first thing of interest is an overview of the gene and its function. This can usually be found in some review article or in the minimal information provided in the gene or protein description provided by Entrez Gene [65] or SwissProt [66]. The next step is to understand what biological processes, pathological processes, interacting proteins, other interested labs/researchers, pathways, and active compounds associated with it. The best way to determine this is to use statistical text mining to generate an association map (somewhat noisy, i.e., high number of false positives but effective) to show all of this related information in a set of tables and network graph representations. This then gives a "map" of information associated with a target. An example of this type of map is seen in figure 6.4. Several groups and commercial entities provide systems to give just this type of overview: BioVista—BEA, IT Omics—LSGraph, InPharmix, IBM—MedTAKMI, Alma Bioinformatica—Alma TextMiner, and others.

6.2.7 Workflow Technologies

Workflow technologies for text mining can make it easier for nonprogrammers to build custom text-mining results. Sophisticated text-mining analyses in support of target validation require significant interaction between the text-mining specialist and the

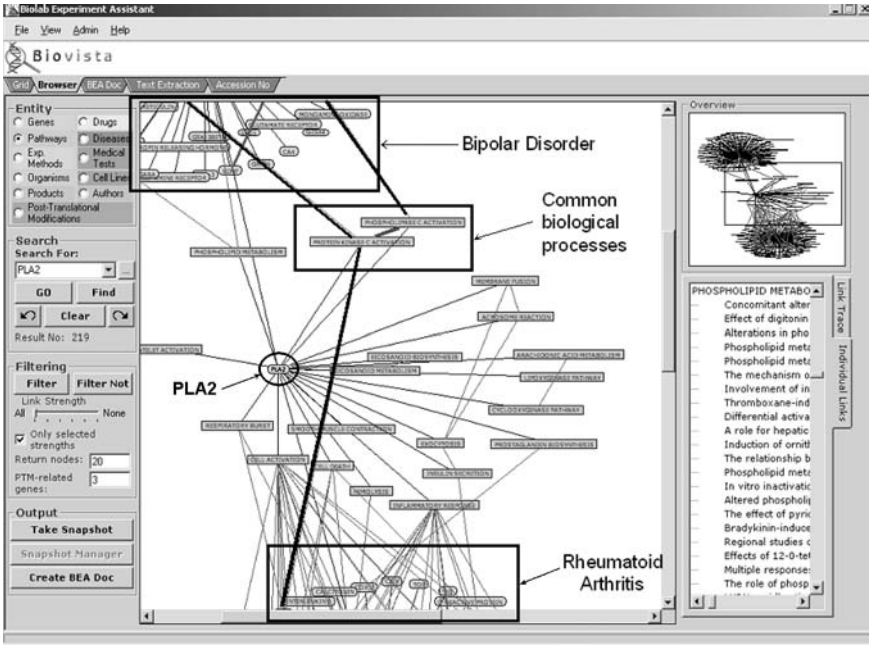


FIGURE 6.4 Network diagram of gene-associated concepts. Landscape map of the gene PLA2 and associated biological processes. Biological processes shared by bipolar disorder (BD) and rheumatoid arthritis (RA) are presented. Protein kinase C activation appears to be integral to the effect of PLA2 on BD and RA. *Courtesy of BioVista*.

domain specialist for reasons indicated in the Introduction. This interaction incurs the cost of an Information System developer to write custom code to generate a text-mining application and results in decreased efficiency of the interaction between the customer and the text-mining specialist. The ideal interaction is for both the text-mining specialist and the domain specialist to work together, interactively analyzing the literature until a result is generated. The result can then undergo final curation by the customer (domain specialist). With workflow technology, such as is available from the Inforsense KDE TextSense product [67] or SciTegic Pipeline Pilot's text analytic workflow module [68], any text-mining specialist can quickly generate custom workflows to provide targeted analyses for customers. Another source of statistical text analysis/workflow technology is SAS Text Miner [69]. See figure 6.5 and figure 6.6.

6.2.8 NLP

NLP is the most sophisticated text-mining technology. NLP has a history of being just a couple of years from being ready, and has been for the last 15 years or so. After a long and unhappy realization that text mining could not be fully automated (and destroying the authors' dreams of lazing on the beach while we burned up racks of computers doing text mining), we realized that NLP technology is ready to use in drug discovery. The fact is these idiot savant programs can work wonders at focusing one's attention on the facts and documents most relevant to a literature study.

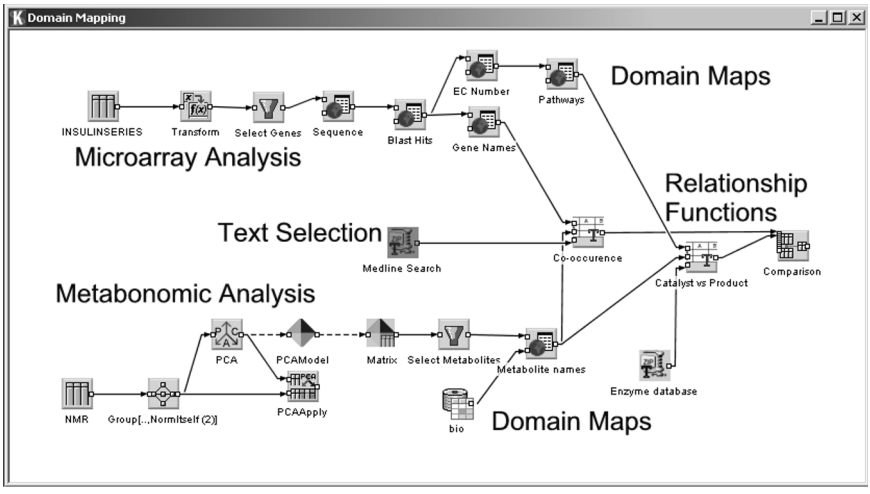


FIGURE 6.5 Metabonomics workflow. The use of text mining to relate two separate data-mining analyses. One part of data mining was microarray analysis of the insulin series of experiments. The other part was a metabonomics analysis using NMR spectroscopy of urine. No database relates metabolic products with gene expression so text mining was used to find co-occurrences of genes and metabolic products as a first step in a follow-up analysis. *Courtesy of InforSense.*

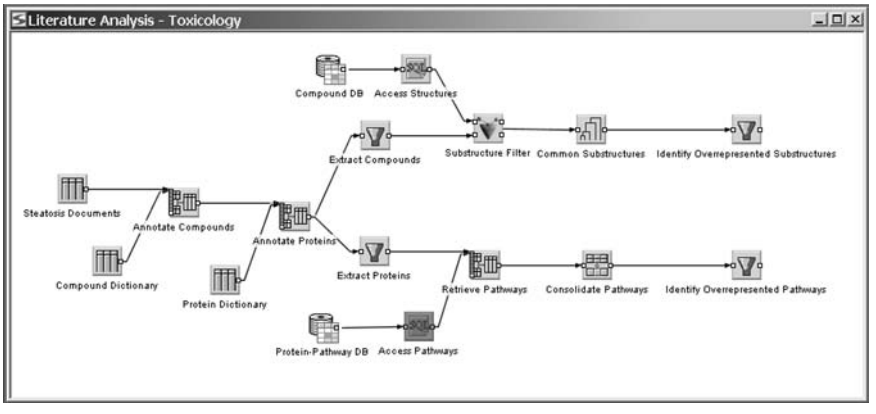


FIGURE 6.6 Toxicology workflow. An analysis of steatosis (a pathological process) is performed by searching for compounds and proteins involved in steatosis-related literature. The compounds identified are subsequently analyzed for common substructures to evaluate any links between steatosis and specific chemical substructures. A parallel workflow identifies proteins and their associated pathways through statistical overrepresentation in association with steatosis. *Courtesy of InforSense.*

NLP technology works by parsing sentences into part-of-speech segments (e.g., noun, verb, preposition, adjective) and then building the part-of-speech segments into noun phrases, verb phrases, prepositional phrases, and so forth. A semantic layer is then added by associating, for example, gene or protein names with the noun

phrases. After this is done, queries can be carried out for noun phrase patterns in every sentence of multiple documents where the noun phrases are semantically typed as protein names. In an analogous way, verb phrases or prepositional phrases can be mapped to biologically relevant concepts such as "interacts with" or "is phosphorylated by." By providing a variety of patterns one can greatly increase recall and precision by making specific patterns with low recall and combining the results. One may find a variety of binding patterns in the text such as seen in table 6.2.

The key difference between agile NLP and standard NLP applications is the workflow. Standard NLP builds many patterns for extracting a focused result set from the literature, such as "collect all protein–protein interactions" or "collect all gene-disease relations." The collected relations go into a database that is queried for results. Agile NLP preparses all of the literature and allows interactive extraction patterns to be built, as seen in next section. Standard NLP can provide both better recall and precision at the cost of a great deal of preparatory work. Extraction templates are built over days, weeks, or months, depending on the level of sophistication. Analysis and evaluations of results are then executed as needed.

The current accuracy of NLP technologies is hard to gauge, but it ranges from 10% recall and 90% precision to 90% recall and 10% precision. The balance most often found in NLP applications is on the order of 25 to 50% for both recall and precision. But this is not the whole picture. NLP technology works by analyzing each sentence syntactically and semantically to extract information (see table 6.3).

TABLE 6.2
Natural Language Processing Extraction Patterns for Protein–Protein Relationship

Sentences
Raf phosphorylates **Mek**.
There is a phosphorylation interaction between **Raf** and **Mek**.
Mek is phosphorylated by **Raf**.

Note: This table presents three ways that the "**Raf** *phosphorylates* **Mek"** relationship can be expressed in the literature. Of course, there are many other ways that this relationship can be presented in the text, and extraction patterns aimed at matching as many instances as possible have to be developed for a successful natural language processing application.

TABLE 6.3
Natural Language Processing

We report here that Cdc2 *interacts* with Orp2, a protein similar to…

Syntactic Layer	Noun	Verb	Noun
Semantic layer	Protein	Protein–protein relation	Protein
Result	Cdc2	Interacts	Orp2

Note: Natural language processing can be thought of as syntactic/semantic querying.

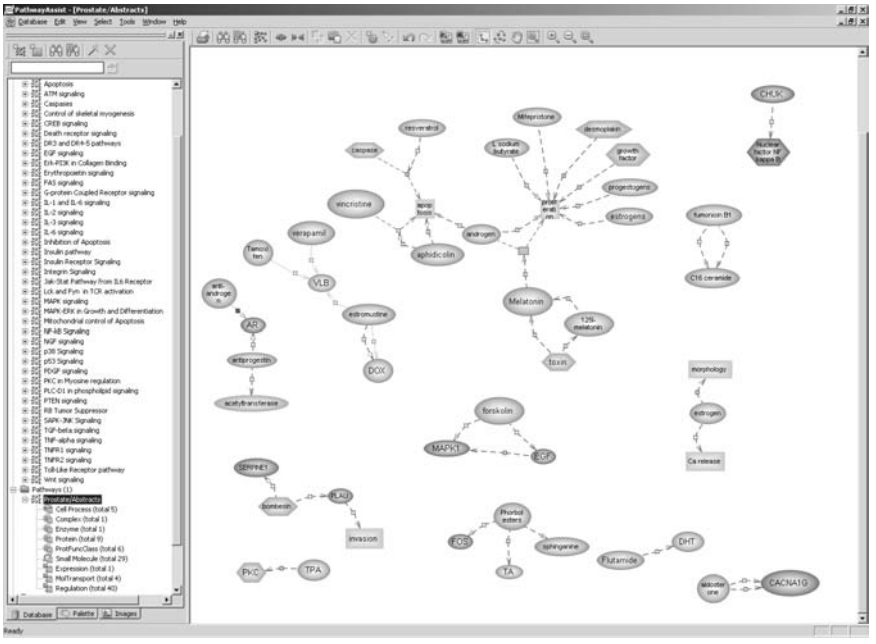


FIGURE 6.7 PathwayAssist. An example of protein versus [protein|cellular process|small molecule] networks generated from full-text journal literature concerning prostate cancer, collected by Quosa. Nodes representing different proteins, small molecules, and cellular processes are represented by different shapes and colors, not clearly identifiable in a grayscale image of a color screenshot.

The recall results are generally shown for each instance of a sentence where a potential relationship can be extracted, such as "Cdc2 interacts with Orp2." However, one is generally interested in whether a fact (as represented by a relationship) exists, not in every instance of the fact being presented in the literature. Redundancy of facts or relations in the text can often greatly increase recall. Often an NLP application will be combined with network visualization as implemented by the GeneWays system [70] or the PathwayAssist application [71]. An example of a NLP result for prostate cancer literature can be seen in figure 6.7. The network interaction presentation of the extracted protein interactions and other associated interactions, such as small molecules, provide much needed context around a given interaction. Another significant benefit is the highlighting of indirect relations between proteins. Indirect relationships are relations where there may exist one or two intermediaries between proteins. A simple table presentation of protein interactions makes it very difficult to determine the networks of interactions that exist between proteins.

6.2.9 Agile NLP: Ontology-Based Interactive Information Extraction

Ontology-Based Interactive Information Extraction (OBIIE) is a new NLP technology. The *interactive* portion of the name implies that the user controls information

extraction "on the fly." Simple keyword searches are often used in early stages of the inquiry. As the user becomes more informed from these queries, the process employs increasingly refined tools and techniques (e.g., NLP and ontologies) to produce the most relevant results focusing in on specific relations to extract.

Ontologies and their structures have been covered in previous sections. But to demonstrate how they are used in OBIIE, an example is used. Imagine a user wants to understand the literature dealing with treatment options for Parkinson's disease. If the user had an ontology of disease phenotypes, the user might extend the search, based on treatment or prophylaxis for *movement disorders* (the Parkinson's disease phenotype). If the user then found that levadopa (a common treatment for Parkinson's disease) belongs to the class of DOPA analogs, he or she might employ a therapeutic compound ontology to explore the relative merits of other DOPA analogs. One could then use a disease ontology to broaden the search to the more general class of neurodegenerative diseases. Conversely, the user might narrow the inquiries, using a protein-naming ontology to learn about specific enzymes and receptors affected by the disease. Whatever path the user takes, the ontologies contain the information about the processes, compounds, and symptoms. Therefore, less in-depth knowledge of the domain is required of the user. The incorporated domain knowledge in an agile NLP application can have a tremendous, practical impact on the usefulness and ease of the search process.

Currently, there is only one example of this agile NLP approach, created by a collaboration between the interactive NLP framework provided by the Linguamatics I2E product and the ontologies of Biowisdom [72]. Ontologies for protein–protein relationships (phosphorylates, forms a complex with, is a substrate of, is proteolyzed by, etc.), protein-naming ontologies, disease, pathological condition, biological process, therapeutic compound ontologies, and others are tightly integrated into the NLP-processing environment. This program uses three major components:

- Linguistic analysis
  - Part of speech information (noun, verb)
  - Morphology (interacts, interacted, … is, was, be …)
  - Syntax ("signaling pathways" "have also been discovered")
- Use of knowledge sources
  - Identify entities: people, places, dates, proteins, and so on
  - Ontologies
- Positional information
  - Searches restricted to words, concepts, and so forth, within the same document, sentence, or phrase

One must provide the corpora for analysis (MEDLINE abstracts, OMIM, proprietary document collections, etc.). The documents are preindexed to provide runtime speed.

An example demonstrates some of the power and scope of this tool for biological literature analysis. Imagine a researcher is studying the apoptotic cysteine protease, caspase 3. To understand how this protease regulates programmed cell death, the researcher might want to explore which serine/threonine kinases are substrates for the enzyme (i.e., are cleaved by caspase 3). Figure 6.8 shows how the researcher

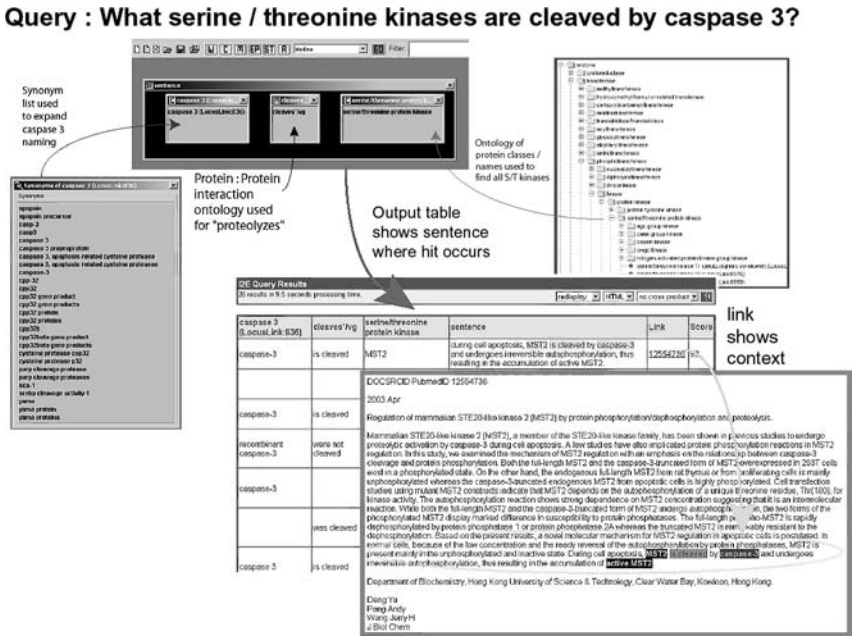


FIGURE 6.8 Interactive natural language processing. An example of the use of OBIIE. A general query is augmented by protein name expansions and protein–protein interaction ontology. The output allows for "drill down" to the specific sentence that meets the search criteria, and the context of the sentence is explored in the original document.

might set up this query using OBIIE. Because caspase 3 has many names in the literature, a thorough search requires expansion of the name to include all known synonyms (cpp32, apopain, YAMA, etc.). These synonyms can all be found with a few keystrokes. Next, an ontology of verb phrases associated with protein–protein interaction can be used to narrow the search to proteolytic interactions. An ontology of protein classes (which also contains the names and synonyms for known proteins) is then used to restrict the search to protein kinases that phosphorylate proteins on serine and/or threonine residues. Finally, the user can confine the search to sentences that contain all three associations (e.g., Caspase 3 cleaves PARP). This query is then run against any corpora that the user has available (in the figure this query is run against the ~13 million abstracts in MEDLINE). Because of the ontologies that are used in this search, all documents are searched for the following pattern: every synonym for caspase 3, any verb phrase that means "is proteolyzed by," and the names (and synonyms) of all the known serine/threonine kinases. This would be a nearly impossible search to run using keyword search due to the massive combinatorial list of possible protein names and verbs. Although the search time is dependent on the computer platform, one can usually expect this search to take only a minute or two. A table showing the database-ready semantically normalized extracted facts and corresponding sentences that match these criteria is returned. In addition, a link to the original document is included to allow the user to understand the context of the sentence. If the search was too narrow, it would only take a few keystrokes to

broaden the search to all protein kinases (serine/threonine kinases and tyrosine kinases), or if the search was too broad it would be simple to add a filter to include only documents that meet the criteria and mention apoptosis (or a disease process) in the same text.

The goal of OBIIE is to produce references and entity relations with the highest relevance to the questions asked. Although ontologies decrease the amount of domain knowledge that the user must have, there is no doubt that a knowledgeable user will be able to ask the best questions. Uninformed users will spend more time educating themselves by means of the query process before they arrive at the same goals.

6.2.10 Visualization

Several different visualization techniques have been applied to the results of text mining. An excellent review of what is currently available for text visualization and the algorithms for data dimensionality reduction is Visualizing Knowledge Domains [73]. Network diagrams are invaluable for presenting indirect relationships and contextual information. Tables are a core technology for presenting facts extracted from the literature. Trees are used for the presentation of thesauri, categorization results, and clustering results. Two-dimensional scatter plots are often used to present text-clustering results in a lower dimensional space. Trend diagrams, such as the number of mentions of a gene against a timeline, are another visualization technique.

Network diagrams are particularly challenging due to the great difficulty in managing the large number of nodes and high connectivity (often greater than 10 edges per node). An example of a protein-interaction map is shown in figure 6.7. The network graph presented contains the relations between the protein, cell processes, and small molecule found in the full text of a set of prostate cancer literature. The power of the network interaction graph is the highlighting of indirect relationships. Of course, it naturally follows that the same representation that is the standard for viewing pathways would be the natural choice for viewing extracted protein network information from the literature.

Several different clustering visualization techniques are available for representing knowledge or textual analysis [73]. The same techniques used to map the domain structure of the literature can also be used for text clustering. Various types of data such as author co-citation, citation links, and themes can be mapped into two or three dimensions to present clusters visually.

An example of the use of document cluster visualization is the theme map, figure 6.9, that can aid in understanding the concepts or themes associated with a set of documents and the similarity between the themes and document clusters. The heatmap presentation of terms versus document clusters in figure 6.10 shows the effectiveness of using heatmaps for more than gene expression. Tying multiple visualizations together can greatly increase the ability to detect trends, as can be seen in figure 6.11.

6.3 EXAMPLES OF TEXT MINING

Some examples of how to use text mining effectively in target validation are discussed next. We do not endorse the specific applications used in these cases as the

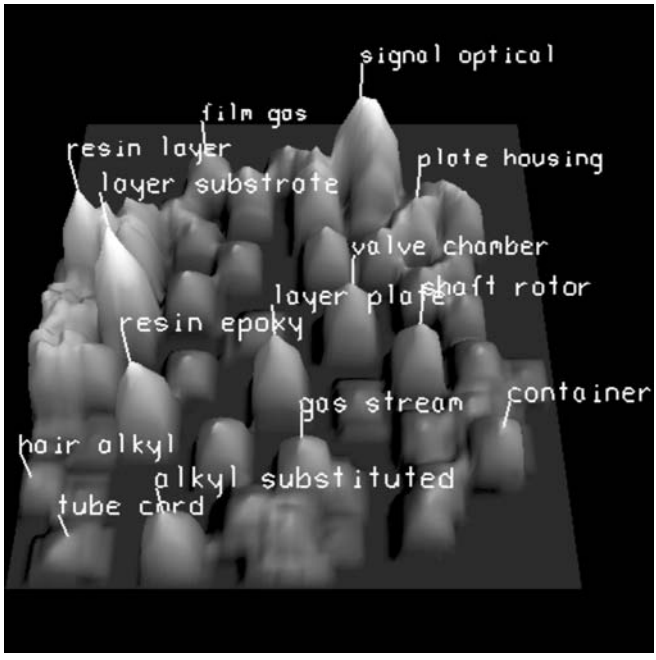


FIGURE 6.9 (See color insert following page 306) Cluster theme visualization with a theme map. OmniViz ThemeMap™ for 5,885 patents in the field of electronics and surface chemistry from 1999. The major themes or concepts are denoted by mountains, which provide a rapid means for seeing the represented concepts. The ThemeMap view also allows the user to build the map according to specific themes to help understand the content in particular subject areas. *Courtesy of OmniViz.*

best text-mining applications for the particular purpose indicated. The tools used are the ones available to the authors at the time of the publication and used to generate real results for the purposes of introducing the reader to actual applications of text mining. Please review table 6.4 for a list of resources that serve as a starting point for further research using the World Wide Web.

6.3.1 Drug-Target Safety Assessment

Acceptance of text mining in the pharmaceutical industry is dependent on its cost effectiveness. Therefore, one of the most compelling areas for the application of text mining is in predictive toxicology. The elucidation of the human genome sequence has lead to plentiful targets, but safety assessment is now the bottleneck. It is estimated that 20% of the cost of all drugs is the result of the cost of failures due to discovery of unacceptable toxicology [74]. Because the cost of development of a new drug is now approximately $800 million [75], even modest improvement of the failure rate by employing text mining to explore and predict potential adverse effects can lead to huge savings to the pharmaceutical industry and to the cost of drugs delivered to the public.

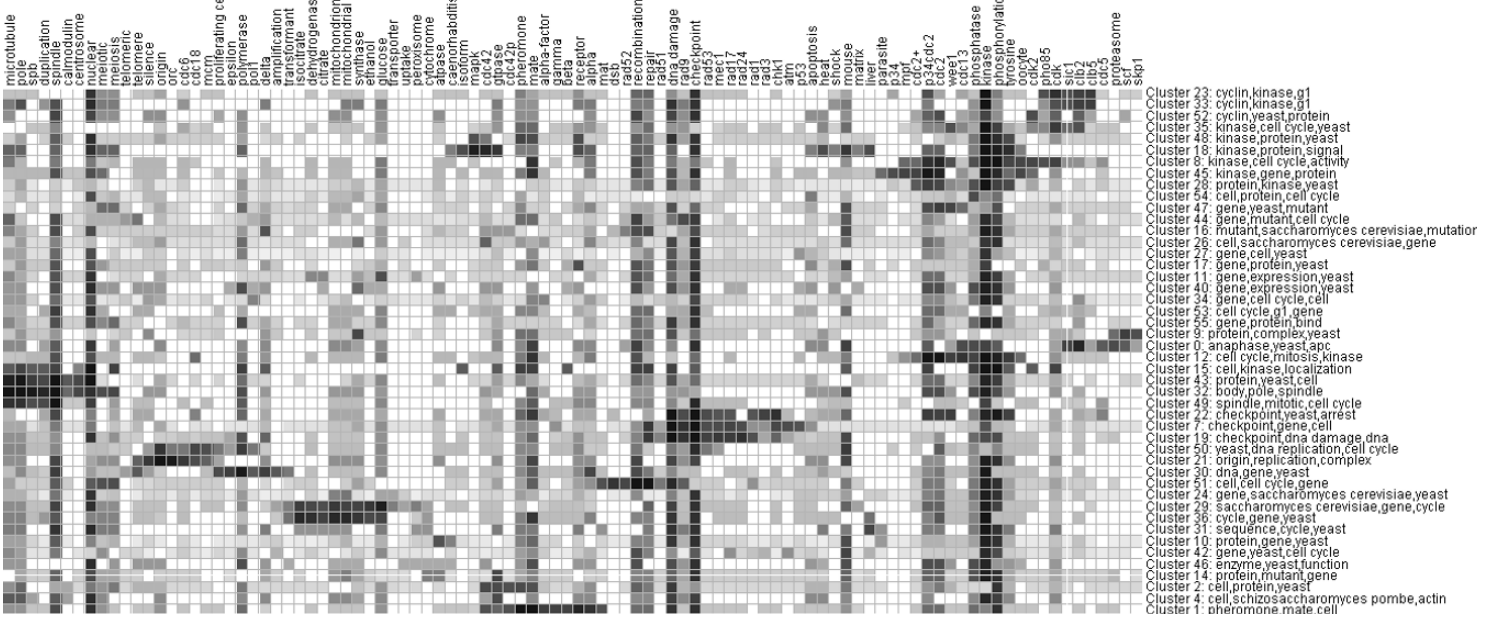


FIGURE 6.10 Cluster theme visualization with a heatmap. The OmniViz CoMet™ visualization provides a quick means to view how attributes are distributed across the dataset. In this example from the biomedical literature on yeast cell cycle regulation, CoMet™ has been configured to show how the major topics (columns) are distributed among the different clusters of documents (rows). The clusters represent over and under representation of co-occurrances between the major topics and documents. Note, the conversion to grayscale from color resulted in lost information in the figure as presented. *Courtesy of OmniViz.*

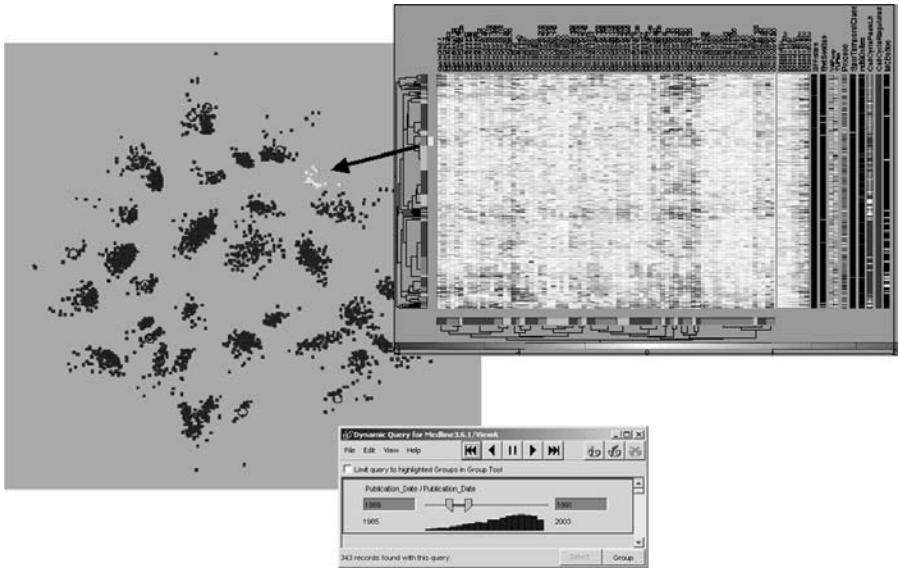


FIGURE 6.11 OmniViz Treescape™. In this composite, a TreeScape™ visualization of gene-expression analysis was linked to an analysis of the relevant literature. Selecting genes (highlight) in the dendrogram causes the relevant documents to also highlight. In this case, the selected genes are referenced in a set of documents that clustered together. This analysis was augmented by the use of a dynamic date query tool, so that trends over time could be uncovered. *Courtesy of OmniViz.*

TABLE 6.4
Text-Mining Web Resources

Resource Name	URL
BLIMP—Biomedical Literature Mining Publications	http://blimp.cs.queensu.ca/
BioNLP.org Web site and mailing list	http://www.ccs.neu.edu/home/futrelle/bionlp/index.html
BNLPB: Bibliography of Natural Processing in Biomedicine	http://textomy.iit.nrc.ca/cgi-bin/BNLPB_ix.cgi
TextMining.org	http://www.textmining.org
Directory of Open Access Journals	http://www.doaj.org

Note: The table presents Web resources that can encourage further learning in regards to text mining. Please be advised that the list will be out of date by the time you read this.

Molecular toxicologists are faced with two problems. The first arises in the target selection process. At this point, the toxicologist is asked to predict if modulation of the function of the protein (receptor, enzyme, ion channel, etc.) will lead to an unacceptable side effect profile. The second type of application of toxicology occurs when the screening process has discovered a compound or compound series. Here,

the molecular toxicologist is asked if the chemical structure or substructure might lead to drug toxicity.

Text mining can aid the safety evaluation in both cases. For the first problem, function and mechanism based toxicity along several lines of inquiry need to be pursued. The user needs to map a toxicology-controlled vocabulary to the target protein name. More practically speaking, the corpora need to be searched for a list of toxicology terms associated with the specific protein name (and synonyms). Second, the protein may be a member of a family (or multiple families) of related proteins. Therefore, the search should be broadened to include toxicological associations to all the members of the family (see fig. 6.6). There may also be information available on the membership of the protein in a signaling cascade or metabolic pathway. The search might also be extended to explore the toxicological association to the pathway name or to all the members of the pathway.

The corpora analyzed are very important. Detailed toxicology information is rarely present in abstracted literature (MEDLINE, Embase, Biosis, etc.). Full-text journal literature will be required for toxicological literature analyses. The authors have had success in using an application, Quosa, [13], to search and retrieve full-text journal articles for additional text-mining analyses. These extended searches will generate a great deal of literature. Therefore, there will be a need to do some form of qualitative meta-analysis of the accumulated sources. This analysis might employ co-occurrence or clustering techniques to show trends and common themes in the compiled literature. It should also be noted that this type of searching could not be done in a fully automated fashion. The searches must be directed by someone with a deep domain understanding to be able to apply supervision. Not all branches of the search are equally likely to achieve results, and as yet, fully automated systems cannot mimic the understanding achieved by trained toxicologists. However, text-mining technologies can grant nearly comprehensive analyses of the literature to a toxicologist and provide significant decision-support capability for their analyses.

The second problem, compound-based toxicity, requires a different set of tools. Although there are many programs for chemical substructure searching, it is difficult to know the meaningful substructural moiety with regards to potential toxicity. Therefore, it is probably most efficacious to be directed at the early stages by chemistry experts. Once the relevant substructure classes are determined, text mining can be initiated. From this point, the analyst searches the literature for linkage between compounds or compound classes and a thesaurus of toxicological terms. If the compounds belong to known pharmaceutical agent families, the search is broadened to explore known adverse side effects of the drug class. If the biochemical mechanism of the undesirable effects is known, the workflow applied to mechanism-based toxicity can be followed (see the aforementioned). There are databases of compound-toxicology associations as well as bibliographic collections in this area, such as the National Library of Medicine MEDLINE/TOXNET [76].

6.3.2 Landscape Map: Disease-to-Gene Linkages

Statistical text analyses can generate the information required to build a "landscape" of related concepts for a text-mining study. Looking at the document or paragraph

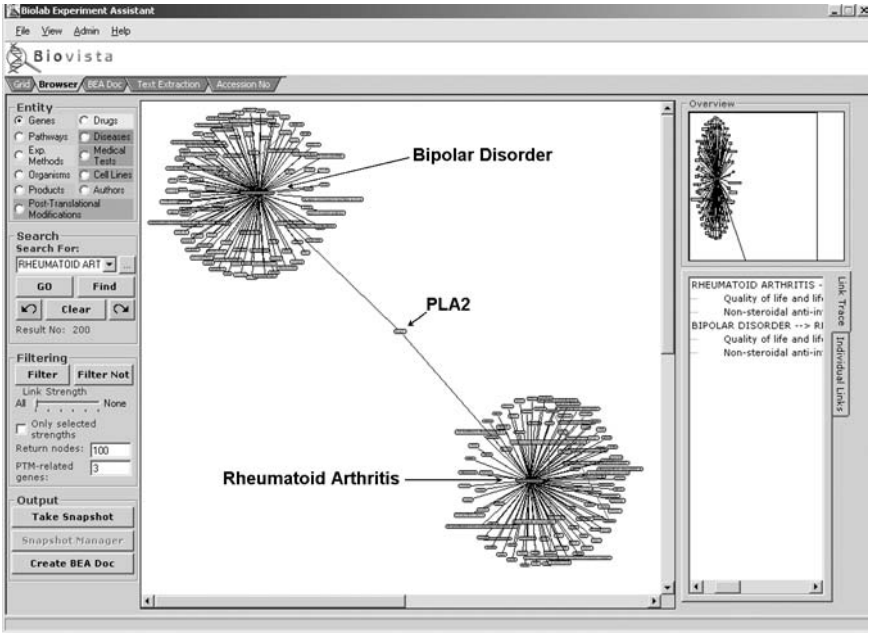


FIGURE 6.12 Landscape of disease-to-gene associations. Associations between bipolar disorder and genes. One gene, Phospholipase A2 (PLA2), has an association with a second concept, rheumatoid arthritis. *Courtesy of BioVista.*

level of associated concepts can show indirect or loose relationships between bioentities or concepts. As an example, the associations between bipolar disorder and genes can be presented as shown in figure 6.12. One gene, PLA2, has an association with a second concept, rheumatoid arthritis. At closer magnification, the thickness of the lines between nodes in the network graph indicates the strength of the association, based on the number of co-occurrences in the literature.

One can also analyze what other diseases or biological process are associated indirectly with bipolar disorder through gene/protein linkages as seen in figure 6.4, which shows the biological processes common to PLA2, bipolar disorder, and rheumatoid arthritis. One can easily navigate the conceptual landscape of the literature through statistical text mining with the appropriate visualization. Of course, many of the suggested relationships are not correct. However, this type of text mining is the fastest way to get an overview on an area of research and is a highly significant source for hypothesis generation.

6.3.3 Applications of Text Mining in the Drug-Discovery and Development Process

A schematic representation of the idealized drug-discovery process is shown in figure 6.1. At present, text mining has been applied mostly to the biological aspects of drug discovery. Due to complex naming systems, the chemical literature is much more

difficult to explore. However, compound and drug ontologies and entity extraction tools are becoming available [55–57,77,78].

There are some areas of the research and development process that have a more or less constant need of text mining throughout drug discovery. Intellectual property and competitive intelligence are constantly monitored to avoid costly legal challenges after a drug has been designed and validated. The patent literature is often searched to understand the intellectual property issues surrounding prior filings for potential targets.

Biomarker selection is another area with broad text-mining applicability. Biomarkers are quite useful in both preclinical and clinical trials to quickly determine efficacy and develop dosing regimens. Biomarkers can be used as surrogate endpoints that can be used to access the efficacy of the test drug in the clinical setting. Text mining can be used to determine gene-to-pathway associations from the literature, which can lead one to potential biomarkers such as excreted proteins downstream in an activated pathway. Text mining can also be used to look for papers with disease-dependent gene expression as another approach for discovering biomarkers. The third area of the drug-discovery process that has broad applicability is safety assessment. (See the Use Case Safety Assessment section earlier.) In the target identification phase, protein function-based toxicities must be considered. After hits have been found, compound-based toxicities become the predominate problem. In the preclinical drug-development stage during animal safety assessment, text mining is needed to determine if any undesirable effects will also be found in humans. In the clinical trial phase, candidate drugs sometimes display adverse effects that were not found in preclinical testing. Text mining is again required to understand the mechanism of the manifested toxicological aberrations.

In the Target Identification phase, the literature is examined for disease to target connections. Use of OBIIE-like (reference Interactive NLP section) systems to determine gene-to-disease, pathway-to-disease, and gene-to-pathway associations have been used to find targets for unmet medical needs [79]. A pharmaceutical company might have years of experience and an extensive compound collection in the area of protease inhibitors and a desire to make obesity control drugs. To leverage this expertise, a text-mining expert might direct efforts to find protease genes related to obesity. The results of this search might find metalloproteases that regulate a process that controls leptins. This search might then be directed toward chemical classes of compounds with good K_i values (K_i is the inhibition constant for the inhibited enzyme) for metalloproteases. This information would then aid the chemists in their synthetic explorations. The time and resource savings of text mining versus launching a complete high-throughput screening campaign are clear. When putative protein targets are identified, text mining can aid in choosing or designing the assays systems and protocols for use in high-throughput screening, secondary screens, and animal models.

Studies to determine absorption, solubility, stability, and bioavailability are typically started after hits have been found. Text mining can be used here to determine appropriate test regimens and if pharmacodynamics and pharmacokinetics information is available for similar structure or on similar targets.

After launch of drugs, literature alerting systems are often used to monitor the acceptance, market position, and competitor status of the drug. WebFountain [80]

is being used to analyze marketing campaigns in the consumer marketplace. One might expect the same technology to be effectively used to monitor drug-marketing results as well.

6.3.4 Systems Biology/Pathway Simulation

Systems biology is an area that academics and pharmaceutical companies are rapidly embracing. Systems biology seeks to understand the complex relationships between signaling cascades, transcriptional control, and disease-relevant phenotypes at the cellular and organism levels. Usually these interrelationships are presented as a computational model. For example, a model might be built of the insulin-signaling pathway as a means to better treat diabetes.

This area, pathway simulation, requires extensive literature validation. The kinetic constants for every reaction in the pathway (usually in the hundreds) and the concentrations of every protein (usually less than one hundred) in the scheme must be informed from the literature. It has been our experience that this kind of detailed information is not available from abstracts. In the past, individuals read papers for the required information to develop these models. Because no one can read literature in its entirety, the readers would stop either when they found the first instance of the necessary fact or after reaching their attention/frustration limit. The manual approach at best does not explore the complete range of values available from different sources and at worst leaves many of the values blank.

The direction that was taken for projects such as the AstraZeneca Epidermal Growth Factor pathway model [79] was to make a list of every protein and other reactants in the pathway (and their synonyms) and search MEDLINE for any abstract that contains one of these names. Kinetic information is rarely in the abstract, but the name of the enzyme for which the kinetic information was derived is almost always found there. The authors then used the semi-automated downloading application, Quosa, to create a local library of all the full-text versions of the papers on the proteins in the pathway. For a moderate-sized pathway, this can entail 50K to 150K full-length articles. These can then be re-searched for the appearance of kinetic terms (e.g., V_{max}, K_m, association rate, etc.). This search usually reduces the 50K to 150K full-length articles to about 100 papers that need to be read, verified, and tabulated by an expert in the field. In addition, the software used in the secondary searches can highlight the sections containing the relevant sentences, so the burden of reading is further reduced. It has been our experience that only about 1 in 50 kinetic constants are available without having access to full-length article sorting. With this workflow, missed data can be mostly eliminated and the coverage of the literature is limited only by how much of the journal articles are available in electronic form. It is difficult to compute comparative time savings for this text-mining approach, as the manual methods are never taken to completion. To complete a project of this size it would take about 1 or 2 weeks for someone with some biological knowledge and reasonable text-mining proficiency.

Another area of pathway analysis that benefits from text mining is in *de novo* pathway generation. This means building a pathway when the cascade is unknown. For example, genetic linkage analysis often finds a gene associated with a disease

but no known signaling cascade associated with the gene. In these cases, text mining can be used to build up a protein: protein association network with the discovered gene. This is followed by associating the protein interactors with known pathways. By inspecting or graphically visualizing the network, it is possible to elucidate likely pathway connections for the new gene.

6.3.5 Text Categorization

Hakenberg et al. [81] presented one example of using a text-classification engine for filtering text in the paper. The paper describes the use of an SVM classifier based on SVMLight [20]. Finding papers describing kinetic parameters is difficult. It is further complicated by the need to collect full-text journal articles to filter, as most of the information required for the text categorization is only found in the full-text article and not in the abstract. For example, the authors randomly downloaded 4,582 documents from several journals published between the years 1993 and 2003. An expert manually reviewed a random sample of 200 papers to determine the frequency of papers containing kinetic parameters. Twelve percent were found to contain kinetic parameters with a 95% confidence interval of 8 to 17.5%. To generate the positive training set, a search was run using keywords such as K_m, V_{max}, kinetic parameter with a resulting set of 791 papers, of which 155 were found through manual curation to be true positives. The accuracy of the system was evaluated using a fivefold cross-validation resulting in 60% precision with 49% recall. The user selects the recall/precision balance; the indicated precision/recall balance was deemed most useful for this particular text-classification filter.

6.3.6 Clustering: Literature Discovery

The following document-clustering examples utilize Oracle Text k-Means clustering, document gist, and theme extraction. Searching MEDLINE with two distinct queries in the article titles and selecting the top 300 documents for each query provided the 600 documents for this clustering example. The queries were "lymphoma" and "Alzheimer's disease."

k-Means clustering was performed on the title and abstract text of each document, using the Oracle Text hierarchical k-Means method. To evaluate the most basic performance of clustering, the total number of desired clusters was specified as 2. The expectation is that each resulting cluster will contain the 300 documents for each distinct topic.

Table 6.5A details each cluster's quality score, size, and defining terms in order of significance to the cluster. The size of each cluster agrees with the sizes of the two topic-based document groups, and quality of each cluster is high (82%–85%). Examination of the cluster descriptive terms provides hints but fails to clearly indicate a biological significance of the clusters. To more clearly elucidate such significance, table 6.5B presents the top themes and sentence-based text "gist" based on the complete corpus of each cluster. Themes represent concepts from a knowledge base represented by numerous hierarchically organized thesauri. Biomedical themes are clearly predominant, with a clear lymphoma versus AD grouping. The reported gist consists of three sentences from all the abstracts in each cluster that best reflect the themes identified.

This simple example can be expanded by requesting 10 clusters from the k-Means algorithm. Because the algorithm is hierarchical in nature, the initial split of the root cluster is identical to the 2-cluster example. Further splitting steps are performed until 10 clusters are obtained while maximizing the relative cluster centroid distances and intercluster quality. Figure 6.13 shows the hierarchy of the final

TABLE 6.5
Document Clustering: Two-Cluster Example

A

Cluster	Quality	Size	Cluster Tokens
1	0.85	300	**CONFERS**, 70.9, WMH, TEMPERATURE, RESTORATIONS, PRECURSORS, ADD, CONFIRMATORY, SUBSCALES, PRESYMPTOMATICALLY, THA, ANOMIA, EASE, SOUGHT, APPROVAL, DYSCALCULIA, CIRCUMFERENTIAL, INTERACTING, 2.74, CVD, METHYLATION, MORTEM, DIABETES, PRESERVE, 8OHDG, PHOSPHOGLYCERATE, DOLICHOLS, SUPERVISORY, BLOT, APOE4, EFFECTIVENESS, 223, CWPS, COS, HOMEMAKER, APP717, INFLUENCES, INCURRED, COMMANDS, RELIANT, AL, OUTCOMES, NONMEMORY, DYADS, PREPARATIONS, SUBSERVE, INSULIN, 0.9, 802, PEDIGREE, DENTATE, WORKGROUP, LIMB…
2	0.82	300	**UNDERSTOOD**, CYTOFLUOROMETRY, SIALOSYL, COCULTIVATED, 3A, WEIGHT, HISTIOCYTOSIS, 1981, ALLO, FLOWER, PERIODIC, VARIABLY, AOTUS, MIDFACIAL, NORMALLY, KANSAS, HISTOPATHOGENESIS, SCHIFF, CIS, CH31, SPACING, IDENTIFIES, TRISOMIES, LOCATIONS, REMARKS, BECOMES, CSI, BENCE, TDT, 51.7, DLBCLS, FAULTS, PROMPTED, M6, CODED, ACTVP, DPDL, GLYCOLIPID, RFS, FUSION, CONTINUING, TRISOMIC, SWOG, ALKALINE, DUPLICATION, EROSIONS, THIE, LPS, PARATHORMONE, 8, 572, SUFFERED, CYTOGENETICALLY, THYROID, SNC, PRIESS, SECRETORY, COMPUTED…

B

Cluster	Themes	Gist
1	Alzheimer's disease (35) studies (20) significance (17) dementias (15) groups (5), learning (4) consideration (1) analysis (1) relation (1) control (1)	As part of an effort by the NIA-Alzheimer's Disease Cooperative Study to evaluate new measures of efficacy for their utility in treatment studies, the Severe Impairment Battery (SIB) was examined in a 1-year evaluation of change across a wide range of AD severity. In studies comparing neuropathology protocols for AD, several groups have found that the CERAD diagnosis most closely correlates with measures of dementia severity, such as the Mini-Mental State Examination (MMSE). We compared the performance of two subgroups of mild Alzheimer's disease patients (e.g., EAD and typical Alzheimer's disease; TAD) on neuropsychological and associated measures.

TABLE 6.5
Document Clustering: Two-Cluster Example (continued)

Cluster	Themes	Gist
2	lymphomata (44) cells (27) lymphomas (16) lymphocytes (15) difference (15) viruses (4) groups (4) relation (1) anatomy (1) immune systems (1)	European phase II study of rituximab (chimeric anti-CD20 monoclonal antibody) for patients with newly diagnosed mantle-cell lymphoma and previously treated mantle-cell lymphoma, immunocytoma, and small B-cell lymphocytic lymphoma. PURPOSE: Mantle-cell lymphoma (MCL), immunocytoma (IMC), and small B-cell lymphocytic lymphoma (SLL) are B-cell malignancies that express CD20 and are incurable with standard therapy. Peripheral T-cell non-Hodgkin's lymphomas of low malignancy: prospective study of 25 patients with pleomorphic small cell lymphoma, lymphoepitheloid cell (Lennert's) lymphoma and T-zone lymphoma.

Note: **A.** Cluster attributes for a two-cluster example. Cluster Quality is the average percent similarity of a document in the cluster from the cluster *centroid*. Cluster tokens are the individual words/terms that statistically distinguish each cluster. The most significant terms are listed first. A non-expert cannot easily define a more general theme of each cluster. **B.** Cluster *themes* and *gist* better defines the biological content of the clustered documents. It becomes clear that cluster 1 is related to Alzheimer's disease while cluster 2 contains Lymphoma documents.

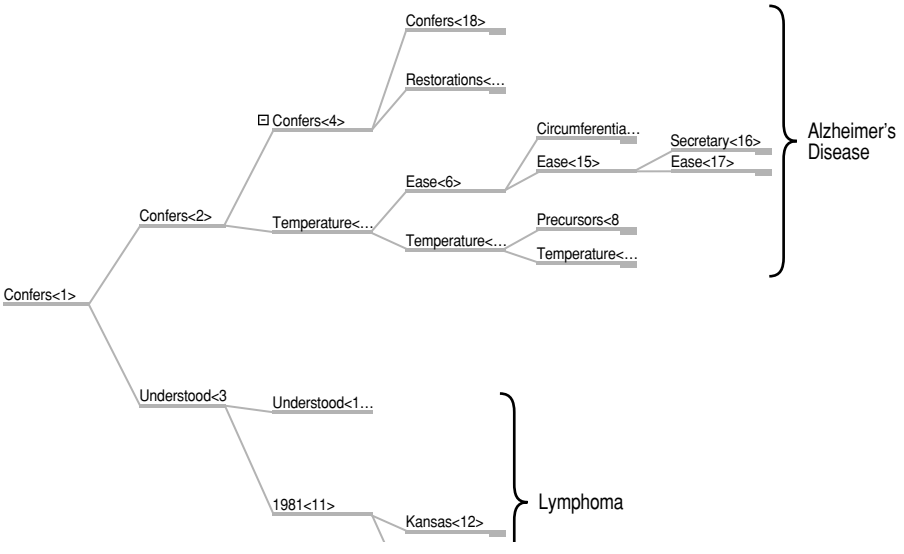


FIGURE 6.13 Clustering example—cluster hierarchy. Document cluster hierarchy separating documents about lymphoma from those about Alzheimer's disease. The cluster node names signify the most significant term for all documents within that cluster.

TABLE 6.6
Document Clustering: Ten-Cluster Example

Cluster	Size	Themes
18	77	Alzheimer's disease (39), studies (21), evaluation (21), dementias (19), control (17), learning (4), consideration (2), analysis (1), groups (1), relation (1)
19	66	Alzheimer's disease (46), genes (21), proteins (18), studies (18), increase (17), change (16), significance (16), relation (8), control (2), groups (1)
14	23	Alzheimer's disease (96), pathology (11), United Kingdom (9), disease (8), APOLIPOPROTEIN (8), vascularity (7), studies (7), dementias (5), genetics (5), groups (2)
16	16	Alzheimer's disease (90), studies (18), disease (12), evaluation (12), benefit (10), psychiatric (8), proteins (7), dementias (7), methods (7), relation (1)
17	19	Alzheimer's disease (87), genetics (18), disease (13), biologicals (11), implication (11), discussion (11), cholinergics (10), therapeutics (9), risk factors (8), groups (1)
8	48	Alzheimer's disease (73), dementias (13), disease (13), groups (5), relation (4), conclusion (1), consideration (1), analysis (1), quality (1), control (1)
9	51	Alzheimer's disease (50), significance (19), proteins (17), change (16), activity (16), studies (16), groups (5), consideration (1), analysis (1), relation (1)
10	138	lymphomata (60), cells (31), leukemia (16), lymphocytes (15), difference (15), B-cells (15), lymphomas (14), viruses (5), biology (1), relation (1)
12	76	lymphomata (93), malignancy (20), Burkitt's lymphoma (14), progression (10), HODGKIN (8), authors (7), intestines (4), disease (4), viruses (4), TRANSL (3)
13	86	lymphomata (80), grading (15), lymphomas (11), evaluation (9), ranks (2), analysis (2), immune systems (2), positions (1), groups (1), relation (1)

Note: Continued grouping of the documents from the two-cluster example identifies a range of subtopics within each disease document set.

clusters. The lymphoma cluster from the basic example was further split into 3 child clusters, whereas the AD cluster split into 7. This indicated that the variety of topics in the AD documents is greater. Table 6.6 presents the sizes and top themes for each cluster (reflecting the vertical ordering of the clusters in the figure). The distinctions between themes of various clusters are more readily apparent when viewed in a heatmap (fig. 6.14). Roughly, lymphoma documents can be grouped into the following topics: biology of B-cell lymphomas, treatment of malignant lymphoma, and lymphoma clinical studies. The AD documents indicate the following topics: clinical studies, psychiatric studies, pathology, and biology.

It is clear that clustering works well for separating clearly distinct topics. Clarity begins to decrease as the representative topics within a group of documents begin to share attributes.

6.4 FINANCIAL VALUE OF TEXT MINING

Several factors that yield objective numbers on the financial benefit of using text-mining approaches have been identified. Surveys by Outsell [82] have shown that researchers

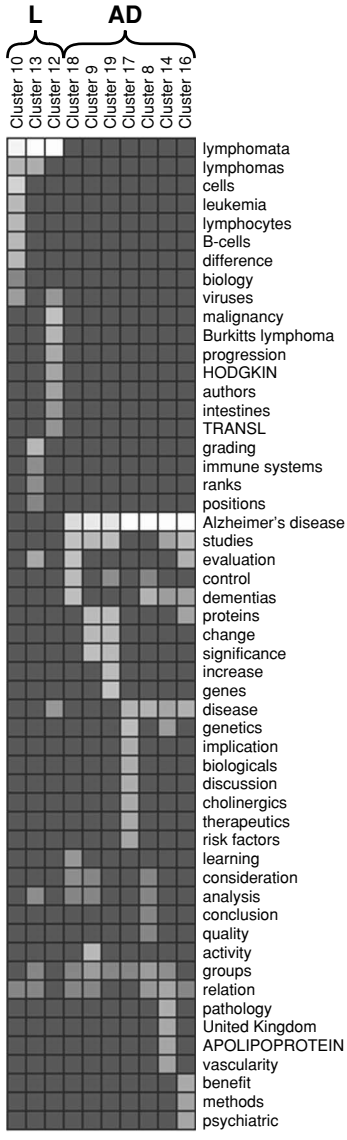


FIGURE 6.14 Clustering example—cluster theme heatmap. The top themes/concepts for each cluster are displayed in a heatmap. Biological themes distinguishing the two main topics are clearly distinct. The few shared themes between clusters of the two unrelated diseases, lymphoma (L) and Alzheimer's disease (AD), are *analysis, disease, evaluation, groups,* and *relation*.

across several industries including biotech spend about nine hours a week accessing and analyzing the literature. The authors' text-mining evaluations have shown better than a factor of 10 time savings compared to manual search and curation. Nine hours is roughly 20% of one's work week. If we assume that half that time is in search of

particular facts and overviews that text mining can help locate, then nearly 10% of a researcher's time overall can be saved and reinvested in other productive work. For 1,000 researchers, the total effort that can be reinvested is 100 man-years per year or at least $10 million/year based on an accounting of $100,000 per researcher per year.

In addition to savings from making each researcher more efficient, text mining may very well lead to savings in more efficient pipeline management, based on better information. Consider, for example, another current topic of great interest: the "front loading" of safety concerns. Industry surveys [75] have indicated about 50% of all potentially therapeutic compounds undergo attrition due to safety concerns. About 50% of the compounds exhibiting toxicity had some indication in the literature that was not noticed until after experimental evidence arose (personal communication, Scott Boyer, AstraZeneca, 1994). So for every four drug projects moving from compound hit to candidate drug, we have one that can potentially end sooner. One should not propose killing a drug project based on the literature but rather reorganize or prioritize toxicology studies based on the literature indications to speed up attrition. This could free up a substantial portion of the expensive lead optimization budget for other projects. It is still an open question whether this potential can be realized, but it is certainly worth pursuing.

It has been suggested (personal communication, Aris Persidis, BioVista, 1994) that better experimental design through the use of text mining can save a lab up to 33% in both time and reagent costs. These numbers come from an informal survey of several academic labs. Many labs developing assays for targets do not realize which cell lines and reagents are most effective. This is due, in large part, to the literature search databases only providing abstracts, and the pertinent information to their experiment design is only found in the full-text journal article.

6.5 DISCUSSION

Text mining is not a task to be undertaken lightly. It requires a significant commitment to manage the documents and applications required to deliver the technology in a useful manner. Even the basic document management is not easy. Managing hundreds of gigabytes to terabytes of text, PDF, Word, PowerPoint, and other document types is a challenging task that has to be appropriately engineered.

There is also a significant cultural challenge to text mining. The corporate library makes a natural partner along with Information Technology (IT) and Informatics in delivering the technology of text mining to the drug-discovery scientist, but it is a significantly different technology and type of result to what has been deployed previously by these groups. There are two challenges for the corporate library: (a) librarians have a great deal of experience with document retrieval but much less with automated information extraction, and (2) they are used to working with outside vendors for tools, not with in-house IT/Informatics personnel. The IT Department tends to have relatively little experience working with text except to store it. The Informatics Department is not used to working with unstructured data. There are challenges for all of the involved parties and a very significant requirement for tight partnerships in delivering text mining.

The interaction required between the text-mining expert and the drug-discovery scientist is much closer than is required of text search. The subtleties of the questions being posed are often so specific to a drug project that even a senior drug-discovery scientist turned text-mining specialist has difficulty narrowing the results to the correct set without the drug scientist customer directly involved. There are many more variables to manage in performing text mining than there are in text search queries.

The authors going into text mining were surprised that text search technology is still difficult to deploy. Text search is a supporting technology for text-mining efforts. Given the extensive development of text search in conjunction with the Internet, we had incorrectly assumed that search technology would be easy to deploy and scale well. As with any computing system, the correct technology has to be selected for the task and has to be appropriately engineered for the application.

Knowledge management is an area that will get more focus in the future. The tidal wave of semantically typed knowledge that is being developed through text mining is going to have to be managed. The multirelational ontology (fig. 6.15) as a database appears to be one method for solving this knowledge management crisis. As with any new high-throughput technology such as gene expression or high-throughput imaging, the chokepoint in the process moves on. As one problem is solved, a new one arises. The idea of mapping extracted knowledge into an ontology structure (fig. 6.16) may avoid some of the pitfalls found in data integration technologies. Mapping columns in a database, which may not be semantically normalized, to a column of data in another database can result in a significant source of errors.

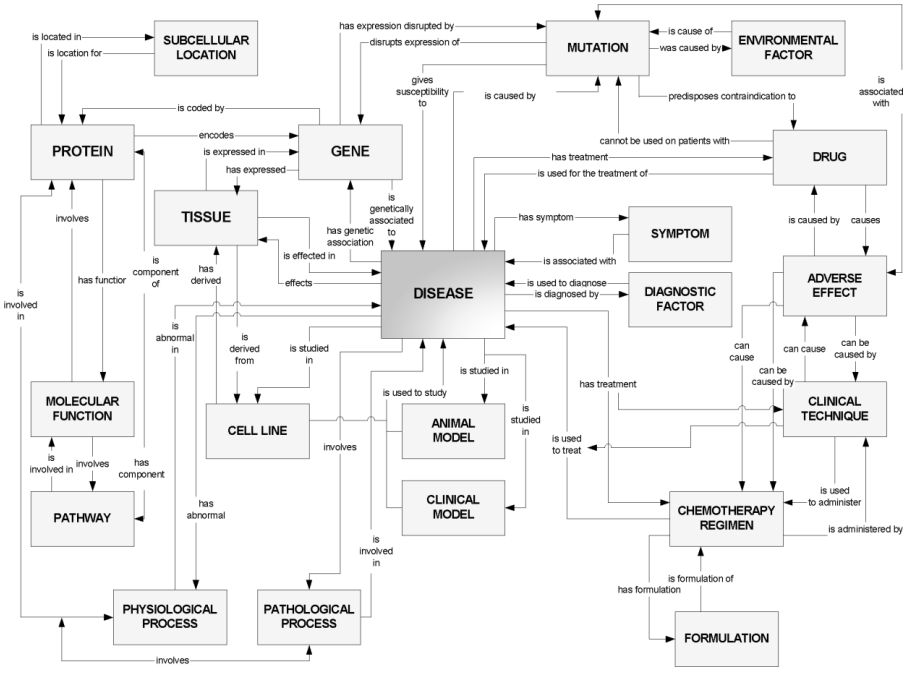


FIGURE 6.15 Multirelational ontology. Representation of multiple, semantically distinct ontologies and the relationships that are mapped between them. *Courtesy of Biowisdom, Ltd.*

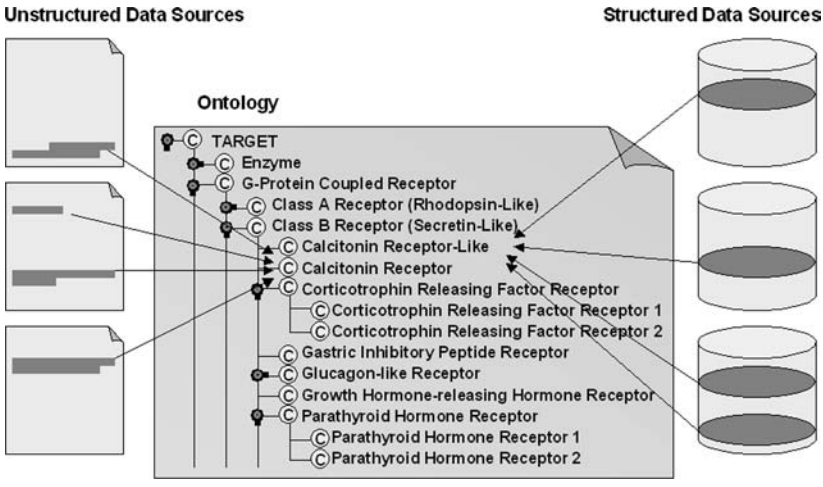


FIGURE 6.16 Data mapping in semantic integration. An example of how both individual relationships and specific items or semantically congruent knowledge in databases can be mapped to an ontology to provide a semantic database structure. *Courtesy of Biowisdom, Ltd.*

All of the information or knowledge can be integrated as individual entities with rules as to what metadata or relations can be used for mapping classes of entities together.

There is too much literature to review or stay current with manually. The knowledge wrapped up in the literature can provide a significant competitive advantage for the companies that are utilizing text mining. Estimates in the industry suggest 90% of drug targets are derived from the literature. Realistically, most of the research for drug discovery is actually produced external to an individual company and is publicly available through published articles and abstracts or patent application background data. Who can afford not to employ methods that allow more efficient and comprehensive surveys of the vast scientific, commercial, and patent literature?

ACKNOWLEDGMENTS

We express our appreciation for the efforts of Jerilyn Goldberg, AstraZeneca, in assisting in formatting and editing the text. Further, we appreciate the materials provided by BioWisdom, Linguamatics, InforSense, BioVista, and OmniViz for the enhancement of this chapter on text mining.

REFERENCES

1. Swanson, D. R., and N. R. Smallheiser. 1994. Assessing a gap in the biomedical literature: Magnesium deficiency and neurologic disease. *Neurosci Res Comm* 15:1–9.

2. BIOBASE. BIOBASE Biological Databases. http://www.biobase.de
3. GVK Biosciences Private Ltd. GVK BIO. htttp://www.gvkbio.com
4. Molecular Connections Private Ltd. Molecular Connections. http://www.molecular-connections.com
5. Jubilant Organosys Inc. Jubilant Biosys. http://www.jubilantbiosys.com
6. Ingenuity Systems, Inc. Ingenuity Systems. http://www.ingenuity.com
7. Copyright Clearance Center. Copyright.com. http://www.copyright.com
8. American Association for Cancer Research. American Association for Cancer Research. http://www.aacr.org
9. Thomson Scientific. BIOSIS. http://www.biosis.com
10. Elsevier. EMBASE.com. http://www.embase.com
11. Thomson. MicroPatent. http://www.micropatent.com
12. Thomson. Delphion. http://www.delphion.com
13. QUOSA. QUOSA. http://www.quosa.com
14. Harris, M. A., J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, et al. 2004. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res* 32:D258–61.
15. U.S. National Library of Medicine. Unified Medical Language System Metathesaurus. http://lhncbc.nlm.nih.gov/csb/CSBPages/UMLSproject.shtml
16. Yahoo! Inc. Yahoo! http://www.yahoo.com
17. Rennie, J. D. M. 2001. Improving multi-class text classification with Naïve Bayes. MA thesis. Massachusetts Institute of Technology.
18. Quinlan, J. R. 1993. *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann.
19. Vapnik, V. N. 1995. The nature of statistical learning theory. Heidelberg, Germany: Springer-Verlag.
20. Joachims, T. 1998. Text categorization with support vector machines: Learning with many relevant features. In *10th European Conference on Machine Learning: 1998,* 137–48. Heidelberg, Germany: Springer-Verlag.
21. Joachims, T. 2002. *Learning to classify text using Support Vector Machines*. Boston: Kluwer Academic.
22. Reel Two. Reel Two. http://www.reeltwo.com
23. van Rijsbergen, C. J. 1989. *Information retrieval*. 2nd ed. London: Buttersworth.
24. Kowalski, G. 1997. *Information retrieval systems: Theory and implementation*. Norwell, MA: Kluwer Academic.
25. Buckley, C., and A. F. Lewit. 1985. Optimizations of inverted vector searches. In *Proceedings of the Eighth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* 97–110. New York: ACM Press.
26. Cutting, D. R., D. R. Karger, J. O. Pedersen, and J. W. Tukey. 1992. Scatter/Gather: A cluster-based approach to browsing large document collections. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 318–29.
27. Zamir, O., O. Etzioni, O. Madani, and R. M. Karp. 1997. Fast and intuitive clustering of web documents. In *Proc. Third Int'l Conf. Knowledge Discovery and Data Mining,* pp. 287–290.
28. Kubat, M., I. Bratko, and R. S. Michalski. 1997. A review of machine learning methods. In *Machine learning and data mining methods and applications,* ed. M. Kubat, I. Bratko, and R. S. Michalski, 3. New York: Wiley.

29. Gennari, J. H., P. Langley, and D. Fisher. 1989. Models of incremental concept-formation. *Artif Intell* 40 (1–3):11–61.
30. Iliopoulos, I., A. Enright, and C. Ouzounis. 2001. Textquest: Document clustering of Medline abstracts for concept discovery in molecular biology. *Pacific Symposium on Biocomputing,* 384–95.
31. Jones, G., A. M. Robertson, C. Santimetvirul, and P. Willett. 1995. Non-hierarchic document clustering using a genetic algorithm. *Information Research* 1 (1). Available at http://InformationR.net/ir/1-1/paper1.html
32. Steinbach, M., G. Karypis, and V. Kumar. 2000. A comparison of document clustering techniques. In *KDD, Workshop on Text Mining*. Available at www.cs.smu.edu/~dunja/KDDpapers/Steinbach_IR.pdf
33. Cover, T. M., and J. A. Thomas. 1991. *Elements of information theory.* New York: Wiley.
34. Lee, D. D., and H. S. Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401:788–91.
35. Oracle. MEDLINE text mining demonstration. http://www.oracle.com/technology/industries/life_sciences/ls_sample_code.html
36. Dumais, S. T. 1991. Improving the retrieval of information from external sources. *Behav Res Methods* 22:229–36.
37. Deerwester, S., S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science* 41:391–407.
38. Hofmann, T. 1999. Probabilistic latent semantic analysis. In *Uncertainty in artificial intelligence*. Stockholm, Sweden.
39. Dempster, A., N. Laird, and D. L. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B* 39:1–38.
40. Recommind, Inc. Recommind white paper. http://www.recommind.com
41. Vivísimo. *How the Vivísimo Clustering Engine works*. http://vivisimo.com/docs/howitworks.pdf
42. Ananyan, S., and A. Kharlamov. *Automated analysis of natural language*. http://www.megaputer.com/tech/wp/tm.php3
43. MacNeil, J. S. 2003. What big pharma wants. *Genome Technology* 29:31–8.
44. Liu, H. F., S. B. Johnson, and C. Friedman. 2002. Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS. *J Am Med Inform Assn* 9:621–36.
45. Resnik, P., and D. Yarowsky. 2000. Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation. *Nat Lang Engineering* 5:113–33.
46. Aronson, A. R. 2001. Ambiguity in the UMLS metathesaurus. In *National Library of Medicine*. Available at http://skr.nlm.nih.gov/papers/references/ambiguity01.pdf
47. Hatzivassiloglou, V., P. A. Duboue, and A. Rzhetsky. 2001. Disambiguating proteins, genes, and RNA in text: A machine learning approach. *Bioinformatics* 17, (Suppl. no. 1):S97–106.
48. Yarowsky, D. 2000. Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities* 34:179–86.
49. Gale, W. A., K. W. Church, and D. Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities* 26:415–39.
50. Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics,* 189–96. Cambridge, MA.

51. Rindflesch, T. C., and A. R. Aronson. 1994. Ambiguity resolution with mapping free-text to the UMLS metathesaurus. *J Am Med Inform Assn* 240–4.
52. Yu, H., and E. Agichtein. 2003. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics,* ISMB Suppl.:340–9.
53. Podowski, R., J. Cleary, N. Goncharoff, G. Amountzias, and W. Hayes. 2005. SureGene, a scalable system for automated term disambiguation of gene and protein names. *Journal of Bioinform Comput Biol* 3 (3):1–29.
54. IUPAC. International Union of Pure and Applied Chemistry. http://www.iupac.org
55. Boyer, S., and J. Cooper. 2004. Automatics chemical structure indexing from plain text. In *2004 International Chemical Information Conference, Annecy, France*. Available at http://www.infonortics.com/chemical/ch04/slides/boyer.pdf
56. Data Harmony, Inc. M.A.I. Chem. http://www.dataharmony.com/products/maichem.htm
57. Reel Two. SureChem. http://surechem.reeltwo.com/
58. Stapley, B. J., and G. Benoit. 2000. Biobibliometrics: Information retrieval and visualization from co-occurrences of gene names in Medline abstracts. *Proceedings of the 2000 Pacific Symposium on Biocomputing* 5:526–537.
59. Jennsen, T. K., A. Laegreid, J. Komorowski, and E. Hovig. 2001. A literature network of human genes for high-throughput analysis of gene expression. *Nat Genet* 28 (1): 21–8.
60. Glenisson, P., B. Coessens, S. Van Vooren, J. Mathys, Y. Moreau, and B. De Moor. 2004. TXTGate: Profiling gene groups with text-based information. *Genome Biol* 5 (6):R43.
61. Raychaudhuri, S., J. T. Chang, P. D. Sutphin, and R. B. Altman. 2002. Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. *Genome Res* 12:203–14.
62. Masys, D. 2001. Linking microarray data to the literature. *Nature Genet* 28:9–10.
63. Raychaudhuri, S., J. Chang, F. Imam, and R. Altman. 2003. The computational analysis of scientific literature to define and recognize gene expression clusters. *Nucleic Acids Res* 31:4553–60.
64. Albert, S., S. Gaudan, K. Heidrun, A. Raetsch, A. Delgado, B. Huhse, H. Kirsch, et al. 2003. Computer-assisted generation of a protein-interaction database for nuclear receptors. *Mol Endocrinol* 17:1555–67.
65. Maglott, D. R., J. Ostell, K. D. Pruitt, and T. Tatusova. 2005. Entrez Gene: Gene-centered information at NCBI. *Nucleic Acids Res* 33, Database Issue:D54–8.
66. Boeckmann, B., A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, J. M. Martin, et al. 2003. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res* 31:365–70.
67. InforSense Ltd. InforSense KDE TextSense. http://[www.inforsense.com
68. SciTegic Pipeline Pilot. 2005. Available at http://www.scitegic.com/products_services/pipeline_pilot.htm
69. SAS Institute Inc. SAS Text Miner. http://www.sas.com/technologies/analytics/datamining/textminer
70. Rzhetsky, A., I. Iossifov, T. Koike, M. Krauthammer, P. Kra, M. Morris, H. Yu, et al. 2004. GeneWays: A system for extracting, analyzing, visualizing, and integrating molecular pathway data. *J Biomed Inform* 37:43–53.
71. Daraselia, N., A. Yuryev, S. Egorov, S. Novichkova, A. Nikitin, and I. Mazo. 2004. Extracting human protein interactions from MEDLINE using a full-sentence parser. *Bioinformatics* 20:604–43.
72. Milward, D., M. Bjareland, W. Hayes, M. Maxwell, L. Oberg, N. Tilford, J. Thomas, et al. 2005. Ontology-based interactive information extraction from scientific abstracts. *Comp Funct Genomics* 6:67–71.

73. Bomer, K., C. Chen, and K. Boyack. 2003. Visualizing Knowledge Domains. In *Annual Review of Information Science & Technology.* Vol. 37, ed. B. Cronin, 179–255. Medford, NJ: Information Today.
74. Tufts Center for the Study of Drug Development. 2005. *Outlook 2005*. Boston: Tufts Center for the Study of Drug Development. http://csdd.tufts.edu/InfoServices/OutlookPDFs/Outlook2005.pdf
75. Tufts Center for the Study of Drug Development. 2001. *Research milestone*. Boston: Tufts Center for the Study of Drug Development.
76. U.S. National Library of Medicine. TOXNET. http://toxnet.nlm.nih.gov/index.html
77. BioWisdom. Biowisdom. http://www.biowisdom.com
78. National Center for Biotechnology Information, U.S. National Library of Medicine. PubChem. http://pubchem.ncbi.nlm.nih.gov/
79. De Graaf, D. 2004. The impact of a detailed mechanistic model of a complex signaling pathway on drug discovery and development. In *Mathematical models in signalling systems: June 17, 2004* Nashville, TN.
80. A tangled web of business information. 2004. Available at http://www.research.ibm.com/thinkresearch/pages/2004/ 20040119_webfountain.shtml
81. Hakenberg, J., S. Schmeier, A. Kowald, E. Klipp, and U. Leser. 2004. Finding kinetic parameters using text mining. *OMICS: A Journal of Integrative Biology* 8:131–52.
82. Commissioned study. *Super information about information managers (Super I-AIM)*. Conducted by Outsell, Inc. Sponsored by Dialog, Factiva and KPMG. 2001. Available at http://www.outsellinc.com

7 Pathways and Networks

Eric Minch
Merck Research Laboratories

Ivayla Vatcheva
German Cancer Research Center

CONTENTS

7.1 INTRODUCTION

7.1.1 What Is a Pathway?

For purposes of this chapter, we define a pathway as a description of the mechanism of the typical process by which a cellular or organismic function is realized. This definition makes two points. First, a pathway is not a mechanism, but a description of mechanisms, that is, it exists as a model subject to our investigative needs and not as an entity itself. Second, a pathway must have functional interpretations, that is, it must correspond to a goal or need we can identify in the cell or organism. Thus this definition does not provide a purely objective criterion but tosses the problem into the laps of people who want to define "function" at the cellular or organismic level. It also removes from consideration any sequence of biochemical interactions that do not have such an interpretation.

Although the discovery and characterization of pathways begin with biochemical investigations yielding data about interactions among molecules and their relation to cellular types and locations, it does not end there. In this chapter, we include some description of the lower level of representation (e.g., yeast two-hybrid results, 2D gels, GC-MS, etc.), but the main focus is on processes at the middle level (e.g., metabolite profiles and flux rates) and higher levels (e.g., reaction sets, transport processes, signaling and regulatory effects), that is, levels involving functional interpretations.

7.1.2 What Are the Relationships among Different Sorts of Pathways?

There are three commonly recognized families of pathways: metabolic, signaling, and regulatory pathways. Metabolic pathways, historically the first to be recognized, are those involving the synthesis (anabolism) and breakdown (catabolism) of compounds (sugars, starches, amino acids, lipids, etc.). They are typically categorized according to their function: producing energy, transferring energy, synthesizing proteins, making toxins harmless, providing transportable forms of substances, providing storable forms of substances, and so on. Signaling pathways involve modifications to successions of proteins, beginning with detection of a condition internal or external to the cell and propagating to an effect on metabolic or regulatory pathways. Regulatory pathways involve interactions between DNA segments, either relatively direct (promoter, enhancer, operon, or short RNA segment) or relatively indirect (in which one or more genes change transcription activity of one or more other genes through effects on signaling or metabolic pathways).

Both signaling and regulatory pathways can be regarded as control mechanisms that modify metabolic activities according to current conditions recognized by the cell. Although these three families are commonly treated separately, in the cell there is extensive interaction among them. Signaling pathways are typically initiated by changes in the concentration of metabolites or other small compounds, and they usually result in either metabolic changes (via activation or inhibition of enzymes) or regulatory changes (via transcription factor binding). Regulatory pathways, in

turn, result in changes in the production of proteins that participate in either metabolic or signaling pathways, and metabolic pathways, in addition to their acknowledged anabolic and catabolic function, are continually exercising upward control on signaling and regulatory pathways via the monitoring of metabolite concentrations. All types of pathways are thus parts of a whole system; treating any type in isolation from the others will lose a part of the whole story.

7.1.3 What Is the Significance of Pathways to Drug Discovery and Development?

One use of pathway identification and analysis is in answering target validation questions, both in disease modeling and in toxicological assessment. If the putative target is a receptor upstream in a signaling pathway, will blocking of the receptor disrupt other vital cell functions? Analysis of this problem involves finding all pathways in which the protein is involved and assessing its role there to predict off-target effects. If the target is suppressed, and thus the targeted pathway is inactivated, are there alternative pathways that can accomplish the function of the disabled pathway? Finding alternative pathways involves determining other trajectories that converge to the same phenotypic attractor. Does the cell possess compensating mechanisms that make a putative target unattractive? Changes in concentration of a protein resulting from drug administration may be neutralized by a feedback control loop. Such a compensation mechanism could be overcome by a high drug dosage, but this could produce undesirable side effects. Analysis of the positive and negative effects of changes is required to address this problem.

Target validation involves more than analysis of signaling and regulatory interactions: metabolomic profiles can be used as markers in toxicological assessment or disease diagnosis. If a particular pattern of concentration changes in a metabolite set is found to be reliably related to a disease condition or to toxicity, this can be followed up by identifying the pathway(s) whose disruption is most likely to be causing the pattern and hence the condition.

Finally, an often overlooked use of pathway analysis is in drug production. Many drugs can be synthesized by genetic engineering (e.g., insulin) or metabolic engineering (e.g., penicillin) of microorganisms or plants [Stephanopoulos, Aristidou, and Nielsen 1998]. The optimization of yield quantity and purity involves many of the same tools as are employed in understanding disease mechanisms.

7.2 PATHWAY DATA

7.2.1 Data Acquisition Techniques

To talk about the use of pathways, we must first describe the experimental methods that are used to infer these pathways, for without the constraint of data, metabolic and control pathways could be hypothesized to mediate any conceivable function. In this section we mention methods for monitoring the activity of genes, proteins, and metabolites. Other clues (phenotype, fMRI, electrochemical, etc.), however, have also been used.

7.2.1.1 Transcriptomics

Various methods are available for detecting and quantifying gene-expression levels, including northern blots, differential display, polymerase chain reaction after reverse transcription of RNA, and serial analysis of gene expression. These techniques are used primarily to measure the expression levels of specific genes or to screen for significant differences in mRNA abundance.

High-throughput expression studies have matured in the form of array-based (or chip) technologies that have mainly been developed along two lines. In cDNA array experiments, many gene-specific polynucleotides are individually arrayed on a single matrix. The matrix is then simultaneously probed with fluorescently tagged cDNA representations of total RNA pools from test and reference cells, allowing one to determine the relative amount of transcript present in the pool by the type of the fluorescent signal generated [Duggan et al. 1999]. In oligonucleotide arrays the basic principle is the same, but the spots on the chip contain short oligos instead of cDNA clones from known genes. The oligos are exposed to a solution containing many copies of the target DNA. If the oligos are tagged, with either fluorescent dye or radioactive label, hybridization between oligos and matching DNA can be detected.

Array technologies require a workflow of activities, including the production of the probes, labeling and hybridization of the target, data extraction from fluorescent images, and subsequent storage and mining of the collected data.

7.2.1.2 Proteomics

Messenger RNA levels contain valuable information about the cell state and the activity of genes. Nevertheless, messenger RNA is only an intermediate on the way to the production of functional protein products. Methods for monitoring protein levels therefore have advanced significantly in the past years.

Western blotting (immunoblotting) is a technique that allows one to measure the amount of a protein in a sample by using antibody specific to that protein. The method is suitable for comparing relative amounts of a protein in different samples. Absolute quantities require suitable calibration and are difficult to obtain. Western blot experiments have supported the development of mathematical models of signal transduction that predict how a cell reacts to external stimuli (e.g., Bentele et al. 2004). Nevertheless, the technique yields average numbers over a cell population and can miss events happening on the level of individual cells. This and other limitations of Western blotting are addressed by various quantitative microscopy techniques.

Western blots and other antibody-based approaches, including enzyme-linked immunosorbent assay and immunoprecipitation, traditionally have been used to study signaling pathways. Although these methods are sensitive and specific, the reliance on high-quality antibodies limits the applicability of these approaches to large-scale studies. To address this deficiency, high-throughput quantitative proteomics technologies are now advancing [Tao and Aebersold 2003].

The most mature approach for quantitative protein profiling is two-dimensional gel electrophoresis (2DE). In 2DE, the sample under investigation is fractioned and

subsequently the complex protein components separated. The 2DE gels may then be stained to reveal the resolved protein spots and imaged to identify protein expression changes between samples. 2DE gel analysis is combined with mass spectrometry (MS) to identify the proteins in the individual spots.

Another approach that has proven promising in a range of proteomics studies is the generation of quantitative protein profile using the isotope-coded affinity tag (ICAT) reagent method [Gygi et al. 1999]. The use of ICAT reagents allows for the relative quantitation of two samples. The samples are isotopically labeled (one heavy, one light) through a reactive group that specifically binds to cysteine residues. The reactive group also contains an affinity tag (biotin), which can be used to selectively isolate the labeled components and is then removed before MS analysis.

The ICAT technique supports the detection and quantification of differences in the protein profiles in cells or tissues in different states or organisms. The method holds significant promise for the discovery of diagnostic protein markers, the detection of new targets, and as a tool for further understanding biological mechanisms and processes [Tao and Aebersold 2003].

To gain more understanding in the information-processing flow of signaling networks, one needs to measure the spatial and temporal distribution of proteins involved. Such data support the development of fine-grained spatial and dynamical models that can be analyzed to obtain detailed spatiotemporal predictions on cellular behavior. Experimentally, such data are obtained by using fluorescent activity reporters (biosensors) that can track local signaling events over time inside individual living cells [Meyer and Teruel 2003].

A common strategy to obtain localization data of a protein is to tag it with green fluorescence protein (GFP) or other fluorescent tags and to monitor the time-course of its translocation in response to signal stimulation. Monitoring is accomplished by taking sequential fluorescence-microscopy images during and after cell stimulation. In addition, fluorescence-recovery after photobleaching (FRAP) can be used to measure the mobility of signaling proteins. In FRAP, a fluorescently tagged protein is photobleached in a small region of the cell. Subsequently, fluorescently tagged molecules surrounding the bleached region diffuse and recover the fluorescence in this region. Important parameters that can be measured with this technique include diffusion constants and the proportion of tightly bound (or immobile) protein.

In a next step, the spatial dynamics of molecular activation and interaction mechanisms can be studied with fluorescence resonance energy transfer (FRET) [Miyawaki 2003]. FRET is the nonradiative transfer of excited-state energy from an initially excited donor fluorophore to an acceptor fluorophore. The donor and acceptor fluorophores are usually tagged to a pair of proteins, the interaction of which is studied. The widely used donor and acceptor fluorophores are from the class of autofluorescent proteins like GFP. FRET is an accurate technique for measuring molecule proximity (< 10 nm) and to monitor the localization and dynamics of protein–protein interactions and conformational changes in living cells. FRET has been used mostly in conjunction with microscopy imaging to visualize signaling events in time and space [Sekar and Periasamy 2003]. There are also a number of ways to quantify the FRET efficiency [Miyawaki 2003]. Quantitative data in terms

of spatiotemporal distribution of the concentration of interacting proteins can thus be obtained.

Single-cell experiments allow key parameters that define the dynamic properties of cellular signaling networks that are not available from cell population experiments to be extracted (e.g., Western blotting). Such parameters include delay time constants between signaling steps or bistability in the activation process.

Better understanding of the connectivity and dynamics of cellular networks requires that quantitative readout of cellular behavior is obtained for a range of physiologically relevant conditions. Ideally, we would like to activate or inhibit all intermediate steps in a network while monitoring the activity profiles (or concentration) profiles of all molecules involved using, for example, FRET. Currently such perturbations are possible using only a limited set of known small-molecule inhibitors and activators. Another promising perturbation strategy is RNA interference [Hannon 2002].

7.2.1.3 Metabolomics

Metabolomics (or metabonomics) refers to the identification of low-weight molecules (metabolites) from samples in a particular physiological or developmental state and quantification of their abundance; helpful reviews are given in [Fiehn 2003, Lindon 2004]. Analysis by hyphenated techniques such as gas chromatography mass spectrometry (GC-MS), liquid chromatography-mass spectrometry (LC-MS), or capillary electrophoresis-mass spectrometry (CE-MS) provides a detailed chromatographic profile of the sample including both identification and measurements of the relative or absolute amounts of the components. Direct spectroscopic techniques such as nuclear magnetic resonance (NMR) and direct mass spectrometry techniques such as quadrupole-time of flight (QToF), Fourier transform-ion cyclotron resonance (FT-ICR), and matrix assisted laser desorption/ionization (MALDI) can be used to classify and determine differences between samples by highlighting the changes occurring in a given biological context, but without prior specification of metabolites to be identified are less useful for pathway analysis.

7.2.2 Databases

Most data acquired by the aforementioned and by more classical methods are maintained in proprietary databases and not made generally accessible. Many academic or government groups, however, and quite a few private efforts willing to share data under license, have made their results available. In this section we list some of the databases that provide pathway descriptions.

7.2.2.1 Primarily Metabolic Databases

PUMA/EMP/WIT [Selkov et al. 1998] include a series of projects for representation of metabolic pathways underway since the early 1990s under the direction of Evgeni Selkov, of Argonne National Laboratories and Integrated Genomics. They provide metabolic databases, search facilities, and visualization capabilities. EcoCyc, MetaCyc, and BioCyc [Karp 2000] include a series of databases and associated ontologies,

based first on annotated sequences and curated functional data from *E.coli* and later on similar data and literature from about 150 species. It was developed under the direction of Peter Karp, of SRI, starting in the early 1990s, and includes facilities for metabolic database search, pathway visualization, and causal inferencing.

KEGG [Kanehisa et al. 2002] is a similar effort underway since the late 1990s under the direction of Minoru Kanehisa at Kyoto University. It also provides metabolic databases, search facilities, and visualization capabilities.

PathDB (table 7.1) has been under development since before 2000 at the National Center for Genome Resources. It focused originally on plant metabolism (*Arabidopsis thaliana*) but is not in principle restricted to plants, and has recently begun including yeast (*Saccharomyces cerevisiae*). It includes a relatively large database of compounds, proteins, reactions, and pathways, plus visualization and search/navigation tools.

Metabolic Pathways of Biochemistry (table 7.1) has been maintained since 1998 by Karl Miller [1998] of The George Washington University's Biochemistry Department. It contains over 20 pathway diagrams with details on the structure and other characteristics of the molecules.

The University of Minnesota has been adding to the Biocatalysis/Biodegradation Database [Ellis et al. 2003] since the early 1990s. It includes 144 pathways of microbial metabolism, mostly for xenobiotic degradation.

7.2.2.2 Signaling, Regulatory, and General Databases

The Biomolecular Interaction Network Database (BIND) [Bader 2001; Bader, Betel, and Hogue 2003] was begun in 2000 by Chris Hogue, of Toronto's Samuel Lunenfeld Research Institute at Mt. Sinai Hospital, and Francis Ouellette, of Vancouver's Center for Molecular Medicine and Therapeutics at the University of British Columbia. Although originally intended to handle protein–protein interactions and complexes, its data model was extended until it became capable of handling quite general objects and processes in cellular metabolism. Its continued development is under the auspices of Blueprint, a collaboration between IBM and MDS Proteomics. BIND archives biomolecular interaction, complex and pathway information accompanied by a graphical analysis tool to view the domain composition of proteins in interaction and complex records.

The Munich Information Center for Protein Sequences [Mewes et al. 2004] is a group established within the Max Planck Institute for Biochemistry that has been developing methods for proteomic and metabolic analysis of gene expression data since the early 1990s.

Signaling PAthway Database (SPAD) (table 7.1) was developed at Kyushu University and includes clickable maps for 16 signal transduction pathways. It is still maintained but has not increased coverage since 1998.

TRANSPATH [Choi et al. 2004], originally developed at the German Research Center for Biotechnology and based on Cell Signaling Network Database [Takai-Igarashi, Nadaoka, and Kaminuma 1998], is currently a commercial offering. It includes signaling and gene-regulation pathway data extracted mostly from primary

TABLE 7.1
Web Resources for Pathway Databases, Standards, and Software Referred to in Text

Name	Citation/Organization	URL
AfCS	*Nature*	http://www.signaling-gateway.org/
aMAZE	van Helden et al. [2000]	http://www.ebi.ac.uk/research/pfmp/
BBD	Ellis et al. [2003]	http://umbbd.ahc.umn.edu/
BBID	Becker et al. [2000]	http://bbid.grc.nia.nih.gov/
BIND	Bader et al. [2001]	http://www.bind.ca/
BioD	Cook, Farley, and Tapscott [2001]	http://www.rainbio.com/BioD_home.html
BioPathways		http://www.biopathways.org/
BioPAX		http://www.biopax.org/
BioUML		http://www.biouml.org/
CellML	Lloyd, Halstead, and Nielsen [2004]	http://www.cellml.org
Cytoscape	Shannon et al. [2003]	http://www.cytoscape.org/
DDLab	Wuensche [2003]	http://www.ddlab.com/
DIP	Salwinski et al. [2004]	http://dip.doe-mbi.ucla.edu/
Discoverer	Bayesware	http://www.bayesware.com/products/discoverer/discoverer.html
EMP	Selkov et al. [1998]	http://emp.mcs.anl.gov/
Kegg	Kanehisa et al. [2002]	http://www.genome.ad.jp/kegg/
Metacore	GeneGo	http://www.genego.com/about/products.shtml#metacore
MIM	Kohn [1999]	http://discover.nci.nih.gov/kohnk/interaction_maps.html
MIPS	Mewes et al. [2004]	http://www.mips.biochem.mpg.de/
MPB		http://www.gwu.edu/~mpb/
PANTHER	Applied Biosystems	https://panther.appliedbiosystems.com/pathway/
PathBLAST	Kelley et al. [2004]	http://www.pathblast.org/
PathDB	NCGR	http://www.ncgr.org/pathdb/
Pathway Articulator	Jubilant	http://jubilantbiosys.com/pd.htm
Pathway Assist	Iobion	http://www.stratagene.com/products/displayProduct.aspx?pid=559
Pathway Enterprise	Omniviz	http://www.omniviz.com/applications/pathways.htm
PathwayLab	Innetics	http://innetics.com/
Pathways Analysis	Ingenuity	http://www.ingenuity.com/products/pathways_analysis.html
Pathways Data System	Ozsoyoglu, Nadeau, and Ozsoyoglu, 2003	http://nashua.cwru.edu/pathways/
PATIKA	Demir et al. [2002]	http://www.patika.org/
Reactome	Joshi-Tope et al. [2005]	http://www.reactome.org
SPAD		http://www.grt.kyushu-u.ac.jp/spad/
STKE	*Science*	http://stke.sciencemag.org/
TAMBIS	Baker et al. [1999]	http://cagpc.cs.man.ac.uk/tambis/manual/
WIT	Selkov et al. [1998]	http://wit.mcs.anl.gov/WIT2/

literature, plus facilities for network visualization and tools for analyzing gene-expression data.

The Database of Interacting Proteins (DIP) [Salwinski et al. 2004] encompasses protein–protein interaction data. DIP started life as an academic project, but since 2001 has been generally available under commercial license.

The Biological Biochemical Image Database [Becker et al. 2000] is an interesting collection of keyword-searchable signaling and regulatory maps from multiple sources (conference presentations, journal articles, books, hand-drawn, etc.). It is supported by the National Institute on Aging, with coverage of over 700 genes.

Reactome [Joshi-Tope 2005] is a publicly accessible effort covering metabolic and signaling pathways in humans and model organisms. The reactions are curated directly by domain expert biologists rather than extracted from the literature. It includes pathway search and visualization capabilities.

Applied Biosystems has recently made PANTHER Pathway (table 7.1) available, which includes over 60 mostly signaling pathways, including keyword search, visualization, and highlighting of gene sets from expression analysis.

The Signal Transduction Knowledge Environment (table 7.1) is another expert-curated database, focusing exclusively on signaling pathways. It currently contains about 40 detailed and annotated pathway descriptions, including citations and graphical "connection maps." Originally a collaboration between Stanford University and the journal *Science*, it is now available by subscription.

The Alliance for Cellular Signaling (AfCS) (table 7.1), sponsored by the journal *Nature,* also has a signaling database. It has assembled a smaller number of pathways, as its focus to date has been on the signaling molecules, for which it has compiled a set of about 4,000 detailed summaries.

7.2.3 Standards

The standards issue is somewhat contentious in this field for several reasons. First, only in the last few years has it been recognized that standards for pathway representation are important, and becoming more so with each passing year. Second, several established databases with incompatible representations have served as *de facto* standards for a number of years. Third, the entry of commercial vendors into the field has upped the ante, making market competition a factor.

Every database schema, every class hierarchy, every ontology embodies an implicit candidate for standards. Here we list only the entities concentrating on developing standard representations explicitly for pathway data.

The Systems Biology Markup Language [Hucka et al. 2003] started in 2000 as part of the ERATO Kitano Systems Biology Project. Its primary purpose is to serve as a representation language for the storage and communication of biochemical models. It is an XML-based medium with structures representing compartments, species, reactions, unit definitions, parameters, and kinetic rate expressions. Its definition has undergone continual refinement by its contributing teams from the

Kitano group, the Caltech group, and the several simulation groups involved in its development.

The Cell Markup Language [Lloyd, Halstead, and Nielsen 2004] has been under development since 1999 by scientists from the University of Aukland and formerly also Physiome Sciences. It is an open XML-based exchange format for the general description of the structure, mathematics, and state of cellular models.

The BioPathways Consortium (table 7.1) is a group founded in 2000 by Eric Neumann and Vincent Schaechter. Its memberships include participants from industry, academia, government, and other research institutes, and its goals include developing and consolidating technologies for representation and communication of information concerning pathways and biochemical interactions. The BioPathways Consortium collaborates with the BioPAX Project.

The BioPAX Project (table 7.1) was begun by Chris Hogue, Peter Karp, and Chris Sander in 2002, with the goal of developing a standard exchange format for pathway data. The Level 1 specification included metabolic pathway information. Level 2, currently under development, will also include more generic molecular interactions.

The TAMBIS Project (Transparent Access to Multiple Bioinformatics Information Sources) [Baker et al. 1991] started in the late 1990s at the University of Manchester and has developed a powerful data model together with a knowledge base affording near-natural language queries to federated distributed biological databases.

The Protein Function and Biochemical Pathways project, and its aMAZE database [van Helden 2000] are an ongoing effort since 1998 under the leadership of Shoshana Wodak of the European Bioinformatics Institute and the Free University of Brussels. It includes an encompassing object-oriented data model and has begun development of pathway visualization and analysis tools.

A number of other proposals include innovative ideas but have not yet achieved broad support (BioUML [table 7.1]; BioD [Cook, Farley, and Tapscott 2001]; Molecular Interaction Maps [Kohn 1999; Kitano 2003]; Diagrammatic Cell Language [Maimon and Browning 2001]).

7.3 PATHWAY ANALYSIS

7.3.1 Data Analysis Techniques

7.3.1.1 Topological Analysis

The most intuitive representations of biochemical networks are graphs [Deville et al. 2003; Newman 2003]. A graph is defined by a set of nodes representing the entities of the network (genes, proteins, or metabolites) connected via a set of edges representing chemical, physical, or functional interactions among the entities. Graphs can be directed, in which case the edges (or arcs) express casual or temporal relationships between the biochemical species (e.g., A activates B, or A is transformed into B). Hypergraphs can be used in cases where two or more compounds react to form a product, or multiple proteins cooperatively regulate the expression of a gene. In bipartite graphs, reactions are also modeled as nodes, and edges signify that a biochemical species is an input or an output of a reaction (see fig. 7.1).

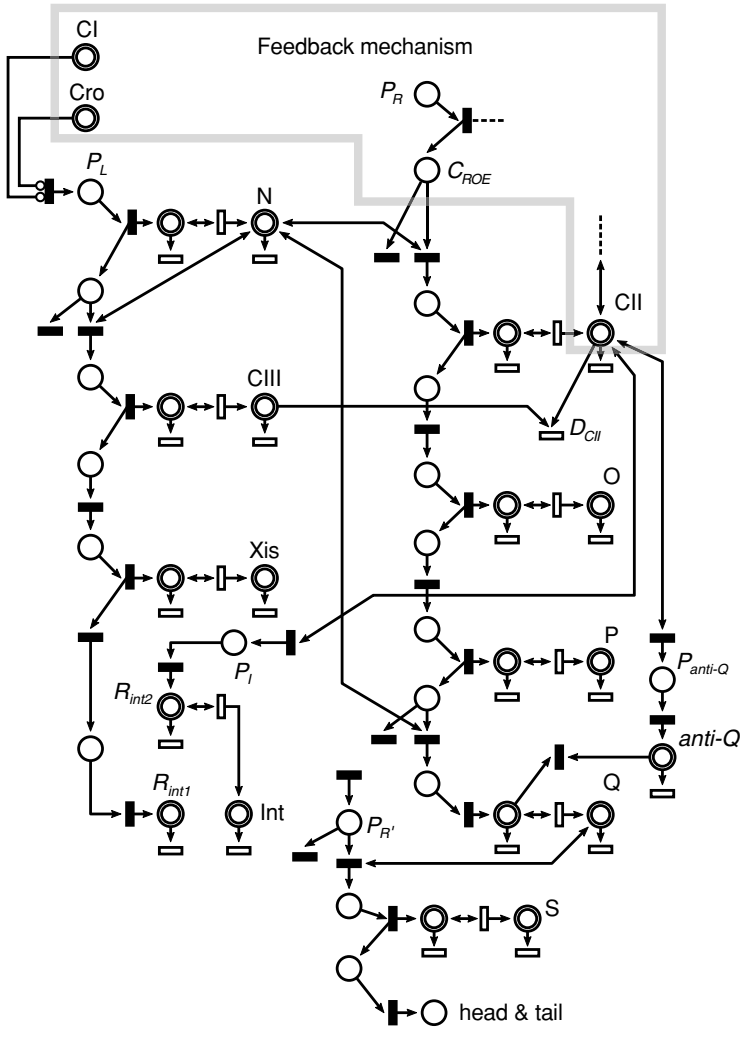


FIGURE 7.1 A bipartite graph in which circles represent interactions and rectangles represent biomolecules. This example is a modified Petri net modeling the phage λ lysis-lysogeny circuit [Matsuno et al. 2000].

Various analyses can be applied on graphs to select new targets or prune a set of already identified targets. Given only the interactions among a set of biomolecules (and often this is all there is, because establishing kinetic parameters is so difficult), it is possible to construct networks allowing comparison of pathways from different organisms, tissues, or different developmental stages, to highlight common features or differences, and to predict missing elements [Bernauer et al. 2004]. Most pathway maps take the form of such static topological representations, and for many explanatory purposes these are adequate. For instance, one can determine the set of genes whose expression is influenced by a target protein directly or indirectly. An undesired

influence may reduce the confidence in a target. By computing all paths between two species one may identify new potential downstream targets. Further, cycles and feedback loops in the graph can be identified. Negative feedback increases the stability of the system and thus makes it robust to perturbations, whereas positive feedback decreases stability [Kitano 2004]. Consequently, targets involved in feedback loops can be accordingly evaluated.

Graphs also allow global connectivity characteristics of networks to be studied. Some exciting graph theoretic work [Fell and Wagner 2000; Jeong et al. 2000; Jeong et al. 2001] has been done in the last five years or so on biochemical (and other) networks, indicating that their robustness or attack tolerance can be predicted from statistics on their connectivity. In particular, it has been shown that cellular networks follow a power-law distribution: there is a small number of molecules with many connections. Removal of highly connected nodes is likely to be very disruptive for the function of the system. Conversely, networks are robust to removal of nodes with a low number of connections. For instance, Jeong et al. [2001] have shown that the connectivity of a protein in yeast is correlated with the likelihood that its removal be lethal to the cell.

Further hints as to the architecture of metabolic and regulatory networks have been found by examining them for common subgraph patterns [Lee et al. 2002; Milo et al. 2002]. Although these lines of research are promising, they have as yet found no application in drug discovery. Moreover, the predictions drawn from graphs are quite restricted. For instance, one cannot infer the quantitative effect of the concentration change of one molecule to another. Static topological analysis of graphs does not allow questions related to the dynamics of the network to be answered, and as it is increasingly believed now, biological function is determined more by dynamics than by structure alone [Kitano 2002b]. It is a simple exercise, for example, to devise networks that have the same topology but because of differing dynamics have opposite function (e.g., activation vs. inhibition; see fig. 7.2).

7.3.1.2 Flux Balance Analysis

Whereas detailed dynamic models can precisely answer questions on cellular behavior, the widespread application of such approaches has been hampered by the lack of kinetic information. In the absence of kinetic information, a method known as flux balance analysis (FBA) has been developed to analyze the metabolic capabilities of a cellular system based on mass balance constraints [Varma and Palsson 1994; Edwards et al. 1999; Edwards and Palsson 2000].

Based on mass balances, a reaction network can be described in terms of linear equations, with parameters representing the stoichiometric coefficients and variables, the metabolic fluxes. The result is a system of ordinary differential equations,

$$d\boldsymbol{X}/dt = \mathbf{S}\boldsymbol{v} + \boldsymbol{b}\,, \tag{7.1}$$

where $\boldsymbol{X}$ is an m dimensional vector defining the quantity of the metabolites within a cell, $\boldsymbol{v}$ is the vector of n metabolic fluxes, $\mathbf{S}$ is the $m \times n$ stoichiometric matrix, and $\boldsymbol{b}$ is the vector of metabolic exchange fluxes. The time constants of metabolic

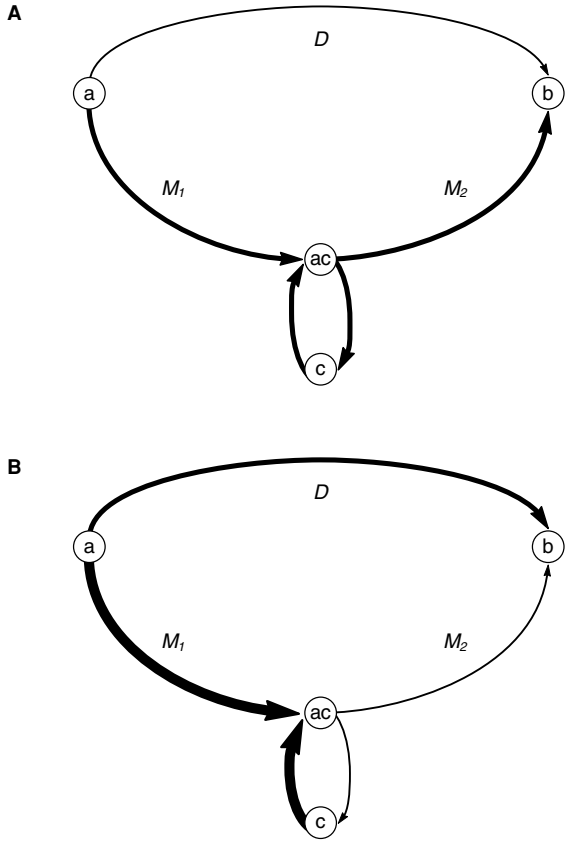


FIGURE 7.2 An isomerization reaction with two parallel paths; a direct path D ($a \rightarrow b$), and a mediated path M_1 ($a + c \rightarrow ac$), M_2 ($ac \rightarrow b + c$). In the first diagram (A), the rate of $M_1 >> D$ and $M_2 >> D$, thus path M bypasses the direct path D, and c accelerates a through the system, acting as a catalyst (reaction rates indicated by arrow thickness). In the second diagram (B), the rate of $M_1 >> D$, but in this case $D >> M_2$, thus c locks up a in complex ac, acting as a competitive inhibitor.

transients are usually much smaller compared to the time constants of cell regulatory dynamics and growth. Hence, the transient mass balances can be simplified to only consider the steady-state behavior, analogous to the system of equations,

$$\mathbf{S}\,\boldsymbol{v} + \mathrm{I}\,\boldsymbol{b} = 0\ , \tag{7.2}$$

where I is the identity matrix. Further rearrangement yields the linear system $\mathbf{S'}\,\boldsymbol{v'} = \mathbf{0}$, with $\mathbf{S'}$ an $m \times n'$ stoichiometric matrix where n' is the total number of fluxes, including fictitious fluxes that transport material across the system boundary. All vectors $\boldsymbol{v'}$, that satisfy equation 7.2 (also called the nullspace of $\mathbf{S'}$) are steady-state metabolic flux distributions that do not violate the mass balance constraints.

This linear system usually has an infinite number of solutions. Its solution space is determined by a set of basis vectors. All solutions of the system can be expressed as linear combination of the basis vectors. The dimension of the nullspace (the number of basis vectors) is given by $n' - rank(\mathbf{S}')$, where $rank(\mathbf{S}')$ is the number of linearly independent rows in $\mathbf{S}'$.

Because many vectors within the nullspace are not physiologically feasible, additional constraints can be placed on the metabolic network to restrict the number of possible solutions [Bonarius, Schmid, and Tramper 1997]. Common constraints are based on capacity and thermodynamic considerations and are realized by imposing a lower and/or an upper limit for each flux ($\alpha_j \leq v_j \leq \beta_j, j = 1 \ldots n'$).

Enforcing stoichiometric, capacity, and thermodynamic constraints simultaneously leads to the definition of a solution space that contains all feasible steady-state flux vectors. Within this set, one can find a particular steady-state metabolic flux vector that optimizes the network behavior toward achieving one or more goals (e.g., maximize or minimize the production of certain metabolites). Mathematically speaking, an objective function has to be defined that needs to be minimized or maximized subject to the imposed constraints. Such optimization problems are typically solved via linear programming techniques.

Somewhat related to FBA is an approach for the determination of so-called elementary flux modes [Schuster, Dandekar, and Fell 1999; Schuster, Fell, and Dandekar 2000]. Generally speaking, an elementary flux mode is a minimal set of enzymes that can operate at steady state. In contrast to FBA, which produces a set of vectors spanning the possible steady-state fluxes, the elementary mode vectors are uniquely determined (up to a multiplication by a positive real number). Any steady-state flux distribution can be then represented as a linear combination of these modes with nonnegative coefficients.

Target identification and validation can profit from FBA and elementary flux modes in a number of ways. If the activity of a putative enzyme target is blocked, a plethora of counterreactions may be evoked to achieve homoeostasis. The new metabolic routes after blocking can be calculated by elementary-mode analysis. That way, the effect of deactivating the target can be estimated. Elementary modes also may contribute to the identification of new targets by determining the most vulnerable parts in a metabolic network where no side paths compensating the effect of a drug exist.

7.3.1.3 Metabolic Control Analysis

The methods of FBA and elementary flux modes study interactions between different routes in a metabolic network and the quantification of flux distributions but do not evaluate how fluxes are controlled. In Metabolic Control Analysis (MCA), the control exerted by the rate of a reaction over a substrate flux or any other system parameter (e.g., metabolite concentration or cell proliferation) can be described quantitatively as a control coefficient. The control coefficient is a relative measure of how much a perturbation affects a system variable and is defined as the fractional change in the system property over the fractional change in the reaction rate [e.g., Burns et al. 1985],

$$C_i^A = \partial A / \partial v_i \cdot v_i / A = \partial \ln A / \partial \ln v_i , \tag{7.3}$$

where A is the variable, i is the step (enzyme), and v_i is the rate of the step perturbed. Because the rate of reaction cannot be perturbed directly, control coefficients are determined by perturbations in parameters that affect the rate linearly, such as enzyme concentration. That is, in equation 7.3, v_i is replaced by the corresponding enzyme concentration $[E_i]$.

It has been demonstrated that for a given flux the sum of its flux-control coefficients of all steps in the metabolic network obeys the theorem [Heinrich and Rapoport 1974; Giersch 1988; Reder 1988]:

$$\Sigma_i C_i^J = 1 , \tag{7.4}$$

where the summations are over all the steps of the system. Hence, increases in some of the flux-control coefficients imply decreases in others so that the total remains unity. Consequently, one can conclude that control is a global property of the system, dependent on all reaction steps.

Similarly to the control coefficients, elasticity coefficients have been defined to quantify the effect of perturbations of a reaction parameter on a reaction rate [e.g., Heinrich and Rapoport 1974; Burns et al. 1985]. The elasticity coefficients are defined as the ratio of relative change in local rate to the relative change in the parameter (usually the concentration of an effector),

$$\varepsilon_p^i = \partial v_i / \partial p \cdot p / v_i = \partial \ln v_i / \partial \ln p , \tag{7.5}$$

where v_i is the reaction rate and p is the perturbation parameter. An elasticity coefficient can be defined for each parameter that affects the reaction rate, such as the concentration of the reaction substrates, products, and effectors.

Unlike control coefficients, elasticity coefficients are local properties because they measure how isolated enzymes are sensitive to changes in parameters. Both the control and elasticity coefficients are not constants but depend on the value of the relevant parameter.

The connectivity theorems are another important feature of MCA. Through these theorems, one can relate the control coefficients to the elasticity coefficients. The connectivity theorem for flux-control coefficients states that, for a common metabolite S, the sum of the products of the flux-control coefficient of all (i) steps affected by S and its elasticity coefficients toward S, is zero [Kacser, Burns, and Davies 1973],

$$\sum_i C_i^J \varepsilon_{[S]}^i = 0 \cdot \tag{7.6}$$

The connectivity theorems describe how perturbations on metabolites of a pathway propagate through the chain of enzymes. The summation theorems, together with the connectivity relations and enzyme elasticities, and possibly some additional relations in the case of branched pathways [e.g., Fell and Sauro 1985],

can be used to derive the control coefficients of a metabolic network. For this purpose, several computational frameworks have been developed [e.g., Fell and Sauro 1985; Westerhoff and Kell 1987].

The classical MCA has been extended in a number of directions. For instance, Reder [1988] introduced a general methodology to calculate control coefficients from elasticities that takes into account moiety-conservation. Several papers have addressed enzyme–enzyme interactions in the context of MCA [e.g., Kacser, Sauro, and Acerenza 1990; Sauro and Kacser 1990].

MCA is often used to correlate individual genes and phenotypic characteristics in metabolic diseases and to identify candidate enzymatic targets for drug development or gene therapy. For instance, enzymes that exert strong control on a system property such as tumor cell proliferation are suitable targets for cancer therapy. It is speculated that MCA will have an increasing impact on the choice of targets for intervention and drug discovery [Cascante et al. 2002]. MCA also holds promise in the identification of combined drug therapies for metabolic disorders. Although the presence of every enzyme in a sequence is essential to a metabolic process, the overall stimulatory or inhibitory effect of a drug is likely achieved with lower concentrations of those enzymes with high flux-control coefficients than for enzymes with low coefficients.

7.3.2 Modeling

By modeling, we mean here the application of mathematical and computational techniques, including but not restricted to those just described, to elucidate the structure and function and to explain the behavior of entire systems as opposed to component subsystems. This includes methods for both simulation (or analytical modeling) and reconstruction (or synthetic modeling).

7.3.2.1 Simulation

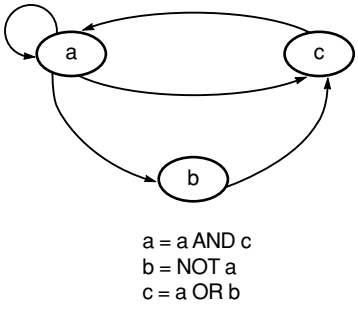
The vast array of molecular processes occurring simultaneously in the cell cannot be comprehended by using experimental techniques alone. In addition to experimental tools, formal methods for the modeling and simulation of biochemical processes are indispensable. Models incorporate the immense amount of experimentally generated data in a systematic fashion and can be used to understand observed phenomena, to simulate *in silico* therapeutic intervention experiments on drug target function, or to predict undesired side effects [Somogyi and Greller 2001].

Generally speaking, a model is an abstract representation that allows inferences about the real network to be made. Different modeling formalisms exist, and the choice of a particular methodology depends on the questions one is trying to answer as well as the type of experimental data available [D'haeseleer, Liang, and Somogyi 2000]. The different algorithms have different strengths; they are focused on the representation of different physical or chemical processes and are often suited to different spatial and temporal scales.

A number of review articles on modeling and simulations of biochemical networks have recently been published. In particular, the comparison of different modeling formalisms [Somogyi and Greller 2001; de Jong 2002; Wiechert 2002], their applications [Endy and Brent 2001; Hasty et al. 2001], and the role of modeling in understanding cellular behavior and its control [Endy and Brent 2001; Kitano 2002a; Sharom, Bellows, and Tyers 2004] have been discussed. In this section we discuss modeling formalisms for the representations of biochemical networks and their impact on the target identification and validation process, and in the next section we elaborate on the problem of how to obtain models depicted in these formalisms. This discussion is not intended to be exhaustive, though. Rather, an emphasis is given to well-established formalisms that have been proven useful in a series of applications. Examples of modeling approaches that have not been included here are Petri Nets [Pinney, Westhead, and McConkey 2003], pi-calculus [Curti et al. 2004], and neural networks [Vohradsky 2001].

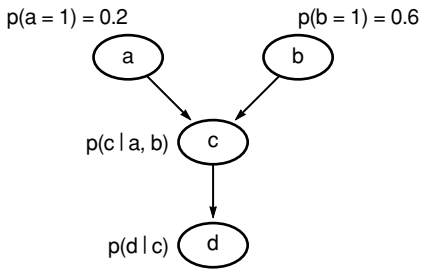
The most elementary formal model for pathways is a graph of biochemical interactions, which can be used for the sort of topological analysis just discussed. In the next level of detail, gene regulatory networks can be modeled by Boolean networks [Kauffman 1969; de Jong 2002]. In this case, each gene is treated as having two states: active (on) or inactive (off). Interactions between genes are represented as Boolean functions (AND, OR, XOR, etc.), which infer the state of a gene from the activity of the other genes in the network. Given the values of the nodes at time t, the Boolean functions are used to update the values of the nodes at time t + 1 (fig. 7.3). Hence, Boolean networks assume synchronous transitions between states—the activities of all genes are updated simultaneously.

Because the number of states of a Boolean network is finite, so is the number of possible trajectories (the sequences of states starting from a given initial state) the system can exhibit. Because of its finite dimension, the space of trajectories can



a	b	c	a	b	c
0	**0**	**0**	**0**	1	**0**
0	**0**	1	**0**	1	**0**
0	1	**0**	**0**	1	1
0	1	1	**0**	1	1
1	**0**	**0**	**0**	**0**	1
1	**0**	1	1	**0**	1
1	1	**0**	**0**	**0**	1
1	1	1	1	**0**	1
t			t + 1		

FIGURE 7.3 An example of a Boolean network representing a 3-gene regulatory network. The Boolean functions define how the expression of the corresponding genes changes based on the current expression values. The transition table on the right specifies all possible expression patterns at time point t + 1 derived from the expression patterns of the genes at time point t and the Boolean functions. At each time point a gene is either expressed (1) or repressed (0).



a	b	p(c = 1 | a, b)
0	0	0.01
0	1	0.15
1	0	0.2
1	1	0.9

c	p(d = 1 | c)
0	0.9
1	0.1

FIGURE 7.4 An example of a Bayesian network representing a 4-gene network. Genes a and b activate gene c, and gene c represses the expression of gene d. This information is coded in the conditional probability tables on the right-hand side. The expression values of a gene are restricted in this example to "on" (1) and "off" (0).

be systematically investigated and the steady states and state cycles determined. In this way, one may predict the consequences of knock-out experiments or determine the set of trajectories converging to the same phenotypic attractor.

An appealing property of Boolean networks is that they are inherently simple but emphasize generic network behavior. Nevertheless, the coarse abstraction to two possible activity values of genes and the synchronous state update are strong assumptions that may not always be justified. Various extensions of Boolean networks have been proposed to cope with these limitations. The formalism of generalized logical networks [Thomas and d'Ari 1990] allows variables to have more than two values and transitions between states to occur asynchronously, whereas probabilistic Boolean networks allow uncertainty in the data and permit interactions between genes to be quantified [Shmulevich et al. 2002].

Genetic regulatory networks have also been extensively modeled by Bayesian networks [Friedman et al. 2000]. Bayesian networks are directed acyclic graphs with vertices representing the concentration (or activity) X_i of the molecules. For each X_i, the conditional distributions p (X_i | parents (X_i)) is defined, where parents (X_i) refers to the direct regulators of molecule i (fig. 7.4). The joint probability of the entire set of variables can then be calculated as the product of all conditional distributions. The probability distributions in a Bayesian net allows one to answer such questions as "What is the probability that the concentration (activity) X_i of the molecule i equals x given the observation on the molecular concentrations of other entities in the network?" By addressing this type of question, one can assess the consequences of modulating the activity of a putative target protein on other molecules in the network.

Bayesian networks offer an attractive modeling formalism because they allow measurement noise and stochastic effects in gene regulation and signal transduction to be incorporated. A major disadvantage, however, is that dynamical aspects of the network are not represented.

Traditionally, dynamical systems are modeled by differential equations. In the case of biochemical networks, Ordinary Differential Equations (ODEs) model the concentration (or activity) of proteins, RNA species, or metabolites by time-dependent

functions. More formally, the rate change of the concentration of a molecule is expressed as a function of the concentration of other molecules,

$$\frac{dc_i}{dt} = f_i(c_1, c_2, \ldots, c_N), \tag{7.7}$$

where c_i is the concentration of the ith element in the network, and f_i is usually a nonlinear function.

Because of the nonlinearity of the functions f_i, analytical solutions of the ODEs just mentioned are seldom available. Instead, one can use numerical simulation to approximate the solution. The result of the numerical simulation is a series of concentration values for each molecule at a sequence of time points. ODEs allow the complete behavior of the system to be simulated over time and to track changes of the behavior due to perturbations. Minor changes in the concentration of a target protein following drug administration might be neutralized by feedback control. On the other hand, high dosages of the drug can cause serious side effects [Kitano 2002a]. Thus, ODE models can help to identify not only target proteins whose inhibition gives rise to desired therapeutic effects but also the appropriate dosage of the inhibiting drug. Moreover, differential equation models allow the search for perturbations on multiple proteins that produce a desired combined effect while reducing the side effects [Endy and Brent 2001]. In a more futuristic view, differential equations can be used not only to identify and validate targets but also to establish the way of drug administration, such as order of usage, timing, and dosage of drugs that would drive a system into a desired state [Kitano 2004].

ODEs have been used extensively to model signal transduction [Kholodenko et al. 1999; Schoeberl et al. 2002; Bentele et al. 2004], gene regulation [Tyson and Novak 2002; Leloup and Goldbeter 2003], and metabolic pathways [Leaf and Srienc 1998; Jamshidi et al. 2001].

Dynamical modeling requires detailed kinetic information on the mechanisms involved that is often not available. To cope with this problem, the formalism of piecewise-linear differential equations abstracts from the biochemical details of the regulatory mechanisms and exploits the switchlike nature of gene expression [de Jong 2002].

A disadvantage of ODE models is that they assume spatially homogeneous systems, an assumption that sometimes may lead to wrong predictions. Although in many cases spatial effects can be incorporated in the function f_i, there are situations where one may need to take into account diffusion and transport of proteins from one compartment to another. For the purpose, reaction-diffusion equations (RDEs) of the form

$$\frac{\partial c_i}{\partial t} = f_i(c_1, c_1, \ldots, c_1) + D_i \nabla^2 c_i \tag{7.8}$$

can be used, where the term f_i is the reaction term, and $D_i \nabla^2 c_i$ the diffusion term, with D_i being a diffusion constant and ∇^2 denoting the Laplacian operator (in Cartesian coordinates, $\nabla^2 c_i$ is the sum of the three second partial derivatives of c_i).

If initial molecule concentrations are specified together with the concentrations at the compartment boundaries, RDEs can be solved numerically using finite-difference or finite-element discretizations of the compartment volume. The solution consists of molecular concentrations across the discretized compartment volume in a series of time points. RDEs and variants thereof have been used in a range of applications, including studies of pattern formation [Meinhardt and Gierer 2000; Myasnikova et al. 2001] and calcium transport [Smith, Pearson, and Keizer 2002].

Although RDEs model biochemical networks more realistically, they are subject to many unknown parameters that prevent their wide usage. In particular, predictions derived from RDEs are quite sensitive to initial and boundary molecular concentrations that are seldom accessible. Quantitative proteomics approaches based on fluorescent microscopy hold a promise in providing the data necessary for the development of such models.

Differential equations allow biochemical networks to be described on the level of individual reaction steps like enzymatic catalysis, or transcription site binding. Differential equations, however, rely on the assumption that molecular concentrations vary continuously and deterministically. These assumptions may not always hold, especially in cases where some types of molecules participate with low number [Arkin, Ross, and McAdams 1998]. As a consequence, the same system with almost identical initial conditions and environmental inputs can exhibit different behavior.

Stochastic models relax these assumptions by taking a nondeterministic approach. Basically, stochastic approaches model the probability the system to be in a certain state at a specific time point by incorporating the probability distribution of the molecular concentrations in the past, the probability that a reaction will occur in a given time frame, and the probability that this reaction will bring the system from one state to another. Stochastic models can be simulated via stochastic simulation, pioneered by Gillespie [1977, 2000]. The computational load of the original algorithm has further motivated various improvements of the method [Gibson and Bruck 2000].

Stochastic modeling and simulation better approximates the biochemical reality, but its usage may not always be beneficial. It requires very detailed knowledge on the underlying reaction mechanisms. Even when such knowledge is available, the computational costs involved may become prohibitive for larger systems.

7.3.2.2 Network Reconstruction

Whereas in the previous section we reviewed various model formalisms and the questions they can answer, in this section we concentrate on the more difficult and challenging problem of how such models can be obtained. Model construction is usually intercepted as a tedious manual task accomplished by experienced professionals. In some cases, however, this task can be automated, and a model can be computationally inferred directly from experimental data, a problem usually referred to as reverse engineering.

The modeling formalism and approach that one chooses for reverse engineering depends on various factors [D'haeseleer, Liang, and Somogyi 2000; Bolouri and Davidson 2002; Greller and Somogyi 2002]. To a large extent, the choice is dictated

by the characteristics of the network being studied and the type of questions one would wish to address through modeling. Furthermore, the quality and quantity of data available for model reconstruction should be evaluated: one should assure that the data are sufficient to support the complexity of the model. For instance, Boolean networks allow the analysis of very large systems but do not yield detailed predictions, whereas stochastic models approximate the reality better but are restricted to smaller systems because of their complexity.

A number of reviews of reverse-engineering approaches can be found in the literature [D'haeseleer, Liang, and Somogyi 2000; Bolouri and Davidson 2002; Hasty et al. 2001]. Here, we discuss well-established techniques for inferring models as those discussed in the previous section.

Typically, nodes in the network (genes, proteins, etc.) are assumed to be linked if there is some correlation between their behaviors. Such structure can often be deduced using clustering or correlation-based techniques. To discover coregulated genes, expression profiles obtained in different perturbation experiments across a series of time points can be grouped according to various metrics, including Euclidean distance, Pearson correlation, and mutual information [D'haeseleer, Liang, and Somogyi 2000]. Different metrics can be employed by different algorithms that group genes according to the similarity of their expression profiles. Examples of such algorithms include hierarchical clustering, K-means, and self-organizing maps [Brazma and Vilo 2000]. Given the multitude of distance metrics and clustering algorithms, it may be very difficult to choose the appropriate algorithm for a given set of data [D'haeseleer, Liang, and Somogyi 2000]. Clustering algorithms can be further complemented by motif discovery algorithms. Coexpressed genes sharing the same motifs (or *cis*-regulatory elements) are likely to be regulated by the same transcription factors. See Li and Wang [2003] for a recent review of identifying *cis*-regulatory elements.

New information can be included in network models to specify the nature and the strength of the relationships using logical or probabilistic formalisms. Boolean networks are among the first formalisms for which automated model reconstruction algorithms have been developed. The REVEAL algorithm [Liang, Fuhran, and Somogyi 1998] pioneered this work by exploiting time series of gene expression and a mutual information measure for network performance. Basically, the method is able to infer the sets of input elements controlling each element in the network and the corresponding Boolean function. Ideker, Thorsson, and Karp [2000] proposed an alternative approach based on the branch-and-bound technique.

Research has been also conducted for learning Bayesian networks from data [Friedman 2004]. The learning process amounts to a search in the space of feasible network structures that are constrained by the independence relationships suggested by the data. Each network is evaluated according to an objective (scoring function)—the posterior network probability given the data [Cooper and Herskovits 1992]. Additional heuristics can be employed to reduce the search space and speed up the learning process [Heckerman, Geiger, and Chickering 1995]. In many cases, the amount of data (gene expression data) is small relative to all variables (genes). Consequently, it is likely that there are many networks that explain the data equally well. To deal with this problem, one can either use model selection techniques to

identify the best network or, for every feature, take the average over all feasible networks (Bayesian averaging):

$$P(F = f \mid D) = \sum_G P(G \mid D)\, P(F = f \mid G). \tag{7.9}$$

This equation simply says that the probability that a feature F has value f (e.g., that a gene has an expression level f), given some data D is a weighted sum of the probabilities that F equals f provided G is the true Bayesian network explaining the data. The weights here are the posterior probabilities P(G | D), the scoring functions obtained in the learning process. Various extensions dealing with the problem of multiple network models have been proposed [e.g., Friedman 2003].

Work has been done to infer differential equation models of cellular networks from time-series data. As we explained in the previous section, the general form of the differential equation model is $dc_i/dt = f_i(c_1, c_2, ..., c_N)$, where f_i describes how each element of the network affects the concentration rate of the i^{th} network element. If the functions f_i are known, that is, the individual reaction and interaction mechanisms in the network are available, a wealth of techniques can be used to fit the model to experimental data and estimate the unknown parameters [Mendes 2002]. In many cases, however, the functions f_i are unknown, nonlinear functions. A common approach for reverse engineering ordinary differential equations is to linearize the functions f_i around the equilibrium [Stark, Callard, and Hubank 2003] and obtain

$$\frac{dc_i}{dt} = a_{i_0} + a_{i_1 1} c_1 + a_{i_2} c_2 + \ldots + a_{i_N} c_N, \tag{7.10}$$

where $a_{ij} = \partial f_i / \partial x_j$. The estimation of the parameters a_{ij} can be accomplished either by a direct approximation of the partial derivatives or by using time-series expression or concentration data and minimizing the prediction error of the model [Holter et al. 2001; Xiong, Li, and Fang 2004].

Dynamical modeling poses more serious requirements on the experiments performed to gather the data fed into the model. Various studies have investigated the kind and number of experiments necessary to estimate the parameters in a linearized model or even to minimize the number of necessary experiments [Kholodenko et al. 2002; Tegner et al. 2003].

7.4 INTEGRATED APPLICATIONS

The line between pathway databases and pathway analysis software packages is becoming increasingly hard to draw. Most databases include some sort of analysis capability (even if only search, filtering, visualization, etc.), and no analysis software is useful without some access to data. In this section we include capsule summaries of software packages the main focus of which addresses the need for pathway analysis rather than pathway data.

Cytoscape [Shannon et al. 2003] is an open-source project with the purpose of developing a platform for visualizing biochemical networks, with support for additional

functionality such as correlation with gene expression or other state data, topological analysis, annotation, database connectivity, and so on. It is written in Java, thus portable, and its plug-in architecture promotes extensibility.

PathBLAST [Kelley et al. 2004] is a network alignment and search tool for comparing protein interaction networks across species to identify protein pathways and complexes that have been conserved by evolution.

DDLab [Wuensche 2002] is an interactive graphics package for researching discrete dynamical networks (Boolean networks and cellular automata), such as those generated by reconstruction of regulatory networks from gene expression data.

Discoverer (table 7.1) is an automated modeling tool able to transform a set of data into a Bayesian network by searching for the most probable model responsible for the data.

The Case Western Reserve University's Pathways Database System [Ozsoyoglu, Nadeau, and Ozsoyoglu 2003] comes with a sample set of metabolic and signaling pathways, most drawn from Michal's Biochemical Pathways [Michal 1999]. It consists of a suite of tools, including query interfaces, viewers, and a pathway editor.

PATIKA [Demir et al. 2002] is an academic project with powerful editing and visualization capabilities. Although it has no advanced analyses (a few involving correlation with expression data), it has a fully functional database query interface, though the data included are rather sparse.

PathwayLab (table 7.1) is a pathway editing package with database links. Compounds and proteins, reactions and interactions, can be added and edited, and some simulation and analysis capability is included.

Pathway Assist (table 7.1) allows navigation and analysis of biological pathways, gene regulation and protein interaction maps. The software enables you to launch remote Entrez searches directly, import search results, assemble results into pathway diagrams, and find connections between disparate data.

Pathway Enterprise (table 7.1) is a package for managing, designing, and visualizing pathways, including annotations, quantitative information, and citation links. It can import data from public and proprietary databases and export pathways globally or selectively.

Pathway Articulator or PathArt (table 7.1) is built around a database of signaling and metabolic pathways derived from manual curation of the literature. Database elements are annotated with links to several vocabularies (GO, Locuslink, CAS, etc.), and displayed in an editable pathway visualizer.

Metacore (table 7.1) is a suite of software oriented toward understanding the function of gene sets discovered by expression analysis. It includes a database of annotated metabolic, signaling, and regulatory pathways and a graphical pathway display/editor.

Pathways Analysis (table 7.1) contains a large set of manually curated gene and protein interactions, together with multiple viewers to display various aspects of the interactions. Both canonical pathways (i.e., generally accepted functional units) and arbitrary interaction networks can be displayed, and links to annotation data and to literature citations are accessible from the viewers.

7.5 FUTURE DIRECTIONS

The current approach to target validation is still heavily dependent on laboratory-based work. This contrasts with other industrial areas, where computer simulations have largely replaced the physical evaluation of prototypes. Mathematical and computational tools applicable to *in silico* validation have been developed in other fields, but their use still requires much more data, at significant cost, than their results so far have justified. As further technological advances yield higher-throughput methods and greater sensitivity and accuracy of detection, the necessary data will become available. Some simple subsystems will then be ripe for modeling and simulation with the required degree of accuracy, but the greater goal of systems biology is to model at the level of the cell, the tissue, or even the organism, based on simulation of events at the molecular level. Because of the vast range of scales—in space and time—this will require development of more powerful methods for hybrid and multiscale modeling.

REFERENCES

Arkin, A., J. Ross, and H. H. McAdams. 1998. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells. *Genetics* 149:1633–48.

Bader, G. D., I. Donaldson, C. Wolting, B. F. Ouellette, T. Pawson, and C. W. Hogue. 2001. BIND: The Biomolecular Interaction Network Database. *Nucleic Acids Res* 29:242–5.

———, D. Betel, and C. W. V. Hogue. 2003. BIND: The Biomolecular Interaction Network Database. *Nucleic Acids Res* 31:248–50.

Baker, P. G., C. A. Goble, S. Bechhofer, N. W. Paton, R. Stevens, and A. Brass. 1999. An ontology for bioinformatics applications. *Bioinformatics* 15:510–20.

Becker, K. G., S. L. While, J. Muller, and J. Engel. 2000. BBID: The Biological Biochemical Image Database. *Bioinformatics* 16:745–6.

Bentele, M., I. Lavrik, M. U. Ulrich, S. Stoesser, D. W. Heermann, K. Kalthoff, P. H. Krammer, and R. Eils. 2004. Mathematical modeling reveals threshold mechanism in CD95-induced apoptosis. *J Cell Biol* 166:839–51.

Bernauer, S., D. Croft, P. Gardina, E. Minch, M. de Rinaldis, and I. Vatcheva. 2004. Case study: Data management strategies in an integrated pathway tool. *Appl Bioinformatics* 3:63–75.

Bolouri, H., and E. H. Davidson. 2002. Modeling transcriptional regulatory networks. *BioEssays* 24:1118–29.

Bonarius, H. P. J., G. Schmid G, and J. Tramper. 1997. Flux analysis of underdetermined metabolic networks: The quest for the missing constraints. *Trends Biotechnol* 15:308–14.

Brazma, A., and J. Vilo. 2000. Gene expression data analysis. *FEBS Lett* 480:17–24.

Burns, J. A., A. Cornish-Bowden, A. K. Groen, R. Heinrich, H. Kacser, J. W. Porteous, S. M. Rapoport, et al. 1985. Control of metabolic systems. *Trends Biochem Sci* 10:16.

Cacsante, M., L. G. Boros, B. Comin-Anduix, P. de Atauri, J. J. Centelles, and PW-N. Lee. 2002. Metabolic control analysis in drug discovery and disease. *Nat Biotechnol* 20:243–9.

Choi, C., M. Krull, A. Kel, O. Kel-Margoulis, S. Pistor, A. Potapov, N. Voss, and E. Wingender. 2004. TRANSPATH (R)—A high quality database focused on signal transduction. *Comp Funct Genomics* 5:163–8. http://www.biobase.de/pages/products/databases.html

Cook, D. L., J. F. Farley, and S. J. Tapscott. 2001. A basis for a visual language for describing, archiving and analyzing functional models of complex biological systems. *Genome Biol* 2:research0012.1–10.

Cooper, G. F., and E. Herskovits. 1992. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning* 9:309–47.

Curti, M., P. Degano, C. Priami, and C. T. Baldari. 2004. Modelling biochemical pathways through enhanced pi-calculus. *Theor Comp Sci* 325:111–40.

Demir, E., O. Babur, U. Dogrusoz, A. Gursoy, G. Nisanci, R. Cetin-Atalay, and M. Ozturk. 2002. PATIKA: An integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics* 18:996–1003.

Deville, Y., D. Gilbert, J. van Helden, and S. J. Wodak. 2003. An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics* 4:246–59.

D'haeseleer, P., S. Liang, and R. Somogyi. 2000. Genetic network inference: From co-expression clustering to reverse engineering. *Bioinformatics* 16:707–26.

Duggan, D. J., M. Bittner, Y. Chen, P. Meltzer, and J. M. Trent. 1999. Expression profiling using cDNA microarrays. *Nat Genet* 21:10–14.

Edwards, J. S., R. Ramakrishna, C. H. Schilling, and B. O. Palsson. 1999. Metabolic flux balance analysis. In *Metabolic engineering,* ed. S. Y. Lee and E. T. Papoutsakis, 13–57. CITY: Marcel Deker.

———, and B. O. Palsson. 2000. Metabolic flux balance analysis and the *in silico* analysis of *Escherichia coli* K-12 gene deletions. *BMC Bioinformatics* 1:1. PMID 11001586.

Ellis, L. B. M,, B. K. Hou, W. Kang, and L. P. Wackett. 2003. The University of Minnesota Biocatalysis/Biodegradation Database: Post-Genomic Datamining. *Nucleic Acids Res* 31:262–5.

Endy, D., and R. Brent. 2001. Modelling cellular behavior. *Nature* 409:391–5.

Fell, D. A., and H. M. Sauro. 1985. Metabolic control analysis: Additional relationships between elasticities and control coefficients. *Eur J Biochem* 148:555–61.

———, and A. Wagner. 2000. The small world of metabolism. *Nat Biotechnol* 18:1121–2.

Fiehn, O., and W. Weckwerth. 2003. Deciphering metabolic networks. *Eur J Biochem* 270:579–588.

Friedman, N., M. Linial, I. Nachman, and D. Pe'er. 2000. Using Bayesian networks to analyze expression data. *J Comput Biol* 7:601–20.

———. 2003. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning* 50:95–125.

———. 2004. Inferring cellular networks using probabilistic graphical models. *Science* 3003:799–805.

Gibson, M. A., and J. Bruck. 2000. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J Phys Chem A* 104:1876–89.

Giersch, C. 1988. Control analysis of metabolic networks. 1. Homogeneous functions and the summation theorems for control coefficients. *Eur J Biochem* 174:509–13.

Gillespie, D. T. 1977. Exact stochastic simulation of coupled chemical reactions. *J Phys Chem* 81:2340–61.

———. 2000. The chemical Langevin equation. *J Chem Phys* 113:297–306.

Greller, L. D., and R. Somogyi. 2002. Reverse engineers map the molecular switching yards. *Trends Biotechnol* 20:445–7.

Gygi, S. P., B. Rist, S. A. Gerber, F. Turecek, M. H. Gelb, and R. Aebersold. 1999. Quantitative analysis of complex protein mixtures using isotope-coded affinity tags. *Nat Biotechnol* 17:994–9.

Hannon, G. J. 2002. RNA interference. *Nature* 418:244–51.

Hasty, J., D. McMillen, F. Isaacs, and J. J. Collins. 2001. Computational studies of gene regulatory networks: In numero molecular biology. *Nat Rev Genet* 2:268–79.

Heckerman, D., D. Geiger, and D. M. Chickering. 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning* 20:197–243.

Heinrich, R., and T. A. Rapoport. 1974. A linear steady-state treatment of enzymatic chains. General properties, control and effector strength. *Eur J Biochem* 42:89–95.

Holter, N. S., A. Maritan, M. Cieplak, N. V. Fedoroff, and J. R. Banavar. 2001. Dynamic modeling of gene expression data. *Proc Natl Acad Sci USA* 98:1693–8.

Hucka, M., A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, et al. 2003. The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics* 19:524–31.

Ideker, T. E., V. Thorsson, and R. M. Karp. 2000. Discovery of regulatory interactions through perturbation: Inference and experimental design. *Proceedings of the Pacific Symposium on Biocomputing* 5:302-13.

Jamshidi, N., J. S. Edwards, T. Fahland, G. M. Church, and B. O. Palsson. 2001. Dynamic simulation of the human red blood cell metabolic network. *Bioinformatics* 17:286–7.

Jeong, H., B. Tombor, R. Albert, Z. N. Oltvai, and A-L. Barabási. 2000. The large-scale organization of metabolic networks. *Nature* 407:651–4.

———, S. P. Mason, Z. N. Oltvai, and A-L. Barabási. 2001. Lethality and centrality in protein networks. *Nature* 411:41–2.

Jong, H. de. 2002. Modeling and simulation of genetic regulatory systems: A literature review. *J Comput Biol* 9:67–103.

Joshi-Tope, G., M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, et al. 2005. Reactome: A knowledgebase of biological pathways. *Nucleic Acids Res* 33, Database Issue:D428–32.

Kacser, H., J. A. Burns, and D. D. Davies. 1973. *Control of biological processes*. Cambridge: Cambridge Univ. Press.

———, H. M. Sauro, and L. Acerenza. 1990. Enzyme–enzyme interactions and control analysis. 1. The case of nonadditivity: Monomer-oligomer associations. *Eur J Biochem* 187:481–91.

Kanehisa, M., S. Goto, S. Kawashima, and A. Nakaya. 2002. The KEGG databases at GenomeNet. *Nucleic Acids Res* 30:42–6.

Karp, P. D. 2000. An ontology for biological function based on molecular interactions. *Bioinformatics* 16:269–85.

Kauffman, S. A. 1969. Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theoret Biol* 22:437–67.

Kelley, B. P., B. B. Yuan, F. Lewitter, R. Sharan, B. R. Stockwell, and T. Ideker. 2004. PathBLAST: A tool for alignment of protein interaction networks. *Nucleic Acids Res* 32, Suppl. no. 2:W83–8.

Kholodenko, B. N., O. V. Demin, G. Moehren, and J. B. Hoek. 1999. Quantification of short term signaling by the epidermal growth factor receptor. *J Biol Chem* 274:30169–81.

———, A. Kiyatkin, F. J. Bruggeman, E. Sontag, H. V. Westerhoff, and J. B. Hoek. 2002. Untangling the wires: A strategy to trace functional interactions in signaling and gene networks. *Proc Natl Acad Sci USA* 99: 12841–6.

Kitano, H. 2002a. Computational systems biology. *Nature* 420:206–10.

———. 2002b. Systems biology: Toward system-level understanding of biological systems. In *Foundations of systems biology*, ed. H. Kitano, 1–36. Cambridge, MA: MIT Press.

———. 2003. A graphical notation for biochemical networks. *Biosilico* 1:169–76.

———. 2004. Cancer as a robust system: Implications for anticancer therapy. *Nat Rev* 4:227–35.

Kohn, K. W. 1999. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Molec Biol Cell* 10:2703–34.

Leaf, T. A., and F. Srienc. 1998. Metabolic modeling of polyhydroxybutyrate biosynthesis. *Biotechnol Bioeng* 57:557–70.

Lee, T. I., N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, et al. 2002. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science* 298:794–9.

Leloup, J-C., and A. Goldbeter. 2003. Toward a detailed computational model for the mammalian circadian clock. *Proc Natl Acad Sci USA* 100:7051–6.

Li, H., and W. Wang. 2003. Dissecting the transcription networks of a cell using computational genomics. *Curr Opin Genet Dev* 13:611–6.

Liang, S., S. Fuhran, and R. Somogyi. 1998. REVEAL: A general reverse engineering algorithm for inference of genetic network architectures. *Proceedings of the Pacific Symposium on Biocomputing* 3:18–19.

Lindon, J. C., Holmes, E., and J. K. Nicholson. 2004. Metabonomics: Systems biology in pharmaceutical research and development. *Curr Opin Mol Ther* 6(3):265–272.

Lloyd, C. M., M. D. B. Halstead, and P. F. Nielsen. 2004. CellML: Its future, present, and past. *Prog Biophys Mol Biol* 85:433–50.

Maimon, R., and S. Browning. 2001. Diagrammatic notation and computational structure of gene networks. In *Proceedings of the Second International Conference on Systems Biology*, Pasadena, CA, ed. H. Kitano, 311–17. Omnipress, Madison, WI.

Matsuno, H., A. Doi, M. Nagasaki, and S. Miyano. 2000. Hybrid Petri net representation of gene regulatory network. *Proceedings of the Pacific Symposium on Biocomputing* 5:338–49.

Meinhardt, H., and A. Gierer. 2000. Pattern formation by local self-activation and lateral inhibition. *BioEssays* 22:753–60.

Mendes, P. 2002. Modeling large biological systems from functional genomic data: Parameter estimation. In *Foundations of systems biology,* ed. H. Kitano, 164–86. Cambridge, MA: MIT Press.

Mewes, H. W., C. Amid, R. Arnold, D. Frishman, U. Guldener, G. Mannhaupt, M. Munsterkotter, et al. 2004. MIPS: Analysis and annotation of proteins from whole genomes. *Nucleic Acids Res* 32, Database Issue:D41–4.

Meyer, T., and M. N. Teruel. 2003. Fluorescence imaging of signaling networks. *Trends Cell Biol* 13:101–6.

Michal, G. 1999. Biochemical pathways. New York: Wiley.

Miller, K. J. 1998. Metabolic pathways of biochemistry. http://www.gwu.edu/~mpb/index.html

Milo, R., S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. 2002. Network motifs: Simple building blocks of complex networks. *Science* 298:824–7.

Miyawaki, A. 2003. Visualization of the spatial and temporal dynamics of intracellular signaling. *Dev Cell* 4:295–305.

Myasnikova, E., A. Samsonova, K. Kozlov, M. Samsonova, and J. Reinitz. 2001. Registration of the expression patterns of Drosophila segmentation genes by two independent methods. *Bioinformatics* 17:3–12.

Newman, M. E. J. 2003. The structure and function of complex networks. *SIAM Review* 45:167–256.

Ozsoyoglu, Z. M., J. Nadeau, and G. Ozsoyoglu. 2003. Pathways database system. *OMICS* 7:123–5, http://nashua.cwru.edu/pathways/

Pinney, J. W., D. R. Westhead, and G. A. McConkey. 2003. Petri Net representations in systems biology. *Biochem Soc Trans* 31:1513–5.

Reder, C. 1988. Metabolic control theory: A structural approach. *J Theoret Biol* 135:175–201.

Salwinski, L., C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg. 2004. The Database of Interacting Proteins: 2004 update. *Nucleic Acids Res* 32:D449–51.

Sauro, H. M., and H. Kacser. 1990. Enzyme–enzyme interactions and control analysis. 2. The case of nonindependence: Heterologous associations. *Eur J Biochem* 187:493–500.

Schoeberl, B., C. Eichler-Jonsson, E. D. Gilles, and G. Müller. 2002. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nat Biotechnol* 20:370–5.

Schuster, S., T. Dandekar, and D. A. Fell. 1999. Detection of elementary flux modes in biochemical networks: A promising tool for pathway analysis and metabolic engineering. *Trends Biotechnol* 17:53–60.

———, D. A. Fell, and T. Dandekar. 2000. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nat Biotechnol* 18:326–32.

Sekar, R. M., and A. Periasamy. 2003. Fluorescence resonance energy transfer (FRET) microscopy imaging of live cell protein localizations. *J Cell Biol* 160:629–33.

Selkov, E. Jr., Y. Grechkin, N. Mikhailova, and E. Selkov. 1998. MPW: The metabolic pathways database. *Nucleic Acids Res* 26:43–5.

Shannon, P., A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, et al. 2003. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Res* 13:2498–504.

Sharom, J. R., D. S. Bellows, and M. Tyers. 2004. From large networks to small molecules. *Curr Opinion Chem Biol* 8:81–90.

Shmulevich, I., E. R. Dougherty, S. Kim, and W. Zhang. 2002. Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18:261–74.

Smith, G. D., J. E. Pearson, and J. E. Keizer. 2002. Modeling intracellular calcium waves and sparks. In *Computational cell biology*, ed. C. P. Fall, E. S. Marland, J. M. Wagner, and J. J. Tyson, 198–227. New York: Springer-Verlag.

Somogyi, R., and L. D. Greller. 2001. The dynamics of molecular networks: Applications to therapeutic discovery. *Drug Discov Today* 6:1267–77.

Stark, J., R. Callard, and M. Hubank. 2003. From the top down: Towards a predictive biology of signalling networks. *Trends Biotechnol* 21:290–3.

Stephanopoulos, G. N., A. A. Aristidou, and J. Nielsen. 1998. *Metabolic engineering: Principles and methodologies*. San Diego, CA: Academic Press.

Takai-Igarashi, T., Y. Nadaoka, and T. Kaminuma. 1998. A database for cell signaling networks. *J Comp Biol* 5:747–54.

Tao, W. A., and R. Aebersold. 2003. Advances in quantitative proteomics via stable isotope tagging and mass spectrometry. *Curr Opinion Biotechnol* 14:110–8.

Tegner, J., M. K. S. Yeung, J. Hasty, and J. J. Collins. 2003. Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proc Natl Acad Sci USA* 100:5944–9.

Thomas, R., and R. d'Ari. 1990. *Biological feedback*. Boca Raton, FL: CRC Press.

Tyson, J. J., and B. Novak. 2002. Cell cycle controls. In *Computational cell biology,* ed. C. P. Fall, E. S. Marland, J. M. Wagner, and J. J. Tyson, 261–84. New York: Springer-Verlag.

van Helden, J., A. Naim, R. Mancuso, M. Eldridge, L. Wernisch, D. Gilbert, and S. J. Wodak. 2000. Representing and analysing molecular and cellular function using the computer. *Biol Chem* 381:921–35.

Varma, A., and B. O. Palsson. 1994. Metabolic flux balancing: Basic concepts, scientific and practical use. *Bio/Technology* 12:994–8.

Vohradsky, J. 2001. Neural model of the genetic network. *J Biol Chem* 276:36168–73.
Westerhoff, H. V., and D. B. Kell. 1987. Matrix method for determining the steps most rate-limiting to metabolic fluxes in biotechnological processes. *Biotechnol Bioengin* 30:101–7.
Wiechert, W. 2002. Modeling and simulation: Tools for metabolic engineering. *J Biotechnol* 94:37–63.
Wuensche, A. 2003. Discrete dynamics lab: Tools for investigating cellular automata and discrete dynamical networks. *Kybernetes* 32:77–104.
Xiong, M., J. Li, and X. Fang. 2004. Identification of genetic networks. *Genetics* 166:1037–52.

8 Molecular Interactions: Learning from Protein Complexes

Ana Rojas, David de Juan, and Alfonso Valencia
Centro Nacional de Biotecnología

CONTENTS

8.1 MOLECULAR INTERACTIONS: LEARNING FROM PROTEIN COMPLEXES

8.1.1 Molecular Interactions Are Essential to Understanding Biology

Biological science now has access to several sequenced genomes, spanning almost all live lineages from prokaryotic to human [1]. Whole-genome analyses have attracted considerable attention in terms of computational resources [2,3] and the

databases devoted to the subject [4]. The increasing amount of available data has led to a constant battle to comprehend the rapid input of information.

All this genomic information is traditionally classified and analyzed in "gene-centric" catalogs, which provide a gene-by-gene view of the genomes. Most of the scientific questions are concerned with complex interactions between genes and proteins [5] as seen in recent developments [6] where studies of biological processes focus on function as the point of reference. Therefore, the challenge ahead is to move from the gene-centric view to other more integrated approaches, in which the isolated gene is no longer the functional unit.

The easiest way of addressing molecular interactions is by monitoring the protein interaction space, although we should keep in mind that molecular interactions include nucleic acids, membranes, and small molecules. This additional spectrum of interaction is crucial to comprehending the dynamics of a living system as well to understanding how the interactions between cellular components are organized. These are features that are critical to the explanation of almost all biological functions, such as signal transduction, metabolism, cellular architecture, and information transfer. Therefore, protein interactions are an essential but small part of the whole repertoire of molecular interactions available in biological systems, whereby the understanding of them can help to develop methodology for future studies in other systems.

Here we review the current state of experimental and computational methods for the study of protein interactions, including prospects for future developments.

8.2 CURRENT STATUS OF EXPERIMENTAL PROCEDURES

8.2.1 Reaching the Proteome: From Standard to Large-Scale Detection of Protein Interactions

Novel proteomic technologies have produced a remarkable amount of data on protein complexes. These days, the current priority from a bioinformatics perspective is to develop efficient systems to create a knowledge space where the incoming data can be organized from a biological perspective.

One of the lessons we have learned from the genomics projects indicates that the "complexity" in terms of gene composition is smaller than anticipated, and genetic features alone do not seem sufficient to explain many of the specific organisms' properties. An example is represented by the remarkably small *Fugu* genome (about 350,000,000 bp), that is about one-tenth of its human counterpart (3,400,000,000 bp) but still accounts for a similar number of genes (probably less than 25,000) in both organisms. Thus it is possible that a considerable part of biological diversity and complexity is encoded in the repertoire of potential interactions, including the complex arrangement of domains typical of higher eukaryotes that increase the potential number of interactions and their potential regulation.

During the last six years several high-throughput proteomics approaches have been published, in what is certainly only the first wave of applications. The current panoply of experimental procedures is discussed in several reviews [7–9]. Traditional

techniques such as chemical cross-linking [10], affinity purification [11], and size-exclusion chromatography [12] are devoted to detect physical interactions and are often used in combination with immunological methods and mass spectrometry.

Gavin et al. [13] made use of a combination of Tandem-Affinity purification (TAP) and mass spectrometry to characterize multiprotein complexes in *S. cerevisiae*. Ho et al. [14] identified protein complexes covering about 25% of the yeast proteome using a similar procedure, with a different purification step.

Yeast two-hybrid approaches [15,16] are based on the modular properties of systems such as the Gal4 protein of *S. cerevisiae*. Five other large collections of data have been produced with variants of this methodology [17–19]. Additional variations on this technique have also been applied to membrane proteins [20], and alternative methods such as directed mutagenesis [21] and phage display [22] are used to detect and identify protein complexes. Yeast protein chips have also been used to screen protein–protein interactions and protein–drug interactions [23]. Several of these experimental systems are capable of producing information on the specific interaction regions using protein fragments.

This first generation of experimental methods has several technical limitations, including bias in interaction preferences (more obvious in the approaches based on the yeast two-hybrid system), overrepresentation of small proteins in complex purification procedures (typical of TAP approaches), unnatural interactions of nonnative proteins, indirect associations, and others [24].

8.2.2 Structural Approaches

Structural approaches including X-ray crystallography, nuclear magnetic resonance (NMR), and the latest electron microscopy have yielded direct detailed structural information on protein complexes. Unfortunately, none of these procedures is able to produce structures rapidly. Attempts to circumvent this problem are launched by Structural Genomics consortiums. For example, the Joint Core for Structural Genomics has released several X-ray structures from organisms such as *Thermatoga maritima* [25] on a large scale. The efficiency of these X-ray structures depends first on the targeted organism (to a great extent) and second on experimental phases of protein purification and crystallization that are heavily dependent on human input. Although there is an increase in the amount of available structures with about 20,000 entries on databases devoted to the subject (http://pqs.ebi.ac.uk/pqs-doc.shtml), these datasets are highly redundant, which reduces the amount of useful information available. To circumvent this problem, an effort has been made to create nonredundant sets that can be used for benchmarking. This issue is complicated by the different possible definitions of redundancy [26–28] that make the composition of the nonredundant datasets vary from 226 in Fariselli et al. [26], to Zhou and Shan [29], to 329 in Keskin et al. [27], to 115 in Yan and Honavar [28]. Recently Bradford and Westhead [30] created a dataset of 180 protein complexes divided manually into transient and obligate interactions.

An additional complication is related to the analysis of transient complexes including flexible protein regions and low-stability interactions. Examples of this type of interactions are homodimerization of the LicT [31] protein and the ribonuclease

Inhibitor-Angiogenin complex [32]. Flexibility and low stability make crystallization experiments difficult to conduct, making NMR the ideal technology to characterize domain interfaces and to detect disordered regions and the disorder–order transition on ligand binding [33].

A promising structure-based approach relies on the combination of electron microscopy and atomic structural information. High-resolution electron microscopy images can be used to build low-resolution models, typically in the 10Å resolution range, that are sufficient to establish the "shape" or envelope of protein and protein complexes. Electron microscopy has several advantages; because minimal amounts of samples are needed, the purification need not be of high quality, and it can be applied to disperse complexes [34] and to complexes that can be obtained in the form of two-dimensional crystals (2D electron microscopy) [35]. The envelopes obtained by electron microscopy can then be used directly to fit individual protein structures (or models) including additional experimental information [36], or it may be possible to use the experimental spatial envelope as a constraint to select correct docking models [8]. Recent examples of electron microscopy are the models of the yeast exosome [37] and the 80S ribosome [38], and actin binding to its chaperon molecule [39].

An illustrative example of combining protein modeling and electron microscopy data is the case of the apoptosome, an Apaf-1 cytochrome c complex that activates procaspase-9 [40]. The data obtained in this work helped to decipher the exact mechanism of a very important apoptosis triggering mechanism. Another interesting example is the use of computational and biochemical methods to conduct structural analyses of the seven proteins that compose the core building block of the nuclear pore complex [41].

As can be deduced from all these examples, solving protein complexes is far from being routine work, much less so if additional features are to be considered, such as the modulation of the interactions in different cellular states, posttranslational modifications, and other dynamic properties of the complexes.

8.3 THE RANGE OF COMPUTATIONAL METHODS

Computational approaches designed to experimentally address difficult biological cases tend to raise high expectations. However, in general, computational approaches and simulations require the existence of extensive sets of examples, good knowledge of the physical basis of the problem, and long periods of development. The prediction of the structure of protein complexes is no exception to these rules, and progress is complicated by the lack of comprehensive benchmark sets. In the case of protein modeling, homology comparative modeling is perhaps the subdiscipline in structural bioinformatics where prediction methods have been most successful, because protein structure modeling of high-quality 3D structures is useful for making high-resolution models (~2.5Å RMS) and has been proven to be efficient based on suitable templates. Generally speaking, although experimental approaches provide more information than predictive methods, in the particular case of protein interactions, computational approaches are at least as accurate as current experimental approaches.

8.3.1 Genomes, Sequences, and Domain Composition

The information obtained about the distribution and organization of bacterial genomes has facilitated the development of various methods for the prediction of interaction partners [42]. These methods include the phylogenetic profiling method, which attempts to identify genes that share the same pattern of presence or absence in a collection of genomes [43,44]. The rationale is that a group of genes sharing the same profile will encode proteins that are necessary for a common process, a relation that does not necessarily imply a physical interaction. The main drawback is that complete genomes are required to establish overall distribution of proteins [45,46]. A second method is the conservation of gene neighborhood, which relies on conservation of the operon structure [47,48]. As observed in prokaryotes, there is a tendency for a gene to have its neighbors conserved through the different lineages. The obvious caveat of this approach is to what extent it can be extended beyond the prokaryotic domain [49].

Third, the "gene-fusion" approach relies on the observation of protein pairs encoded by separate genes that are encoded by a single gene in other species, where they could have been originated by a gene-fusion event. These cases reveal a close relationship indicating an underlying functional relation between the corresponding proteins, which in some cases could correspond to a direct physical interaction [50,51]. Gene-fusion events are considered to be critical for the evolution of the species, as seen in the rerooting of the eukaryotic tree based on a derived gene-fusion tree analysis [52].

As an alternative to these three approaches, two other sequence-based methods have been developed to predict interaction events. The "mirror tree" method is based on extracting information from the possible coevolution of interacting proteins. Consequently the phylogenetic trees of coevolving proteins are expected to have a significant similarity, as detected in examples such as insulin and its receptors [53], and dockerins and cohexins [54]. Current methods are based on the direct comparison of the distance matrices of pairs of protein families [55] and its extension to large data sets [56]. Improvements have been described to predict interaction specificity [57].

Finally, the *in silico* two-hybrid system implies that traces of the coevolution of interacting proteins can be detected in the form of patterns of possible compensatory mutations in the interface of the two proteins. The so-called correlated mutations have been useful for predicting the tendency of interacting residue pairs to be located in the proximity of protein interfaces [58]. As in the case of mirror tree, this method has also been applied to large datasets [59].

Along with this type of approach, based on genome organization, a different approach is related to combinatorial domains of interacting proteins, analyzing the statistical tendency of proteins sharing related domains to associate. The rationale behind this concept is that proteins sharing a common domain will have a functional relation created by this domain [60–62]. This is expected to happen in lineages having a high degree of either domain shuffling and/or accretion during evolutionary time. For instance, eukaryotic proteins connected with immunological processes have been acquired in this way [63].

Considering the large number of multidomain proteins in eukaryotes, the association of domains is expected to generate a dense and complex network of interactions. Studies conducted on yeast MIPS, MYGD, and DIP protein-interacting databases extracted the domains defined in InterPro [64] that are involved in these interactions to infer conclusions. Another method considers proteins as collections of domains in which each domain is responsible for a specific interaction with another domain. This methodology is done in order to estimate the probabilities of interaction between the corresponding proteins [60,65].

8.3.2 Structure: What Is Known about Interacting Surfaces?

Understanding interaction properties is essential for developing reliable tools. Starting in the 1970s the physico-chemical features of protein complex interfaces have been analyzed, including statistical characteristics such as solvent exposure, type of amino acids involved, evolutionary conservation, and geometry of the sites [66]. For example, the amino acids for which the exposed surface area decreases by more than a given threshold (~1 Å^2) upon interaction are usually considered to be part of the interaction region [67]. These first studies were limited by the small number of available structures [66]. With the increase of information on protein complexes, it has been possible to establish better divisions of the interaction types, even if the number of known complexes is still very small compared to the potential number of interactions, which could be around 10,000 nonredundant domain interactions considering a space of approximately 1,000 folds [68].

Vadja and Camacho [69] concluded that in rigid-body docking the best docking results were obtained for complexes with a standard interface area of 1400 Å^2 < I ΔSÅ < 2000 Å^2 (where ΔSAÅ indicates changes in solvent accessible surface after separation of the complex).

Important facts for the classification of protein interactions include intrinsic and transient complexes, homo- and heterocomplexes, enzyme-inhibitor complexes, and protein–antibody complexes [67,69,70].

An obvious complication regarding the statistics on interaction sites is the existence of a potentially large number of protein interaction sites, which make the results obtained for the set of known sites less reliable. To circumvent this problem various methods calculate statistics by sampling surface-exposed patches of neighbor residues to represent the statistical properties of protein surfaces [67,71]. Similar strategies have been used for protein-DNA complexes [72,73], protein-RNA complexes [74], and carbohydrate binding sites [75]. In these cases the properties of the binding sites have shown little statistical significance, which complicates their use for the prediction of interacting surfaces [76]. However, there are a few rough, general rules that characterize interaction sites. For example heterodimmer surfaces are relatively flat, with an average surface of 600 to 3000 Å^2, and they do not show any special amino acid preferences, whereas in homodimmers the interfaces are slightly less planar and have an average surface of 400 to 4800 Å^2, showing a preference for nonpolar groups.

8.3.3 Predicting Structure from Protein Complexes: The Docking Problem

Docking algorithms are designed to simulate the physical interaction of two molecules of known structure, including the prediction of corresponding molecular structure [77–82]. The most common methods consider the proteins as rigid bodies, disregarding the conformational changes required for the adaptation of proteins upon binding. Of course, this is an oversimplification, because, first, flexibility is a driving force in the assembly of large complexes and, second, structurally disordered regions frequently contribute to the interacting regions, where they gain structure and contribute to the binding energy [83–85]. This relation between conformational changes and binding corresponds to what is known in biochemistry as allosteric signal transmission and induced fit movements [86].

Other criteria include the classification of protein complexes based on docking difficulty as described by Vajda and Camacho [69]. According to these authors there are five types of complexes that are defined according to surface type. Type I complexes include small (rigid interface) conformational changes such as trypsin and trypsin inhibitors; type II complexes also include small changes but have a surface size of over 2000 $Å^2$, as, for example, the Ras protein and its activating domain RasGAP. Type III complexes include moderate conformational changes but have larger surfaces than type I, like the Hyhel-5Fab with lysozyme. Type IV is restricted to side-chains and is represented by the CheY and Che-Y-binding protein CheA; type V involves substantial backbone changes, as in the case of cyclin A and cyclin dependent kinase 2.

It is not surprising that the most difficult cases are the transient complexes that undergo severe conformational changes, whereas the easiest one is the docking of enzymes and inhibitors.

A small number of approaches are addressing these problems by simulating flexibility and conformational changes during the interaction reaction. These algorithms are expected to perform better in those cases in which conformational flexibility is important, whereas the various rigid docking approaches might be sufficient for docking proteins in cases in which the binding does not involve big conformational changes [87].

The typical protocol for a docking experiment first involves the generation of a large set of putative complexes covering as many conformations as possible at a given resolution level (e.g., defined by the minimal angle for the orientation of the two structures). This collection of potential solutions is then analyzed in terms of their energies and a simple evaluation of the electrostatic complementarity.

Variations of this procedure include the application of statistical scoring potentials derived from known complexes, similar to those developed for threading methods. For instance, Kortemme, Morozov, and Baker [88] developed an orientation-dependent hydrogen bonding potential deduced from known crystal structures to discriminate the correct relative orientation of protein in protein complexes. Other methods use residue pair potentials previously extracted from interacting surfaces of known complexes, extending their predictions to homologs of known structure [89,90].

The evaluation of docking methods is a major concern, and over the last few years a significant community-wide evaluation has been organized (Critical Assessment of Prediction of Interactions, or CAPRI; http://capri.ebi.ac.uk) [76,91]. Although there is a general tendency to improve docking methods and the systems for evaluating their results, they are still a long way from providing satisfactory solutions [76]. The general experience of the CAPRI community indicates that the application of docking methods to cases of biological significance requires considerable human expertise to evaluate the available experimental information, such as point mutations or low-resolution NMR data. For instance, additional information was crucial to establish the structural model of the N-terminal region of the prokaryotic enhancer-binding protein XylR [92]. Lu, Lu, and Skolnick [90,93] extended their protein-structure prediction method (Multiprospector) to the prediction of protein complexes by trying all combinations of protein sequences in the structure of known complexes, searching for compatibility, assuming that proteins in the framework of the right complex will be more stable if isolated.

8.3.4 Hybrid Methods Based on Sequence and Structure

A widely used approach for building interaction networks is to extrapolate the experimental information from model systems (i.e., *S. cerevisiae*, *C. elegans*, *D. melanogaster,* and *H. pylori*), assuming that orthologous sequences will participate in similar interactions. For instance, an *E. coli* network was inferred from *H. pylori* data using sequence similarity and further clustering strategies [94]. Mathews et al.'s [95] method searches for *interologs* (potentially conserved interactions) in *C. elegans* using experimentally verified *S. cerevisiae* interacting partners. Although these studies are very interesting, this extrapolation involves a substantial risk, first because the conservation of interactions over a long evolutionary time has not yet been proven, and second because domain shuffling, characteristic of eukaryotes, increases interaction complexity. Aloy et al. [96] calculated the degree of conservation of interacting regions and concluded that similar interaction sites can be assigned to proteins if the sequence similarity is better than 30 to 40.

An alternative to filter docking solutions is to include external information derived from the analysis of evolutionary properties of protein families. The analyses of interaction interfaces have shown that the degree of conservation for these areas is not significantly higher than other areas in the protein [97] (see section 8.3.3).

Even if conserved residues can be part of interaction surfaces, other residues conserved in subfamilies, the so-called tree-determinants, can also be important for tracing functional interfaces. These tree-determinants point to positively selected changes in a protein family that potentially indicate the presence of functional-specific sites. In this position it is possible to find protein–protein interaction related sites [94,98]. The ability of these methods to predict sites that are specific [99] and the state-of-the-art in current methods for predicting functional sites [100] has been described in recent publications, including experiments demonstrating the capacity of those methods to predict residues that once swapped can produce a exchange of functional specificity between two protein subfamilies [65,101,102].

The "correlated mutations" method incorporates evolutionary information allowing the prediction of the overall trends of change in residues located in close proximity [58,103].

Our group has used a combination of evolutionary information with standard docking in the context of the CAPRI competition (section 8.3.3). Our methods were able to correctly predict 55% of laminin and 70% of nidogen interface residues in the laminin–nidogen complex (fig. 8.1). A typical problem in this type of approach is the difficulty of correctly establishing the structure of the complex even after determining with sufficient precision the regions of interaction. A good example of this problem was the prediction of the Rcc1-Ran complex [104], where the best scoring prediction showed the right regions in interaction but in the wrong relative orientation once the structure of the complex was available [105] (fig. 8.2).

Zhou and Shan [29] and Fariselli et al. [26] employed neural networks to combine sequence and structural information for the prediction of whether a residue is located in an interaction site of a protein with a known structure. Bradford et al. [30] used a Support Vector Machine approach to identify interface residues using sequence neighbors. In both cases, the interaction surfaces are represented as surface patches of neighboring residues with their associated sequence profiles (derived from multiple alignments). The accuracy of these methods is 70% for interaction-surface prediction.

FIGURE 8.1 Laminin–nidogen complex. The X-ray structure of the complex is available (pdb code 1NPE). Laminin is represented in dark gray and nidogen is represented in light gray. The contact residues are indicated in the space fill-in. For the best model, 55% and 70% of the interface residues were correctly predicted by our group using solely sequence-based methods, in laminin and nidogen, respectively.

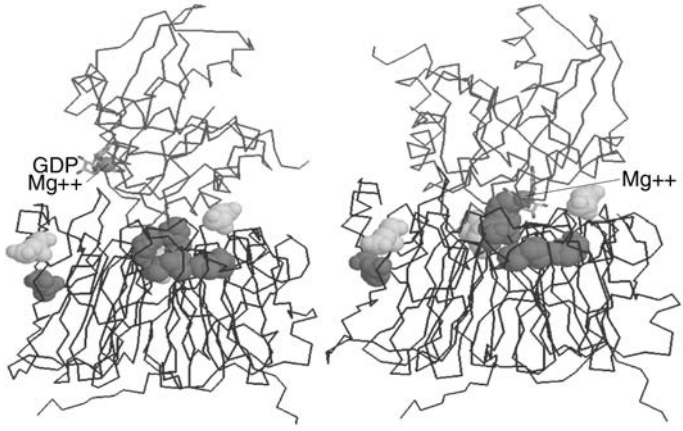


FIGURE 8.2 Prediction of the rcc1-ran complex. The left panel is the real structure (pdb code: 1I2M), whereas the right panel shows the model [104]. The figure shows the difference in orientation between the real structure and the model. Light gray indicates ran and dark gray indicates the Rcc1. Residues involved in the interface are indicated as spacefill models. The critical His 334 (critical for binding and catalysis) is close to the GTP and other catalytic residues in the model (right), whereas in the real structure this His 334 is located far away from the GTP and other catalytic residues. Therefore, the most feasible explanation is better described by the model. This figure illustrates the caveats of predicting a given interaction when the biology is complex.

8.4 MERGING EXPERIMENTAL AND COMPUTATIONAL METHODS

Genomic sequencing, proteome characterization, and structural genomics projects are providing a wealth of information about genomes, genes, and proteins. The recent advances in proteomics offer novel possibilities for understanding protein networks. Experimental and computational approaches developed in the last five years have provided useful information to address questions about properties, organization, evolution, and complexity of protein–protein interactions.

The promise for the future is that the integration of the information provided by network connectivity will be useful for overcoming some difficulties in the assignment of the protein function [106]. Gavin and collaborators [13] conducted the first survey of the functional organization of the yeast proteome by computationally analyzing 589 purified protein assemblies. Later on, Aloy et al. [107] used a large set of purified yeast protein complexes and obtained electron microscopy envelopes for 102 of them, for which they were able to build 45 three-dimensional models, using computational methods and all available experimental information.

Lappe and Holm [108] used a computational strategy ("pay-as-you-go") to exploit the scale-free property of networks representing biological systems to estimate that about 10,000 TAP-MS experiments would be enough to cover the full interactome. A rational design of these experiments by following known interactions was estimated to require four times fewer experiments than a pure random selection of the proteins used for the pull-down experiments.

Finally, Russell et al. [109] compared the available experimental and computational approaches, and proposed that, given their small overlap, only hybrid approaches would be able to provide a sufficient coverage of the interactome.

Hoffmann and Valencia [110] proposed a different view of the scope of computational and experimental methods by comparing the structure of the corresponding networks. They showed that methods with similar experimental or computational logic tended to find similar proteins to be the main interactors, even if the details of the interactions were different.

These and other studies point to the need to merge different experimental and computational techniques, because they represent largely orthogonal views of the interaction space.

8.5 WHERE IS THE INFORMATION?

With the emergence of high-throughput projects, the importance of data management is increasing rapidly, and a number of resources on protein interactions and protein complexes have been recently organized (table 8.1).

The Human Proteome Organization has initiated an effort to establish standards for the interchange of information between the various interaction databases. The

TABLE 8.1
Main Databases on Protein–Protein Interactions

Database	Site and Description
DIP	http://dip.doe-mbi.ucla.edu/
[118]	Stores experimentally determined interactions between proteins. Currently, it includes 18,488 interactions for 7,134 proteins in 104 organisms.
MINT	http://cbm.bio.uniroma2.it/mint/
[120]	Designed to store functional interactions between biological molecules (proteins, RNA, DNA). It is now focusing on experimentally verified direct and indirect protein–protein interactions.
BIND	http://www.bind.ca/
[121]	Contains full descriptions of interactions, molecular complexes, and pathways.
MIPS	http://www.mips.biochem.mpg.de/
[122]	Large collection of various types of interactions. Used commonly as equivalent to "hand-curated" set of interactions.
PathCalling Yeast Interaction Database	http://portal.curagen.com/extpc/com.curagen.portal.servlet.Yeast
[17]	Identifies protein–protein interactions on a genome-wide scale for functional assignment and drug-target discovery.
TheGRID	http://biodata.mshri.on.ca/grid/servlet/Index
	A database of genetic and physical interactions that contains interaction data from several sources, including MIPS and BIND.
IntAct	http//www.ebi.ac.uk/intact
[123]	The project aims to define a standard for the representation and annotation of protein–protein interactions and to develop a public database of experimentally identified and predicted interactions.

consortium behind these initiatives has chosen the basic XML layer for the exchange and has prepared a vocabulary for the description of experimental and computational techniques.

In addition to the data standardization issue, the distribution of information is also an important problem, particularly regarding the need for sharing data with the different groups involved, which are often at various locations and multidisciplinary in nature [111]. An example of this type of technological initiative is the PLANET project (http://eu-plant-genome.net), which makes different data repositories available through a single interface using BioMOBY technology [112].

The building and maintenance of the protein interaction databases is a major effort, which offers the possibility of overcoming some of the limitations of the traditional sequence databases. In particular, most current protein interaction databases are linked to text-mining projects with the aim of not only facilitating the annotation process but also (and perhaps more important) maintaining the links between the interactions stored in the database and the basic experiments described in the literature. During the last few years the technology in the text-mining field has improved considerably [113–117] (see chapter 6), even if key problems such as the identification of protein and gene names are still a challenge. For example, in 2001 it was only possible to link 30% of the DIP database entries to the available literature [118], and 20% of the missing links were explained by inaccuracies in the text-mining system. Surprisingly, the remaining 80% occurred because the protein names used were not found in any of the available Medline entries or there was a lack of information about particular interactions in the literature.

The new generation of text-mining tools has overcome many of these problems, hence leading to more efficient navigation of the literature [119] (http://pdg.cnb.uam.es/UniPub/iHOP/). The iHOP system is now being integrated with the INTACT database, creating direct links between the database and the literature.

8.6 PERSPECTIVES

Genomic sequencing, proteome characterization, and structural genomics projects are providing a remarkable amount of information about the mechanics of living cells. A remarkable set of seven high-throughput proteomics experiments now offers the first overall view of the organization of individual proteins in interactions and complexes. Parallel to the experimental approaches, a number of computational approaches have addressed the problems of identification of protein–protein interaction partners and the detailed description of the protein interaction sites. Given the potentially large size of the interaction space and its complexity (in terms of space, time, and modifications), current trends indicate that a well-designed combination of experimental and theoretical data will be necessary for deciphering the biological characteristics of the interaction networks.

For all this work in protein interaction networks to be useful to modern molecular biology, not only should the information be provided not only in general terms (network properties), but it should also include molecular details of the interactions (i.e., reproducing the structure of protein complexes using docking methods, including protein flexibility and conformational changes).

Finally, it is important to keep in mind that the characterization of protein interaction networks is only the first step toward an understanding of cellular systems that includes localization and timing of the interactions, as well as the influence of the various posttranslational modifications and gene control steps.

ACKNOWLEDGMENTS

We acknowledge Iakes Ezkurdia and Gonzalo Lopez for their valuable input.

REFERENCES

1. Lander, E. S., L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, et al. 2001. Initial sequencing and analysis of the human genome. *Nature* 409:860–921.
2. Venter, J. C., M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, et al. 2001. The sequence of the human genome. *Science* 291:1304–51.
3. Jasny, B. R., and L. Roberts. 2004. Solving gene expression. *Science* 306:629.
4. Galperin, M. 2005. The Molecular Biology Database collection: 2005 update. *Nucleic Acid Res* 33:D5–24.
5. Ng, S., and S. H. Tan. 2004. Discovering protein–protein interactions. *J Bioinform Comp Biol* 1:711–41.
6. Joshi-Tope, G., I. Vastrik, G. R. Gopinath, L. Matthews, E. Schmidt, M. Gillespie, P. D'Eustachio, et al. 2003. *The genome knowledgebase: A resource for biologists and bioinformaticists*. Paper presented at the Cold Spring Harbor Symposium on Quantitative Biology, Cold Spring Harbor, NY.
7. Golemis, E. 2002. Protein–protein interactions: A molecular cloning manual. New York: Cold Spring Harbor Laboratory Press.
8. Sali, A., R. Glaeser, T. Earnest, and W. Baumeister. 2003. From words to literature in structural proteomics. *Nature* 422:216–25.
9. Seraphin, B. 2002. Identification of transiently interacting proteins and of stable protein complexes. *Adv Protein Chem* 61:99–117.
10. Rappsilber, J., S. Siniossoglou, E. C. Hurt, and M. Mann. 2000. A generic strategy to analyze the spatial organization of multi-protein complexes by cross-linking and mass spectrometry. *Anal Chem* 72:267–75.
11. Aebersold, R., and M. Mann. 2003. Mass spectrometry-based proteomics. *Nature* 422:198–207.
12. Wen, J., T. Arakawa, and J. S. Philo. 1996. Size-exclusion chromatography with on-line light-scattering, absorbance, and refractive index detectors for studying proteins and their interactions. *Anal Biochem* 240:155–66.
13. Gavin, A. C., M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz. et al. 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* 415:141–7.
14. Ho, Y., A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, S. L. Adams, A. Millar, et al. 2002. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature* 415:180–3.
15. Fields, S., and O. Song. 1989. A novel genetic system to detect protein–protein interactions. *Nature* 340:245–6.

16. Phizicky, E., P. I. Bastiaens, H. Zhu, M. Snyder, and S. Fields. 2003. Protein analysis on a proteomic scale. *Nature* 422:208–15.
17. Uetz, P., L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, and D. Lockshon. 2000. A comprehensive analysis of protein–protein interactions in Saccharomyces cerevisiae. *Nature* 403:623–7.
18. Rain, J. C., L. Selig, H. De Reuse, V. Battaglia, C. Reverdy, S. Simon, G. Lenzen, et al. 2001. The protein–protein interaction map of Helicobacter pylori. *Nature* 409:211–5.
19. Li, S., C. M. Armstrong, N. Bertin, H. Ge, S, Milstein, M. Boxem, P. O. Vidalain, et al. 2004. A map of the interactome network of the metazoan C. elegans. *Science* 303:540–3.
20. Stagljar, I., and S. Fields. 2002. Analysis of membrane protein interactions using yeast-based technologies. *Trends Biochem Sci* 27:559–63.
21. Wells, J. A. 1991. Systematic mutational analyses of protein–protein interfaces. *Methods Enzymol* 202:390–411.
22. Tong, A. H., B. Drees, G. Nardelli, G. D. Bader, B. Brannetti, L. Castagnoli, M. Evangelista, et al. 2002. A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science* 295:321–4.
23. Zhu, H., M. Bilgin, R. Bangham, D. Hall, A. Casamayor, P. Bertone, N. Lan, et al. 2001. Global analysis of protein activities using proteome chips. *Science* 293:2101–5.
24. Grunenfelder, B., and E. A. Winzeler. 2002. Treasures and traps in genome-wide data sets: Case examples from yeast. *Nat Rev Genet* 3:653–61.
25. Lesley, S. A., P. Kuhn, A. Godzik, A. M. Deacon, I. Mathews, A. Kreusch, G. Spraggon, et al. 2002. Structural genomics of the Thermotoga maritima proteome implemented in a high-throughput structure determination pipeline. *Proc Natl Acad Sci USA* 99:11664–9.
26. Fariselli, P., F. Pazos, A. Valencia, and R. Casadio. 2002. Prediction of protein–protein interaction sites in heterocomplexes with neural networks. *Eur J Biochem* 269:1356–61.
27. Keskin, O., C. J. Tsai, H. Wolfson, and R. Nussinov. 2004. A new, structurally nonredundant, diverse data set of protein–protein interfaces and its implications. *Protein Sci* 13:1043–55.
28. Yan C, D. Dobbs, and V. Honavar. 2003. Identification of residues involved in protein–protein interaction from amino acid sequence: A support vector machine approach. In *Intelligent systems design and applications,* ed. A. Abraham, K. Franke, and M. Köppen, 53–62. Berlin: Springer-Verlag.
29. Zhou, H. X., and Y. Shan. 2001. Prediction of protein interaction sites from sequence profile and residue neighbor list. *Proteins* 44:336–43.
30. Bradford, J. R., and D. R. Westhead. 2005. Improved prediction of protein–protein binding sites using a support vector machines approach. *Bioinformatics* 21:1487–94.
31. van Tilbeurgh, H., D. Le Coq, and N. Declerck. 2001. Crystal structure of an activated form of the PTS regulation domain from the LicT transcriptional antiterminator. *Embo J* 20:3789–99.
32. Papageorgiou, A. C., R. Shapiro, K. R. Acharya. 1997. Molecular recognition of human angiogenin by placental ribonuclease inhibitor—An X-ray crystallographic study at 2.0 A resolution. *Embo J* 16:5162–77.
33. Hiller, S., A. Kohl, F. Fiorito, T. Herrmann, G. Wider, J. Tschopp, M. G. Grutter, and K. Wuthrich. 2003. NMR structure of the apoptosis- and inflammation-related NALP1 pyrin domain. *Structure* (Camb) 11:1199–1205.

34. Martin-Benito, J., E. Area, J. Ortega, O. Llorca, J. M. Valpuesta, J. L. Carrascosa, and J. Ortin. 2001. Three-dimensional reconstruction of a recombinant influenza virus ribonucleoprotein particle. *EMBO Rep* 2:313–7.
35. Mitsuoka, K., T. Hirai, K. Murata, A. Miyazawa, A. Kidera, Y. Kimura, and Y. Fujiyoshi. 1999. The structure of bacteriorhodopsin at 3.0 A resolution based on electron crystallography: Implication of the charge distribution. *J Mol Biol* 286:861–82.
36. Chacon, P., and W. Wriggers. 2002. Multi-resolution contour-based fitting of macromolecular structures. *J Mol Biol* 317:375–84.
37. Aloy, P., F. D. Ciccarelli, C. Leutwein, A. C. Gavin, G. Superti-Furga, P. Bork, B. Bottcher, and R. B. Russell. 2002. A complex prediction: Three-dimensional model of the yeast exosome. *EMBO Rep* 3:628–35.
38. Spahn, C. M., R. Beckmann, N. Eswar, P. A. Penczek, A. Sali, G. Blobel, and J. Frank. 2001. Structure of the 80S ribosome from Saccharomyces cerevisiae—tRNA-ribosome and subunit-subunit interactions. *Cell* 107:373–86.
39. Llorca, O., E. A. McCormack, G. Hynes, J. Grantham, J. Cordell, J. L. Carrascosa, K. R. Willison, J. J. Fernandez, and J. M. Valpuesta. 1999. Eukaryotic type II chaperonin CCT interacts with actin through specific subunits. *Nature* 402:693–6.
40. Acehan, D., X. Jiang, D. G. Morgan, J. E. Heuser, X. Wang, and C. W. Akey. 2002. Three-dimensional structure of the apoptosome: Implications for assembly, procaspase-9 binding, and activation. *Mol Cell* 9:423–32.
41. Devos, D., S. Dokudovskaya, F. Alber, R. Williams, B. T. Chait, A. Sali, and M. P. Rout. 2004. Components of coated vesicles and nuclear pore complexes share a common molecular architecture. *PLoS Biol* 2:E380.
42. von Mering, C., R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork. 2002. Comparative assessment of large-scale data sets of protein–protein interactions. *Nature* 417:399–403.
43. Enault, F., K. Suhre, O. Poirot, C. Abergel, and J. M. Claverie. 2003. Phydbac (phylogenomic display of bacterial genes): An interactive resource for the annotation of bacterial genomes. *Nucleic Acids Res* 31:3720–2.
44. Enault, F., K. Suhre, O. Poirot, C. Abergel, and J. M. Claverie. 2004. Phydbac2: Improved inference of gene function using interactive phylogenomic profiling and chromosomal location analysis. *Nucleic Acids Res* 32:W336–9.
45. Gaasterland, T., and M. A. Ragan. 1998. Microbial genescapes: Phyletic and functional patterns of ORF distribution among prokaryotes. *Microb Comp Genomics* 3:199–217.
46. Pellegrini, M., E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. 1999. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proc Natl Acad Sci USA* 96:4285–8.
47. Dandekar, T., B. Snel, M. Huynen, and P. Bork. 1998. Conservation of gene order: A fingerprint of proteins that physically interact. *Trends Biochem Sci* 23:324–8.
48. Overbeek, R., M. Fonstein, M. D'Souza, G. D. Pusch, and N. Maltsev. 1999. The use of gene clusters to infer functional coupling. *Proc Natl Acad Sci USA* 96:2896–901.
49. Morett, E., J. O. Korbel, E. Rajan, G. Saab-Rincon, L. Olvera, M. Olvera, S. Schmidt, B. Snel, and P. Bork. 2003. Systematic discovery of analogous enzymes in thiamin biosynthesis. *Nat Biotechnol* 21:790–5.
50. Enright, A. J., I. Iliopoulos, N. Kyrpides, and C. A. Ouzounis. 1999. Protein interaction maps for complete genomes based on gene fusion events. *Nature* 402:86–90.
51. Marcotte, E. M., M. Pellegrini, H. L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg. 1999. Detecting protein function and protein–protein interactions from genome sequences. *Science* 285:751–3.

52. Stechmann, A., and T. Cavalier-Smith. 2002. Rooting the eukaryote tree by using a derived gene fusion. *Science* 297:89–91.
53. Fryxell, K. J. 1996. The coevolution of gene family trees. *Trends Genet* 12:364–9.
54. Pages, S., A. Belaich, J. P. Belaich, E. Morag, R. Lamed, Y. Shoham, and E. A. Bayer. 1997. Species-specificity of the cohesin-dockerin interaction between Clostridium thermocellum and Clostridium cellulolyticum: Prediction of specificity determinants of the dockerin domain. *Proteins* 29:517–27.
55. Goh, C. S., A. A. Bogan, M. Joachimiak, D. Walther, and F. E. Cohen. 2000. Co-evolution of proteins with their interaction partners. *J Mol Biol* 299:283–93.
56. Pazos, F., and A. Valencia. 2001. Similarity of phylogenetic trees as indicator of protein–protein interaction. *Protein Eng* 14:609–14.
57. Ramani, A. K., and E. M. Marcotte. 2003. Exploiting the co-evolution of interacting proteins to discover interaction specificity. *J Mol Biol* 327:273–84.
58. Pazos, F., M. Helmer-Citterich, G. Ausiello, and A. Valencia. 1997. Correlated mutations contain information about protein–protein interaction. *J Mol Biol* 271:511–23.
59. Pazos, F., and A. Valencia. 2002. In silico two-hybrid system for the selection of physically interacting protein pairs. *Proteins* 47:219–27.
60. Gomez, S. M., S. H. Lo, and A. Rzhetsky. 2001. Probabilistic prediction of unknown metabolic and signal-transduction networks. *Genetics* 159:1291–8.
61. Wojcik, J., and V. Schachter. 2001. Protein–protein interaction map inference using interacting domain profile pairs. *Bioinformatics* 17, Suppl. no. 1:S296–305.
62. Gomez, S. M., W. S. Noble, and A. Rzhetsky. 2003. Learning to predict protein–protein interactions from protein sequences. *Bioinformatics* 19:1875–81.
63. Koonin, E. V., and L. Aravind. 2002. Origin and evolution of eukaryotic apoptosis: The bacterial connection. *Cell Death Differ* 9:394–404.
64. Sprinzak, E., and H. Margalit. 2001. Correlated sequence-signatures as markers of protein–protein interaction. *J Mol Biol* 311:681–92.
65. Morillas, M., P. Gomez-Puertas, A. Bentebibel, E. Selles, N. Casals, A. Valencia, F. G. Hegardt, G. Asins, and D. Serra. 2003. Identification of conserved amino acid residues in rat liver carnitine palmitoyltransferase I critical for malonyl-CoA inhibition: Mutation of methionine 593 abolishes malonyl-CoA inhibition. *J Biol Chem* 278:9058–63.
66. Chothia, C., and J. Janin. 1975. Principles of protein–protein recognition. *Nature* 256:705–8.
67. Jones, S., and J. M. Thornton. 1996. Principles of protein–protein interactions. *Proc Natl Acad Sci USA* 93:13–20.
68. Aloy, P., and R. B. Russell. 2004. Ten thousand interactions for the molecular biologist. *Nat Biotechnol* 22:1317–21.
69. Vajda, S., and C. J. Camacho. 2004. Protein–protein docking: Is the glass half-full or half-empty? *Trends Biotechnol* 22:110–6.
70. Nooren, I. M., and J. M. Thornton. 2003. Diversity of protein–protein interactions. *Embo J* 22:3486–92.
71. Jones, S., and J. M. Thornton. 1997. Analysis of protein–protein interaction sites using surface patches. *J Mol Biol* 272:121–32.
72. Jones, S., P. van Heyningen, H. M. Berman, and J. M. Thornton. 1999. Protein-DNA interactions: A structural analysis. *J Mol Biol* 287:877–96.
73. Shanahan, H. P., M. A. Garcia, S. Jones, and J. M. Thornton. 2004. Identifying DNA-binding proteins using structural motifs and the electrostatic potential. *Nucleic Acids Res* 32:4732–41.

74. Luscombe, N. M., R. A. Laskowski, and J. M. Thornton. 2001. Amino acid-base interactions: A three-dimensional analysis of protein-DNA interactions at an atomic level. *Nucleic Acids Res* 29:2860–74.
75. Taroni, C., S. Jones, and J. M. Thornton. 2000. Analysis and prediction of carbohydrate binding sites. *Protein Eng* 13:89–98.
76. Janin, J., K. Henrick, J. Moult, L. T. Eyck, M. J. Sternberg, S. Vajda, I. Vakser, and S. J. Wodak. 2003. CAPRI: A Critical Assessment of PRedicted Interactions. *Proteins* 52:2–9.
77. Halperin, I., B. Ma, H. Wolfson, and R. Nussinov. 2002. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins* 47:409–43.
78. Halperin, I., H. Wolfson, and R. Nussinov. 2004. Protein–protein interactions: Coupling of structurally conserved residues and of hot spots across interfaces. Implications for docking. *Structure* (Camb) 12:1027–38.
79. Smith, G. R., and M. J. Sternberg. 2002. Prediction of protein–protein interactions by docking methods. *Curr Opin Struct Biol* 12:28–35.
80. Verkhivker, G. M., D. Bouzida, D. K. Gehlhaar, P. A. Rejto, S. T. Freer, and P. W. Rose. 2002. Complexity and simplicity of ligand-macromolecule interactions: The energy landscape perspective. *Curr Opin Struct Biol* 12:197–203.
81. Baker, N. A., and J. A. McCammon. 2003. Electrostatic interactions. *Methods Biochem Anal* 44:427–40.
82. Krumrine J., F. Raubacher, N. Brooijmans, and I. Kuntz. 2003. Principles and methods of docking and ligand design. In *Structural bioinformatics,* ed. P. E. Bourne and H. Weissig, 443–76, Hoboken, NJ: Wiley-Liss.
83. Dunker, A. K., C. J. Brown, J. D. Lawson, L. M. Iakoucheva, and Z. Obradovic. 2002. Intrinsic disorder and protein function. *Biochemistry* 41:6573–82.
84. Tompa, P. 2003. Intrinsically unstructured proteins evolve by repeat expansion. *Bioessays* 25:847–55.
85. Uversky, V. N. 2002. Natively unfolded proteins: A point where biology waits for physics. *Protein Sci* 11:739–56.
86. Luque, I., and E. Freire. 2000. Structural stability of binding sites: Consequences for binding affinity and allosteric effects. *Proteins* Suppl. no. 4:63–71.
87. Fernandez-Recio, J., M. Totrov, and R. Abagyan. 2002. Soft protein–protein docking in internal coordinates. *Protein Sci* 11:280–91.
88. Kortemme, T., A. V. Morozov, and D. Baker. 2003. An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein–protein complexes. *J Mol Biol* 326:1239–59.
89. Aloy, P., and R. B. Russell. 2003. InterPreTS: Protein interaction prediction through tertiary structure. *Bioinformatics* 19:161–2.
90. Lu, L., H. Lu, and J. Skolnick. 2002. MULTIPROSPECTOR: An algorithm for the prediction of protein–protein interactions by multimeric threading. *Proteins* 49:350–64.
91. Janin, J. 2005. Assessing predictions of protein–protein interaction: The CAPRI experiment. *Protein Sci* 14:278–83.
92. Devos, D., J. Garmendia, V. de Lorenzo, and A. Valencia. 2002. Deciphering the action of aromatic effectors on the prokaryotic enhancer-binding protein XylR: A structural model of its N-terminal domain. *Environ Microbiol* 4:29–41.
93. Lu, H., L. Lu, and J. Skolnick. 2003. Development of unified statistical potentials describing protein–protein interactions. *Biophys J* 84:1895–1901.
94. Lichtarge, O., and M. E. Sowa. 2002. Evolutionary predictions of binding surfaces and interactions. *Curr Opin Struct Biol* 12:21–27.

95. Matthews, L. R., P. Vaglio, J. Reboul, H. Ge, B. P. Davis, J. Garrels, S. Vincent, and M. Vidal. 2001. Identification of potential interaction networks using sequence-based searches for conserved protein–protein interactions or "interologs." *Genome Res* 11:2120–6.
96. Aloy, P., H. Ceulemans, A. Stark, and R. B. Russell. The relationship between sequence and interaction divergence in proteins. *J Mol Biol* 332:989–98.
97. Grishin, N. V., and M. A. Phillips. 1994. The subunit interfaces of oligomeric enzymes are conserved to a similar extent to the overall protein sequences. *Protein Sci* 3:2455–8.
98. Armon, A., D. Graur, and N. Ben-Tal. 2001. ConSurf: An algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information. *J Mol Biol* 307:447–63.
99. del Sol Mesa, A., F. Pazos, and A. Valencia. 2003. Automatic methods for predicting functionally important residues. *J Mol Biol* 326:1289-302.
100. Lopez-Romero, P., M. J. Gomez, P. Gomez-Puertas, and A. Valencia. 2004. Prediction of functional sites in proteins by evolutionary methods. In *Methods in proteome and protein analyses,* ed. R. M. Kamp, J. Calvete, and T. Choli-Papadopoulou, 319–336. Heidelberg, Germany: Springer-Verlag.
101. Bauer, B., G. Mirey, I. R. Vetter, J. A. Garcia-Ranea, A. Valencia, A. Wittinghofer, J. H. Camonis, and R. H. Cool. 1999. Effector recognition by the small GTP-binding proteins Ras and Ral. *J Biol Chem* 274:17763–70.
102. Stenmark, H., A. Valencia, O. Martinez, O. Ullrich, B. Goud, and M. Zerial. 1994. Distinct structural elements of rab5 define its functional specificity. *EMBO* 13:575–83.
103. Carettoni, D., P. Gomez-Puertas, L. Yim, J. Mingorance, O. Massidda, M. Vicente, A. Valencia, E. Domenici, and D. Anderluzzi. 2003. Phage-display and correlated mutations identify an essential region of subdomain 1C involved in homodimerization of Escherichia coli FtsA. *Proteins* 50:192–206.
104. Azuma, Y., L. Renault, J. A. Garcia-Ranea, A. Valencia, T. Nishimoto, and A. Wittinghofer. 1999. Model of the ran-RCC1 interaction using biochemical and docking experiments. *J Mol Biol* 289:1119–30.
105. Renault, L., J. Kuhlmann, A. Henkel, and A. Wittinghofer. 2001. Structural basis for guanine nucleotide exchange on Ran by the regulator of chromosome condensation (RCC1). *Cell* 105:245–55.
106. Devos, D., and A. Valencia. 2000. Practical limits of function prediction. *Proteins* 41:98–107.
107. Aloy, P., B. Bottcher, H. Ceulemans, C. Leutwein, C. Mellwig, S. Fischer, A. C. Gavin, et al. Structure-based assembly of protein complexes in yeast. *Science* 303:2026–9.
108. Lappe, M., and L. Holm. 2004. Unraveling protein interaction networks with near-optimal efficiency. *Nat Biotechnol* 22:98–103.
109. Russell, R. B., F. Alber, P. Aloy, F. P. Davis, D. Korkin, M. Pichaud, M. Topf, and A. Sali. 2004. A structural perspective on protein–protein interactions. *Curr Opin Struct Biol* 14:313–24.
110. Hoffmann, R., and A. Valencia. 2003. Life cycles of successful genes. *Trends Genet* 19:79–81.
111. Chicurel, M. 2002. Bioinformatics: Bringing it all together. *Nature* 419:751, 753, 755.
112. Wilkinson, M. D., and M. Links. 2002. BioMOBY: An open source biological web services proposal. *Brief Bioinform* 3:331–41.

113. Blaschke C, M. A. Andrade, C. Ouzounis, and A. Valencia. 1999. Automatic extraction of biological information from scientific text: Protein–protein interactions. Paper presented at the International Conference in Intelligent Systems for Molecular Biology, 1999.
114. Thomas, J., D. Millward, C. Ouzounis, S. Pulman, and M. Carroll. 2000. Automatic extraction of protein interactions from scientific abstracts. Paper presented at the Pacific Symposium on Biocomputing, Oahu, Hawaii.
115. Frieman, C., P. Kra, M. Krauthammer, H. Yu, and A. Rzhetsky. 2001. GENIS: A natural language processing system for the extraction of molecular pathways from journal articles. Paper presented at the 9th International Conference on Intelligent Systems for Molecular Biology, Copenhagen, Denmark.
116. Blaschke, C., and A. Valencia. 2001. Can bibliographic pointers for known biological data be found automatically? Protein interactions as a case study. *Comp Funct Genom* 2:196–206.
117. Blaschke, C., L. Hirschman, and A. Valencia. 2002. Information extraction in molecular biology. *Briefings in Bioinformatics* 3:154–65.
118. Xenarios, I., E. Fernandez, L. Salwinski, X. J. Duan, M. J. Thompson, E. M. Marcotte, and D. Eisenberg. 2001. DIP: The Database of Interacting Proteins: 2001 update. *Nucleic Acids Res* 29:239–41.
119. Hoffmann, R., and A. Valencia. 2004. A gene network for navigating the literature. *Nat Genet* 36:664.
120. Zanzoni, A., L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni. 2002. MINT: A Molecular INTeraction database. *FEBS Lett* 513:135–40.
121. Bader, G. D., D. Betel, and C. W. Hogue. 2003. BIND: The Biomolecular Interaction Network Database. *Nucleic Acids Res* 31:248–50.
122. Mewes, H. W., D. Frishman, U. Guldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Munsterkotter, S. Rudd, and B. Weil. 2002. MIPS: A database for genomes and protein sequences. *Nucleic Acids Res* 30:31–4.
123. Hermjakob, H., L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, et al. IntAct: An open source molecular interaction database. *Nucleic Acids Res* 32:D452–5.

9 *In Silico* siRNA Design

Darryl León
SciTegic

CONTENTS

9.1 INTRODUCTION

9.1.1 RNAi Biology

Sequencing the human genome was one of the most important scientific milestones in the last century. This feat has inspired others to take the next steps in understanding how a complex set of genes affects specific phenotypes. Many pharmaceutical and biotechnology companies now use a variety of mRNA expression technologies to help them with understanding how a target gene is associated with a disease. During this target validation step, scientists would like to confirm gene regulation and be able to find a direct association of a target expression with a disease or health condition.

RNA interference (RNAi) is a relatively new tool researchers can use for targeting mRNA expression levels. RNAi can be applied to target validation, protein knockdown studies, gene function, pathway elucidation, and therapeutic development [1].

Because this technique is relatively inexpensive to perform, many academics have introduced this new and simple tool into their own research labs. This exciting technology has not only intrigued target validation scientists in academic and biopharmaceutical laboratories but also caught the attention of many entrepreneurs who believe this technology has specific therapeutic application for key diseases. Several companies have been formed that specialize in looking for ways to silence target genes related to specific diseases, or they supply unique reagents and delivery methods for RNAi protocols. *In silico* small interferring RNA (siRNA) design can contribute to an RNAi experiment, but before exploring this topic, it is important to understand how siRNA is used to silence a target gene.

The cellular response to RNAi is an intrinsic reaction that is implicated in modulating mRNA expression and preventing viral infection. Essentially, RNAi is mediated by small interfering RNAs that are derived from long double-stranded RNAs (dsRNAs) [2] (fig. 9.1). The long dsRNAs are cleaved with an enzyme complex called Dicer (DCR-1) [3]. An siRNA is created, and it appears to arbitrate the degradation of the corresponding single-stranded mRNA using the RNA-induced silencing complex (RISC) [4]. The result is a cleaved mRNA molecule, and this degradation leads to the down-regulation of a target gene [5]. One of the first RNA interference studies was done by injecting double-stranded RNA into a cell from *Caenorhabditis elegans* [6]. Later, it was shown that a set of 21-nucleotide small interfering RNA duplexes could specifically suppress expression of genes in several mammalian cell lines [7].

The introduction of synthesized siRNA molecules or the expression of short hairpin RNA (shRNA) precursor structures is important in mammalian RNAi experiments. However, the introduction of dsRNA longer than 30 nucleotides (in mammalian cells) causes an apoptotic response. This interferon response is initiated by

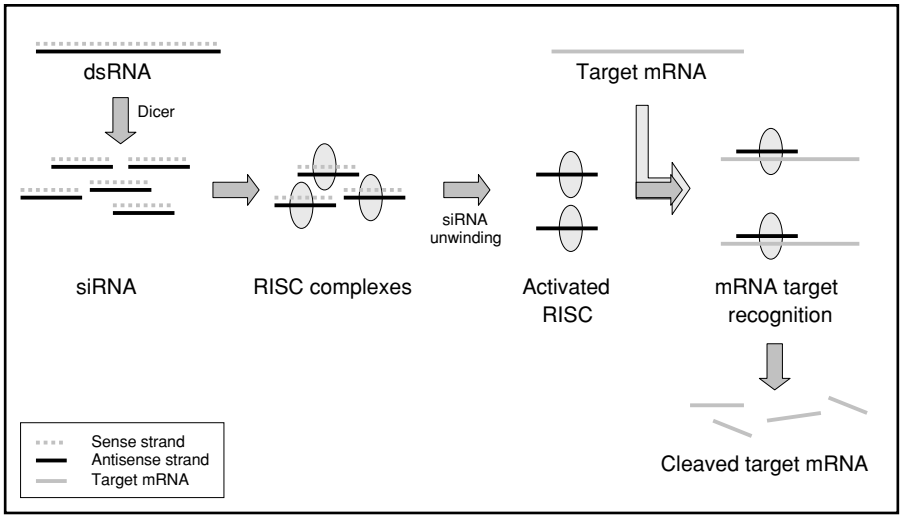


FIGURE 9.1 Target mRNA silencing. A simple schematic showing how siRNA molecules are involved in the RNA interference process [9].

the dsRNA-dependent kinase (PKR) [8]. There are several techniques used to induce RNA interference in mammalian cells with siRNA [9]:

- Chemical synthesis of siRNA
- *In vitro* transcription of siRNA
- Preparation of a population of siRNAs by digestion of long dsRNA with RNase II or Dicer
- *In vivo* expression of a hairpin siRNA from an expression vector
- *In vivo* expression of a hairpin siRNA from a PCR-derived expression cassette

Ultimately, no matter which approach is used, the potency and efficiency of the siRNA molecule is one of the main elements of a successful silencing experiment. Another key step in creating a highly effective RNAi experiment is to avoid the off-target effects that occur when siRNA molecules bind to sequences similar to the target gene in a genome. Microarray experiments have demonstrated that siRNA can produce nonspecific off-target effects. Other approaches have shown induction of nonspecific interferon response, being most obvious by shRNAs. This is where *in silico* approaches can be used to help to decrease the chances of accidentally silencing the wrong gene. Computational methods for avoiding this problem are discussed later in this chapter.

9.1.2 siRNA Technology

Several groups have performed various experiments to determine the preferred characteristics when designing siRNA molecules and the important considerations when selecting a target sequence. For example, designing a specific and potent siRNA is very important for any RNA interference study. The benefits of generating potent siRNAs may include the following [1]:

- A lower amount of siRNA material is required to trigger the RNAi response.
- The lower amount of siRNA helps minimize the number of off-target effects.
- The lower amount of siRNA can minimize the interferon activation pathway.
- One can avoid toxic concentrations with a lower amount of siRNA material.

Scientists have also measured the extent of RNA knockdown in several genes, and their analysis showed the following parameters affect knockdown effectiveness [10].

- Duplexes that targeted the middle of a coding sequence were less efficient at silencing target genes.
- Duplexes that targeted the 3'UTR and coding sequence were comparable.
- Pooling of duplexes was significantly efficient in gene expression knockdown.
- Duplexes that achieved over a 70% knockdown of the mRNA showed nucleotide preferences at positions 11(G or C) and 19 (T), respectively.

Because the physical attributes of siRNA molecules are important for a successful RNAi experiment, one should use siRNA molecules at their lowest effective amount [11]. As the reader will learn in the next several sections, the suggested guidelines for designing a specific and potent siRNA have been exploited and automated using a variety of *in silico* methods. It should also be noted that these are simply guidelines, and they may still lead to off-target effects or other undesired experimental results.

9.2 siRNA DESIGN

9.2.1 Designing an Optimized siRNA

Many design rules from various groups have been created to optimize the specificity of a siRNA molecule (fig. 9.2). Many of these rules have evolved as the RNAi technology has become more prevalent in mRNA expression studies. How an siRNA duplex sequence is related to its target mRNA sequence can be illustrated with an example target region from Lamin A/C [7].

```
Lamin A/C Target
Targeted region (cDNA): 5'-AACTGGACTTCCAGAAGAACATC-3'

mRNA region              5'-AACUGGACUUCCAGAAGAACAUC-3'

siRNA
Sense siRNA:              5'-CUGGACUUCCAGAAGAACAdTdT-3'
Antisense siRNA:     3'-dTdTGACCUGAAGGUCUUCUUGU-5'

Interaction
antisense siRNA:     3'-dTdT GACCUGAAGGUCUUCUUGU-5'
mRNA region           5'-AA  CUGGACUUCCAGAAGAACAUC-3'
```

Because the antisense siRNA strand interacts with the mRNA of the target gene, one should find a region on the target sequence that interacts well with the antisense siRNA strand, and one must also use suitable design rules for creating a specific and stable siRNA molecule for a successful RNAi experiment.

The first set of rules for designing potent siRNA molecules was established by Tuschl's group (http://www.rockefeller.edu/labheads/tuschl/sirna.html [12]). Many current siRNA designers and software programs use these rules to help scientists generate potent siRNA molecules. The rules are simple to understand and have been well summarized by Ding and Lawrence [13].

1. siRNA duplexes should be composed of 21-nt sense and 21-nt antisense strands, paired so that each has a 2-nt 3' dTdT overhang.
2. The targeted region is selected from a given cDNA sequence beginning 50 to 100 nt downstream of the start codon (3' untranslated regions [UTRs] also have been successfully targeted).
3. The target motif is selected in the following order of preferences:
   a. NAR(N17)YNN, where N is any nucleotide, R is purine (A or G), and Y is pyrimidine (C or U)

   b. AA(N19)TT
   c. NA(N21)
4. Nucleotides 1 to 19 of the sense siRNA strand correspond to positions 3 to 21 of the 23-nt target motif.
5. The target sequence is selected to have approximately 50% GC content.
6. Selected siRNA sequences should be aligned against Expressed Sequence Tag (EST) libraries to ensure that only one gene will be targeted.

More recently, a few academic research labs and companies have examined the properties of effective siRNA molecules and created algorithms to help them extend some of the simple rules first established by Tuschl's group. Ding and Lawrence [13] expanded these rules to include secondary structure and accessibility prediction in their program called SiRNA. Their basic siRNA design steps include the following:

1. Selection of accessible sites based on the probability profile of the target RNA structure.
2. For such selected accessible sites, siRNAs are chosen based on the requirements of the empirical rules and the favorable (low) binding energy between the antisense siRNA strand and its target sequence.

To try to improve and automate the prediction of effective siRNAs, another group created a database of siRNAs of known efficacy [14]. They used this database to create a scoring scheme (based on energy parameters), which allowed them to evaluate siRNAs.

Not only is the academic community creating better rules and algorithms to design efficient siRNA molecules to be used in RNA interference experiments, but commercial organizations are also using their own internal research studies to develop criteria to help improve potent siRNA selection. For example, one company expanded the basic rules of siRNA design to include avoidance of 5' and 3' UTRs of a target sequence [15]. The siRNA supply company, Dharmacon, Inc., performed a systematic analysis of 180 siRNAs that targeted two mRNAs, and they identified 25 properties associated with siRNA functionality [16]. Eight of these characteristics associated with siRNA functionality can be summarized as follows [17]:

1. Highly functional siRNAs have G/C content between 30 and 52%
2. At least 3 A/U bases at positions 15 to 19
3. Absence of internal repeats
4. The presence of an A base at position 19 of the sense strand
5. The presence of an A base at position 3 of the sense strand
6. The presence of a U base at position 10 of the sense strand
7. A base other than G or C at position 19 of the sense strand
8. A base other than G at position 13 of the sense strand

Each factor contributes to a final score, and the siRNAs with higher scores are predicted to be better than lower-scoring siRNAs. The determination of these characteristics suggested the usefulness of rational design of potent siRNAs for RNA interference experiments.

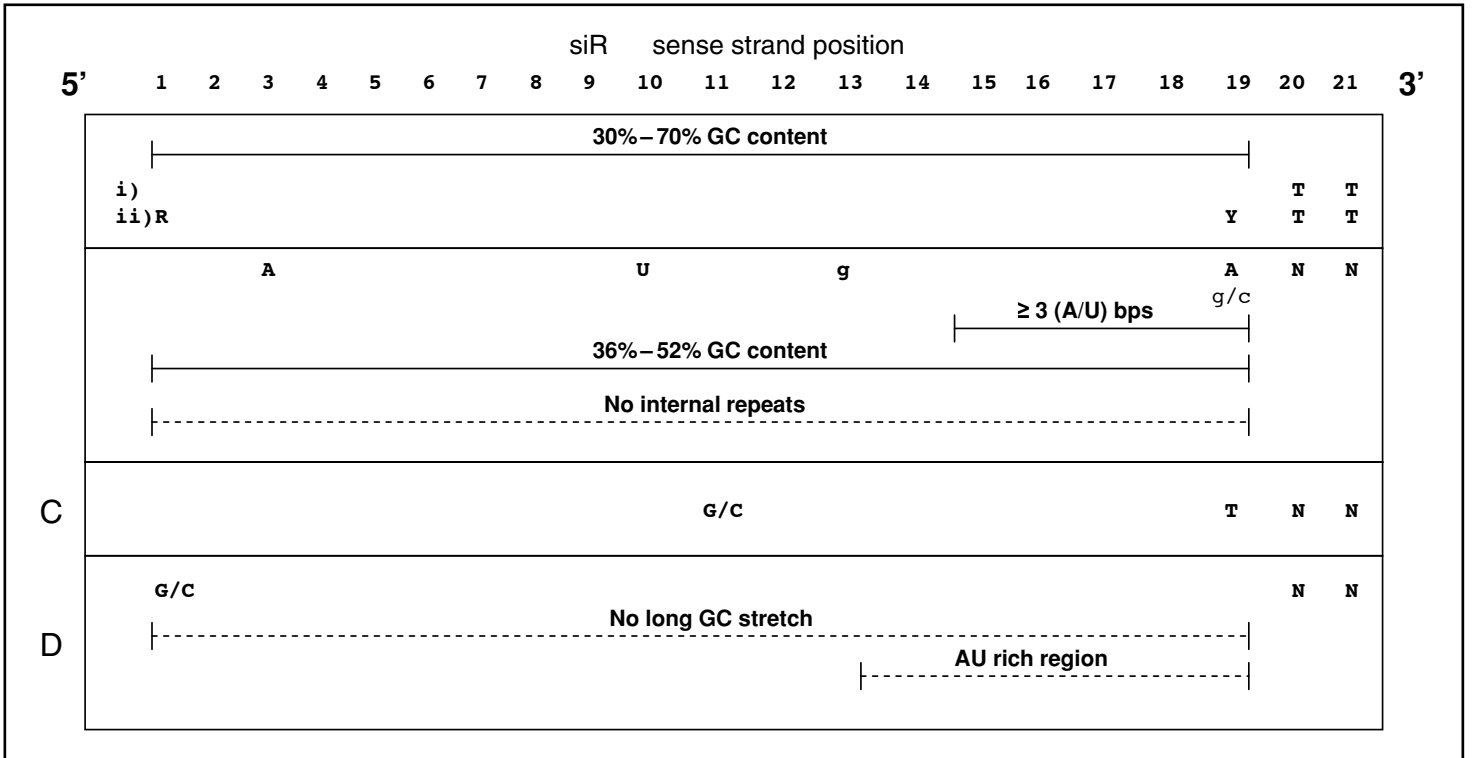


FIGURE 9.2 siRNA design comparison. The base positions of a 19-nt molecule are designated as 5' to 3'. Preferred siRNA positions (based on published findings) are illustrated in sections A [12], B [17], C [10], and D [28]. Uppercase letters are preferred or acceptable bases, and lowercase letters are not acceptable bases at designated positions. In section A, i) and ii) are two siRNA design options, and R and Y denote purine and pyrimidine nucleotides, respectively.

Another commercial group (GenScript) has created a suite of tools for designing vector-based siRNAs and siRNA cassettes [18]. The siRNA target finder tool is used to locate candidate siRNA target sites in mRNA sequences. The design steps can be summarized as follows:

1. A free energy (ΔG1) is calculated to determine the internal stability of the siRNA duplex.
2. BLAST is used to determine the specificity of the siRNA target sequence. To minimize the nonspecific effect, candidates are further analyzed with an additional free energy value (ΔG2) calculation. The final ranking is based on a sum (ΔE) of internal stability (ΔG1) and siRNA specificity (ΔG2). This can be represented with the equation $\Delta E = \Delta G1 + \Delta G2$.
3. siRNA target sites with minimum free energy lower than –5 kcal/mol are filtered.
4. Poly (A) and poly (T) (> 3mers) are removed to avoid premature termination of transcription. Poly (G) and poly (C) (> 2mers) and poly (G, C) (> 6mer) are removed to decrease the RNA duplex internal stability.
5. A check for documented single nucleotide polymorphisms sites is performed.
6. Other adjustable parameters are included in the siRNA determination:
   a. A default of a 21mer target sequence
   b. A default of 35 to 60% GC content
   c. A default open reading frame for the target region
   d. cDNA sequence
   e. Organism type

Although designing a potent siRNA molecule is important for many RNAi experiments, it may also be important to consider the attributes of the target gene sequence.

9.2.2 Selecting siRNA Targets

Everyone agrees that the design of a potent and specific siRNA molecule is very important for a successful silencing experiment; however, there is disagreement about the importance of how the physical aspects of a site on a target mRNA affect siRNA binding. For example, Reynolds et al. [17] reported that functionality is governed not by the local mRNA target but by the specific properties of the siRNA molecule. Conversely, other researchers seem to believe the target mRNA is important and have suggested guidelines for selecting a target site [2]:

- Consider a site 100 nucleotides downstream from the translation start site.
- Avoid regions where secondary structure could develop.
- Avoid regions where mRNA-binding proteins may interfere with siRNA binding.

It is also possible to scan an entire genome for RNAi elements and to determine the presence of cellular genes that are degraded by RNAi elements. Horesh et al.

[3] examined two methods for determining RNAi control using a suffix tree algorithm. They looked at ESTs for evidence that RNAi control elements are expressed, and they used synteny between *C. elegans* and *C. briggsae* to search for similar genes that may be under RNAi control in both organisms. The authors concluded that about 70 genes were under RNAi control.

9.2.3 siRNA and Sequence Similarity Searching

Although creating a stable and potent siRNA molecule is important for a successful RNAi experiment, it is also important to avoid the effects of off-target gene silencing. To identify potential off-target sites, many siRNA design programs incorporate some type of sequence similarity search algorithm in their siRNA selection workflow. For example, a sequence similarity search tool such as Basic Local Alignment Search Tool (BLAST) is incorporated in the siRNA design workflow to find these off-target sequences. This crucial step in siRNA design is typically performed after several siRNA sequences have been identified based on the target sequence. Of course, it is ideal to be able to search the entire genomic sequence of an organism with the candidate target sequence; one can also search against known cDNAs or clustered ESTs and find potential off-target sites.

When using BLAST for siRNA design, the typical default parameters of BLAST (for nucleotides) will not find off-target sequences; therefore, one needs to use special settings for short nucleotide sequences.

Parameter	Normal Default	Suggested Settings
Filter query sequence (-F)	T	F
e-value (-e)	10	1000
word size (-W)	11	7

However, even if one uses these BLAST parameters for short sequences, BLAST many times overlooks deleterious off-target sequence alignments. For instance, a single mismatch between the siRNA and the target RNA can lead to the wrong gene being silenced. Hence, it is imperative that an siRNA sequence is specific to the target mRNA sequence and does not show significant similarity to any other mRNA sequence. Other search applications may be better suited for short oligonucleotides. For example, a tool called *AOsearch* searches for matches and close matches to oligonucleotides (http://sonnhammer.cgb.ki.se/AOsearch/). It allows the submitter to select the number of acceptable mismatches and the desired database (human or mouse) to search. One can also try programs based on the Smith–Waterman algorithm, but they could be slow to generate results. Even though silencing a few off-target sites is inevitable, one can minimize the possibility by using some type of sequence similarity search program.

9.3 DATABASES IN siRNA

There are numerous collection databases that have been derived from primary sequence databases such as GenBank and Swiss-Prot. These special collection databases include

such information as structure motifs, promoter regions, and mutation variants. As more information about siRNA sequences and RNAi experiments becomes available, we will see an increasing number of databases tracking and storing data related to specific siRNA sequences and their corresponding effects on genes (table 9.1). In 2004, there were two searchable worm RNAi databases available on the Internet. One Web site is available at *WormBase* (table 9.1). The Web interface allows you to select a phenotype and search for associated RNAi experiments performed on *C. elegans*. The search returns information such as the gene identifier, the RNAi experiment, and resulting phenotype(s). The other worm database is the *RNAi Database*. It was designed to archive and distribute phenotypic data from large-scale RNAi analyses, and it provides information about experimental methods and phenotypic results [19]. The RNAi Database Web site allows you search the database with query terms such as gene name, gene identifier, and phenotype. A small but potentially useful database is the *siRNA Database* by McManus (table 9.1). This site is a single Web page table, but it contains helpful information such as the gene name, gene identifier (if available), siRNA strand, and reference of selected RNAi experiments. Another small database is available at the *Tronolab siRNA Database*. It contains a table of information (gene name, species, target sequence, and reference) for selected siRNA experiments. Finally, a group at the Whitehead Institute has created a public database called *sirBank* that records sequences known to suppress gene activity [20].

TABLE 9.1
List of siRNA Databases

Database Name	Organization	URL
RNAi Database	New York University	http://nematoda.bio.nyu.edu/
RNAi Phenotype Search	Wormbase	http://www.wormbase.org/db/searches/rnai_search
siRNA Database	MIT	http://web.mit.edu/mmcmanus/www/siRNADB.html
siRNA Database	Protein Lounge	http://www.proteinlounge.com/sirna_home.asp
siRNA DataBase Thermodynamic and Composition Information	Joint Center for Computional Biology and Bioinformatics	http://www.jcbi.ru/EN/sirna/index.shtml
sirBank	Whitehead	http://jura.wi.mit.edu/siRNAext/
Tronolab siRNA Database	Université de Genève	http://www.tronolab.com/sirna_database.php

9.4 siRNA SOFTWARE

9.4.1 Public Tools

Once the rules of siRNA design and target selection became more accepted, *in silico* siRNA prediction came to the fore. Bioinformaticists now use empirical and theoretical design rules to generate siRNA molecules as described earlier. However, because these design approaches use different algorithms, their siRNA design

TABLE 9.2
List of Academic siRNA Design Tools

Tool Name	Organization	URL
OptiRNAi	University of Delaware	http://bioit.dbi.udel.edu/rnai/
RNAit	TrypanoFAN	http://www.trypanofan.org
SIDE	Centro Nacional de Investigaciones Oncológicas	http://side.bioinfo.cnio.es/
siDirect	University of Tokyo	http://design.RNAi.jp/
SiRNA	CPAN	http://search.cpan.org/dist/bioperl/Bio/Tools/SiRNA.pm
siRNA	Center for Computational Research University of Buffalo	http://bioinformatics.ccr.buffalo.edu/cgi-bin/biotool/EMBOSS/emboss.pl?_action=input& _app=sirna&_section=Nucleic
*Si*RNA	Wadsworth Bioinformatics Center	http://www.bioinfo.rpi.edu/applications/sfold/sirna.pl
siRNA Elite	Feng and Zhenbiao	http://www.sirnadesign.com/
siRNA Selection Program	Whitehead Institute for Biomedical Research	http://www.protocol-online.org/prot/Research_Tools/Online_Tools/SiRNA_Design/
siRNA Selector	Wistar Institute	http://hydra1.wistar.upenn.edu/Projects/siRNA/siRNAindex.htm
siSearch-siRNA Design	Center for Genomics and Bioinformatics Karolinska Institutet	http://sonnhammer.cgb.ki.se/siSearch/siSearch_1.5.html
TROD	Université de Genevé	http://websoft2.unige.ch/sciences/biologie/bicel/RNAi.html

programs may return varying siRNA results. For example, one design approach predicts the efficacy of oligonucleotides used in siRNA experiments by using a genetic programming-based machine learning system [21]. Other approaches may exploit published RNAi experiments and collect successful siRNAs in a database and then make use of this information to facilitate the design of effective siRNA molecules.

There are many Web sites and downloadable programs that are freely available to the public (table 9.2). An early Web-based tool, *RNAit*, was an application created for the selection of RNAi targets in *Trypanosoma brucei* [22]. Chalk, Wahlestedt, and Sonhammer [14] developed a software tool that incorporates a set of "Stockholm" rules for siRNA design. The tool is called *siSearch* and allows the user to select from the following list of design rules for selecting siRNAs:

1. %GC content of the siRNA
2. Stockholm rules score
3. Regression tree classification (trained on the Khvorova et al. dataset) [23]
4. Reynolds et al. rules score (without the oligo 6.0 Tm calculation) [17]
5. Ui-tei et al. rules score [24]
6. Amarzguioui and Prydz rules score [25]
7. Special motifs (AA(N19), AA(N19)TT, NAR(N17)YNN, and custom motifs)

8. Disallow certain motifs (AAAA/TTTT, CCC/GGG, and long stretches of consecutive GC)
9. Schwarz et al. energy difference sense and antisense ends [26]

In 2004, there was an explosion of siRNA design tools. *DEQOR* is a program available via the Internet that uses a scoring system based on siRNA design parameters. It predicts gene regions that show high silencing capacity and siRNAs with high silencing potential for chemical synthesis [27]. Another siRNA design software tool called *siDirect* computes siRNA sequences with target-specificity for mammalian RNAi (28). The software avoids off-target gene silencing to reduce potential cross-hybridization sequences. Cui et al. [29] developed a program called *OptiRNAi* that uses the Elbashir et al. criteria to predict target sequences for siRNA design. A Web tool that screens siRNAs for gene specificity is called *siRNA Selector* [30]. This software tool uses rules from several siRNA groups and allows the user to adjust siRNA length, GC content, and so forth. Other tools have been created to help design DNA oligonucleotides with an attached T7 promoter sequence. The *T7 RNAi Oligo Designer* (TROD) aids in the design of oligodeoxynucleotide sequences for the *in vitro* production of siRNA duplexes with T7 RNA polymerase [31].

9.4.2 Commercial Efforts

Although there are a number of public siRNA design tools available on the Internet, many siRNA supply companies offer "free" siRNA design tools for their customers. These companies hope customers visit their Web sites, use their design tools, and generate siRNA molecules that can be ordered directly from their Web site. This Web site approach has been adopted by several companies such as Ambion, Dharmacon, and Invitrogen (see table 9.3). Because commercial Web sites may use different siRNA

TABLE 9.3
List of Commercial siRNA Design Tools

Tool Name	Company	URL
BLOCK-iT RNAi Designer	Invitrogen	https://rnaidesigner.invitrogen.com/sirna/
Complete RNAi	Oligoengine	http://www.oligoengine.com/Tools_Temp/Tools_Main.html
Deqor	Scionics Computer Innovation	http://cluster-1.mpi-cbg.de/Deqor/deqor.html
iRNAwiz	Ocimum Biosolutions	http://www.ocimumbio.com/web/Bioinformatics/prod_details.asp?prod_id=31&prodtype=1
RNAi Design	IDT	http://biotools.idtdna.com/rnai/
siRNA Designer	IRIS Genetics	http://www.irisgenetics.com/Navigation.html
sirna Designer Program	Imgenex	http://www.imgenex.com/sirna_resources.php
siRNA Search	Ambion	http://www.ambion.com/catalog/sirna_search.php
siRNA Target Designer	Promega	http://www.promega.com/siRNADesigner/program/
siRNA Target Finder	GenScript	https://www.genscript.com/ssl-bin/app/rnai

design rules and off-target verification approaches, it may be helpful to visit the sites and determine if they give similar siRNA results.

One bioinformatics company that offers specialized siRNA software tools is called Ocimum Biosolutions. Their software, *iRNAwiz*, provides an environment for the design of successful siRNA molecules and is composed of several components. These components include a *siRNA Search* tool, a *BLAST* tool, a *Motif Search* tool, a *Stemloop* search, and a *Statistical Analysis* tool [32]. They claim the combination of these tools will result in the design of siRNA molecules with high efficiency.

9.5 PRACTICAL APPLICATIONS OF siRNA

9.5.1 Drug-Target Validation

Early on, RNAi tools were created to study gene function in mammalian cells. For example, a vector system was created that directs the synthesis of siRNAs, and it was shown that the expression of these siRNAs caused efficient and specific down-regulation of a gene target [33]. It was reported in 2004 that several major pharmaceutical companies (e.g., Merck and Pfizer) had started to use RNAi technology for their target-validation studies [20]. RNAi is not only employed for *in vitro* target identification and validation, but there is an increasing use of *in vivo* RNAi methodologies for finding potential novel drug targets. One study reported the establishment of an *in vivo* siRNA delivery process that identified, validated, and confirmed potential drug targets [34]. A summary of several siRNA knockdown approaches is seen in table 9.4 [35]. These knockdown methods include using synthetic siRNAs or short hairpin RNAs (shRNAs) expressed in a cell. One common delivery method is to infect cells with viruses encoding shRNAs. Unfortunately, this approach may not achieve sufficient shRNA expression for the gene silencing to be effective. Although RNAi methods have been very useful for drug-target validation, they have their own experimental challenges. For instance, one survey reported that the top reasons RNAi experiments generate unsatisfactory results include insignificant knockdown, off-target effects, and poorly designed oligonucleotides [36].

9.5.2 Functional Genomics

Functional genomics has benefited significantly from the introduction of siRNA techniques and related RNAi approaches. Researchers are able to selectively silence a single gene of interest and determine its function in a cell, or they can examine an entire genome to elucidate how the silencing of selected genes can affect global gene-expression profiles. Several good reviews have been written that summarize how RNAi has been applied as tools for genome-wide screening [37–39]. Genome-wide or multigene RNAi experiments have been performed on several nonmammalian and mammalian organisms, and the silencing technique has been used to screen thousands of genes in a genome. For example, one group used an RNAi-based phenotypic screening approach to identify known and previously uncharacterized genes that affect the apoptosis pathway [40]. Another team was able to apply genome-wide screening of a siRNA expression cassette library that targeted over 8,000 genes

TABLE 9.4
Summary of Various Methods of siRNA Knockdown [35]

siRNA Approach	Delivery Method	Main Advantages	Main Disadvantages
Synthetic siRNA duplex reagents	Lipids or electroporation	Delivery of high siRNA concentrations. Delivery can be monitored. Reagent configuration control and base modifications.	Delivery may be difficult in some primary cells. May need to manage dsRNA responses.
siRNA duplex reagents generated by *in vitro* transcription or DICER	Lipids or electroporation	Delivery of high concentrations. Inexpensive.	As above. May be difficult to control reagent quality.
Transient shRNA expression from pol II promoter constructs	Transient transfection of plasmid vectors or PCR fragments	Rapid generatation of vectors. Inexpensive.	May not achieve enough expression.
shRNA expressed from integrated pol III promoter constructs	Stable transfection of cells with selectable vectors	Selectable permanent integration and expression.	Selection and compensation effects. Clonal variation.
shRNA expressed from pol III promoters in viral vectors	Infection with viral vectors (e.g., adenovirus, lentivirus, retrovirus)	Virus tropism enhances delivery options. Can be stable or transient expression.	May not achieve enough expression. Viral effects.
shRNA expressed from inducible pol III promoter constructs	Stable transfection or infection with lentiviral or retroviral vectors	siRNA levels can be regulated.	May not achieve enough expression.

[41]. The screening of this library uncovered a set of distinct genes involved in the NF-κB signaling pathway. Not only has genome-wide high-throughput screening with siRNAs been exploited in the last couple of years, but there is also an increase in experimental design and performance. For instance, to support the use of RNAi in mammals, Paddison et al. [42] constructed a large-scale library of RNAi-inducing shRNA expression vectors that targeted human and mouse genes. A total of almost 40,000 shRNAs were designed to target nearly 10,000 human and 6,000 mouse genes. They tested almost 7,000 shRNA expression vectors, and nearly half of these affected proteosomal proteins. As more genome sequences become available, more laboratories will report how they are using RNAi methodologies and genome-wide screening to elucidate the functional role of key genes in many types of organisms.

9.5.3 Clinical Therapeutics

The application of RNAi beyond the laboratory is probably the most exciting aspect of this relatively new technology. Using RNAi methods for finding therapeutics will

increase the number of "druggable" genes. Specifically, synthesized siRNAs and siRNA expression systems will help accelerate the use of RNAi techniques in new therapeutic areas. Some examples of human disease targets for RNA interferences include leukemia, carcinomas, malaria, HIV, hepatitis, and influenza [43]. A variety of start-up companies are beginning to focus their attention on using RNAi therapies (e.g., siRNA or shRNA) for such conditions as age-related macular degeneration, Huntington's disease, Alzheimer's disease, obesity, diabetes, Lou Gehrig's disease, and cytomegalovirus [44].

Even though life science entrepreneurs are enthusiastic about the success of this technology for therapeutic applications, there are many important obstacles that need to be overcome before RNAi becomes a realistic tool in a clinical setting. The main limitations of siRNAs as therapeutic agents are related not to RNAi mechanisms but to how to deliver the molecules to the appropriate tissues. Others have noted that the delivery of rationally designed siRNAs must overcome half-life, uptake, longevity, and off-target effects [45]. In addition, understanding how much siRNA therapeutic solution should be delivered into target cells and how to control and maximize the sequence specificity for target genes will need to be addressed [46]. Nonetheless, companies have already started to bring siRNA therapeutics into human clinical trials. For example, in November 2004, a company called Acuity Pharmaceuticals announced that it is the first company to bring siRNA therapeutics into human Phase I clinical trials. They claimed siRNA therapeutics will be used to treat age-related macular degeneration.

9.6 CONCLUSION

The discovery that siRNAs can be used effectively as gene-silencing tools has been a boon for researchers doing target-validation experiments. The use of siRNAs is new relative to other approaches in molecular biology, but we are already seeing the benefits and promises of this simple technique in functional genomics, drug-target identification, and medically relevant therapeutics. The design of siRNAs is key to a successful RNAi experiment, and this is definitively supported by *in silico* tools. Specifically, bioinformatics has stepped in to supply the computational tools to help researchers quickly use the results of the human genome sequence to create potent and specific siRNA molecules and to help them avoid off-target sequences.

ACKNOWLEDGMENTS

I thank Alison Poggi León at Illumina and Pedro Aza-Blanc at Genomics Institute of the Novartis Research Foundation for their useful comments and review of this manuscript.

REFERENCES

1. Jain, K. K. 2004. RNAi and siRNA. *Pharmacogenomics* 5:239–42.
2. Elbashir, S. M., W. Lendeckel, and T. Tuschl. 2001. RNA interference is mediated by 21- and 22-nucleotide RNAs. *Genes Dev* 15:188–200.

3. Horesh, Y., A. Amir, S. Michaeli, and R. Unger. 2003. A rapid method for detection of putative RNAi target genes in genomic data. *Bioinformatics* 19, Suppl. no. 2:73–80.
4. Lindsay, M. A. 2003. Target discovery. *Nat Rev Drug Discov* 2:831–8.
5. Singer, O., A. Yanai, and I. M. Verma. 2004. Silence of the genes. *Proc Natl Acad Sci USA* 101:5313–4.
6. Fire, A., S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello. 1998. Potent and specific genetic interference by double-stranded RNA in Caenorhabditis elegans. *Nature* 391:806–11.
7. Elbashir, S. M., J. Harborth, W. Lendeckel, A. Yalcin, K. Weber, and T. Tuschl. 2001. Duplexes of 21-nucleotide RNAs mediate RNA interference in cultured mammalian cells. *Nature* 411:4948.
8. Gil, J., and M. Esteban. 2000. Induction of apoptosis by the dsRNA-dependent protein kinase (PKR): Mechanism of action. *Apoptosis* 5:107–14.
9. Coty, C. 2003. Mining genomic data to identify function. *Genomics Proteomics* 3:32–5.
10. Hsieh, A. C., R. Bo, J. Manola, F. Vazquez, O. Bare, A, Khvorova, S. Scaringe, and W. Sellers. 2004. A library of siRNA duplexes targeting the phosphoinositide 3-kinase pathway: Determinants of gene silencing for use in cell-based screens. *Nucleic Acids Res* 32:893–901.
11. Schutze, N. 2004. siRNA technology. *Mol Cell Endocrinol* 213:115–9.
12. Elbashir, S. M., J. Martinez, A. Patkaniowska, W. Lendeckel, and T. Tuschl. 2001. Functional anatomy of siRNAs for mediating efficient RNAi in Drosophila melanogaster embryolysate. *Eur Mol Biol Org J* 20:6877–88.
13. Ding, Y., and C. Lawrence. 2005. Rational design of siRNAs with the Sfold software. In *RNA interference technology: From basic science to drug development,* ed. K. Appasani, 129–138. Cambridge, UK: Cambridge Univ. Press.
14. Chalk, A. M., C. Wahlestedt, and E. L. L. Sonhammer. 2004. Improved and automated prediction of effective siRNA. *Biochem Biophys Res Comm* 319:264–74.
15. Poncelet, D. 2002. RNAi: The review. White paper. Eurogentec. www.eurogentec.com. Seraing, Belgium.
16. Toner, B. 2004. The tools may be free, but rivalry is brewing among RNA interference software providers. *Bioinform* Q1:4–5.
17. Reynolds, A., D. Leake, Q. Boese, S. Scaringe, W. S. Marshall, and A. Khvorova. 2004. Rational siRNA design for RNA interference. *Nat Biotechol* 22:326–30.
18. Wang, L., and F. Y. Mu. 2004. A Web-based design center for vector-based siRNA and siRNA cassette. *Bioinformatics* 20:1818–20.
19. Gunsalus, K. C., W-C. Yueh, P. MacMenamin, and F. Piano. 2004. RNAiDB and PhenoBlast: Web tools for genome-wide phenotypic mapping projects. *Nucleic Acids Res* 32:D406–10.
20. Burke, A. J. 2003. Short interfering sensation. *Genome Technol* 33:36–44.
21. Saestrom, P. 2004. Predicting the efficacy of short oligonucleotides in antisense and RNAi experiments with boosted genetic programming. *Bioinformatics* 20(17):3055–63.
22. Redmond, S., J. Vadivelu, and M. C. Field. 2003. RNAit: An automated Web-based tool for the selection of RNAi targets in Trypanosoma brucei. *Mol Biochem Parasitol* 128:115–8.
23. Khvorova, A., A. Reynolds, and S. D. Jayasena. 2003. Functional siRNAs and miRNAs exhibit strand bias. *Cell* 115:209–16.
24. Ui-Tei, K., Y. Naito, F. Takahashi, T. Haraguchi, H. Ohki-Hamazaki, A. Juni, R. Ueda, and K. Saigo. 2004. Guidelines for the selection of highly effective siRNA sequences for mammalian and chick RNA interference. *Nucleic Acids Res* 32:936–48.

25. Amarzguioui, M., and H. Prydz. 2004. An algorithm for selection of functional siRNA sequences. *Biochem Biophys Res Commun* 316:1050–8.
26. Schwarz, D. S., G. Hutvagner, T. Du, Z. Xu, N. Aronin, and P. D. Zamore. 2003. Asymmetry in the assembly of the RNAi enzyme complex. *Cell* 115:199–208.
27. Henschel, A., F. Buchholz, and B. Habermann. 2004. DEQOR: A web-based tool for the design and quality control of siRNAs. *Nucleic Acids Res* 32:W113–20.
28. Naito, Y., T. Yamada, K. Ui-Tei, S. Morishita, K. Saigo. 2004. siDirect: Highly effective, target-specific siRNA design software for mammalian RNA interference. *Nucleic Acids Res* 32:W124–9.
29. Cui, W., J. Ning, U. P. Naik, and M. K. Duncan. 2004. OptiRNAi, and RNAi design tool. *Comput Methods Programs Biomed* 75:67–73.
30. Levenkova, N., Q. Gu, and J. J. Rux. 2004. Gene specific siRNA selector. *Bioinformatics* 20:430–2.
31. Dudek, P., and D. Picard. 2004. TROD: T7 RNAi Oligo Designer. *Nucleic Acids Res* 32:W121–3.
32. *UIRNAwiz*. 2004. White paper. Ocimum Biosolutions. Indianapolis, Indiana.
33. Brummelkamp, T., R. Bernards, and R. Agami. 2002. A system for stable expression of short interferring RNAs in mammalian cells. *Science* 296:550–2.
34. Xie, F., Y. Liu, J. Xu, Q. Q. Tang, P. V. Scaria, Q. Zhou, M. C. Woodle, and P. Y. Lu. 2004. Delivering siRNA to animal disease models for validation of novel drug targets in vivo. *PharmaGenomics* 4:32–8.
35. Cocks, B. G., and T. P. Theriault. 2003. Developments in effective application of small inhibitory RNA (siRNA) technology in mammalian cells. *Drug Discov Today: Targets* 3:165–71.
36. RNAi roundup. 2004. *Genome Technol* 46:37.
37. Dorsett, Y., and T. Tuschl. 2004. siRNAs: Applications in functional genomics and potential as therapeutics. *Nat Rev Drug Discov* 3:318–29.
38. Hannon, G. J., and J. J. Rossi. 2004. Unlocking the potential of the human genome with RNA interference. *Nature* 431:371–8.
39. Friedman, A., and N. Perrimon. 2004. Genome-wide high-throughput screens in functional genomics. *Curr Opin Genet Dev* 14:470–6.
40. Aza-Blanc, P., C. L. Cooper, K. Wagner, S. Batalov, Q. L. Deveraux, and M. P. Cooke. 2003. Identification of modulators of TRAIL-induced apoptosis via RNAi-based phenotypic screening. *Mol Cell* 12:627–37.
41. Zheng, L., J. Liu, S. Batalov, D. Zhou, A. Orth, S. Ding, and P. Schultz. 2004. An approach to genomewide screens of expressed small interfering RNAs in mammalian cells. *Proc Natl Acad Sci USA* 101:135–40.
42. Paddison, P. J., J. M. Silva, D. S. Conklin, M. Schlabach, M. Li, S. Aruleba, V. Balija, et al. 2004. A resource for large-scale RNA-interference-based screens in mammals. *Nature* 428:427–31.
43. Kumar, R., D. S. Conklin, and V. Mittal. 2003. High-throughput selection of effective RNAi probes for gene silencing. *Genome Res* 13:2333–40.
44. Marx, V. 2004. Silence could be golden. *Chem Engin News* 82:18–23.
45. Karpilow, J., D. Leake, and B. Marshall. 2004. siRNA: Enhanced functionality through rational design and chemical modification. *PharmaGenomics* March/April: 32–40.
46. Kim, N. V. 2003. RNA interference in functional genomics and medicine. *J Korean Med Sci* 18:309–18.

10 Predicting Protein Subcellular Localization Using Intelligent Systems

Rajesh Nair and Burkhard Rost
Columbia University

CONTENTS

10.1 INTRODUCTION

10.1.1 Decoding Protein Function: A Major Challenge for Modern Biology

The genetic information for life is stored in the nucleic acids, while proteins are the workhorses that are responsible for transforming this information into physical reality. Proteins are the macromolecules that perform most important tasks in organisms, such as the catalysis of biochemical reactions, transport of nutrients, and recognition and transmission of signals. The plethora of aspects of the role of any particular protein is referred to as its *function*. The genome (DNA) sequences of over 180 organisms, including a draft sequence of the human genome [1,2], has now been completed. For over 105 of these, these data are publicly available and contribute about 413,000 protein sequences, that is, about one-fourth of all currently known protein sequences [3–5]. The number of entirely sequenced genomes is expected to continue growing exponentially for at least the next few years. With the availability of genome sequences of entire organisms, we are for the first time in a position to understand the expression, function, and regulation of the entire set of proteins encoded by an organism. This information will be invaluable for understanding how complex biological processes occur at a molecular level, how they differ in various cell types, and how they are altered in disease states [6]. Identifying protein function is a big step toward understanding diseases and identifying novel drug targets [7]. However, experimentally determining protein function continues to be a laborious task requiring enormous resources. For example, more than a decade after its discovery, we still do not know the precise and entire functional role of the prion protein [8]. The rate at which expert annotators add experimental information into more or less controlled vocabularies of databases snails along at an even slower pace. This has left a huge and rapidly widening gap between the amount of sequences deposited in databases and the experimental characterization of the corresponding proteins [9,10]. Bioinformatics plays a central role in bridging this sequence-function gap through the development of tools for faster and more effective prediction of protein function [11–13].

10.1.1.1 Protein Function Has Myriad Meanings

The function of a protein is hard to define. Proteins can perform molecular functions like catalyzing metabolic reactions and transmitting signals to other proteins or to DNA. At the same time they can also be responsible for performing physiological functions as a set of cooperating proteins, such as the regulation of gene expression, metabolic pathways, and signaling cascades [11]. What makes matters worse,

although many biologists may assume that they "know it when they see it," in fact, their conclusion is likely to be biased by the department with which they are affiliated; for example, geneticists attach a different meaning to the word *function* than do chemists; pharmacologists; or medical, structural, or cell biologists. This Babylonian confusion comes about since function is a complex phenomenon that is associated with many mutually overlapping levels: chemical, biochemical, cellular, organism mediated, developmental, and physiological [14]. These levels are related in complex ways; for example, protein kinases can be related to different cellular functions (such as cell cycle) and to a chemical function (transferase) plus a complex control mechanism by interaction with other proteins. The same kinase may also be the culprit that leads to misfunction, or disease. The variety of functional roles of a protein often results in confusing database annotations, which makes it difficult to develop tools for predicting protein function [15]. Computer-readable hierarchical descriptions of function are needed for reliable automatic predictions [11,16,17]. But defining an ontology for protein function has proved to be an extremely difficult task.

10.1.1.2 What Makes Subcellular Localization Ideal for Function Prediction Experiments?

Since biological cells are subdivided into membrane-bound compartments, the subcellular localization of a protein is much more easily identifiable than its other roles in a cell. In contrast with other functional features, the protein-trafficking mechanism is relatively well understood, and computer-readable subcellular localization data are available for large numbers of proteins. Proteins must be localized in the same subcellular compartment to cooperate toward a common physiological function. Though some proteins can localize in multiple compartments, the majority of proteins are localized within a single compartment for the largest part of their lifetime. Knowledge of the subcellular localization of a protein can significantly improve target identification during the drug-discovery process [18,19]. For example, secreted proteins and plasma membrane proteins are easily accessible by drug molecules because of their localization in the extracellular space or on the cell surface. A purified secreted protein or a receptor extracellular domain can be utilized directly as a therapeutic (e.g., growth hormone) or may be targeted by specific antibodies or small molecules. Important therapeutics have been created that target proteins present on the cell surface in a specific cell type or disease state [20]. Rituxan is an antibody therapeutic targeting the B lymphocyte-specific CD20 protein and is an effective therapeutic in the treatment of non-Hodgkin's lymphoma. Aberrant subcellular localization of proteins has been observed in the cells of several diseases, such as cancer and Alzheimer's disease. Therefore, unraveling the native compartment of a protein is an important step on the long way to determining its role [11,21]. Using experimental high-throughput methods for epitope and green fusion protein tagging, two groups have recently reported localization data for most proteins in *Saccharomyces cerevisiae* (baker's yeast) [22,23]. So far, the majority of large-scale experiments suggesting localization have been restricted to yeast, or to particular compartments, such as a recent analysis of chloroplast proteins in *Arabidopsis thaliana* (grass) [24]. As of now, these large-scale experiments cannot be repeated for mammalian or other

higher eukaryotic proteomes. One significant obstacle is that large-scale production of a collection of cell lines each with a defined gene chromosomally tagged at the 3'-end is not yet possible [25]. In contrast, computational tools can provide fast and accurate localization predictions for any organism [9,26]. As a result, subcellular localization prediction is becoming one of the central challenges in bioinformatics [27–29].

10.1.1.3 Protein Trafficking Proceeds via Sorting Signals

Bacterial cells generally consist of a single intracellular compartment surrounded by a plasma membrane. In contrast, eukaryotic cells are elaborately subdivided into functionally distinct, membrane-bounded compartments. The major constituents of eukaryotic cells are extracellular space, cytoplasm, nucleus, mitochondria, Golgi apparatus, endoplasmic reticulum (ER), peroxisome, vacuoles, cytoskeleton, nucleoplasm, nucleolus, nuclear matrix, and ribosomes [30]. Most eukaryotic proteins are encoded in the nuclear genome and synthesized in the cytosol, and many need to be further sorted before they reach their final destinations (fig. 10.1). The localization of a protein is largely determined by a trafficking system that is reasonably well understood for some organelles [28,31–34]. The system has two main branches [35]. On one, proteins are synthesized on cytoplasmic ribosomes and from there can go to the nucleus, mitochondria, or peroxisomes. The second branch leads from the ER-ribosomes to the Golgi apparatus and from there to lysosomes, or secretory vesicles, and on to the extracellular space. At each branch point, a "decision" must be made for each protein—either retain the protein in the current compartment or transport it to the next. These decisions are made by membrane transport complexes, which respond to targeting signals on the proteins themselves. In most cases, these targeting signals are short stretches of amino acid residues. The best understood branch point is the second one leading to secretion. Many proteins destined for this branch have an N-terminal signal peptide, which is cleaved off proteolytically either during or after protein translocation through the membrane. Proteins lacking this signal are retained in the cytoplasm. The targeting signals used at the other branch points are not always so clear for two reasons. First, the signals are presented by folded proteins and hence are not always contiguous in sequence. Second, even where the signals are contiguous in sequence, not all signal peptides have been documented. In the absence of a clear understanding of the principles governing protein translocation, computational methods for predicting subcellular localization have pursued a number of conceptually distinct approaches.

10.2 *IN SILICO* APPROACHES TO PREDICTING SUBCELLULAR LOCALIZATION

10.2.1 No Straightforward Strategy for Predicting Localization

Methods for predicting the subcellular localization of proteins have primarily explored four avenues: (a) annotation transfer from homologous sequences, (b) predicting the

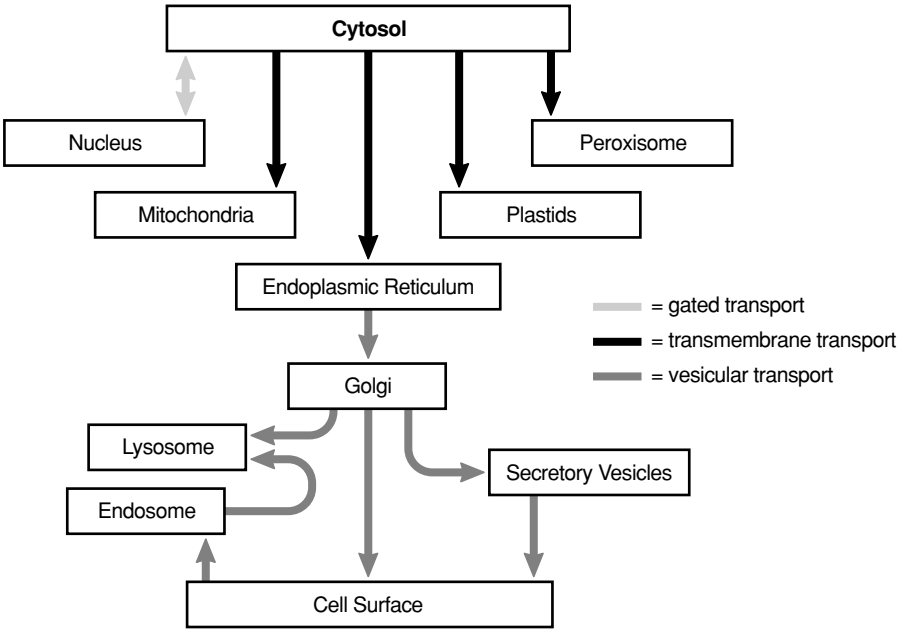


FIGURE 10.1 A simplified "roadmap" of protein traffic. Proteins can move from one compartment to another by gated transport *(white),* transmembrane transport *(dark gray),* or vesicular transport *(light gray)*. The signals that direct a given protein's movement through the system, and thereby determine its eventual location in the cell, are contained in each protein's amino acid sequence. The journey begins with the synthesis of a protein on a ribosome in the cytosol and terminates when the final destination is reached. At each intermediate station *(boxes),* a decision is made as to whether the protein is to be retained in that compartment or transported further. In principle, a signal could be required for either retention in or exit from a compartment. Proteins are synthesized in the cytosol from where they are sorted to their respective localizations. (From Alberts et al. [126]; used with permission.)

sorting signals that the cell uses as "address labels," (c) mining the functional information deposited in databases and scientific literature, and (d) using the observation that the subcellular localization depends in subtle ways on the amino acid composition (table 10.1). In addition, there are metamethods, which combine the outputs from a number of primary methods in an optimal way to enhance accuracy and coverage. Sequence similarity is perhaps the most frequently used method to annotate function for unknown proteins and accounts for the majority of annotations about function in public databases [26,36,37]. A major limitation of sequence-homology-based methods is that they are only applicable when another sequence-similar protein with experimentally known function is available. Hence, only a small fraction of known sequences can be annotated using this approach [27]. Since protein trafficking relies on the presence of sorting signals, ideally we would like to predict the signals responsible for targeting. However, our current knowledge of sorting signals is far from perfect, and recent cell biological studies seem to indicate that the protein-sorting mechanism is far more complex than previously thought. This makes it extremely difficult to accurately identify sorting signals [38]. In spite of

TABLE 10.1
Services for Subcellular Localization Prediction

Method	URL
	Sequence Homology-Based Localization Annotations
LOChom [50]	cubic.bioc.columbia.edu/db/LOChom/
	Methods Based on N-Terminal Sorting Signals
SignalP [127]	www.cbs.dtu.dk/services/SignalP/
ChloroP [73]	www.cbs.dtu.dk/services/ChloroP/
TargetP [68]	www.cbs.dtu.dk/services/TargetP/
iPSORT [125]	biocaml.org/ipsort/iPSORT/
MitoProt [128]	www.mips.biochem.mpg.de/cgibin/proj/medgen/mitofilter/
Predotar [129]	www.inra.fr/Internet/Produits/Predotar/
	Prediction and Analysis of Nuclear Localization Signals
PredictNLS [62]	cubic.bioc.columbia.edu/predictNLS/
	Inferring Localization Using Text Analysis
LOCkey [92]	cubic.bioc.columbia.edu/services/LOCkey/
Proteome Analyst	www.cs.ualberta.ca/~bioinfo/PA/
GeneQuiz [44]	jura.ebi.ac.uk:8765/ext-genequiz/
Meta_A [94]	mendel.imp.univie.ac.at/CELL_LOC/
	Methods Based on Amino Acid Composition
LOCnet [116]	cubic.bioc.columbia.edu/services/LOCnet/
SubLoc [109]	www.bioinfo.tsinghua.edu.cn/SubLoc/
PLOC [111]	www.genome.jp/SIT/ploc.html
ProtComp	www.softberry.ru/berry.phtml?topic=index&group=programs&subgroup=proloc
	General Methods
PSORT II [61]	psort.nibb.ac.jp/
PSORT-B [54]	www.psort.org/psortb/
LOCtarget [119]	cubic.bioc.columbia.edu/services/LOCtarget/

their limited applicability, methods that predict sorting signals provide the most useful predictions, since by pinpointing the "targeting signal," they shed light on the molecular mechanisms of protein translocation. Traditionally, expert human annotators have been responsible for interpreting experimental data in the scientific literature and annotating protein function in public databases [39,40]. However, recent advances in data-mining techniques have made it possible to deploy automatic methods to complement the role of "expert annotators" and extract functional information directly from biological databases, MEDLINE abstracts [41], and even full scientific papers. Due to the exponential growth in the size of biological databases, a number of methods have recently been developed that infer subcellular localization using automatic text analysis. Many recent advances in predicting subcellular localization have been the result of using the amino acid composition and other sequence-derived features. These *ab initio* methods utilize only the amino acid composition and features predicted from the primary sequence, hence they have the advantage of being applicable to all protein sequences. A method for accurately predicting subcellular localization from the amino acid sequence alone would be invaluable in

interpreting the wealth of data provided by large-scale sequencing projects. Furthermore, predictions of localization can assist high-throughput techniques to determine localization from cDNAs [42]. However, prediction accuracy for *ab initio* methods still lags behind other approaches.

Next, we review the different approaches for predicting subcellular localization and describe the state-of-the-art methods for predicting localization.

10.3 INFERRING LOCALIZATION THROUGH SEQUENCE HOMOLOGY

10.3.1 Most Annotations of Function through Homology Transfer

Traditionally, the first approach for annotating function of an unknown protein relies on sequence similarity to proteins of known function [43,44]. The method works by first identifying a database protein of experimentally known function with significant sequence similarity to a query protein, U, and then transferring the experimental annotations of function from the homologue to the unknown query U. Understanding the relation between function and sequence is of fundamental importance, since it provides insights into the underlying mechanisms of evolving new functions through changes in sequence and structure [45]. Several studies have explored the relationship of sequence and structure similarity to conservation of various aspects of protein function [46–49]. One major observation is the existence of sharp "conservation thresholds" for sequence similarity: above the threshold, sequence-similar pairs of proteins share the same function, and below it, they have dissimilar functions. In practice, ad hoc thresholds of 50 to 60% sequence identity are often used for transferring functional annotations. Recent studies indicate that these levels of sequence similarity may not be sufficient to accurately infer function [48,50]. Several pitfalls in transferring annotations of function have been reported, for example, inadequate knowledge of thresholds for "significant sequence similarity," using only the best database hit, or ignoring the domain organization of proteins [9,36,51,52]. In spite of this, homology-based approaches continue to be among the most reliable for annotating subcellular localization [50,53,54].

10.3.2 LOChom: Database of Homology-Based Annotations

By performing a large-scale analysis of the relationship between sequence similarity and subcellular localization, Nair and Rost [50] were able to establish sequence-similarity thresholds for the conservation of subcellular localization. They observed a sharp transition separating the regions of conserved and nonconserved localization, although this transition was less well defined than those previously observed for the conservation of protein structure and enzymatic activity [48]. To their surprise, they found that pairwise sequence identities of over 80% were needed to safely infer localization based on homology. A simple measure for sequence similarity accounting

for pairwise sequence identity and alignment length, the HSSP-distance [55,56], was found to accurately distinguish between protein pairs of identical and different localizations. In fact, BLAST expectation values [57,58] outperformed the HSSP-distance only for sequence alignments in the subtwilight zone, which is the region of sequence similarity where structure and function can no longer be safely inferred from sequence similarity alone. LOChom [50] is a comprehensive database containing homology-based subcellular localization annotations for nearly a quarter of all proteins in the Swiss-Prot database [59] and around 20% of sequences from five entirely sequenced eukaryotic genomes [50].

10.4 PREDICTING SEQUENCE MOTIFS INVOLVED IN PROTEIN TARGETING

10.4.1 Prediction Possible for Some Cellular Classes

A number of methods have tried to predict localization by identifying local sequence motifs, such as signal peptides [60,61] or nuclear localization signals (NLS)[28,62] that are responsible for protein targeting. The prediction of N-terminal sorting signals has a long history originating from the early work on secretory signal peptides of von Heijne [63,64]. N-terminal signal peptides are responsible for the transport of proteins between the ER and the Golgi apparatus and also for targeting proteins to the mitochondria [65] and to chloroplasts [66]. Early methods for predicting signal peptides were essentially based on consensus signals, using linear discriminant functions with weight matrices. Modern machine-learning techniques can predict whether a protein contains an N-terminal targeting peptide by automatically extracting correlations from the sequence data without any prior knowledge of targeting signals, which makes it impossible to gain any idea about the protein-sorting mechanism by looking at the output from these predictors. The introduction of machine-learning techniques like neural networks (NNs) and hidden Markov models (HMMs) [67,68] has resulted in spectacular improvements in prediction accuracy. Machine-learning methods like NNs and HMMs learn to discriminate automatically from the data, using only a set of experimentally verified examples as input. It is now possible to predict secretory signal peptides (SPs) [69,70], mitochondrial targeting peptides (mTPs) [71,72], and chloroplast targeting peptides (cTPs) [73] quite reliably using machine-learning techniques. A particular problem for methods detecting N-terminal signals is that start codons are predicted with less than 70% accuracy by genome projects [2,74,75]. For additional details, the reader can consult a number of excellent reviews on N-terminal sorting signal prediction [67,76,77]. Sorting signals also mediate the import of proteins into the nucleus. A protein is imported into the nucleus if it contains an NLS, which is a short stretch of amino acids. Extensive experimental research on nucleo-cytoplasmic transport [31] indicates that NLSs can occur anywhere in the amino acid sequence and in general have an abundance of positively charged residues [78,79]. Efforts at NLS prediction started with the work of Cokol, Nair, and Rost [62], who successfully applied "*in silico* mutagenesis" to discover new NLSs. Since the entire protein sequence must be searched for NLSs, application of machine-learning techniques

has proved difficult. Overall, known and predicted sequence motifs enable annotating about 30% of the proteins in six eukaryotic proteomes [3,80]. Here, we review TargetP and PredictNLS, which are the most accurate tools for predicting signal peptides and nuclear localization signals.

10.4.2 TargetP: Predicting N-Terminal Signal Peptides

TargetP is a neural-network-based tool for predicting N-terminal sorting signals. The neural network can discriminate between proteins destined for the secretory pathway, mitochondria, chloroplast, and other localizations with an accuracy of 85% (plant) or 90% (nonplant). The N-terminal signal peptide is proteolytically cleaved either during or after protein translocation. TargetP predicts the cleavage site, though cleavage site prediction accuracy is lower, with 40% to 50% sites correctly predicted for chloroplastic and mitochondrial presequences and above 70% for secretory signal peptides. The neural network architecture consists of two layers. The first layer contains one dedicated network for each type of presequence (SP, mTP, cTP, Other), while the second is a "decision neural network" that makes the final choice between the different compartments (fig. 10.2). The signal peptide problem was posed to the neural networks in two ways: (a) recognition of the cleavage sites against the background of all other sequence positions, and (b) classification of amino acids as

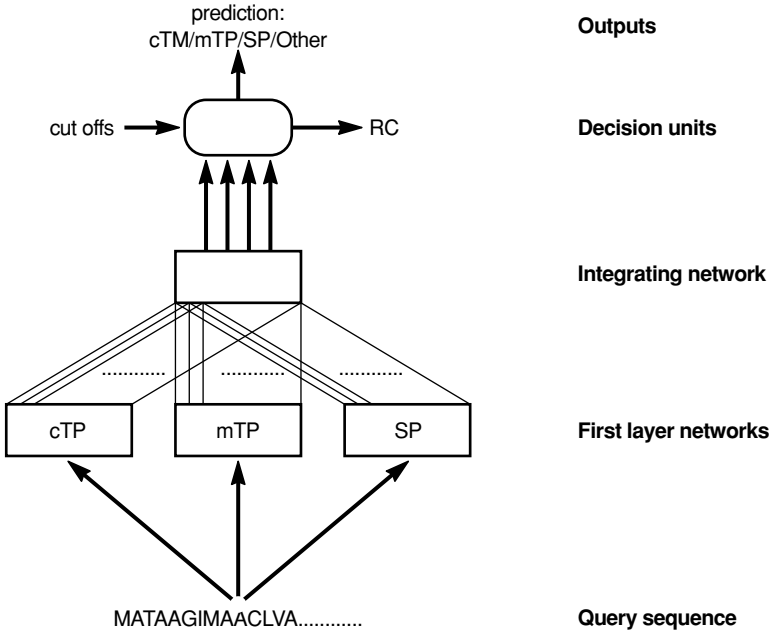


FIGURE 10.2 TargetP localization predictor architecture. TargetP is built from two layers of feed-forward neural networks and, on top, a decision-making unit, taking into account cutoff restrictions (if opted for) and outputting a prediction and a reliability class, RC, which is an indication of prediction certainty (see the text). The nonplant version lacks the cTP network unit in the first layer and does not have cTP as a prediction possibility.

belonging to the signal peptide or not. Sequence data were presented to the neural networks using a sliding-window technique: a window of residues is presented to the neural network, and the network is trained to predict the state of the central residue. The sliding-window approach is remarkably successful at capturing sequence features correlated over long stretches of residues [81]. The window is then moved along the amino acid sequence, and predictions are made in turn for each successive residue. Window sizes ranged from 27 residues for the SP networks to 56 residues for the cTP networks. A dataset consisting of 269 SP, 368 mTP, and 141 cTP sequences (for the plant version of TargetP), and 715 SP and 371 mTP sequences (for the nonplant version) was used to train pairwise feed-forward neural networks to accurately identify each type of targeting presequence. The scores for the 100 N-terminal residues were then fed to the second layer integrating network, which determines the type of N-terminal targeting peptide. From a TargetP analysis of *Arabidopsis Thaliana* and *Homo Sapiens*, 10% of all plant proteins were estimated to be mitochondrial and 14% chloroplastic, and the abundance of secretory proteins in both *Arabidopsis* and *Homo* was estimated to be 10%.

10.4.3 PredictNLS: Predicting Nuclear Localization Signals

Over the last few years a large number of distinct NLSs have been experimentally implicated in nuclear transport [31,78]. NLSdb [82] is the largest publicly available database of experimental NLSs. However, known experimental NLSs can account for fewer than 10% of known nuclear proteins. To remedy this, PredictNLS [62] uses a procedure of *in silico* mutagenesis to discover new NLSs. Briefly, this procedure works as follows:

1. Change or remove some residues from the experimentally characterized NLS motifs and monitor the resulting true (nuclear) and false (nonnuclear) matches. Obviously, allowing alternative residues at particular positions increased the number of nuclear proteins found. However, often this also increased the number of matching nonnuclear proteins.
2. Discard any potential NLSs that are found in known nonnuclear proteins (false matches).
3. Require that potential NLSs be found in at least two distinct nuclear protein families. The 194 potential NLSs discovered using this procedure increased the coverage of known nuclear proteins to 43%. All proteins in the PDB [83] and Swiss-Prot databases were annotated using the full list of experimental and potential NLSs. NLSdb contains over 6,000 predicted nuclear proteins and their targeting signals from the PDB and Swiss-Prot databases. The database also contains over 12,500 predicted nuclear proteins from six entirely sequenced eukaryotic proteomes (*Homo sapiens, Mus musculus, Drosophila melanogaster, Caenorhabditis elegans*, *Arabidopsis thaliana,* and *Saccharomyces cerevisiae*). Approximately 20% of the NLS motifs were observed to co-localize with experimentally determined DNA-binding region of proteins [62,84]. This observation was also

used to annotate over 1,500 DNA-binding proteins. We also annotated all sequences in the yeast, worm, fruit fly, and human proteomes.

10.5 AUTOMATIC LEXICAL ANALYSIS OF CONTROLLED VOCABULARIES

10.5.1 Mining Databases to Annotate Localization

Automatic text analysis methods can be classified into two broad categories: extracting information directly from scientific literature and inferring function from controlled vocabularies in protein databases. New experimental discoveries are first published in scientific journals. Mining scientific literature to automatically retrieve information is an appealing goal, and a number of groups have worked on different aspects of this problem: machine selection of articles of interest [85], automated extraction of information using statistical methods [86,87], and natural language processing techniques for extracting pathway information [88,89]. However, usefulness of this class of methods for annotating protein function is hampered by a crucial bottleneck: the mapping of gene/protein names [37,90]. To date no attempts have been made to directly annotate subcellular localization from scientific publications. The second class of methods has proved more successful for annotating function. Functional annotations in protein databases are written mostly in plain text using a rich biological vocabulary that often varies in different areas of research, which makes it difficult to parse using computer programs. In addition, databases like Swiss-Prot usually contain functional annotations at a very detailed level of biochemical function, for example, a given sequence is annotated as a cdc2 kinase but not as being involved in intracellular communication [91]. A number of text-analysis tools have been implemented that infer various aspects of cellular function from database annotations of molecular function. Many methods explore the functional annotations in SWISS-PROT, especially the keyword annotations [12,44,92–94]. Swiss-Prot currently contains over 800 keyword functional descriptors. Semantic analysis of the keywords is used to categorize proteins into classes of cellular function [95,96]. Both fully automated and semiautomated methods have been applied to predicting subcellular localization. The fully automatic methods extract rules from keywords by using statistical learning methods like, probabilistic Bayesian models [97], symbolic rule learning [98], and M-ary (multiple category) classifiers like the k-Nearest Neighbour [99]. Some of the major methods in this category are LOCkey [92], Proteome Analyst [93], Spearmint [100,101], and the SVM-based approach of Stapley et al. [102]. The semiautomated methods are based on building dictionaries of rules. Keywords characteristic of each of the functional classes are first extracted from a set of classified example proteins. Using these keywords, a library of rules is created associating a certain pattern of occurrence of keywords to a functional class. The major methods in this category are EUCLID [44], Meta_A [94], and RuleBase [12]. Function annotations from RuleBase and Spearmint have been integrated into UniProt [50], which is the world's most comprehensive catalog of information on proteins. Next we review the LOCkey algorithm for predicting subcellular localization.

10.5.2 LOCkey: Information–Theory-Based Classifier

The LOCkey system [92] is a novel M-ary classifier that predicts the subcellular localization of a protein based on Swiss-Prot keywords. The LOCkey algorithm can be divided into two steps (fig. 10.3): building datasets of trusted vectors for known proteins and classifying unknown proteins. First, a list of keywords is extracted from Swiss-Prot for all proteins with known subcellular localization. On average most proteins have between two and five keywords. A dataset of binary vectors [103] is generated for each protein by representing the presence of a certain keyword in the protein by 1 and its absence by 0. Second, to infer subcellular localization of an unknown protein U, all keywords for U are read from SWISS-PROT. These keywords are translated into a binary keyword vector. From this original keyword vector, LOCkey generates a set of all possible combinations of alternative vectors by flipping vector components of value 1 (presence of keyword) to 0 in all possible combinations. For example, for a protein with three keywords, there are $2^3 - 1 = 7$ possible subvectors: 111, 110, 101, 011, 100, 010, and 001. These subvectors constitute all possible keyword combinations for protein U. The keyword combination (i.e., subvector) that yields the best classification of U into one of 10 classes of subcellular

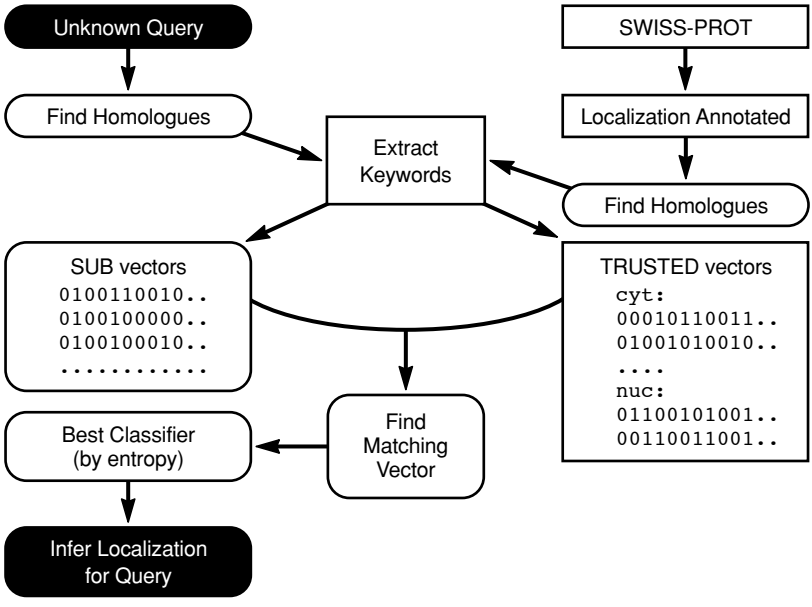


FIGURE 10.3 The LOCkey algorithm. A sequence unique data set of localization annotated SWISS-PROT proteins was first compiled. Keywords were extracted for these proteins and merged with any keywords found in homologues. The keywords were represented as binary vectors in the Trusted Vector Set. An unknown query was first annotated with keywords through identification of SWISS-PROT homologues. Keywords for the query were represented as binary vectors. All possible keyword combinations were constructed (the SUB vectors). The best matching vector was found based on entropy criteria (see Methods). This vector was used to infer localization for the query.

localizations is then found. This is done by retrieving all exact matches of each of the subvectors to any of the proteins in the trusted set, that is, by finding all proteins in the trusted set that contain all the keywords present in the subvector. By construction, the proteins retrieved in this way may also contain keywords not found in U.

The next task is to estimate the "surprise value" of the given assignment. Toward this end, LOCkey simply compiles the number of proteins belonging to each type of subcellular localization. This procedure is repeated in turn for each of the subvectors, and localization is finally assigned to a protein by minimizing an entropy-based objective function. The system accurately solves the classification problem when the number of data points (proteins) and dimensionality of the feature space (number of keywords) are not too large. LOCkey reached a level of more than 82% accuracy in a full cross-validation test. However, due to a lack of functional annotations, the coverage was low and the system failed to infer localization for more than half of all proteins in the test set. For five entirely sequenced proteomes, namely yeast, worm, fly, plant (*Arabidopsis thaliana*), and a subset of all human proteins, the LOCkey system automatically found about 8,000 new annotations about sucellular localization. LOCkey has been optimized to provide fast annotations. Anotating the entire worm proteome took less than four hours on a PIII 900 MHz machine. The algorithm is limited to problems with relatively few data points (proteins) in the vector set ($n < 1{,}000{,}000$) and with few keywords ($n < 10{,}000$).

10.6 *AB INITIO* PREDICTION FROM SEQUENCE

10.6.1 *Ab Initio* Methods Predict Localization for All Proteins at Lower Accuracy

The breakthrough for *ab initio* prediction came from the pioneering works of Nishikawa and colleagues (Nishikawa and Ooi [104]; Nakashima and Nishikawa [105]). They observed that the total amino acid composition of a protein is correlated with its subcellular localization. An explanation for this observation was provided by Andrade, O'Donoghue, and Rost [106], who observed that the signal for subcellular localization was almost entirely due to the surface residues. Throughout evolution each subcellular compartment has maintained its characteristic physico-chemical environment, so it is not surprising that protein surfaces have evolved to adapt to these conditions. A wide array of methods has been developed to exploit this correlation of subcellular localization with sequence composition. The first tool to use amino acid composition was the PSORT expert system from Nakai and Kanehisa [107], which used standard statistical methods. However, it is only with the recent applications of machine-learning techniques that composition-based methods have started approaching the prediction accuracy of other methods. One of the earliest methods to use a machine-learning approach was the NNPSL predictor [108], which used feed-forward NNs trained on the amino acid composition. The network classified proteins from eukaryotic organisms into one of four possible subcellular compartments with an accuracy of 66% and prokaryotic proteins into one of three compartments with an accuracy of 81%. They also showed that the neural network predictions were fairly insensitive to sequencing errors near the N-terminal, adding

weight to the importance of the predictions. Hua and Sun [109] showed that support vector machines (SVMs) are even better at predicting localization from the amino acid composition. This is so since SVMs in general are better at extracting correlations when the dataset is relatively small and noisy [110]. By training SVMs on the dataset of Reinhardt and Hubbard [108], their SubLoc system was able to improve prediction accuracy by over 13%. Park and Kanehisa [111] have shown that adding residue pair compositions to the amino acid composition can improve prediction accuracy by over 5%. Their PLOC system classifies proteins into one of nine subcellular compartments with an accuracy of over 79%. Cai and colleagues [112–114] have tried to incorporate higher-order correlations among the amino acid residues (residues i and (i + n), n = 2,3,4) by using pseudo-amino acid composition. The pseudo-amino acid composition accounts for sequence-order effects by defining a correlation factor based on various biochemical properties, for every residue and its sequence neighbors. However, these methods are not publicly available, and their prediction accuracy is hard to assess. With the availability of large numbers of completely sequenced genomes, phylogenetic profiles have been employed to identify sucellular localization [115]. So far, this approach has been much less accurate in predicting localization than methods based solely on composition. By incorporating predicted secondary structure, solvent accessibility, and amino acid composition with evolutionary information into a multilevel neural network architecture, Nair and Rost [116] were able to significantly improve prediction accuracy over existing methods. Their LOCnet system is one of the most accurate *ab initio* methods for predicting localization from sequence.

10.6.2 LOCnet: Improving Predictions Using Evolution

The LOCnet [116] system consists of three layers of neural networks and sorts proteins into one of four subcellular classes (fig. 10.4). The first layer consists of dedicated neural networks that use particular features from protein sequences, alignments, and structure to presort proteins into L/not-L (where L = cytoplasmic, nuclear, extracellular, mitochondrial). Output from the first-layer networks, which are trained on different sequence features, is combined using a second layer of networks. The third layer uses a simple jury decision [117] to assign one of four localization-states to each protein. Major sources of improvement over publicly available methods originated from using predicted secondary structure (from PROFsec [118]), improved predictions of solvent accessibility (from PROFacc [118]), and evolutionary information from sequence profiles. LOCnet has a module that implicitly predicts generic signal peptides (but not the cleavage sites) and target peptides [119]. Although LOCnet performs better for extracellular proteins with signal peptides, it can identify proteins that are secreted using alternative pathways, such as fibroblast growth factors and the interleukin family of cytokines. In combination with other methods, it can distinguish between proteins with signal peptides that are retained in the Endoplasmic reticulum or Golgi apparatus and those that are actually secreted [53]. LOCnet was found to be over 7% more accurate than the best publicly available system [119] on an independent test set of newly annotated proteins in the Swiss-Prot database. The LOCnet system has been applied to

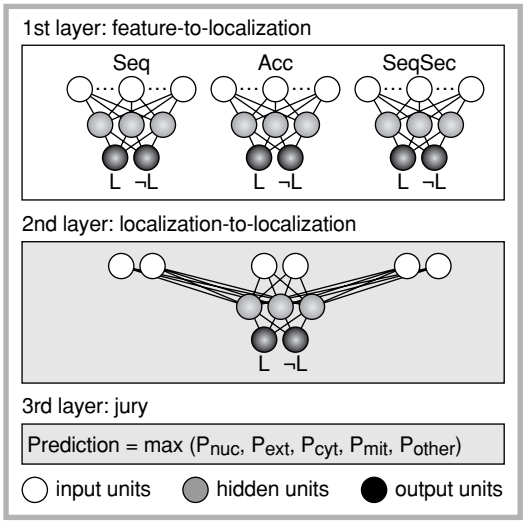


FIGURE 10.4 Neural network architecture of LOCnet. The first level of pairwise neural networks use an architecture of 20 to 60 input units and 2 output units with a hidden layer consisting of 3 to 9 units. The output from the different first-level pairwise neural networks are used as input to the second-level integrating neural network. The second-level pairwise networks consist of 6 input units and 2 output units with a hidden layer consisting of 3 units. The final localization prediction is based on a jury decision of the outputs from the different pairwise integrating networks.

annotate subcellular localization for all proteins in the PDB [120] and in TargetDB [119]. TargetDB [121] is a database of structural genomics targets and provides registration and tracking information for the National Institutes of Health structural genomics centers.

10.7 INTEGRATED METHODS FOR PREDICTING LOCALIZATION

10.7.1 Improving Accuracy through Combinations

The different strategies for predicting localization have their own strengths and weaknesses. High-accuracy methods like those based on sequence motifs and homology are plagued by the problem of low coverage and can provide annotations for less than one-third of known sequences. In this era of whole-genome sequencing, high-quality annotations for all proteins in an organism are needed. Currently the best solution available is to combine low-coverage methods with state-of-the-art high-coverage methods, like those based on composition. This approach was pioneered by Nakai and colleagues [61,122,123] with their PSORT system. PSORT II is an expert system that combines a comprehensive database of sorting signals with predictions based on composition. The LOCtarget [119] system combines predictions based on sequence motifs, homology, text analysis, and neural networks, and it can

distinguish between nine localization classes. In its current implementation the only sequence motifs used by LOCtarget are those responsible for sorting to the nucleus. Drawid and Gerstein [124] proposed a system that uses Bayesian statistics for integrating multiple kinds of information. (It integrates 30 different features, which include everything from SignalP predictions to microarray expression profiles.) They applied their method to predicting localization of the full *Saccharomyces cerevisiae* proteome and provided estimates of the fraction of all yeast proteins found in different compartments. Next we review PSORT II, which is one of the most widely used methods for predicting localization.

10.7.2 PSORT II: Expert System for Predicting Localization

The PSORT system [61] predicts the localization of proteins from gram-negative bacteria, gram-positive bacteria, yeasts, animals, and plants. For a query sequence the program calculates the values of feature variables that reflect various characteristics of the sequence (table 10.2). Next, it uses the k-nearest-neighbor algorithm to interpret the set of values obtained and estimates the likelihood of the protein being sorted to each candidate site. Finally, it displays some of the most probable sites.

TABLE 10.2
Features Detected by PSORT II

Feature	Criteria
N-terminal signal peptide	Modified McGeoch's method and the cleavage-site consensus
Mitochondrial-targeting signal	Amino acid composition of the N-terminal 20 residues and some weak cleavage site consensus
Nuclear-localization signals	Combined score for various empirical rules
ER-lumen-retention signal	The KDEL-like motif at the C-terminus
ER-membrane-retention signal	Motifs: XXRR-like (N-terminal) or KKXX-like (C-terminal)
Peroxisomal-targeting signal	PTS1 motif at the C-terminus and the PTS2 motif
Vacuolar-targeting signal	[TIK]LP[NKI] motif
Golgi-transport signal	The YQRL motif (preferentially at the cytoplasmic tail)
Tyrosine-containing motif	Number of tyrosine residues in the cytoplasmic tail
Dileucine motif	At the cytoplasmic tail
Membrane span(s)/topology	Maximum hydrophobicity and the number of predicted spans; charge difference across the most N-terminal transmembrane segment
RNA-binding motif	RNP-1 motif
Actinin-type actin-binding motifs	From PROSITE
DNA-binding motifs	63 motifs from PROSITE
Ribosomal-protein motifs	71 motifs from PROSITE
Prokaryotic DNA-binding motifs	33 motifs from PROSITE
N-myristoylation motif	At the N-terminus
Amino acid composition	Neural network score that discriminates between cytoplasmic and nuclear proteins
Coiled coil structure length	Number of residues in the predicted coile-coil state
Length	Length of sequence

The program achieved an overall prediction accuracy of 57% and can distinguish 11 subcellular classes. One reason for the lower accuracy of PSORT is our current incomplete knowledge of sorting signals. Extensions to PSORT II have been proposed: iPSORT [125] for extensive feature detection of N-terminal sorting signals and PSORT–B [54] for predicting localization of gram-negative bacteria.

10.8 CONCLUSION

10.8.1 Several Pitfalls in Assessing Quality of Annotations

To draw reliable inferences from a prediction, it is essential that the accuracy of the method be properly established. To obtain accurate estimates of performance, the testing procedure should mimic a blindfold prediction exercise as far as possible. One way of ensuring this is to choose the training data such that the test sequences have no sequence similarity to proteins in the training set. However this is often not the case, and many methods test their performance only on a small sample of selected proteins, resulting in overestimates of prediction accuracy. Another problem that affects prediction accuracy is the number of redundant sequences in public databases. Adequate care must be taken during development to avoid biased predictions toward large families of redundant protein sequences by using sequence unique test sets. Otherwise, estimated accuracy is likely to be much higher than the true prediction accuracy. Benchmarking prediction methods proves to be a difficult task, since the methods have been developed at different times and database annotations of function are constantly growing. In addition, there are no standard procedures for reporting prediction accuracy, with some methods only reporting the overall prediction accuracy, which can be quite uninformative because of the large differences in the sizes of the datasets for the different subcellular classes. Functional annotations in standard databases usually contain large numbers of incorrect annotations, which makes development of prediction tools all the more difficult. Another problem without any obvious solution is choosing an appropriate trade-off: sensitivity or specificity. Depending on the application, either high specificity or sensitivity might be desirable. Hence, caution should be exercised when using predictions from automatic servers, especially in cases where little is known about the function of the protein and the sequence signals that are involved in sorting. It is sometimes instructive to compare predictions from multiple servers that use different prediction strategies. Similar predictions from the servers might indicate some propensity of the protein for the predicted localization, while conflicting predictions might call for further research.

10.8.2 Prediction Accuracy Continues to Grow

In spite of the difficulties in correctly assessing the accuracy of prediction methods, during the last few years significant strides have been made in tackling the problem of subcellular localization prediction. One reason for the progression is the application of advanced machine-learning techniques, which can recognize subtle correlations among different kinds of sequence features. A second reason is the steady growth in the amount of functional information deposited in databases. Already

prediction tools are proving useful for automatic annotations of sequence databases and for screening potentially interesting genes from genome data. In the near future it might be possible to predict the subcellular location of almost any given protein with high confidence. Future improvements are likely to result through the use of integrated prediction methods that cleverly combine the output from programs that predict different functional features to provide a comprehensive prediction of subcellular localization. Integrated prediction methods better capture biological reality, since events affecting the fate of proteins are interrelated. For example, it is evident that a modification enzyme will not modify its potential substrates when the membrane separates them. Moreover, combination methods can be designed to naturally fall into an ontological scheme, which would help us achieve the goal of a unified framework for protein function prediction.

ACKNOWLEDGMENTS

Thanks to the following members of our group for helpful discussions: Jinfeng Liu, Dariusz Przybylski, and Kazimierz Wrzeszczynski. RN would also like to thank Christina Schlecht for proofreading the manuscript. Last, but not least, thanks to all those who deposit their experimental data in public databases and to those who maintain these databases, and to the World Wide Web for making so many resources easily accessible.

REFERENCES

1. Istrail, S., G. G. Sutton, L. Florea, A. L. Halpern, C. M. Mobarry, R. Lippert, B. Walenz, et al. 2004. Whole-genome shotgun assembly and comparison of human genome assemblies. *Proc Natl Acad Sci USA* 101:1916–21.
2. Venter, J. C., M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, et al. 2001. The sequence of the human genome. *Science* 291:1304–51.
3. Carter, P., J. Liu, and B. Rost. 2003. PEP: Predictions for Entire Proteomes. *Nucleic Acids Res* 31:410–3.
4. Liu, J., and B. Rost. 2001. Comparing function and structure between entire proteomes. *Protein Sci* 10:1970–9.
5. Pruess, M., W. Fleischmann, A. Kanapin, Y. Karavidopoulou, P. Kersey, E. Kriventseva, V. Mittard, et al. 2003. The proteome analysis database: A tool for the in silico analysis of whole proteomes. *Nucleic Acids Res* 31:414–7.
6. Zhu, H., M. Bilgin, and M. Snyder. 2003. Proteomics. *Annu Rev Biochem* 72:783–812.
7. Brutlag, D. L. 1998. Genomics and computational molecular biology. *Curr Opin Microbiol* 1:340–5.
8. Harrison, P. M., P. Bamborough, V. Daggett, S. Prusiner, and F. E. Cohen. 1997. The prion folding problem. *Curr Opin Struct Biol* 7:53–9.
9. Bork, P., and E. V. Koonin. 1998. Predicting functions from protein sequences—Where are the bottlenecks? *Nat Genet* 18:313–8.
10. Smith, T. F. 1998. Functional genomics—Bioinformatics is ready for the challenge. *Trends Genet* 14:291–3.
11. Bork, P., T. Dandekar, Y. Diaz-Lazcoz, F. Eisenhaber, M. Huynen, and Y. Yuan. 1998. Predicting function: From genes to genomes and back. *J Mol Biol* 283:707–25.

12. Fleischmann, W., S. Moller, A. Gateau, and R. Apweiler. 1999. A novel method for automatic functional annotation of proteins. *Bioinformatics* 15:228–33.
13. Luscombe, N. M., D. Greenbaum, and M. Gerstein. 2001. What is bioinformatics? A proposed definition and overview of the field. *Methods Inf Med* 40:346–58.
14. Rost, B., Liu, J., Nair, R., Wrzeszczynski, K. O., and Ofran, Y. 2003. Automatic prediction of protein function. *Cell Mol Life Sci* 60:2637–50.
15. Apweiler, R., T. K. Attwood, A. Bairoch, A. Bateman, E. Birney, M. Biswas, P. Bucher, et al. 2000. InterPro—An integrated documentation resource for protein families, domains and functional sites. *Bioinformatics* 16:1145–50.
16. Overbeek, R., N. Larsen, W. Smith, N. Maltsev, and E. Selkov. 1997. Representation of function: The next step. *Gene* 191:GC1–GC9.
17. Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, et al. 2000. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* 25:25–9.
18. Ohlstein, E. H., R. Ruffolo, Jr., and J. D. Elliott. 2000. Drug discovery in the next millennium. *Annu Rev Pharmacol Toxicol* 40:177–91.
19. Maliepaard, M., G. L. Scheffer, I. F. Faneyte, M. A. van Gastelen, A. C. Pijnenborg, A. H. Schinkel, M. J. van De Vijver, et al. 2001. Subcellular localization and distribution of the breast cancer resistance protein transporter in normal human tissues. *Cancer Res* 61:3458–64.
20. Clark, H. F., A. L. Gurney, E. Abaya, K. Baker, D. Baldwin, J. Brush, J. Chen, et al. 2003. The secreted protein discovery initiative (SPDI), a large-scale effort to identify novel human secreted and transmembrane proteins: A bioinformatics assessment. *Genome Res* 13:2265–70.
21. Bork, P., C. Ouzounis, and C. Sander. 1994, From genome sequences to protein function. *Curr Opin Struct Biol* 4:393–403.
22. Kumar, A., S. Agarwal, J. A. Heyman, S. Matson, M. Heidtman, S. Piccirillo, L. Umansky, et al. 2002. Subcellular localization of the yeast proteome. *Genes Dev* 16:707–19.
23. Huh, W. K., J. V. Falvo, L. C. Gerke, A. S. Carroll, R. W. Howson, J. S. Weissman, and E. K. O'Shea. 2003. Global analysis of protein localization in budding yeast. *Nature* 425:686–91.
24. Kleffmann, T., D. Russenberger, A. von Zychlinski, W. Christopher, K. Sjolander, W. Gruissem, and S. Baginsky. 2004. The Arabidopsis thaliana chloroplast proteome reveals pathway abundance and novel protein functions. *Curr Biol* 14:354–62.
25. Davis, T. N. 2004. Protein localization in proteomics. *Curr Opin Chem Biol* 8:49–53.
26. Koonin, E. V. 2000. Bridging the gap between sequence and function. *Trends Genet* 16:16.
27. Eisenhaber, F., and P. Bork. 1998. Wanted: Subcellular localization of proteins based on sequence. *Trends Cell Biol* 8:169–70.
28. Nakai, K. 2000. Protein sorting signals and prediction of subcellular localization. *Adv Protein Chem* 54:277–344.
29. Schneider, G., and U. Fechner. 2004. Advances in the prediction of protein targeting signals. *Proteomics* 4:1571–80.
30. Lodish, H., A. Berk, D. Baltimore, and J. Darnell. 2000. *Molecular cell biology.* 4th ed. New York: Freeman.
31. Mattaj, I. W., and L. Englmeier. 1998. Nucleocytoplasmic transport: The soluble phase. *Annu Rev Biochem* 67:265–306.
32. Schatz, G., and B. Dobberstein. 1996. Common principles of protein translocation across membranes. *Science* 271:1519–26.

33. Bar-Peled, M., D. C. Bassham, and N. V. Raikhel. 1996. Transport of proteins in eukaryotic cells: More questions ahead. *Plant Mol Biol* 32:223–49.
34. Bauer, M. F., S. Hofmann, W. Neupert, and M. Brunner. 2000. Protein translocation into mitochondria: The role of TIM complexes. *Trends Cell Biol* 10:25–31.
35. Darnell, J., H. Lodish, and D. Baltimore. 1990. *Molecular cell biology.* 2nd ed. New York: Freeman.
36. Devos, D., and A. Valencia. 2001. Intrinsic errors in genome annotation. *Trends Genet* 17:429–31.
37. Valencia, A., and F. Pazos. 2002. Computational methods for the prediction of protein interactions. *Curr Opin Struct Biol* 12:368–73.
38. Nakai, K. 2001. Review: Prediction of in vivo fates of proteins in the era of genomics and proteomics. *J Struct Biol* 134:103–16.
39. Apweiler, R., A. Gateau, S. Contrino, M. J. Martin, V. Junker, C. O'Donovan, F. Lang, N. Mitaritonna, S. Kappus, and A. Bairoch. 1997. Protein sequence annotation in the genome era: The annotation concept of SWISS-PROT+TREMBL. *Proc Int Conf Intell Syst Mol Biol* 5:33–43.
40. Bairoch, A., and R. Apweiler. 1997. The SWISS-PROT protein sequence data bank and its new supplement TrEMBL. *Nucleic Acids Res* 25:31–6.
41. Airozo, D., R. Allard, B. Brylawski, K. Canese, D. Kenton, L. Knecht, S. Krasnov, et al. 1999. MEDLINE. *National Library of Medicine (NLM),* Vol. 1999.
42. Simpson, J. C., R. Wellenreuther, A. Poustka, R. Pepperkok, and S. Wiemann. 2000. Systematic subcellular localization of novel proteins identified by large-scale cDNA sequencing. *EMBO Rep* 1:287–92.
43. Koonin, E. v., R. L. Tatusov, and K. E. Rudd. 1996. Protein sequence comparison at genome scale. *Methods Enzymol* 266:295–322.
44. Tamames, J., C. Ouzounis, G. Casari, C. Sander, and A. Valencia. 1998. EUCLID: Automatic classification of proteins in functional classes by their database annotations. *Bioinformatics* 14:542–3.
45. Thornton, J. M., C. A. Orengo, A. E. Todd, and F. M. Pearl. 1999. Protein folds, functions and evolution. *J Mol Biol* 293:333–42.
46. Orengo, C. A., A. E. Todd, and J. M. Thornton. 1999. From protein structure to function. *Curr Opin Struct Biol* 9:374–82.
47. Wilson, C. A., J. Kreychman, and M. Gerstein. 2000. Assessing annotation transfer for genomics: Quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. *J Mol Biol* 297:233–49.
48. Rost, B. 2002. Enzyme function less conserved than anticipated. *J Mol Biol* 318:595–608.
49. Pawlowski, K., and A. Godzik. 2001. Surface map comparison: Studying function diversity of homologous proteins. *J Mol Biol* 309:793–806.
50. Nair, R., and B. Rost. 2002. Sequence conserved for subcellular localization. *Protein Sci* 11:2836–47.
51. Galperin, M. Y., and E. V. Koonin. 2000. Who's your neighbor? New computational approaches for functional genomics. *Nat Biotechnol* 18:609–13.
52. Doerks, T., A. Bairoch, and P. Bork. 1998. Protein annotation: Detective work for function prediction. *Trends Genet* 14:248–50.
53. Wrzeszczynski, K. O., and B. Rost. 2004. Annotating proteins from endoplasmic reticulum and Golgi apparatus in eukaryotic proteomes. *Cell Mol Life Sci* 61:1341–53.
54. Gardy, J. L., C. Spencer, K. Wang, M. Ester, G. E. Tusnady, I. Simon, S. Hua, et al. 2003. PSORT-B: Improving protein subcellular localization prediction for Gram-negative bacteria. *Nucleic Acids Res* 31:3613–7.

55. Sander, C., and R. Schneider. 1991. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins* 9:56–68.
56. Rost, B. 1999. Twilight zone of protein sequence alignments. *Protein Eng* 12:85–94.
57. Altschul, S. F., T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. 1997. Gapped Blast and PSI-Blast: A new generation of protein database search programs. *Nucleic Acids Res* 25:3389–402.
58. Altschul, S. F., and W. Gish. 1996. Local alignment statistics. *Methods Enzymol* 266:460–80.
59. Boeckmann, B., A. Bairoch, R. Apweiler, M. C. Blatter, A. Estreicher, E. Gasteiger, M. J. Martin, et al. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res* 31:365–70.
60. von Heijne, G. 1995. Protein sorting signals: Simple peptides with complex functions. *Exs* 73:67–76.
61. Nakai, K., and P. Horton. 1999. PSORT: A program for detecting sorting signals in proteins and predicting their subcellular localization. *Trends Biochem Sci* 24:34–6.
62. Cokol, M., R. Nair, and B. Rost. 2000. Finding nuclear localization signals. *EMBO Rep* 1:411–5.
63. von Heijne, G. 1981. On the hydrophobic nature of signal sequences. *Eur J Biochem* 116:419–22.
64. von Heijne, G. 1985. Signal sequences. The limits of variation. *J Mol Biol* 184:99–105.
65. Voos, W., H. Martin, T. Krimmer, and N. Pfanner. 1999. Mechanisms of protein translocation into mitochondria. *Biochim Biophys Acta* 1422:235–54.
66. Bruce, B. D. 2000. Chloroplast transit peptides: Structure, function and evolution. *Trends Cell Biol* 10:440–7.
67. Nielsen, H., S. Brunak, and G. von Heijne. 1999. Machine learning approaches for the prediction of signal peptides and other protein sorting signals. *Protein Eng* 12:3–9.
68. Emanuelsson, O., H. Nielsen, S. Brunak, and G. von Heijne. 2000. Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J Mol Biol* 300:1005–16.
69. Kall, L., A. Krogh, and E. L. Sonnhammer. 2004. A combined transmembrane topology and signal peptide prediction method. *J Mol Biol* 338:1027–36.
70. Nielsen, H., J. Engelbrecht, S. Brunak, and G. von Heijne. 1997. A neural network method for identification of prokaryotic and eukaroytoic signal peptides and prediction of their cleavage sites. *Internationl Journal of Neural Systems* 8:581–99.
71. Fujiwara, Y., M. Asogawa, and K. Nakai. 1997. Prediction of mitochondrial targeting signals using hidden Markov model. *Genome Inform Ser Workshop Genome Inform* 8:53–60.
72. Emanuelsson, O., G. von Heijne, and G. Schneider. 2001. Analysis and prediction of mitochondrial targeting peptides. *Methods Cell Biol* 65:175–87.
73. Emanuelsson, O., H. Nielsen, and G. von Heijne. 1999. ChloroP, a neural network-based method for predicting chloroplast transit peptides and their cleavage sites. *Protein Science* 8:978–84.
74. Gaasterland, T., and M. Oprea. 2001. Whole-genome analysis: Annotations and updates. *Curr Opin Struct Biol* 11:377–81.
75. Lander, E. S., L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, et al. 2001. Initial sequencing and analysis of the human genome. *Nature* 409:860–921.
76. Emanuelsson, O., and G. von Heijne. 2001. Prediction of organellar targeting signals. *Biochim Biophys Acta* 1541:114–9.

77. Durbin, R., S. R. Eddy, A. Krogh, and G. Mitchison. 1998. *Biological sequence analysis*. Cambridge, UK: Cambridge Univ. Press.
78. Jans, D. A., C. Y. Xiao, and M. H. Lam. 2000. Nuclear targeting signal recognition: A key control point in nuclear transport? *Bioessays* 22:532–44.
79. Moroianu, J. 1999. Nuclear import and export: Transport factors, mechanisms and regulation. 1999. *Crit Rev Eukaryot Gene Expr* 9:89–106.
80. Liu, J., and B. Rost. 2002. Target space for structural genomics revisited. *Bioinformatics* 18:922–33.
81. Qian, N., and T. J. Sejnowski. 1988. Predicting the secondary structure of globular proteins using neural network models. *J Mol Biol* 202:865–84.
82. Nair, R., P. Carter, and B. Rost. 2003. NLSdb: Database of nuclear localization signals. *Nucleic Acids Res* 31:397–9.
83. Berman, H. M., J. Westbrook, Z. Feng, G. Gilllliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. 1995. The Protein Data Bank. *Nucleic Acids Res* 28:235–42.
84. LaCasse, E. C., and Y. A. Lefebvre. 1995. Nuclear localization signals overlap DNA- or RNA-binding domains in nucleic acid-binding proteins. *Nucleic Acids Res* 23:1647–56.
85. Iliopoulos, I., A. J. Enright, and C. A. Ouzounis. 2001. Textquest: Document clustering of Medline abstracts for concept discovery in molecular biology. *Pac Symp Biocomput* 6:384–95.
86. Stephens, M., M. Palakal, S. Mukhopadhyay, R. Raje, and J. Mostafa. 2001. Detecting gene relations from Medline abstracts. *Pac Symp Biocomput* 6:483–95.
87. Stapley, B. J., and G. Benoit. 2000. Biobibliometrics: Information retrieval and visualization from co-occurrences of gene names in Medline abstracts. *Pac Symp Biocomput* 6:529–40.
88. Friedman, C., P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. 2001. GENIES: A natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics* 17, Suppl. no. 1:S74–82.
89. Ng, S. K., and M. Wong. 1999. Toward routine automatic pathway discovery from on-line scientific text abstracts. *Genome Inform Ser Workshop Genome Inform* 10:104–12.
90. Hatzivassiloglou, V., P. A. Duboue, and A. Rzhetsky. 2001. Disambiguating proteins, genes, and RNA in text: A machine learning approach. *Bioinformatics* 17 Suppl. no. 1: S97–106.
91. Apweiler, R. 2001. Functional information in SWISS-PROT: The basis for large-scale characterisation of protein sequences. *Brief Bioinform* 2:9–18.
92. Nair, R., and B. Rost. 2002. Inferring sub-cellular localization through automated lexical analysis. *Bioinformatics* 18, Suppl. no. 1:S78–86.
93. Lu, Z., D. Szafron, R. Greiner, P. Lu, D. S. Wishart, B. Poulin, J. Anvik, C. Macdonell, and R. Eisner. 2004. Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics* 20:547–56.
94. Eisenhaber, F., and P. Bork. 1999. Evaluation of human-readable annotation in biomolecular sequence databases with biological rule libraries. *Bioinformatics* 15:528–35.
95. Andrade, M. A., C. Ouzounis, C. Sander, J. Tamames, and A. Valencia. 1999. Functional classes in the three domains of life. *J Mol Evol* 49:551–7.
96. Ouzounis, C., G. Casari, C. Sander, J. Tamames, and A. Valencia. 1996. Computational comparisons of model genomes. *Trends Biotechnol* 14:280–5.

97. Lewis, D. D., and M. Ringuette. 1994. Comparison of two learning algorithms for text categorization. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, 81–93.
98. Apte, C., F. Damerau, and S. Weiss. 1994. Towards language independent automated learning of text categorization models. In *Proceedings of the 17th Annual ACM/SIGIR conference* 23–30. New York: ACM/Springer.
99. Dasarathy, B. V. 1991. *Nearest Neighbor (NN) norms: NN pattern classification techniques*. Las Alamitos, CA: IEEE Computer Society Press.
100. Kretschmann, E., W. Fleischmann, and R. Apweiler. 2001. Automatic rule generation for protein annotation with the C4.5 data mining algorithm applied on SWISS-PROT. *Bioinformatics* 17:920–6.
101. Bazzan, A. L., P. M. Engel, L. F. Schroeder, and S. C. Da Silva. 2002. Automated annotation of keywords for proteins related to mycoplasmataceae using machine learning techniques. *Bioinformatics* 18, Suppl. no. 2:S35–43.
102. Stapley, B. J., L. A. Kelley, and M. J. Sternberg. 2002. Predicting the sub-cellular location of proteins from text using support vector machines. *Pac Symp Biocomput* 7:374–85.
103. Salton, G. 1989. *Automatic text processing*. Reading, MA:Addison-Wesley.
104. Nishikawa, K., and T. Ooi. 1982. Correlation of the amino acid composition of a protein to its structural and biological characteristics. *J Biochem* 91:1821–4.
105. Nakashima, H., and K. Nishikawa. 1994. Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies. *J Mol Biol* 238:54–61.
106. Andrade, M. A., S. I. O'Donoghue, and B. Rost. 1998. Adaptation of protein surfaces to subcellular location. *J Mol Biol* 276:517–25.
107. Nakai, K., and M. Kanehisa. 1991. Expert system for predicting protein localization sites in gram-negative bacteria. *Proteins: Struct Funct Genet* 11:95–110.
108. Reinhardt, A., and T. Hubbard. 1998. Using neural networks for prediction of the subcellular location of proteins. *Nucleic Acids Res* 26:2230–6.
109. Hua, S., and Z. Sun. 2001. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics* 17:721–8.
110. Vapnik, V. N. 1995. *The nature of statistical learning theory.* New York: Springer-Verlag.
111. Park, K. J., and M. Kanehisa. 2003. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics* 19:1656–63.
112. Cai, Y. D., X. J. Liu, X. B. Xu, and K. C. Chou. 2002. Support vector machines for prediction of protein subcellular location by incorporating quasi-sequence-order effect. *J Cell Biochem* 84:343–8.
113. Chou, K. C., and Y. D. Cai. 2003. Prediction and classification of protein subcellular location-sequence-order effect and pseudo amino acid composition. *J Cell Biochem* 90:1250–60.
114. Pan, Y. X., Z. Z. Zhang, Z. M. Guo, G. Y. Feng, Z. D. Huang, and L. He. 2003. Application of pseudo amino acid composition for predicting protein subcellular location: Stochastic signal processing approach. *J Protein Chem* 22:395–402.
115. Marcotte, E. M., I. Xenarios, A. M. van Der Bliek, and D. Eisenberg. 2000. Localizing proteins in the cell from their phylogenetic profiles. *Proc Natl Acad Sci USA* 97:12115–20.
116. Nair, R., and B. Rost. 2003. Better prediction of sub-cellular localization by combining evolutionary and structural information. *Proteins* 53:917–30.

117. Rost, B., and C. Sander. 1993. Prediction of protein secondary structure at better than 70% accuracy. *J Mol Biol* 232:584–99.
118. Rost, B., G. Yachdav, and J. Liu. 2004. The PredictProtein server. *Nucleic Acids Res* 32, Web Server Issue:W321–6.
119. Nair, R., and B. Rost. 2004. LOCnet and LOCtarget: Sub-cellular localization for structural genomics targets. *Nucleic Acids Res* 32, Web Server Issue:W517–21.
120. Nair, R., and B. Rost. 2003. LOC3D: Annotate sub-cellular localization for protein structures. *Nucleic Acids Res* 31:3337–40.
121. Westbrook, J., Z. Feng, L. Chen, H. Yang, and H. M. Berman. 2003. The Protein Data Bank and structural genomics. *Nucleic Acids Res* 31:489–91.
122. Nakai, K., and M. Kanehisa. 1992. A knowledge base for predicting protein localization sites in eukaryotic cells. *Genomics* 14:897–911.
123. Horton, P., and K. Nakai. 1997. Better prediction of protein cellular localization sites with the k nearest neighbors classifier. *Ismb* 5:147–52.
124. Drawid, A., and M. Gerstein. 2000. A Bayesian system integrating expression data with sequence patterns for localizing proteins: Comprehensive application to the yeast genome. *J Mol Biol* 301:1059–75.
125. Bannai, H., Y. Tamada, O. Maruyama, K. Nakai, and S. Miyano. 2002. Extensive feature detection of N-terminal protein sorting signals. *Bioinformatics* 18:298–305.
126. Alberts, B., D. Bray, K. Roberts, and J. Watson. 1994. *Molecular biology of the cell*. 3rd ed. New York: Garland.
127. Nielsen, H., J. Engelbrecht, S. Brunak, and G. von Heijne. 1997. A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Int J Neural Syst* 8:581–99.
128. Claros, M. G. 1995. MitoProt, a Macintosh application for studying mitochondrial proteins. *Comput Appl Biosci* 11:441–7.
129. Small, I., N. Peeters, F. Legeai, and C. Lurin. 2004. Predotar: A tool for rapidly screening proteomes for N-terminal targeting sequences. *Proteomics* 4:1581–90.

11 Three-Dimensional Structures in Target Discovery and Validation

Seán I. O'Donoghue
Mandala IT

Robert B. Russell
European Molecular Biology Laboratory

Andrea Schafferhans
Lion Bioscience AG

CONTENTS

11.1 INTRODUCTION

For nearly half of all known protein sequences, some information can be inferred about their three-dimensional (3D) structure. In spite of the wealth of structural data and tools available, many scientists still fail to benefit from this information, because it can be somewhat difficult to access and use.

In the first section of this chapter we present a practical approach to accessing and using structural data in target identification and validation, and we have chosen to concentrate on only the most practical and easy-to-apply methods. In the second section, we discuss some specialized methods such as multiple-structure comparison, binding site comparisons, and docking. In the final section, we discuss some emerging methods that treat the problem of using structural information in a more "systemic" way. These emerging methods are not so easy to use, but we believe they will become more important and easier to use in the future.

As with other areas in the biosciences, such as sequencing, the methodology of structure determination has been increasingly automated and accelerated. However, since structure determination is intrinsically so much more complex than sequencing, it will always be much more expensive and slower. Determining a single protein structure can often take ½ man-year and may cost $100,000. However, both in industry and in the academic community structure determination continues primarily due to the wealth of insightful and precise information that can be obtained.

Because of the complexity of structure determination, the gap between the number of known structures and known sequences is huge. In September 2005 there were 60 million nucleotide sequences in the complete EMBL [1] and 2.3 million proteins sequences in UniProt [2]. But the Protein Databank (PDB) [3]—containing almost all experimentally determined 3D structures of proteins and nucleic acids—contained only 33,000 entries.

Thus, less than 1% of protein sequences in UniProt are directly linked to a known 3D structure. However, the situation is much improved because of our ability to infer structural similarity based on sequence homology. Many protein sequences are similar in sequence to a known 3D structure, thus some structural information can be inferred by homology for about half of all UniProt sequences [4]. Much of this information about structure is easy to access and can help in many stages of the drug-discovery process. One of our goals in this chapter is to outline the easiest methods available to encourage scientists who are not experts in structure to take advantage of structural information.

As just mentioned, knowing the 3D structure of a protein gives immediate and valuable insight into function. A dramatic illustration of this value is the increasingly common practice of determining 3D structures as a way to understand function for newly discovered proteins, where little or no other information is known about function. Whereas 10 years ago structure determination was done only when a protein's function was already well understood, today it has become increasingly routine.

Thus, during many drug-discovery projects, it is quite likely that new, related structures will appear. We encourage scientists to mine these structures for as much information as can be easily and automatically determined, in much the same way as when a new potential target sequence appears.

To date, the greatest impact of any structure determined is the structure of DNA [5]. The insight gained from this structure spawned modern molecular biology. However, DNA replication is just one process among millions of other biological processes. When the first protein structure was solved 7 years later [6], it became clear that proteins are much more irregular and complex than DNA. As each new protein structure was solved, a multitude of different structural systems was discovered, often requiring considerably more effort to understand than the comparatively simple DNA replication mechanism. In the last 10 years, the multitude is beginning to become easier to understand and classify, with the emergence of domains and binding site motifs.

There remains the tantalizing possibility that some key protein structure determined in the future may have even wider implications than unlocking the mechanism behind DNA replication.

11.2 FROM SEQUENCE TO STRUCTURES

At the early stage of a target identification and validation process, when a new target sequence has been identified as a potential drug target, it is often useful to find out as much information about the target sequence as possible, including any structural information that may be available. In the first section of this chapter, we focus on online resources that can be used to provide immediate information about the structure properties of a target sequence.

11.2.1 How to Find Related Structures

A good place to begin is to find all 3D structures with significant sequence similarity to the target sequence and then select the most relevant. What is the best way to do this? There are many options. Here we discuss only a selection of methods that share the following characteristics: are accessible via the Web, can deliver all related structures in a few seconds or minutes, and are easy to use even for those who are not expert in protein structure.

The National Center for Biotechnology Information (NCBI) Web site allows users to easily enter a protein sequence and run a BLAST [7] sequence search against all sequences in PDB (e.g., navigate to http://ncbi.nlm.nih.gov/BLAST, choose protein–protein BLAST, then select "pdb" from "Choose database"). This method is easy and quick, and the results are presented in an intuitive graphical display, clearly showing all matching structures aligned onto the target sequence, ranked in order of similarity (fig. 11.1A).

Another site, Swiss-Model [8] (http://swissmodel.expasy.org/), also uses BLAST to allow users to easily and quickly search PDB sequences for a matches. Currently, however, the matches are presented only in a table, so it is not easy to see where the structures match the target sequence. An advantage of Swiss-Model is that it allows the user to then select a structure to be used as a template and automatically generates a homology model. Of course, this process can take some time.

However, BLAST is not the most accurate method for finding similarities and will miss some matching structures that other methods can detect.

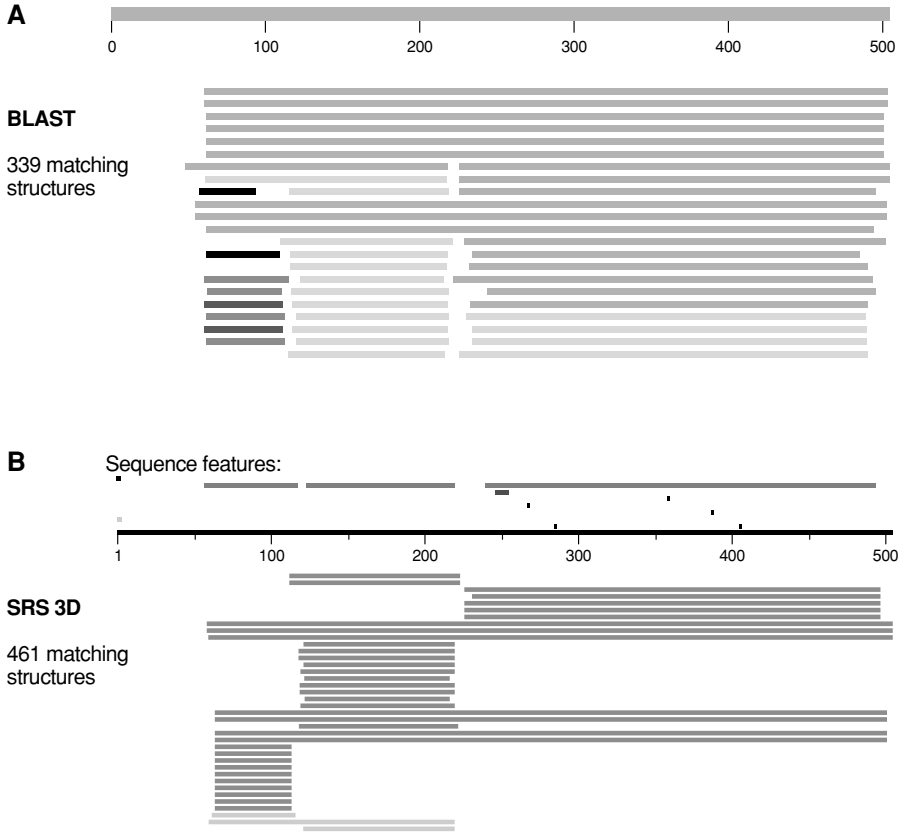


FIGURE 11.1 (See color insert following page 306) All matching structures for the target sequence B-lymphocyte kinase (UniProt P51451) found by NCBI/BLAST[75] (2a) and EBI/SRS 3D[13] (2b). In both cases the target sequence is represented by the bar numbered 1–500. BLAST finds 339 matching structures, of which the top-ranking matches are shown here ranked by alignment score. SRS 3D, like other profile methods, finds more matches than BLAST; in this case it finds 461 matching structures, of which the top ranking matches are shown here ranked by sequence identity. Views like this make it easy to see where in the target sequence a structure matches.

Some examples of more sensitive similarity detection methods using domain profiles are Pfam [9] (http://www.sanger.ac.uk/Pfam), SMART [10] (http://smart.embl-heidelberg.de), InterPro [11] (http://www.ebi.ac.uk/interpro/), and CDD [12] (http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml). At these sites you can enter either a sequence or an accession number of a target sequence, useful only if the sequence is stored in one of the main public databases. The site then shows domains that are found in your sequence. For each domain, you can navigate to find a list of all structures that are available for each domain. Using these profile methods will usually find more matching 3D structures than simple sequence searching, since very remote similarity can be picked up only by transitivity (i.e., if A is homologous to B, B is homologous to C, and C is homologous to known structure X, then one

can infer that A is homologous to X). In addition, profile methods give a higher-quality alignment than using BLAST. Of these sites, we prefer CDD, as it allows users to easily launch a 3D viewer program, where one can see the target sequence aligned onto a chosen structure. However, these sites were not designed specifically to allow easy access to related structures. In particular, the graphical display showing all matching structures aligned onto the target sequence, ranked in order of similarity, is lacking. It is not easy, for example, to see all matching structures ranked in order of their sequence similarity to the target sequence.

The SRS 3D site [13] (http://srs3d.ebi.ac.uk) has been specifically designed to make it easy to find all related structures for a given sequence. For all sequences in UniProt, SRS 3D precalculates all related structures using sequence profile methods. For each UniProt entry with related 3D structures, a link is added to an entry in the Protein Sequence-to-Structure Homologies (PSSH) database. Opening the PSSH entry for a UniProt sequence shows all matching structures aligned onto the target sequence, ranked in order of similarity, giving a concise summary of all known structural information for that sequence (fig. 11.1B). A quick way to access these views is via the URL template:

http://srs3d.ebi.ac.uk/srsbin/cgi-bin/wgetz?-id+-e+-e+[PSSH:'P51451']

changing P51451 to the UniProt accession number of your target sequence (similarly, you can change PSSH:P51451 to PDB:5AIY to display a PDB structure in SRS 3D). However, for sequences not in UniProt, the user must paste the sequence into the user-defined database, then run a BLAST search against the PDBSEQ database, with View set to SeqSimple3D. Since this uses BLAST and not a profile search, these cases will have lower-quality alignments and may miss some hits. SRS 3D also allows the user to easily view any matching 3D structure together with the sequence-to-structure alignment, colored by similarity to the target sequence.

Another site of interest is ModBase [14] (http://ModBase.compbio.ucsf.edu/), which is a large database of homology models, precalculated and of high quality, covering essentially all of UniProt. This database's advantage is that it allows users fast access to homology models. A drawback is that each sequence has an average of only three homology models. This means that the user cannot see all structures related to his sequence, and homology models are available only for a small number of matching structures that have been chosen automatically by the method, not the user.

One more site that can easily generate homology models is 3D-Jigsaw [15] (http://www.bmm.icnet.uk/~3djigsaw/). Unfortunately it cannot presently be used interactively to obtain a list of structures.

Finally, if no matching structures can be found for your target sequence using these methods (sequence or profile search), there is another group of methods, namely fold recognition methods, such as threading.

Even though a target structure may not match any known sequence pattern, it is still possible—even likely—that the target protein adopts a 3D fold very similar to a known structure. In fact most pairs of sequences that have the same fold have no detectable sequence similarity [16]. The aim of fold recognition methods is to detect cases when a protein has a known fold even though its sequence has no

detectable similarity to known sequences. Fold recognition is an extremely difficult problem; however, significant progress has been made in these methods [17]. Several of these methods are available via the Web, and a good place to start is the Structure Prediction Meta Server [18] (http://BioInfo.PL/meta/). The site gives the user an overview of several prediction methods as well as the best guess as to the right answer. Several other "metaservers" also offer similar functionality [19,20], and blind trials have shown these consensus prediction methods to be better than any single method [21,22].

11.2.2 Which Structures to Choose

Assuming you have found several structures with significant homology to your target sequence, which of these structures should you choose for further analysis? Some of the methods for finding structure can be used in a mode that automatically selects the "best" structures (e.g., ModBase and SwissModel). However, we strongly recommend always choosing your own template structure, as this choice greatly affects the relevance of the information you can get out of structures. Our reasons for this are outlined next, where we list some of the major issues to consider in deciding which of the several structures are the most relevant, based on the questions you would like to address.

11.2.2.1 Identical Sequences Are Not Always Equal

If you are fortunate, you will find several structures that have been solved for your target sequence. In this case, it pays to look in more detail at the exact region(s) of match between the structure and the target sequence. Typically, a structure will not span the full length of the sequence; it will have at least a part of the sequence deleted or span only one of the several domains in a protein.

Among the matching structures, there can also be subtle differences in sequence. Sometimes conservative point mutations are introduced to help solve the structure or specifically to study the effect of natural variations.

Often, several residues or entire loop regions may have been deleted from the sequence, or may be present in the sequence, but not present in the structure due to lack of experimental data. This happens especially for highly mobile parts of crystal structures with poor resolution.

We are not suggesting that the aforementioned considerations are always critical. Generally, the structure and function of a domain are not altered by conservative point mutations or even by removing parts of the sequence not involved in the domain. However, there is always the chance that even a conservative change in the sequence can have a dramatic impact. Thus, the more the structure differs from the full-length target sequence, the less certain the conclusions that you can draw from it.

If you are faced with a choice of deciding which structure(s) to work with, one consideration should be whether the mutations or missing residues are in areas that are relevant to the question to be solved using the structure. A mutation in the binding site, for example, would certainly affect any information to be gained about ligands binding at that site.

11.2.2.2 Sequence Similarity Is Best Guide—Usually!

For most target sequences there will be no structures found with a 100% sequence match. However, in about half of all cases there will be several protein structures that are significantly similar in sequence, so some structural information can be inferred about the target sequence [4]. Usually, this means that the overall fold of one or more domains can be inferred, together with the location of binding site residues and other special sites.

Assuming that the target sequence has several related structures with significantly similar homology, which structures should you choose? Which are the most relevant?

As previously discussed, the first criterion is to evaluate where the structure matches your target sequence. In a typical case, your target protein has several domains, and these are matched by different structures, but no matching structure contains all domains in the target sequence (fig. 11.1).

The next criterion is the level of sequence similarity between the target sequence and each structure. This level determines the degree of confidence you can have that the target sequence is likely to adopt a structure similar to the structure found. A generally used but arbitrary rule is that sequence identity in the range of 60 to 100% is considered to be high, that is, sequences with this similarity are highly likely to have the same fold. Sequence identity in the range of 30 to 59% is considered medium, and below 30% is low or not detectable [23]. However, these ranges are an oversimplification. For very short alignments (say 16 residues), 60% is usually not significant, and for long alignments (say more than 100 residues), the threshold for high similarity is only 40% [24].

11.2.2.3 Complexes and Oligomers

Most protein structures contain at least one small molecule forming a complex with the protein. Often these small molecules are natural ligands, or synthetic analogs that are very similar to the natural ligand. However, this is not always the case. Especially when viewing X-ray crystal structures, it is important to realize that crystal structures often include heavy metal ions that were needed to allow the 3D structure to be determined but that otherwise do not have biological significance and never occur with the protein *in vivo*. It is always possible that these metal ions may distort the structure or interfere with binding to other molecules.

Many structures contain not just one protein but several different protein molecules combined in a complex. To understand the relevance of a given complex structure to the questions you would like to address, it is usually important to find out the specific reasons the particular complex was solved. This can be done by referring to at least the abstract of the publication describing the structure.

Often, structures in the PDB are symmetric oligomers; however, not all of these oligomer states occur *in vivo*. Some are a consequence of crystal packing or solution conditions, or they may even arise due to mutations or small molecules. The PDB entry sometimes contains a statement indicating if the oligomer state is biologically relevant. In other cases, the oligomer state in the structure may occur *in vivo*, but it may be only one of several states that occur. However, it is highly recommended to

refer to the publication describing the structure and not simply assume that the oligomer state in the structure is biologically relevant or that it is the only state that can occur.

11.2.2.4 Differences Because of Experimental Method

In addition to the factors already discussed, the experimental methods used to derive a structure can also be significant. A basic introduction to the experimental methods used in structure determination is given at http://www.rcsb.org/pdb/experimental_methods.html. Here we summarize a few points to keep in mind when choosing structures.

Over 80% of available protein structures are derived from X-ray crystallography. However, not all are of equal quality. X-ray crystal structures have a resolution, which is a widely accepted indicator of quality. The lower the numeric value is for resolution the better. For example, a resolution of 5.0 Å indicates a very low-quality structure, 2.5 Å is about average, and 1.2 Å is very high quality [25]. All other factors being equal, it is preferable to choose structures with good resolution. In some cases, it may be preferable to choose a high-resolution structure with medium sequence similarity to the target sequence over a structure with high sequence similarity but very poor resolution.

About 16% of structures are derived from NMR spectroscopy. NMR structure entries usually contain a small ensemble of 10 to 50 structures. Looking at the ensemble gives an impression of the range of motions in the protein [26]; however, you do not always want this additional level of information in the first instance. In some cases, the authors deposit an "average" structure as well, often as a separate PDB entry. In these cases, it is often sufficient just to use the average structure. If no average structure is available, a single structure is often chosen arbitrarily from the ensemble, typically the first structure. Unlike the case with X-ray crystallography, there is not yet any consensus measurement that is used to assess quality of NMR structures.

Currently, there is no agreed-upon way to compare the quality of crystal and solution structures. Crystal structures are generally more precise; however, since most NMR structures are solved in solution state, they may be closer to the native state than crystal structures.

11.2.3 How to View 3D Structures

In many cases, the structures you have selected were derived from proteins similar but not identical to the target protein. One approach would be to use these as template structures to build homology models and to examine the homology models only, without first looking at the template structures. However, we believe it is faster, simpler, and more accurate to look first at the template structures using a molecular graphics program that can show the structure together with an integrated representation of the sequence-to-structure alignment. Cn3D and SRS 3D are two useful programs.

Cn3D [27] (http://www.ncbi.nlm.nih.gov/Structure/CN3D/cn3d.shtml) is integrated into the BLAST and CDD servers at the NCBI site, making it easy to access and use. Cn3D is also easy to install, intuitive to use, and provides an integrated

view of structure together with sequence information, helping the user to navigate between sequence and structure (fig. 11.2A). After running BLAST on a target sequence, Cn3D lets you choose any matching structure and immediately see how the target sequence aligns onto the 3D structure, with a coloring scheme that highlights where the target sequences differ from the chosen structure.

The SRS 3D Viewer [28] is part of the SRS 3D server at the European Bioinformatics Institute (EBI) site (http://srs3d.ebi.ac.uk). Similar to Cn3D, SRS 3D is intuitive and easy to use. It enables the user to easily find all related structures, select one, and immediately see where the target sequence aligns onto the 3D structure (fig. 11.2B). Key differences between Cn3D and SRS 3D are that SRS 3D shows sequence features from the UniProt, InterPro, and PDB entries of the target sequence and enables users to easy map these features onto the structures (fig. 11.2C). SRS 3D also has a richer set of mouse and keyboard commands to help navigate and select parts of the sequence and structure. However, a drawback of the SRS 3D Viewer is that it can be difficult to install on some computers. SRS 3D is a commercial product but can be used freely via the Web to view any 3D structures in the PDB.

There are many other molecular graphics programs available; two of the most popular are Chime (http://www.mdli.com/products/framework/chime/index.jsp) and RasMol [29] (http://www.umass.edu/microbio/rasmol/index2.htm); however, Cn3D and SRS 3D are our first choice for gaining an initial view of structures because of their ability to present sequence and structure in such an integrated way.

In cases where no available structure has exactly the same sequence as the target protein, it will eventually be useful to calculate one or more homology models, especially when you are interested in looking at binding sites. For viewing homology models, two useful programs are DeepView [30] (http://www.expasy.org/spdbv/) for models from Swiss-Model and Chimera [31] (http://www.cgl.ucsf.edu/chimera/) for ModBase models. Chimera has a sophisticated multiple-sequence alignment (MSA) viewer, and a rich command set. However, it has a steep learning curve and has problems installing on some computers.

11.3 FROM STRUCTURE TO FUNCTION

So, having found all structures related to your target sequence and having selected the most appropriate ones, how can you use this information in practical ways to help to identify or validate a target sequence?

Before we can answer this question, it is worth reiterating the need for healthy caution in using 3D structural information. When you look at a static, concrete, and colorful model in a molecular graphics program, it is all too easy to forget that you are looking not at the molecule you are interested in but only at a *model* of the molecule.

If you are using homology models, you need to be especially cautious in the conclusions you draw. The accuracy and usefulness of homology models will depend partly on how closely the template structure you are using matches to the target sequence [32]. Clearly the larger the difference in sequence, the more caution you need in using your model. Even single-point mutations can cause dramatic changes to the protein structure.

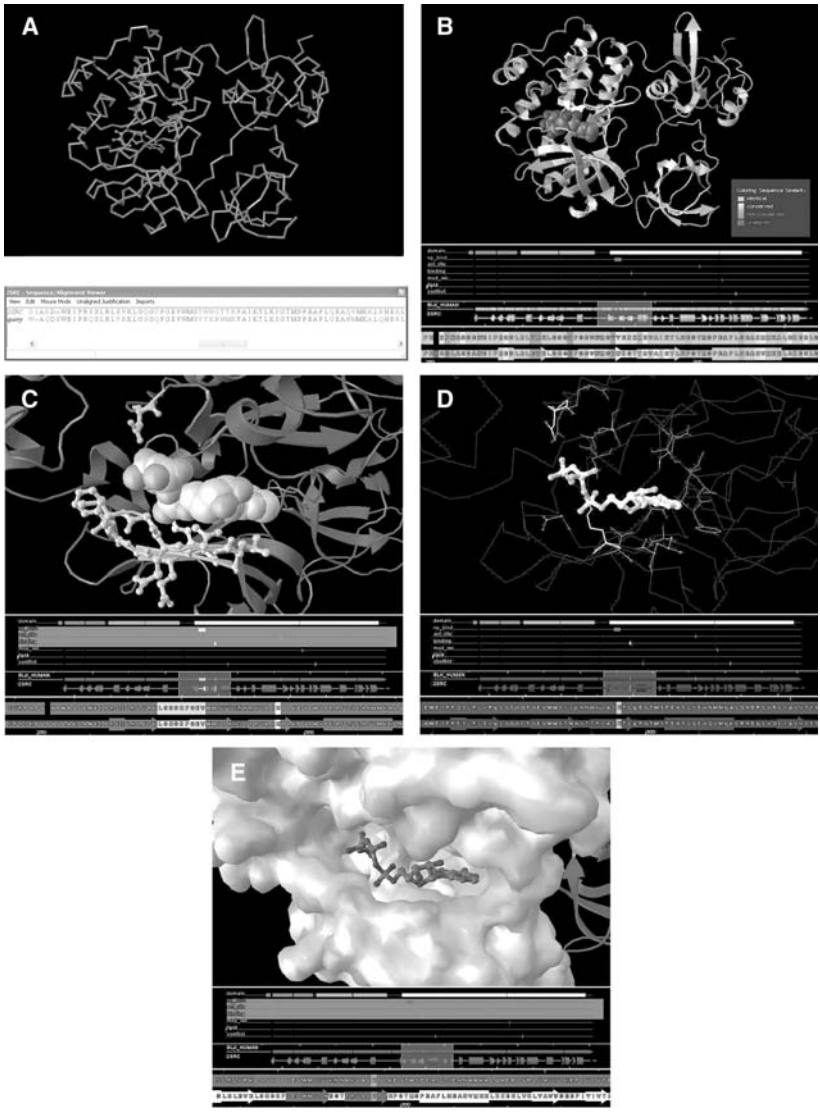


FIGURE 11.2 (See color insert) Different views of a target sequence (UniProt P51451) aligned onto a structure (PDB 2SRC, 63% sequence identity). By default, the Cn3D program (11.2A) colors residues by sequence identity. In contrast, default coloring used by SRS 3D (11.2B) is a scale of sequence similarity. SRS 3D also shows UniProt or InterPro features of the target sequence. In 11.2C, the UniProt features for binding site have been used to highlight the binding site. In 11.2D, another view of the binding site is shown by highlighting any residue that has at least one atom within 4 Å of any ligand atom. In SRS 3D, this view can be created in one step by selecting the "Binding Site" style. In 11.2E, a solvent-accessible surface has been calculated and clearly showing the binding cavity. Bindings site residues are often highly conserved, as in this case where the overall sequence identity between target sequence and structure is 63% compared to 94% identity in the binding site (only 1 of the 16 residues that define the binding differs).

However, even if you are using structures with the same sequence as the target sequence, you need to remember that these structures are also just models. In reality proteins are dynamic and flexible, and some parts of a protein are much more dynamic and flexible than others. These regions may be hinges, which allow secondary structure elements or entire domains to move relative to each other, or they may be loops that extend into the solution and can flip. Often, some of these movements are relevant to the function of the protein (e.g., the protein may have quite different structural states with or without a ligand).

11.3.1 Using Structures in the Lab

Keeping in mind that protein structures are models, how can these models help? First, simply having a model can greatly help to organize your data and ideas about a target protein. You have a frame of reference, a place where you can put these data, grouping together, for example, data about a given domain or binding site. Seeing where domains, binding sites, conserved sequence motifs, intron boundaries, single nucleotide polymorphisms (SNPs), and posttranslational modifications occur in the 3D structure can help to clarify and organize information about a target sequence.

Not only does the model help to organize your data, but it also helps to propose new hypotheses about how a target protein may be affected by such things as point mutations or by the presence of an analog to the natural ligand.

For example, many scientists involved in designing primers and making constructs find it useful to check a structure occasionally to see which regions correspond to the constructs they make. Does the construct begin or end in the middle of a secondary structure element? Does it include residues involved in the binding site?

Scientists examining polymorphisms find it useful to map these onto structures. If an SNP results in mutation in a loop region well away from the binding site, it is less likely to have an effect than a mutation to a binding-site residue. Similarly, for scientists interested in understanding splice variants, it is useful to see which parts of the structure are affected in a given variant.

For these uses, it is crucial to be able to see an integrated view of sequence and structure as in Cn3D, SRS 3D, DeepView, and Chimera. It is also very useful to easily access sequence features, as in SRS 3D.

11.3.2 Finding Binding Sites

A key to gaining insight into function is to identify the binding sites, that is, the dedicated 3D sites in the protein where other molecules bind. Typically, as part of its biological role, a protein will bind to several other proteins via comparatively large but flat binding surfaces and will bind to several small molecules (e.g., receptor agonists or antagonists) via pronounced cavities that provide enough physicochemical interactions to sufficiently stabilize the complex. In proteins that function as enzymes, the active site is usually a cavity or cleft, since the enzymatic activity is achieved by stabilizing the transition state of the reaction. This stabilization requires a precise geometric orientation of the reaction partners while shutting out external interferences and, thus, again requires enough geometry-defining interaction sites to be provided by the protein.

Specifically, given a structure, what methods are available to identify its binding sites and the molecules that bind there?

11.3.2.1 Using Existing Annotations

For most proteins known to be pharmaceutically relevant targets, the residues involved in and important for interactions with other (bio-)molecules have already been elucidated, either from a known 3D structure of the protein in complex with its interaction partner or a more stable analogue, by sequence homology to a known structure, or via experimental methods such as mutation studies. Increasingly, this information is available not just in the literature but also in databases as sequence annotations (e.g., the BINDING, METAL, SITE or ACT_SITE records in UniProt entries or SITE records in PDB entries). Several molecular graphics programs are able to display PDB SITE. However, SRS 3D is currently unique in its ability to easily map sequence features from UniProt, InterPro, and PDB onto 3D structures. Especially useful is the ability to transfer these sequence features from a target sequence and map them onto any matching structure, greatly extending the range of features that can be accessed.

For example, the target sequence used in figure 11.2 has ACT_SITE and BINDING sequence features in the UniProt entry. There is no structure determined for this sequence yet; these features were derived by similarity to other known structures with the same fold. In figure 11.2C we used these features to highlight the binding site.

11.3.2.2 Using Structures Directly

Independent of whether binding site annotations are available for the target sequence or matching structures, it is often informative to directly inspect any matching structures to see where binding sites may be. Several different methods can be used.

Probably the quickest and simplest method is illustrated in figure 11.2D. Here, we identify a binding site residue as one in which at least one atom is near (e.g., within 4 Å) any ligand atom.

To gain more insight into what might bind to a binding site, a next step could be to calculate and display the solvent-accessible surface and color it by various properties such as charge and hydrophobicity. This step can help to obtain a quick overview of the structure and to locate cavities where small molecules may bind (see figure 11.2E). There are different methods for calculating solvent-accessible surfaces; most molecular graphics programs use basic methods designed to give users a quick insight into how the surface might look (e.g., SRS 3D [13], DeepView [30], Chimera [31], and Chime [32] [http://www.mdli.com/products/framework/chime/index.jsp]). However, more accurate calculation methods and options are available from dedicated programs; Grasp [33] (http://trantor.bioc.columbia.edu/ grasp/) is probably the best known.

11.3.2.3 Using Sequence Profiles

A useful approach to searching for possible binding sites is to look at sequence profile information from MSAs and to map this information onto 3D structures.The

basis behind this group of methods is the observation that binding site residues are normally more strongly conserved than other residue positions.

These methods are particularly useful when not much is known about the protein or for structures that have novel folds. However, the methods can also be applied even if the protein is well characterized, as it may find potential binding sites that have not yet been noticed.

Several methods are available for mapping conservation values onto 3D structures and viewing the results. STING Millenium [34] is perhaps the best known. This approach can be informative; however, it is difficult to decide what level of conservation is significant just by looking at these profiles.

Fortunately, in the past 10 years this approach has been developed to yield a number of automatic methods for predicting functional sites. The pioneer Evolutionary Trace method [35] (http://www-cryst.bioc.cam.ac.uk/~jiye/evoltrace/evoltrace.html) maps phylogenetic information onto structures and often reveals accurate information about functional regions. Many similar approaches have since been devised and applied in various contexts [36,37], and several are available over the Internet (e.g., JEvTrace [38], http://www.cmpharm.ucsf.edu/~marcinj/JEvTrace/). See Campbell et al. [39] for a timely review.

Related approaches also probe more deeply into function, suggesting residues involved in specificity by highlighting those that are conserved only in part of a sequence family (e.g., PINTS [40], http://pints.embl.de and SDPpred [41], http://math.genebee.msu.ru/~psn/).

11.3.2.4 Using Structure Patterns

Another possibility is to search for constellations of amino acids that can confirm a similar function even when the proteins containing them adopt completely different folds. Probably the best known example of this type of similarity is the serine protease catalytic triad (Ser, His, Asp), which occurs in at least 10 different folds. Several approaches have been developed to search for these, including PINTS [42] (http://pints.embl.de), the Catalytic Site Atlas [43] (http://www.ebi.ac.uk/thornton-srv/databases/CSA), and Spasm/Rigor [44]. For some protein structures these similarities can be very illuminating as to function by either finding a convergently evolved site on a new fold [45] or confirming a prediction of function based on a weak similarity to a known fold [46,47].

11.3.3 Finding Function and Improving MSA

Using structural alignment methods, it is sometimes possible to find relationships between proteins that cannot be determined from sequence alone. In such cases, structural alignments can improve MSAs of the family of related sequences and reveal functional roles of the target protein that are not apparent from the sequence alone.

It is generally known that many proteins with similar structure have no similarity of sequence. However, the extent to which this is true is not yet so widely realized: most pairs of proteins that share a common fold have no detectable sequence similarity [24]. The consequence is that structural alignments can very often find similarities between proteins that are not detectable from sequence alone.

If the structure you are considering is already in the PDB, then these similarities can be found in a number of databases including SCOP [48] (http://scop.mrc-lmb.cam.ac.uk/scop/index.html), CATH [49] (http://www.biochem.ucl.ac.uk/bsm/cath/cath.html), and FSSP/Dali [50] (http://ekhidna.biocenter.helsinki.fi/dali/). However, if the structure is not yet in the PDB, you will need to perform your own structure search to compare your structure to all other known structures. In a broad sense, structure searching is similar to sequence searches; however, the methods used are very different and quite a bit slower. Some standard methods are Dali [51] (http://ekhidna.biocenter.helsinki.fi/dali/), VAST [52] (http://www.ncbi.nlm.nih.gov/Structure/VAST/vastsearch.html), and SSM [53] (http://www.ebi.ac.uk/msd/Services.html).

If the new structure matches previously known structures, it can then be useful to examine the family of structures that share this fold to see what range of functions this fold can have and if the similarities can shed light on the possible function of the new structure.

Many folds are promiscuous "superfolds" that perform many different functions [54], whereas other folds appear to perform only one function. Complicating matters slightly, even some superfolds still show some similarity in binding site location [55].

These additional structures can then be used to greatly increase the information contained in the MSA of the family of related sequences. The MSA can be a critical piece of information about a sequence, as it contains a wealth of data. In addition, several programs such as T-COFFEE [56] can directly use structural alignment information to further improve MSAs.

The improved MSAs can then help in deciding the function of the target sequence. High similarity in sequences usually implies similar function; however, this is not always the case. There are important exceptions when, for example, the target sequence may lack some key functional residues. The MSA can help to detect these cases. At present, there are no general rules to assessing when structural similarity implies a similarity in function, but certain key insights (e.g., conservation of active site residues or unusual structural features unlikely to arise by chance) can provide support [57,58].

11.3.4 Assessing Druggability

Another use of structures is in assessing a group of candidate target proteins to determine which is most likely to be "druggable." By druggable, we mean that a molecule can be found that can be taken orally, will reach the target protein, and will have the desired effect with no side effects. While this question can only be decided definitively by experiment, examining structures to choose which experiments are most likely to succeed may save considerable time and expense.

So far, proteins that interact only with other proteins or DNA/RNA have been found to be generally very poor drug targets, even if other aspects (such as a pivotal role in a pathway that the drug should control) would suggest them as suitable targets. Conversely, the binding sites of proteins that have a small natural ligand (e.g., hormone receptors), as well as the active sites of enzymes, are usually good candidates for small drug molecules that can pass the various barriers from gut to cell.

Thus, one criterion for good drug targets is having a pronounced cavity that can fit small molecules.

In addition to requiring that a target protein simply have a cavity, we can go one step further and examine this cavity in detail to determine if it is likely to accept a drug molecule. Such an approach is based on the concept originally known as Lipinski's Rule of Five [59], a commonly applied first rule of thumb for assessing whether a compound has the potential to become a drug. The binding site can be seen as an imprint of the compounds it binds; its size and the number and location of lipophilic, charged, and H-bond interaction partners it provides determine what a ligand should look like. Therefore, to determine the likelihood of a protein-binding site to bind a druglike compound, you can apply the "inverse rule of five." This rule is similar to Lipinski's, but it applies the size, lipophilicity, and H-bond donors and acceptors of the binding site.

11.3.5 Docking

Having identified a potential target protein, the ultimate proof that the target is valid is that a viable drug is found against it. Structures often play a key role in the drug-discovery process, so much so that several companies, such as StructuralGenomiX (http://www.stromix.com), have based their business model on using structures to accelerate drug discovery. It is outside the scope of this book to discuss screening in detail, however, we briefly mention the role of structures in the screening process.

At the beginning of the screening process, an analysis of the protein-binding site and natural ligands can help to determine properties of a potential ligand and thus to preselect a set of compounds for initial screening. If a compound is known to bind to the desired drug-binding site, then its overall physiochemical properties as well as its interactions can be taken as exemplary for the searched artificial ligand. The interaction geometry may best be analyzed if a structure of the protein-ligand complex has been solved. In absence of an experimental structure, one will resort to models as may be generated by docking methods [60–62]. If more than one ligand is known, a comparison of the different compounds and their respective binding constants (e.g., using QSAR [63–66]) helps to highlight features that are positive or detrimental for ligand binding. In the case of receptors that have agonists as well as antagonists, care has to be taken to also elucidate the difference in their interaction patterns to be able to design the desired type of actor.

In the absence of a known natural ligand of the target protein, one can resort to applying the aforementioned analysis to known ligands for homologous proteins. In this case, the ligand analysis needs to be complemented by a comparison of the homologous binding sites to infer how the ligands of the target protein will differ from those of the homologous proteins. Not only must differences to homologous proteins be taken into consideration to ensure sufficient strength in binding, but such differences must be exploited to avoid side effects.

The methods currently used for docking studies are not applicable to homology models. However, recent methods have been developed to combine information from homology modeling with QSAR ligand data to dock ligands into homology protein models (e.g., DragHome [67]).

11.3.6 Comparing Structures and Binding Sites

A crystal structure is frozen into one state, but, in favorable cases, several different structures are available for the target protein. For example, one structure may be a complex with the native ligand bound, another structure may have no ligand, and yet another may have a synthetic analog of the native ligand. In such cases, it can be interesting to superimpose and compare the different structures to study the changes that occur in these different states, especially changes in the binding site.

Another interesting analysis of multiple structures is to compare the binding site structure of the target protein with all other similar binding sites. Such a comparison should mainly rely on physicochemical properties of the binding sites (charges, H-bond interaction partners, etc.), since these are the determining factors for the strength of binding. A tool that is specialized in such property-based comparisons of cavities is CavBase/Relibase+ [68,69] (http://www.ccdc.cam.ac.uk). This tool can offer insights that may help the screening process. However, potentially even more interesting is the possibility that such analysis can help screen not only for specificity but also for selectivity.

Both specificity and selectivity are important features of a drug, since they ensure that the drug will have only the desired effect and will avoid side effects. Although these terms are often used synonymously, they refer to two distinct features: selectivity is concerned with site of action (e.g., a defined cell type or protein); specificity is concerned with the kinds of action at a site (e.g., agonistic or antagonistic). A good drug target should allow the design of a specific and selective drug; the binding site has to provide distinct features, significantly differing from other similar binding sites, which can be exploited to ensure preferential binding to the drug target. Furthermore, if the drug target may elicit different kinds of action, the agonistic and antagonistic reactions must be triggered by significantly different interactions.

11.4 DISCUSSION

The experimental method of structure determination is continuing to improve, thus increasing the number of structures appearing per year. Whereas sequencing went through a major milestone in the year 2000 with the human genome completion, no equivalent milestone has yet been reached with structures. The closest milestone on the horizon is set by some structural genomics initiatives, to have at least one structure for each fold domain in human. This milestone is clearly a long way off; some estimate that today we may have covered half of all folds. Unfortunately, some folds are much harder than others to solve, and some are perhaps impossible (membrane proteins, very large flexible proteins), so this goal will probably never be achieved.

Each structure contains a wealth of information, and the more than 30,000 structures that we have now in the PDB has created a new problem: how to deal with all these data effectively. The field of structural bioinformatics is more dynamic than ever; new methods are tried and developed all the time. In this chapter we presented a selection of the currently available methods, mostly those we know are already well-developed enough to be accessible, not just by experts but by the wider

audience of biochemists and molecular biologists. It is our belief that many of these scientists could benefit substantially in their research if they knew how to take advantage of some of these tools.

The wealth of structure data means it is now possible to ask questions that we could not hope to answer previously. Earlier, we had so few data that we were practically forced to ask only "reductionist" questions focusing on one structure, one protein at a time. With the wealth of data now available we can dare to ask systemic questions and hope to find answers. Given that we have multiple structures for one protein bound to different ligands, how can we use all this information to improve screening for more specific drugs? Given that we have many structures for groups of similar proteins, how can we use all this information to find more selective drugs? If we know the structures for most enzymes in a pathway, can we use this information to decide which enzyme in the pathway is the best to target? How can we combine data on multiple structures with screening data to improve selectivity and specificity?

Preliminary studies along these lines have begun to appear in the literature, particularly for some of the pharmaceutically relevant target classes [70–74]. We expect more.

In summary, while it is clear that structure can sometimes give critical insight into understanding the function of a target, it is far from decided how best to do this.

REFERENCES

1. Kanz, C., P. Aldebert, N. Althorpe, W. Baker, A. Baldwin, K. Bates, P. Browne, et al. 2005. The EMBL nucleotide sequence database. *Nucleic Acids Res* 33, Database Issue:D29–33.
2. Bairoch, A., R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, et al. 2005. The Universal Protein Resource (UniProt). *Nucleic Acids Res* 33, Database Issue:D154–9.
3. Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. 2000. The Protein Data Bank. *Nucleic Acids Res* 28:235–42.
4. Schafferhans, A., J. E. W. Meyer, and S. I. O'Donoghue. 2003. The PSSH database of alignments between protein sequences and tertiary structures. *Nucleic Acids Res* 31:494–8.
5. Watson, J. D., and F. H. Crick. 1953. Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature* 171:737–8.
6. Kendrew, J. C., R. E. Dickerson, B. E. Strandberg, R. G. Hart, D. R. Davies, D. C. Phillips, and V. C. Shore. 1960. Structure of myoglobin. *Nature* 185:422–7.
7. Altschul, D., A. M. Lesk, A. C. Bloomers, and A. Klug. 1987. Correlation of co-ordinated amino acid substitutions with function in viruses related to tobacco mosaic virus. *J Mol Biol* 193:693–707.
8. Schwede, T., J. Kopp, N. Guex, and M. C. Peitsch. 2003. SWISS-MODEL: An automated protein homology-modeling server. *Nucleic Acids Res* 31:3381–5.
9. Bateman, A., E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. R. Eddy, S. Griffiths-Jones, K. L. Howe, M. Marshall, and E. L. Sonnhammer. 2002. The Pfam protein families database. *Nucleic Acids Res* 30:276–80.

10. Letunic, I., R. R. Copley, S. Schmidt, F. D. Ciccarelli, T. Doerks, J. Schultz, C. P. Ponting, and P. Bork. 2004. SMART 4.0: Towards genomic data integration. *Nucleic Acids Res* 32, Database Issue:D142–4.
11. Mulder, N. J., R. Apweiler, T. K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bradley, et al. 2005. InterPro, progress and status in 2005. *Nucleic Acids Res* 33, Database Issue:D201–5.
12. Marchler-Bauer, A., J. B. Anderson, P. F. Cherukuri, C. DeWeese-Scott, L. Y. Geer, M. Gwadz, S. He, et al. 2005. CDD: A Conserved Domain Database for protein classification. *Nucleic Acids Res* 33, Database Issue:D192–6.
13. O'Donoghue, S. I., J. E. W. Meyer, A. Schafferhans, and K. Fries. 2004. The SRS 3D module: Integrating sequence, structure, and annotation data. *Bioinformatics* 20:2476–8.
14. Pieper, U., N. Eswar, H. Braberg, M. S. Madhusudhan, F. P. Davis, A. C. Stuart, N. Mirkovic, et al. 2004. MODBASE, a database of annotated comparative protein structure models, and associated resources. *Nucleic Acids Res* 32, Database Issue:D217–22.
15. Bates, P. A., L. A. Kelley, R. M. MacCallum, and M. J. Sternberg. 2001. Enhancement of protein modeling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. *Proteins*, Suppl. no. 5:39–46.
16. Rost, B. 1997. Protein structures sustain evolutionary drift. *Fold Des* 2:S19–24.
17. Godzik, A. 2003. Fold recognition methods. *Methods Biochem Anal* 44:525–46.
18. Ginalski, K., A. Elofsson, D. Fischer, and L. Rychlewski. 2003. 3D-Jury: A simple approach to improve protein structure predictions. *Bioinformatics* 19:1015–8.
19. Lundstrom, J., L. Rychlewski, J. Bujnicki, and A. Elofsson. 2001. Pcons: A neural-network-based consensus predictor that improves fold recognition. *Protein Sci* 10:2354–62.
20. Kurowski, M. A., and J. M. Bujnicki. 2003. GeneSilico protein structure prediction meta-server. *Nucleic Acids Res* 31:3305–7.
21. Tramontano, A., and V. Morea. 2003. Assessment of homology-based predictions in CASP5. *Proteins* 53, Suppl. no. 6:352–68.
22. Kinch, L. N., J. O. Wrabl, S. S. Krishna, I. Majumdar, R. I. Sadreyev, Y. Qi, J. Pei, H. Cheng, and N. V. Grishin. 2003. CASP5 assessment of fold recognition target predictions. *Proteins* 53, Suppl. no. 6:395–409.
23. Marti-Renom, M. A., A. C. Stuart, A. Fiser, R. Sanchez, F. Melo, and A. Sali. 2000. Comparative protein structure modeling of genes and genomes. *Annu Rev Biophys Biomol Struct* 29:291–325.
24. Rost, B. 1999. Twilight zone of protein sequence alignments. *Protein Eng* 12:85–94.
25. Rhodes, G. 2000. *Crystallography made crystal clear: A guide for users of macromolecular models*. 2nd ed. New York: Academic.
26. Chalaoux, F-R., S. I. O'Donoghue, and M. Nilges. 1999. Molecular dynamics and accuracy of NMR structures: Effects of error bounds and data removal. *Proteins* 34:453–63.
27. Wang, Y., L. Y. Geer, C. Chappey, J. A. Kans, and S. H. Bryant. 2000. Cn3D: Sequence and structure views for Entrez. *Trends Biochem Sci* 25:300–2.
28. Fries, K., and S. I. O'Donoghue. 2002. Navigating around the building blocks of life. *Advanced Imaging* 17:18–9, 39.
29. Sayle, R. A., and E. J. Milner-White. 1995. RasMol: Biomolecular graphics for all. *Trends Biochem Sci* 20:374–6.
30. Guex, N., and M. C. Peitsch. 1997. SWISS-MODEL and the Swiss-PdbViewer: An environment for comparative protein modeling. *Electrophoresis* 18:2714–23.

31. Pettersen, E. F., T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin. 2004. UCSF Chimera—a visualization system for exploratory research and analysis. *J Comput Chem* 25:1605–12.
32. Sanchez, R., U. Pieper, F. Melo, N. Eswar, M. A. Marti-Renom, M. S. Madhusudhan, N. Mirkovic, and A. Sali. 2000. Protein structure modeling for structural genomics. *Nat Struct Biol* 7, Suppl.:986–90.
33. Nicholls, A., K. A. Sharp, and B. Honig. 1991. Protein folding and association: Insights from the interfacial and thermodynamic properties of hydrocarbons. *Proteins* 11:281–96.
34. Neshich, G., R. C. Togawa, A. L. Mancini, P. R. Kuser, M. E. Yamagishi, G. Pappas, Jr., W. V. Torres, et al. 2003. STING Millennium: A web-based suite of programs for comprehensive and simultaneous analysis of protein structure and sequence. *Nucleic Acids Res* 31:3386–92.
35. Lichtarge, O., H. R. Bourne, and F. E. Cohen. 1996. An evolutionary trace method defines binding surfaces common to protein families. *J Mol Biol* 257:342–58.
36. Aloy, P., E. Querol, F. X. Aviles, and M. J. Sternberg. 2001. Automated structure-based prediction of functional sites in proteins: Applications to assessing the validity of inheriting protein function from homology in genome annotation and to protein docking. *J Mol Biol* 311:395–408.
37. Landgraf, R., I. Xenarios, and D. Eisenberg. 2001. Three-dimensional cluster analysis identifies interfaces and functional residue clusters in proteins. *J Mol Biol* 307:1487–502.
38. Joachimiak, M. P., and F. E. Cohen. 2002. JEvTrace: Refinement and variations of the evolutionary trace in JAVA. *Genome Biol* 3:RESEARCH0077.
39. Campbell, S. J., N. D. Gold, R. M. Jackson, and D. R. Westhead. 2003. Ligand binding: Functional site location, similarity and docking. *Curr Opin Struct Biol* 13:389–95.
40. Hannenhalli, S. S., and R. B. Russell. 2000. Analysis and prediction of functional sub-types from protein sequence alignments. *J Mol Biol* 303:61–76.
41. Kalinina, O. V., A. A. Mironov, M. S. Gelfand, and A. B. Rakhmaninova. 2004. Automated selection of positions determining functional specificity of proteins by comparative analysis of orthologous groups in protein families. *Protein Sci* 13:443–56.
42. Stark, A., and R. B. Russell. 2003. Annotation in three dimensions. PINTS: Patterns in Non-homologous Tertiary Structures. *Nucleic Acids Res* 31:3341–4.
43. Porter, C. T., G. J. Bartlett, and J. M. Thornton. 2004. The Catalytic Site Atlas: A resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Res* 32, Database Issue:D129–33.
44. Kleywegt, G. J. 1999. Recognition of spatial motifs in protein structures. *J Mol Biol* 285:1887–97.
45. Stark, A., A. Shkumatov, and R. B. Russell. 2004. Finding functional sites in structural genomics proteins. *Structure* 12:1405–12.
46. Lorentzen, E., E. Pohl, P. Zwart, A. Stark, R. B. Russell, T. Knura, R. Hensel, and B. Siebers. 2003. Crystal structure of an archaeal class I aldolase and the evolution of (betaalpha) 8 barrel proteins. *J Biol Chem* 278:47253–60.
47. Sanishvili, R., A. F. Yakunin, R. A. Laskowski, T. Skarina, E. Evdokimova, A. Doherty-Kirby, G. A. Lajoie, et al. 2003. Integrating structure, bioinformatics, and enzymology to discover function: BioH, a new carboxylesterase from Escherichia coli. *J Biol Chem* 278:26039–45.
48. Andreeva, A., D. Howorth, S. E. Brenner, T. J. Hubbard, C. Chothia, and A. G. Murzin. 2004. SCOP database in 2004: Refinements integrate structure and sequence family data. *Nucleic Acids Res* 32, Database Issue:D226–9.

49. Pearl, F. M., C. F. Bennett, J. E. Bray, A. P. Harrison, N. Martin, A. Shepherd, I. Sillitoe, J. Thornton, and C. A. Orengo. 2003. The CATH database: An extended protein family resource for structural and functional genomics. *Nucleic Acids Res* 31:452–5.
50. Holm, L., and C. Sander. 1998. Touring protein fold space with Dali/FSSP. *Nucleic Acids Res* 26:316–9.
51. Holm, L., and C. Sander. 1995. Dali: A network tool for protein structure comparison. *Trends Biochem Sci* 20:478–80.
52. Gibrat, J. F., T. Madej, and S. H. Bryant. 1996. Surprising similarities in structure comparison. *Curr Opin Struct Biol* 6:377–85.
53. Krissinel, E., and K. Henrick. 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallogr D Biol Crystallogr* 60 (Pt 12 Pt 1): 2256–68.
54. Orengo, C. A., D. T. Jones, and J. M. Thornton. 1994. Protein superfamilies and domain superfolds. *Nature* 372:631–4.
55. Russell, R. B., P. D. Sasieni, and M. J. Sternberg. 1998. Supersites within superfolds. Binding site similarity in the absence of homology. *J Mol Biol* 282:903–18.
56. Notredame, C., D. G. Higgins, and J. Heringa. 2000. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol* 302:205–17.
57. Dietmann, S., and L. Holm. 2001. Identification of homology in protein structure classification. *Nat Struct Biol* 8:953–7.
58. Murzin, A. G. 1996. Structural classification of proteins: New superfamilies. *Curr Opin Struct Biol* 6:386–94.
59. Lipinski, C. A., F. Lombardo, B. W. Dominy, and P. J. Feeney. 1997. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv Drug Del Rev* 23:3–25.
60. Schneider, G., and H. J. Bohm. 2002. Virtual screening and fast automated docking methods. *Drug Discov Today* 7:64–70.
61. Taylor, R. D., P. J. Jewsbury, and J. W. Essex. 2002. A review of protein-small molecule docking methods. *J Comput Aided Mol Des* 16:151–66.
62. Brooijmans, N., and I. D. Kuntz. 2003. Molecular recognition and docking algorithms. *Annu Rev Biophys Biomol Struct* 32:335–73.
63. Akamatsu, M. 2002. Current state and perspectives of 3D-QSAR. *Curr Top Med Chem* 2:1381–94.
64. Selassie, C. D., S. B. Mekapati, and R. P. Verma. 2002. QSAR: Then and now. *Curr Top Med Chem* 2:1357–79.
65. Mekenyan, O. 2002. Dynamic QSAR techniques: Applications in drug design and toxicology. *Curr Pharm Des* 8:1605–21.
66. Winkler, D. A. 2002. The role of quantitative structure—activity relationships (QSAR) in biomolecular discovery. *Brief Bioinform* 3:73–86.
67. Schafferhans, A., and G. Klebe. 2001. Docking ligands onto binding site representations derived from proteins built by homology modelling. *J Mol Biol* 307:407–27.
68. Schmitt, S., M. Hendlich, and G. Klebe. 2001. From structure to function: A new approach to detect functional similarity among proteins independent from sequence and fold homology. *Angew Chem Int Ed Engl* 40:3141–4.
69. Schmitt, S., D. Kuhn, and G. Klebe. 2002. A new method to detect related function among proteins independent of sequence and fold homology. *J Mol Biol* 323:387–406
70. Sud, M., A. Schafferhans, and S. I. O'Donoghue. 2003. Sequence and ligand similarity relationships: A case study of the nuclear receptors. Paper presented at the ACS 225th national meeting, New Orleans.

71. Brown, J. R., K. K. Koretke, M. L. Birkeland, P. Sanseau, and D. R. Patrick. 2004. Evolutionary relationships of Aurora kinases: Implications for model organism studies and the development of anti-cancer drugs. *BMC Evol Biol* 4:39.
72. Vieth, M., R. E. Higgs, D. H. Robertson, M. Shapiro, E. A. Gragg, and H. Hemmerle. 2004. Kinomics-structural biology and chemogenomics of kinase inhibitors and targets. *Biochim Biophys Acta* 1697:243–57.
73. Dekker, L. V., R. H. Palmer, and P. J. Parker. 1995. The protein kinase C and protein kinase C related gene families. *Curr Opin Struct Biol* 5:396–402.
74. Xu, J., and Q. Li. 2003. Review of the in vivo functions of the p160 steroid receptor coactivator family. *Mol Endocrinol* 17:1681–92.
75. Altschul, S. F., T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res* 25:3389–402.

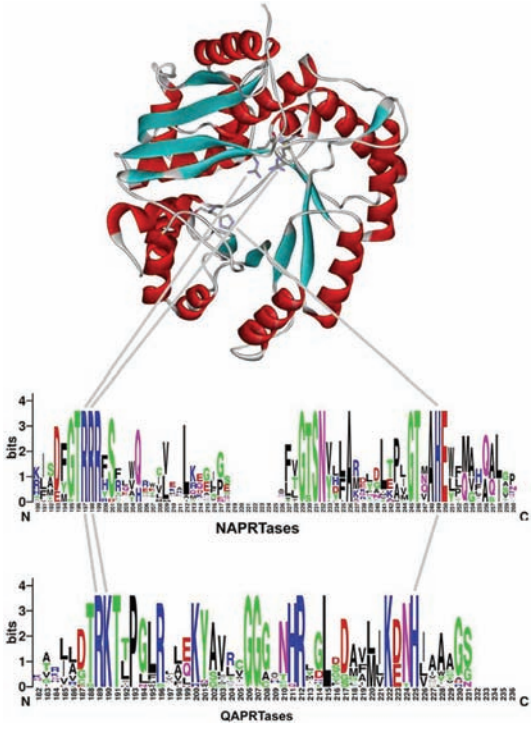


FIGURE 3.4 Integration of sequence and structure information. Two large multiple sequence alignments including residues catalytically important for two protein families have been condensed into two sequence logos. The logos describe the quinolinic acid phosphoribosyltransferase (QAPRTase) and nicotinic acid phosphoribodyltransferase (NAPRTase) families. Critical residue conservation has been mapped between the logos, and those residues have been subsequently mapped from the logos to the three-dimensional structure of yeast NAPRTase.

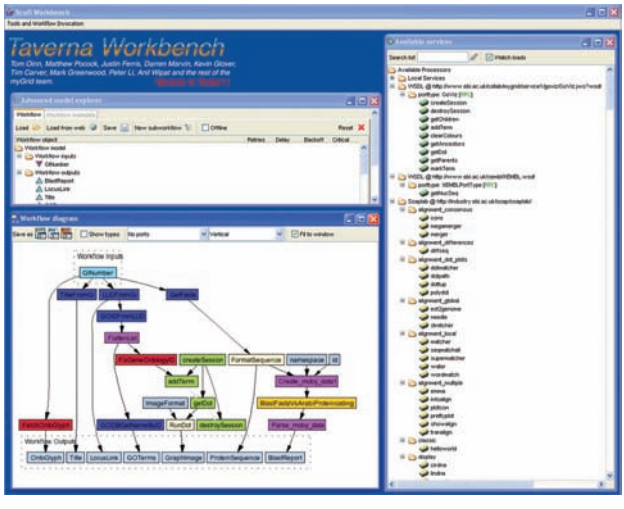


FIGURE 3.6 Graphical user interface of Taverna Workbench, a tool for the composition and execution of bioinformatics workflows. The workflows are written in a new language called Scufl (Simple conceptual unified flow language).

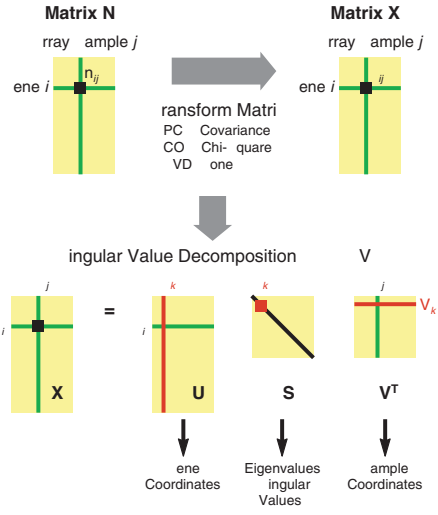


FIGURE 5.4 Data matrix transformations in advance of SVD. A gene expression matrix N will be transformed to obtain a transform matrix X, which is the one that will be actually displayed in a reduced dimension, achieved using SVD. In this figure, U_k and V_k are eigenvectors and S_k is an eigenvalue. U_k is an eigenassay (or eigenarray) and V_k is an eigengene.

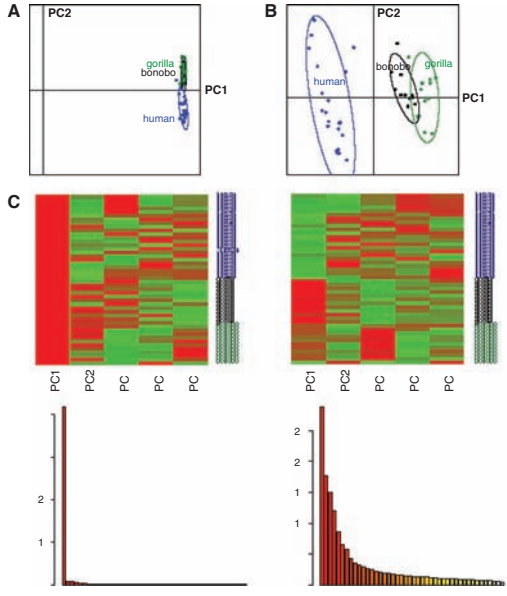


FIGURE 5.5 PCA and COA analysis of gene expression data (12,625 probesets) from fibroblasts isolated from human, gorilla, and bonobo. These figures demonstrate the difference between a PCA and COA, which ask different questions of data. PCA presents the trends in the data with the most variance. A and B show a scatter plot of the first two principal components (PC1, PC2) of a PCA and a COA, respectively. C shows a heatmap of the scores' first five principal components, where red to green is positive-to-negative ends of the axis from the PCA. It is clear that the first component, which represents the 90% of variance in the data, is not associated with samples groupings. In B, the strongest correspondences between genes and samples are analyzed. PC1 represent 19% and PC2 represents 12% of the total chi-square association between samples and gene expression profiles. In COA high chi-squares will be associated with increased gene expression in samples. Thus, if a gene is increased in expression in a many samples, there will be a high chi-square value showing this association. On the scatter plot the positive end of PC1 represents genes that are up-regulated in gorilla and down-regulated in human.

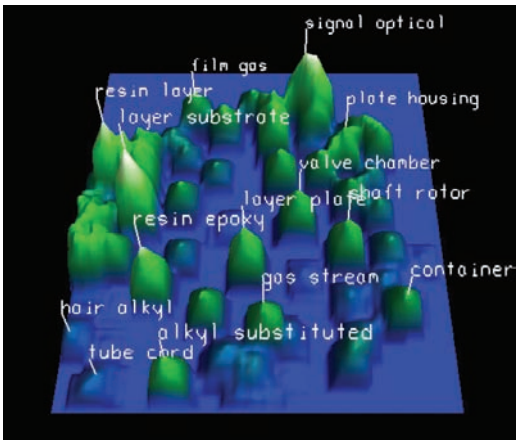


FIGURE 6.9 Cluster theme visualization with a theme map. OmniViz ThemeMap™ for 5,885 patents in the field of electronics and surface chemistry from 1999. The major themes or concepts are denoted by mountains, which provide a rapid means for seeing the represented concepts. The ThemeMap view also allows the user to build the map according to specific themes to help understand the content in particular subject areas. *Courtesy of OmniViz.*

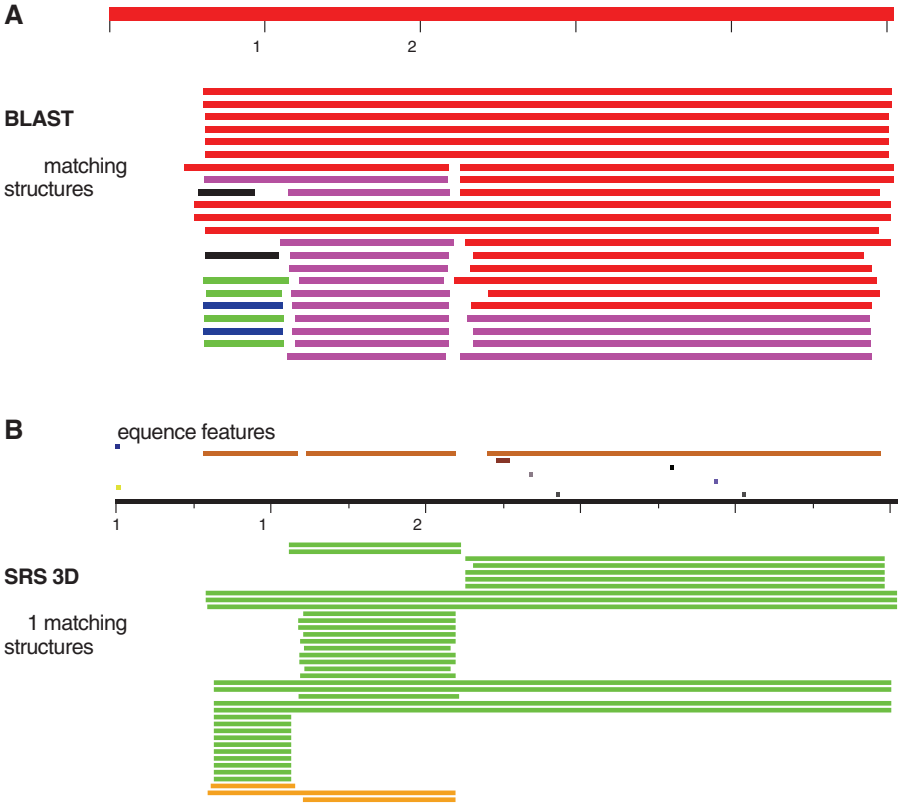


FIGURE 11.1 Shows all matching structures for the target sequence B-lymphocyte kinase (UniProt P51451) found by NCBI/BLAST75 (2A) and EBI/SRS 3D13 (2B). In both cases the target sequence is represented by the bar numbered 1-500. BLAST finds a total 339 matching structures, of which the top ranking matches are shown here, colored and ranked by alignment score (red = >200, magenta = 80–200, green = 50–80, blue = 40–50, black = <40). SRS 3D, like other profile methods, finds more matches than BLAST, in this case it finds 461 matching structures, of which the top ranking matches are shown here colored and ranked by sequence identity (green >60%, orange = 30–60%). Views like this make it very easy to see where in the target sequence a structure matches.

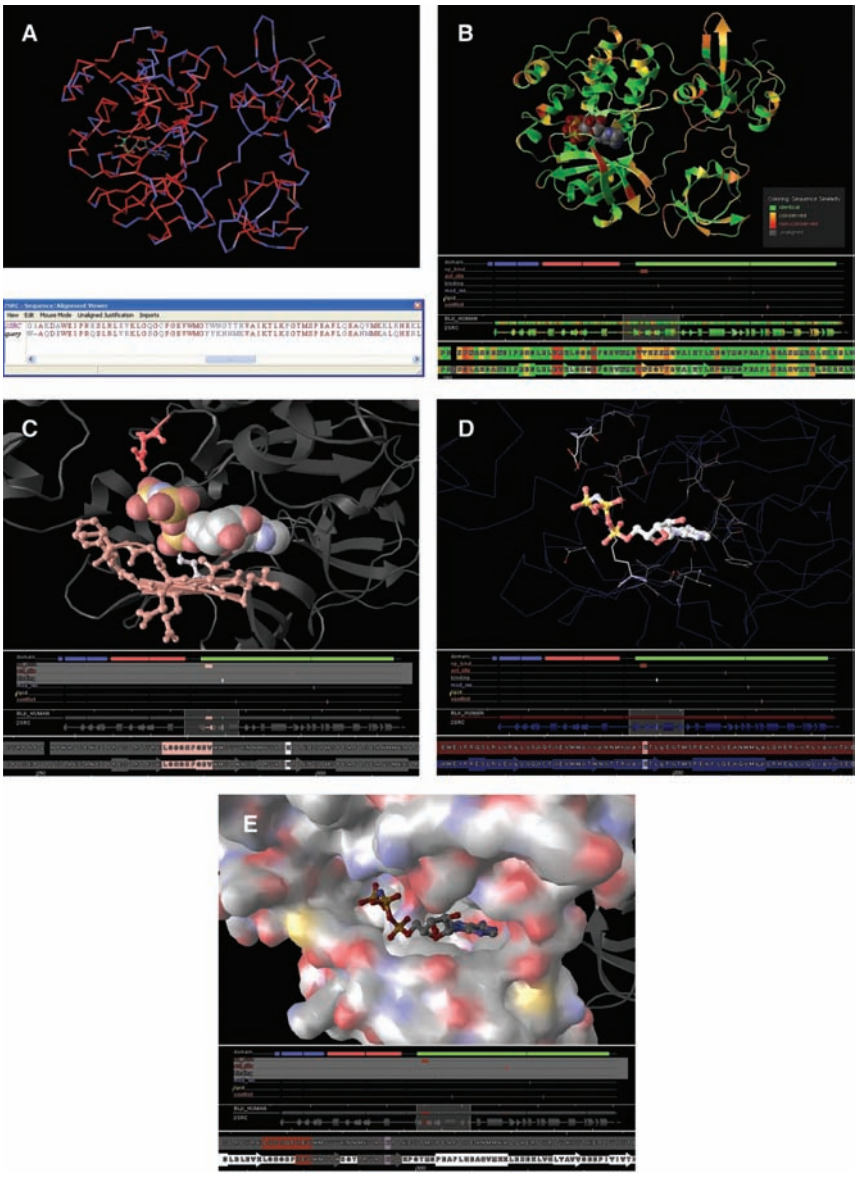


FIGURE 11.2 Shows different views of a target sequence (UniProt P51451) aligned onto a structure (PDB 2SRC, 63% sequence identity). By default, the Cn3D program (11.2A) colors residues by sequence identity (red = identical, blue = different). In contrast, default coloring used by SRS 3D (11.2B) is a scale of sequence similarity (green = identical, yellow to red = similar to non-conserved). SRS 3D also shows UniProt or InterPro features of the target sequence. In 11.2C, the UniProt features for binding site have been used to color the structure and highlight the binding site. In 11.2D, another view of the binding site is shown by highlighting any residue that has at least one atom within 4 Å of any ligand atom. In SRS 3D, this view can be created in one step by selecting the "Binding Site" style. In 11.2E, a solvent-accessible surface has been calculated and colored by atom type, clearly showing the binding cavity. Bindings site residues are often highly conserved, as in this case where the overall sequence identity between target sequence and structure is 63% compared to 94% identity in the binding site (only one of the 16 residues that define the binding is differs).

Part III

Recent Trends

12 Comparative Genomics

Viviane Siino, Bruce Pascal, and Christopher Sears
Biosift, Inc.

CONTENTS

12.1 INTRODUCTION

Genetic diversity and novel function are achieved in nature through incremental modifications. The contribution of comparative genomics to understanding the fundamental principles underlying the processes and consequences of evolutionary changes is considerable and steadfastly growing. The availability of multiple genome sequence databases and computational methods for analyzing the information has proven essential to our well-being. From fostering our understanding of infectious diseases, thus helping scientists create better drugs, vaccines, and effective methods for controlling pathogens, to identifying genes linked to specific human disorders, which has led to the identification and development of better diagnostic tools and drug therapies, comparative genomics' contribution to human health is indispensable. From a purely basic science perspective, comparative genomics has furthered our understanding of regulatory and metabolic pathways and has resulted in inferences about function. Moreover, with an ever-increasing worldwide human population,

agricultural genomics offers the promise of feeding the hungry through improved crops' resistance to pestilence or animal disease. Although still in its infancy, comparative genomics has already had an inestimable impact on the advancement of science and the betterment of life.

12.2 INFECTIOUS DISEASE

Comparative genomics technologies are helping to unravel the molecular basis of pathogenesis, host range, epidemiology, evolution, and phenotypic differences of infectious agents. One aim is to elucidate the differences between pathogenic and nonpathogenic infectious agents as well as between pathogen and host. Such research will identify molecular targets for future investigation: genes that code for pathogenicity factors or genes essential for survival in the host. The most attractive targets include those that are nonfunctional or redundant in the host as well as genes absent from the host but essential in the pathogen [1]. The genetic basis for pathogenicity can also be studied by using microarray-based comparative genomics to characterize and quantify the extent of genetic variability within natural populations at the gene level of resolution [2].

This pathogenic-specific approach has been applied to tuberculosis, a major cause of transmissible morbidity and mortality. The research has led to the identification of essential genes within the infectious organism (*Mycobacterium tuberculosis*), and it has furthered tuberculosis vaccine development by pinpointing potentially antigenic proteins as well as providing better diagnostic tools to detect infection [3]. In one instance, genomic analyses have suggested that loss of genes is part of the ongoing evolution of the slow-growing mycobacterial pathogens and could explain how the tuberculosis vaccine strain *Bacillus Calmatte Guerin* (*BCG*; perhaps the most widely used live vaccine in the world) became attenuated [4]. On comparison of the complete genomes of *M. tuberculosis H37Rv* and the vaccine strain, two major rearrangements were identified in the genome of the vaccine strain *BCG*. Knowledge of their existence will facilitate quality control of BCG vaccine lots [5].

Research for infectious diseases also seeks to identify the critical steps of host defense to infection. One recent approach has been to isolate and characterize a mouse gene, *Bcg* (*Nramp1*), which controls natural susceptibility to infection with Mycobacteria, as well as Salmonella and Leishmania and test the alleles of the human homologue for linkage with tuberculosis and leprosy [6]. Understanding the mechanism of action will enable scientists to design better drugs to mimic the action of the resistant allele.

Tropical pathogens of medical and veterinary importance, many of which are responsible for causing widespread morbidity and mortality in peoples of developing countries, are also of pressing concern. Uncovering the complete gene complement of these organisms is proving to be of immense value in the development of novel methods of parasite control, such as insecticides, antiparasitic drugs, and vaccines, as well as the development of new diagnostic tools [7]. Resistance to insecticides among mosquitoes that act as vectors for malaria (*Anopheles gambiae*) and West Nile virus (*Culex pipiens*) is frequently due to a loss of sensitivity of the insect's acetylcholinesterase enzyme to organophosphate and carbamate compounds, which

compose the majority of insectides. Recent work has shown that this insensitivity results from a single amino-acid substitution in the enzyme, which were found in 10 highly resistant strains of *C. pipiens* from tropical (Africa and Caribbean) and temperate (Europe) areas, as well as in one resistant African strain of *A. gambiae*. Identification of such mutations may pave the way for designing new insecticides [8].

12.3 HUMAN DISORDERS

One function of comparative genomics is to facilitate gene discovery for polygenic disorders such as psychiatric disorders, heart disease, diabetes, and some cancers. Comparative mapping can be used to select target regions in the human genome for large-scale association studies and linkage disequilibrium testing in clinical populations [9]. Current data from the use of these methods are found for a number of human disorders. Linkage and association studies have suggested a number of candidate loci on the short and long arms of chromosome 18 for the psychiatric disorders schizophrenia, affective disorder, and autism [10].

Diabetes research is an ideal candidate field for genomics research. The aim of an insulin-resistance disease program is to identify targets for therapeutic intervention within pathways that control glucose homeostasis. Resistance to the normal action of insulin contributes to the pathogenesis of a number of common human disorders, including type 1 (insulin-dependent) and type 2 (non-insulin-dependent) diabetes mellitus, hypertension, and the Metabolic Syndrome X. Genes regulating lipolysis are prime candidates for susceptibility toward the metabolic syndrome. Lake, Krook, and Zierath [11] surveyed the analysis of insulin-signaling pathways through comparative genomics. They described the genomics approaches that have led to definition of the critical sterol response element binding protein pathway, furthering our knowledge of the development and maintenance of insulin resistance, obesity, and diabetes. Odom et al. [13] recently used a high-throughput genomic technology to identify complex regulatory circuits by a series of transcription factors operating in the pancreas and liver. Their results demonstrate the process by which misregulation of HNF-α (Hepatocyte Nuclear Factor) expression, a key transcriptional regulatory protein found in the liver and pancreatic islets, can lead to type 2 diabetes.

Cancer research will benefit from comparative genomic analysis. It is believed that disease states are epigenetically determined and, thus, each tumor type and stage will be characterized by a gene-expression fingerprint. Genes that are differentially expressed in early tumor stages can be diagnostic of neoplastic transformation and in later stages of transition from *in situ* to invasive cancers. Such approaches have been used in breast cancer and melanoma research [13,14]. Model organisms will serve as systems for exploring the prognosis and treatment of tumors. Yeung et al. [15] showed that a rat gene *Mot1* can act as a modifier of renal tumor size.

Mammalian model organisms are key systems for disease and disorder research, but the contribution of comparative analysis with nonmammalian organisms can be equally critical. The common fruit fly has orthologs to 177 human disease genes and provides the foundation for rapid analysis of some of the basic processes involved in human disease [16]. Conserved chromosomal regions associated with complex human

phenotypes are now known in model organisms. For example, genes that encode related immunoglobulin superfamily molecules have been coordinately mapped to human chromosome 15 and to the syntenic region on mouse chromosome 9. These genes presumably are derived from gene duplications and are similar to the disease-associated genes for *Deleted in Colorectal Cancer*. This interval overlaps a genetic locus for Bardet–Biedl syndrome (BBS4) in humans, a syndrome characterized by poly/syn/brachydactyly, retinal degeneration, hypogonadism, mental retardation, obesity, diabetes, and kidney abnormalities. A detailed map of this locus in several organisms will help identify candidate genes for this disorder, and we hope it will provide a model organism for investigation of the mechanism [17]. The combination of such a comparative analysis with a visualization tool holds the potential for accelerating the process of discovery. For example, in a search for genes associated with inflammatory bowel disease (IBD), a mapping of logarithmic odds score, as illustrated in figure 12.1, which indicate the likelihood of association with the disease,

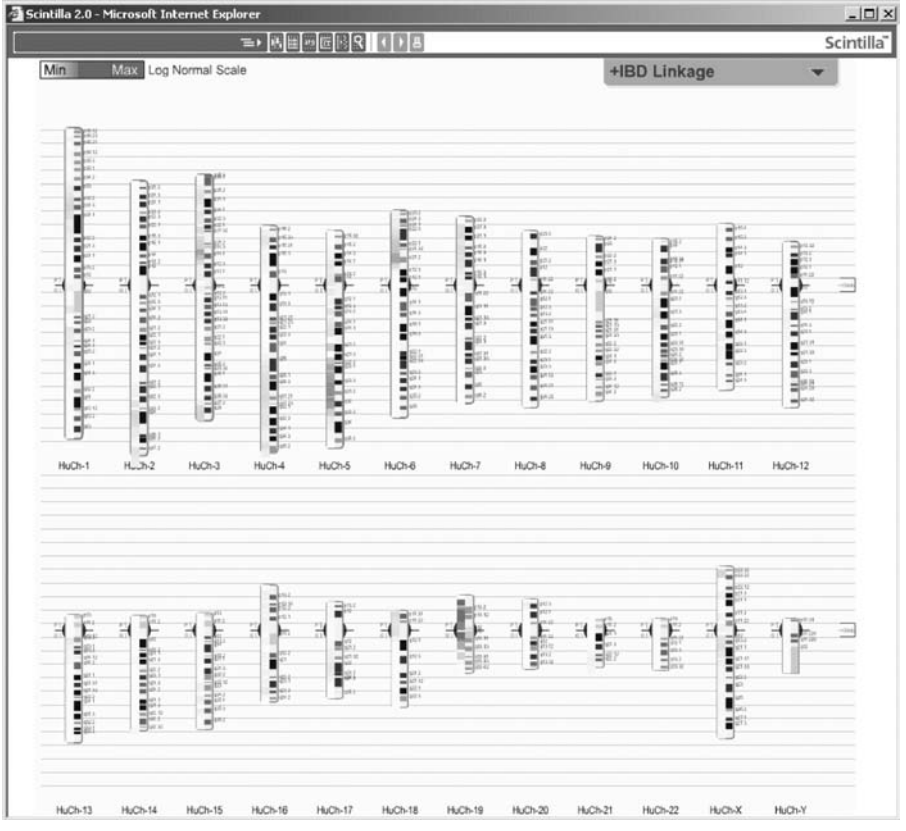


FIGURE 12.1 Mapping of logarithmic odds scores for inflammatory bowel disease against the human genome. The colored bands indicate the likelihood of association with the disease. Regions for which LOD scores are suggestive of an association with IBD are painted blue, whereas regions that are significantly associated with the disease are painted red. (Copyright © 2001 by Biosift, Inc. All rights reserved)

could accelerate the process of identifying new targets for therapeutics. In the case of IBD, improved therapeutics would impact the quality of life of an estimated 1 million individuals in the United States, which would also result in tremendous economical benefits.

12.4 EVOLUTION

Novel methods of evolutionary analysis are now possible through the availability of organisms' genomic blueprints, including detection of domain shuffling and lateral gene transfer, reconstruction of the evolutionary diversification of gene families, tracing of evolutionary change in protein function at the amino acid level, and prediction of structure–function relationships [18]. With the available high-quality draft genomes of model organisms, we are able to obtain a picture of the consequences of evolution over timescales ranging from approximately 1 billion years (human and invertebrate), 75 million years (human and mouse), and 12 to 24 million years (mouse and rat).

Using visualization, as shown in figure 12.2, homologous regions between the *S. cerevisiae* and *H. sapiens* genomes can be obtained at a glance. This multigenome view allows for the visual quantification of positive and negative selection in these genomes.

One aim of evolutionary research is the identification of a minimal set of genes that is necessary and sufficient for sustaining a functional cell. The smallest known genomes belong to Mycoplasma organisms that are adapted to a parasitic lifestyle and thus subvert the host organism's genome to complement their own. Attempts have been made to identify the minimal set of genes that is required for independent life using computational approaches or studies of deletion mutants [19]. For most essential cellular functions, multiple unrelated or distantly related proteins have evolved; only about 60 proteins, primarily those involved in translation, are common to all cellular life [20]. Approximately one-half of protein domains involved in RNA metabolism are present in most, if not all, species from all three primary kingdoms and are traceable to the last universal common ancestor [21]. An interesting hypothesis deriving from the observed complex phylogenetic patterns and for the irregular distribution of metabolic pathways is that the last common ancestor of Bacteria and Archaea contained several members of every gene family as a consequence of previous gene or genome duplications, while different patterns of gene loss occurred during the evolution of every gene family [22].

Features of the universal tree of life, such as the division of the cellular living world into three domains, have been confirmed by genome-sequencing efforts [23]. Comparative genomics has revealed that gene transfers have been frequent in genome evolution. The role of viruses in gene transfers is the subject of considerable investigation. Recent work suggests that DNA and DNA-replication mechanisms appeared first in the virus world before being transferred into cellular organisms [24].

The rat genome was published in early 2004, providing a powerful three-way comparison of mammalian genomes with human as the outlier (and mouse completing the trio). It is found that genes are not distributed randomly along the chromosomes and that there are clusters of high gene density in species with large genomes (from humans to rodents to plants) [25]. Furthermore genomic repeat elements are

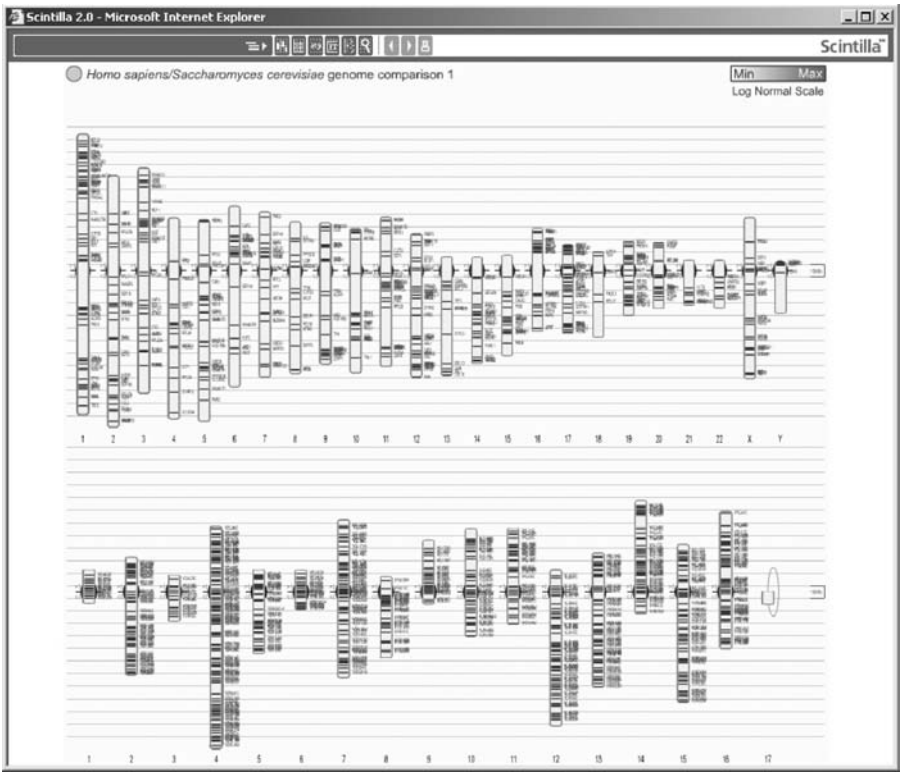


FIGURE 12.2 Genome-level homology between *H. sapiens* and *S. cerevisiae*. (Copyright © 2001 by Biosift, Inc. All rights reserved)

observed to coincide with regions of greater or lesser gene density. Poor protein-encoding regions in humans coincide with low G+C content, low short interspersed elements coverage, and high long interspersed elements coverage. By contrast to the protein coding genes, noncoding RNA genes are evenly distributed throughout the genome [26,27], but it is clear that repetitive elements within the eukaryotic genome appear to be one important engine of evolution. Researchers are working to determine whether repeat elements are attracted to particular genomic regions or whether they are preferentially preserved. Their presence today is a fossil record of the mechanism of evolution, and understanding their distribution will contribute to our understanding of population genetics and a number of genetic diseases.

In reviewing the three available mammalian genomes, no compelling evidence was found of genes arising from noncoding sequences, but numerous examples of gene duplication leading to species-specific functionality are observed [28]. These species-specific gene expansions are found to be due to genomic duplications, often in pericentromeric regions—pointing to another significant process in genome evolution.

Alternative splicing in the human, mouse, and rat genomes has been associated with increased evolutionary change [29]. Whereas most exons in the mouse, rat, and

human genomes are strongly conserved, exons that are included only in alternative splice forms (as opposed to the constitutive or major transcript form) are mostly not conserved and thus are the product of recent exon creation or loss events. While species-specific splicing events can lead to different and specialized protein products, a recent report demonstrates a striking example of convergent evolution in three different ion-channel gene families between humans and *Drosophila melanogaster* [30].

Genomic analysis is casting new light onto the mechanism by which highly specialized structures such as the human brain and nervous system evolve from precursor components. Examination of expressed sequence tag (EST) clones from the complete genome sequences of the human, fruit fly, and nematode showed that over 100 nervous-system-related genes, including genes involved in brain or neural morphogenesis, were commonly shared among these organisms, thus providing evidence at the molecular level for the existence of a common ancestral central nervous system. Approximately 30% of planarian nervous-system-related genes had homologous sequences in Arabidopsis and yeast, which do not possess a nervous system. This implies that the origin of nervous-system-related genes greatly predated the emergence of the nervous system and that these genes might have been recruited toward the nervous system [31]. Based on samples of three kinds of tissue—brain cortex, liver, and blood from humans, chimps, and rhesus macaques—researchers have identified 165 genes that showed significant differences between at least two of the three species and in at least one type of tissue. The brain contained the greatest percentage of such genes, about 1.3%. Gene expression in liver and blood tissue is very similar in chimps and humans and markedly different from that in rhesus macaques. The analysis shows that the human brain has undergone three to four times the amount of change in genes and expression levels than the chimpanzee brain since the two split off from a common ancestor [32].

12.5 REGULATION AND PATHWAYS

By several estimates, the fraction of the mammalian genome that is under some level of purifying selection is between 4 and 7% [28,33,34]. However, only a minority of that fraction comprises protein-coding sequence: the majority are noncoding; human–mouse–rat data support the presence of a large number of potentially functional nongenic sequences, probably regulatory and structural [35]. Their prevalence underscores the importance of such elements in genomic architecture.

Comparative genomics have provided computational approaches for understanding the transcriptional regulatory network, including promoter prediction, transcription factor binding site identification, combinatorial regulatory elements prediction, and transcription factor target gene identification [36]. Differential gene transcription is a fundamental regulatory mechanism of biological systems during development, body homeostasis, and disease. Comparative analysis is proving to be a rapid means for the identification of regulatory sequences in genomes [37]. Identifying the complete transcriptional regulatory network for an organism involves understanding every gene that is affected by each regulatory protein. Because regulatory systems tend to be conserved through evolution, researchers can use comparisons between species to increase the reliability of binding site predictions by combining the

prediction of transcription units having orthologous genes with the prediction of transcription factor binding sites based on probabilistic models [38]. The first step in this process is the localization of regulatory sequences in large anonymous DNA sequences. Once those regions are located, the second step is the identification of individual transcriptional control elements and correlation of a subset of such elements with transcriptional functions. Leung et al. [39] employed this approach to examine the TNF-α (tumor necrosis factor) promoters in primate lineages. A striking conservation was observed in a 69 base pair region corresponding to the well-characterized transcriptionally active nucleoprotein-DNA complex, whereas little conservation was found in regions not believed to have a role in regulation of the gene [39]. A similar methodology demonstrated conservation of repressor regions for the β-like globin locus control region in an evolutionary panel covering 370 million years [40]. Because this approach relies on evolutionary conservation of the regulatory sites, it should be noted that highly specialized structures that may be determinants of species differentiation will not likely be found.

Large-scale regulatory network reconstructions can be converted to *in silico* models that allow systematic analysis of network behavior in response to changes in environmental conditions. These models can be combined further with genome-scale metabolic models to build integrated models of cellular function including both metabolism and its regulation [41].

12.6 AGRICULTURAL GENOMICS

The use of comparative genomics techniques stands to make significant economical contributions to the agricultural industry. Agricultural genomics could lead to new strategies for industrial strain improvements in crops and dairy products.

A better understanding of the physiological processes and regulatory networks associated with dairy bacteria is of considerable commercial importance. Bacteriophages of lactic acid bacteria are a threat to industrial milk fermentation. Owing to their economical importance, dairy phages therefore have become the most thoroughly sequenced phage group in the database [42]. This information will lead researchers to better inhibitors.

Comparative genomics reveals that cereal genomes are composed of similar genomic building blocks (linkage blocks). By stacking these blocks in a unique order, it is possible to construct a single ancestral "chromosome," which can be cleaved to give the basic structure of the 56 different chromosomes found in wheat, rice, maize, sorghum, millet, and sugarcane [43]. A comparative analysis of the functions of rice and orthologous genes in other species involved in these processes revealed that orthologous genes can also display divergent functions [44].

The discovery of novel genes and the corollary expression patterns in response to stress can lead to improved plant tolerance to nature's temperamental nature and help them survive events such as droughts and low temperatures [45]. Preharvest sprouting results in significant economic loss for the grain industry around the world. Lack of adequate seed dormancy is the major reason for preharvest sprouting in the field under wet weather conditions. Although this trait is governed by multiple genes, it is also highly heritable. A major quantitative trait loci controlling both preharvest

sprouting and seed dormancy has been identified on the long arm of barley chromosome 5H, and it explains over 70% of the phenotypic variation [46].

12.7 COMPUTATIONS AND DATABASES

The availability of multiple genome sequence databases, combined with growing computational availability, has enabled significant advances in the identification of new structural and functional properties of DNA sequences, from changes in protein functions to the identification of repeats, conserved sequence regions, and noncoding RNA sequences. The growing size and complexity of these data calls for novel visualization methods as well as the development of powerful and cost-effective computing. These methods have unraveled data, something that simply would not be possible with a single genome analysis.

Recent developments in the identification of repetitive DNA sequences hope to shed new light in understanding the basis of genomic instability and a variety of regulatory functions. One such method, a program called the Spectral Repeat Finder program, significantly improves the repeats identification process by overcoming a number of major technical hurdles through computational techniques. The difficulty in identification of repeats resulting from variability in perfection, length, and dispersion was overcome by using a discrete Fourier Transformation to identify significant sequence periodicities [47]. The identification of these repeat regions extends our understanding of chromosome structure and dynamics and offers new insights into the evolutionary processes that have led to genomic divergence.

Database searches using sequence comparison programs have been successful in the identification of regions of sequence conservation. Because there are important sequence areas with an absence of protein homologs, new computational approaches in comparative genomics need to step outside conventional sequence comparison techniques. Analysis of phylogenetic profiles of protein families, domain fusions, gene adjacency in genomes, and expression patterns have been able to predict functional interactions between proteins and help deduce specific functions for many proteins [48] even in the absence of strong sequence conservation.

Another promising technique in comparative genomics involves the combination of computational and experimental methods to identify novel areas of interest in the sequences of multiple genomes. Fulton et al. [49] identified a large set of genes that were single or low copy in both the tomato and Arabidopsis genomes, which displayed a high probability for ortology. This was accomplished by screening a large tomato EST database and the Arabidopsis genomic sequence. These genes were annotated, and a large portion of them were assigned to putative functional categories with basic metabolic processes. Further computational screens against other genomes revealed that these markers were conserved in genomes of other plant families [49]. This work will further phylogenic studies in plants and provide the groundwork for more robust studies in comparative genomics, and is also applicable to the understanding of nonplant genomes.

A further example of the potential for discovery from combining multiple comparative genomics methods involves the principle of phylogenetic shadowing, a lineage-specific gene-finding technique, and feature-based annotation methods to

detect conserved sequence regions from multiple closely related organisms. This model is applicable where the principle of evolutionary constraint infers regional functions. The system has been successfully used to identify shared gene sequences between multiple primate sequences [50]. The results of these primate studies have obvious implications to humans, and these principles can likely be adapted to provide insight into other closely related genomes. Additional techniques, such as phylogenetic profiling, chromosomal proximity, and domain fusion methods, have been used in combination to provide functional linkage data. The merging of these data has uncovered a large number of conserved pathways and identified clusters of genes that are functionally related [51]. Many comparative genome analyses are designed to reveal important sequences that cannot be detected in a single genome analysis. One method used computational comparative genomic screens to reveal novel noncoding RNA sequences by taking advantage of mutational patterns in pairwise sequence alignments. A whole genome screen of *E. coli* revealed 275 candidate structural RNA loci out of more than 23K conserved interspecies pairwise alignments. Forty-nine were assayed experimentally, and more than 11 expressed small noncoding RNA transcripts of unknown function [52]. This methodology would have not been possible prior to the elucidation of RNA sequences in multiple species.

The increasing volume and complexity of multiple genome data have necessitated the development of visualization techniques to represent these data. Several tools have been developed to provide visual means of navigating multiple genomes simultaneously by retrieving annotated sequences from multiple genomes and generating an interactive display [53]. Some tools take advantage of the newly available data by visualizing annotations and conserved sequence regions in multiple genomes using multiple dimensional representations and provide a compact display of cross-sequence comparisons [54]. Other visualization systems integrate all of the publicly available annotations from multiple genomes and display them in a browsable fashion. In some systems the data can be navigated using a top-down approach, that is, the user displays the entire genome at once and then progressively increases the resolution to navigate down to the nucleotide sequence. At each of these levels of resolution, annotations from several genomes can be displayed simultaneously. These are just a few examples that address the need to represent large-scale comparative analyses in a visual manner. The tremendous increase in data will only be as valuable as the ability for researchers to extract meaningful information, which these tools are making possible.

The increase in genomic sequence database data has also necessitated the development of efficient computational tools running on parallel cluster computers in a cost-effective manner. A recent study ported and optimized the analyses programs of FASTA and Smith Waterman on PARAM 1000, a parallel cluster of workstations, to show significant performance increase over single-processor configurations [55]. The ever-increasing need for high-performance computing and the high cost of supercomputers render such solutions both relevant and essential.

12.8 CONCLUSION

In the last few years comparative genomics has experienced an explosive growth. The vast amount of multigenomic data coupled with the development of new technologies

has enabled a proliferation of discoveries in a diversity of domains from agricultural genomics, to evolutionary principles, to metabolic and functional pathways. But its most striking achievements have been associated with our understanding of human diseases, which has led to novel drugs, vaccines, and diagnostic tools and holds the promise of many more such discoveries.

REFERENCES

1. Fritz, B., and G. A. Raczniak. 2002. Bacterial genomics: Potential for antimicrobial drug discovery. *BioDrugs* 16:331–7.
2. Schoolnik, G. K. 2002. Microarray analysis of bacterial pathogenicity. *Adv Microb Physiol* 46:1–45.
3. Cole, S. T. 2002. Comparative mycobacterial genomics as a tool for drug target and antigen discovery. *Eur Respir J Suppl* 36:78–86.
4. Brosch, R., A. S. Pym, S. V. Gordon, and S. T. Cole. 2001. The evolution of mycobacterial pathogenicity: Clues from comparative genomics. *Trends Microbiol* 9:452–8.
5. Brosch, R., S. V. Gordon, C. Buchrieser, A. S. Pym, T. Garnier, and S. T. Cole. 2000. Comparative genomics uncovers large tandem chromosomal duplications in Mycobacterium bovis BCG Pasteur. *Yeast* 17:111–23.
6. Skamene, E., E. Schurr, and P. Gros. 1998. Infection genomics: Nramp1 as a major determinant of natural resistance to intracellular infections. *Annu Rev Med* 49:275–87.
7. Carlton, J. M. 2003. Genome sequencing and comparative genomics of tropical disease pathogens. *Cell Microbiol* 5:861–73.
8. Weill, M., G. Lutfalla, K. Mogensen, F. Chandre, A. Berthomieu, C. Berticat, N. Pasteur, A. Philips, P. Fort, and M. Raymond. 2003. Comparative genomics: Insecticide resistance in mosquito vectors. *Nature* 423:136–7.
9. Stoll, M., and H. J. Jacob. 2001. Genetic rat models of hypertension: Relationship to human hypertension. *Curr Hypertens Rep* 3:157–64.
10. Kamnasaran, D. 2003. Genetic analysis of psychiatric disorders associated with human chromosome 18. *Clin Invest Med* 26:285–302.
11. Lake, S., A. Krook, and J. R. Zierath. 2003. Analysis of insulin signaling pathways through comparative genomics. Mapping mechanisms for insulin resistance in type 2 (non-insulin-dependent) diabetes mellitus. *Exp Clin Endocrinol Diabetes* 111:191–7.
12. Odom, D. T., N. Zizlsperger, D. B. Gordon, G. W. Bell, N. J. Rinaldi, H. L. Murray, T. L. Volkert, et al. 2004. Control of pancreas and liver gene expression by HNF transcription factors. *Science* 303(5662):1378–81.
13. Seth, A., R. Kitching, G. Landberg, J. Xu, J. Zubovits, and A. M. Burger. 2003. Gene expression profiling of ductal carcinomas in situ and invasive breast tumors. *Anticancer Res* 23:2043–51.
14. Bastian, B. C. 2003. Understanding the progression of melanocytic neoplasia using genomic analysis: From fields to cancer. *Oncogene* 22:3081–6.
15. Yeung, R. S., H. Gu, M. Lee, and T. A. Dundon. 2001. Genetic identification of a locus, Mot1, that affects renal tumor size in the rat. *Genomics* 78:108–12.
16. Rubin, G. M., M. D. Yandell, J. R. Wortman, G. L. Gabor Miklos, C. R. Nelson, I. K. Hariharan, M. E. Fortini, et al. 2000. Comparative genomics of the eukaryotes. *Science* 287:2204–15.

17. Kappen, C., and J. M. Salbaum. 2001. 09/15: Comparative genomics of a conserved chromosomal region associated with a complex human phenotype. *Genomics* 73:171–8.
18. Thornton, J. W., and R. DeSalle. 2000. Gene family evolution and homology: Genomics meets phylogenetics. *Annu Rev Genomics Hum Genet* 1:41–73.
19. Klasson, L., and S. G. Andersson. 2004. Evolution of minimal-gene-sets in host-dependent bacteria. *Trends Microbiol* 12:37–43.
20. Koonin, E. V. 2003. Comparative genomics, minimal gene-sets and the last universal common ancestor. *Nat Rev Microbiol* 1:127–36.
21. Anantharaman, V., E. V. Koonin, and L. Aravind. 2002. Comparative genomics and evolution of proteins involved in RNA metabolism. *Nucleic Acids Res* 30:1427–64.
22. Castresana, J. 2001. Comparative genomics and bioenergetics. *Biochim Biophys Acta* 1506:147–62.
23. Wolf, Y. I., I. B. Rogozin, N. V. Grishin, and E. V. Koonin. 2002. Genome trees and the tree of life. *Trends Genet* 18:472–9.
24. Forterre, P. 2001. Genomics and early cellular evolution: The origin of the DNA world. *C R Acad Sci III* 324:1067–76.
25. Feuillet, C., and B. Keller. 2002. Comparative genomics in the grass family: Molecular characterization of grass genome structure and evolution. *Ann Bot (Lond)* 89:3–10.
26. Dunham, A., L. H. Matthews, J. Burton, J. L. Ashurst, K. L. Howe, K. J. Ashcroft, D. M. Beare, et al. 2004. The DNA sequence and analysis of human chromosome 13. *Nature* 428:522–8.
27. Grimwood, J., L. A. Gordon, A. Olsen, A. Terry, J. Schmutz, J. Lamerdin, U. Hellsten, et al. 2004. The DNA sequence and biology of human chromosome 19. *Nature* 428:529–35.
28. Gibbs, R. A., G. M. Weinstock, M. L. Metzker, D. M. Muzny, E. J. Sodergren, S. Scherer, G. Scott, et al. 2004. Genome sequence of the Brown Norway rat yields insights into mammalian evolution. *Nature* 428:493–521.
29. Modrek, B., and C. J. Lee. 2003. Alternative splicing in the human, mouse and rat genomes is associated with an increased frequency of exon creation and/or loss. *Nat Genet* 34:177–80.
30. Copley, R. R. 2004. Evolutionary convergence of alternative splicing in ion channels. *Trends Genet* 20:171–6.
31. Mineta, K., M. Nakazawa, F. Cebria, K. Ikeo, K. Agata, and T. Gojobori. 2003. Origin and evolutionary process of the CNS elucidated by comparative genomics analysis of planarian ESTs. *Proc Natl Acad Sci USA* 100:7666–71.
32. Normile, D. 2001. Comparative genomics. Gene expression differs in human and chimp brains. *Science* 292:44–5.
33. Chinwalla, A. T., L. L. Cook, K. D. Delehaunty, G. A. Fewell, L. A. Fulton, R. S. Fulton, T. A. Graves, et al. 2002. Initial sequencing and comparative analysis of the mouse genome. *Nature* 420:520–62.
34. Cooper, G. M., M. Brudno, E. A. Stone, I. Dubchak, S. Batzoglou, and A. Sidow. 2004. Characterization of evolutionary rates and constraints in three Mammalian genomes. *Genome Res* 14:539–48.
35. Dermitzakis, E. T., A. Reymond, R. Lyle, N. Scamuffa, C. Ucla, S. Deutsch, B. J. Stevenson, et al. 2002. Numerous potentially functional but non-genic conserved sequences on human chromosome 21. *Nature* 420:578–82.
36. Qiu, P. 2003. Recent advances in computational promoter analysis in understanding the transcriptional regulatory network. *Biochem Biophys Res Commun* 309:495–501.

37. Dickmeis, T., C. Plessy, S. Rastegar, P. Aanstad, R. Herwig, F. Chalmel, N. Fischer, and U. Strähle. 2004. Expression profiling and comparative genomics identify a conserved regulatory region controlling midline expression in the zebrafish embryo. *Genome Res* 14:228–38.
38. Tan, K., G. Moreno-Hagelsieb, J. Collado-Vides, and G. D. Stormo. 2001. A comparative genomics approach to prediction of new members of regulons. *Genome Res* 11:566–84.
39. Leung, J. Y., F. E. McKenzie, A. M. Uglialoro, et al. 2000. Identification of phylogenetic footprints in primate tumor necrosis factor-alpha promoters. *Proc Natl Acad Sci USA* 97(12):6614–8.
40. Shelton, D. A., L. Stegman, R. Hardison, W. Miller, J. H. Bock, J. L. Slightom, M. Goodman, and D. L. Gumucio, et al. 1997. Phylogenetic footprinting of hypersensitive site 3 of the b-globin locus control region. *Blood* 89:3457–69.
41. Herrgard, M. J., M. W. Covert, and B. O. Palsson. 2004. Reconstruction of microbial transcriptional regulatory networks. *Curr Opin Biotechnol* 15:70–7.
42. Brussow, H. 2001. Phages of dairy bacteria. *Annu Rev Microbiol* 55:283–303.
43. Moore, G., L. Aragon-Alcaide, M. Roberts, S. Reader, T. Miller, and T. Foote. 1997. Are rice chromosomes components of a holocentric chromosome ancestor? *Plant Mol Biol* 35:17–23.
44. Shimamoto, K., and J. Kyozuka. 2002. Rice as a model for comparative genomics of plants. *Annu Rev Plant Biol* 53:399–419.
45. Cushman, J. C., and H. J. Bohnert. 2000. Genomic approaches to plant stress tolerance. *Curr Opin Plant Biol* 3:117–24.
46. Li, C., P. Ni, M. Francki, A. Hunter, Y. Zhang, D. Schibeci, H. Li, et al. 2004. Genes controlling seed dormancy and pre-harvest sprouting in a rice-wheat-barley comparison. *Funct Integr Genomics* 4:84–93.
47. Sharma, D., B. Issac, G. P. Raghava, and R. Ramaswamy. 2004. Spectral Repeat Finder (SRF): Identification of repetitive sequences using Fourier transformation. *Bioinformatics* 20:1405–12.
48. Galperin, M. Y., and E. V. Koonin. 2000. Who's your neighbor? New computational approaches for functional genomics. *Nat Biotechnol* 18:609–13.
49. Fulton, T. M., R. Van der Hoeven, N. T. Eannetta, and S. D. Tanksley. 2002. Identification, analysis, and utilization of conserved ortholog set markers for comparative genomics in higher plants. *Plant Cell* 14:1457–67.
50. McAuliffe, J. D., L. Pachter, and M. I. Jordan. 2004. Multiple-sequence functional annotation and the generalized hidden Markov phylogeny. *Bioinformatics* 20:1850–60.
51. Yanai, I., and C. DeLisi. 2002. The society of genes: Networks of functional links between genes from comparative genomics. *Genome Biol* 3:research0064.
52. Rivas, E., R. J. Klein, T. A. Jones, and S. R. Eddy. 2001. Computational identification of noncoding RNAs in E. coli by comparative genomics. *Curr Biol* 11:1369–73.
53. Hoebeke, M., P. Nicolas, and P. Bessieres. 2003. MuGeN: Simultaneous exploration of multiple genomes and computer analysis results. *Bioinformatics* 19:859–64.
54. Montgomery, S. B., T. Astakhova, M. Bilenky, E. Birney, T. Fu, M. Hassel, C. Melsopp, et al. 2004. Sockeye: A 3D environment for comparative genomics. *Genome Res* 14:956–62.
55. Janaki, C., and R. R. Joshi. 2003. Accelerating comparative genomics using parallel computing. *In Silico Biol* 3:429–40.

13 Pharmacogenomics

Bahram Ghaffarzadeh Kermani
Illumina, Inc.

CONTENTS

Along with the emergence of genomics in the past two decades, pharmacogenomics has attracted much attention. The goal is often referred to as "personalized medicine," that is, to provide the right drug, at the right dosage, to the right patient. Personalized medicine is propelled by the observation that genetics make a real contribution to drug response. Since its inception, pharmacogenomics has been viewed as the future of medicine and at the same time has become the topic of much controversy in the scientific community. While its enthusiasts await the promised revolution of pharmacogenomics [1], neither drug companies nor drug approval agencies have, as yet, made any giant leaps in replacing traditional toxicological studies with toxicogenomic approaches, as genomic approaches have not been rigorously validated [2]. Nevertheless, as is discussed in this chapter, there is evidence that there might be a realizable, albeit limited, scope in which pharmacogenomics could prevail.

13.1 INTRODUCTION

Pharmacogenomics is the art of analyzing various genomic information (e.g., polymorphisms, gene expression, copy number, methylation, protein profiles) in assessing differential response to drugs. The objective of such analyses is to detect evidence of variation in response to drug action and factors influencing the absorption, distribution, metabolism, and excretion of these chemical agents [3]. Pharmacogenomics can be viewed from two perspectives: pharmacokinetics and pharmacodynamics [4]. Pharmacokinetics deals with drug metabolism, which in turn determines the optimal drug dose to maximize effectiveness and minimize toxicity. Pharmacodynamics, on the other hand, describes the drug mechanism of action on a target. This latter branch of pharmacogenomics is the main focus of drug discovery. Both pharmacokinetics and pharmacodynamics work in tandem to describe the drug effect in the patient.

The goal associated with pharmacogenomics is described as "personalized medicine." The belief that there is a one-size-fits-all approach ignores the fact that humans show significant genetic diversity [5]. "Personalized" medicine should not be misinterpreted as describing an ultimate goal of providing a unique drug for each individual for a given disease. Since genetic variations between individual humans are large, to the extent that individuals are unique, providing completely personalized medicine to each individual is unrealistic; such an approach would require multiplying the current number of medicines by the world population. Such a constraint is unrealizable from multiple aspects, including the cost of development, the development cycle, the method of distribution, and so forth. What is truly meant by the term *personalized medicine* is to stratify individuals into a few classes and to design a unique drug for each class. In this context, the current state-of-the-art in the pharmaceutical industry

is the equivalent of having one class. The number of classes is not required to be the same for each disease. The binning of individuals into different classes is guided by considering statistical information about a population, as viewed by a certain disease, and applying it to individuals [4]. With this objective in mind, pharmacogenomically focused drugs could achieve a higher penetration in a selected population, and those individuals could be given a more effective dose of the drug [5].

In parallel to stratification of individuals, one could also consider the stratification of diseases. What is today known to us as a certain illness may indeed be a class of several diseases that result in similar macroscopic (and not necessarily microscopic) symptoms and phenotypes. One exemplary basis for such an argument is the fact that a biological pathway has different constituents or blocks. If any of these blocks malfunctions, the outcome would be disruption of the overall function of the pathway, resulting in a similar phenotype. While the idea of dividing illnesses into smaller subcategories is inspired by the divide-and-conquer method of solving complex problems, it suffers from the linear growth in the amount of validation that is needed. Indeed, most researchers believe that pharmacogenomics could result in subdividing some complex diseases into a large number of classes [6]. The problem is that for each disease subcategory, different (albeit simpler) clinical trials must be performed, which results in increased complexity and cost in a drug-approval process.

Another promise of pharmacogenomics is the reduction of adverse drug reactions (ADR). ADR is one of the prominent causes of illness and fatality. It has been reported that in a certain year, more than 2 million North Americans have been hospitalized for experiencing serious adverse drug reactions, resulting in more than 100,000 fatalities [7,8]. This number has placed ADR as the fifth leading cause of death [7]. Many ADRs arise from genetic differences in drug metabolism, transporters, ion channels, receptors, and other drug targets [7]. Thus, there are reasons to believe that pharmacogenomics could play a major role in reducing ADRs. One such role would be reduction of ADR through accurate labeling of contraindications [6].

Perhaps the largest motivation of pharmacogenomics for drug companies is its promise for reducing the cost of drug development. Drug development is a time-consuming and costly process, taking 1 to 2 decades and costing several hundreds of million dollars, with a success rate of approximately 1 in 10, only a fraction of which are observed to be blockbusters [11]. Despite the complicated development machinery, many drugs are ineffective, as a result of underdosing, overdosing, or missed dosing. Such conditions have been associated with the cost of more than 100 billion dollars a year [5]. Therefore, there is much room for improving the effectiveness of drugs, their development, and the testing process. Pharmacogenomics is a view from different angles, albeit shared by one name, which promises enhancements in drug efficacy and safety. The analysts predict that by using pharmacogenomically enhanced diagnostics and drugs, the pharmaceutical companies could benefit from extra revenue on the order of 200 to 500 million dollars for each drug [9].

An alternative perspective from which to view the cost of the drugs is the overall cost of medicine for the patients. Many patients end up trying several drugs (and withstanding their side effects) before finding the one that works best for them, if any. The promise of pharmacogenomics is identifying the effective drug the first time, which would result in a significant cost reduction and reduced risk. Moreover,

with some diseases, the window of treatment is so limited that there is no room for trial and error. In other words, if the correct drug is not the first drug prescribed, the disease might have progressed to an untreatable state; various cancers and Alzheimer's disease, which has a rather narrow window of treatment (18 months), are examples of this scenario [3].

Among the other potentials for the role of pharmacogenomics in the future of medicine, one could list customized therapy, screening for disease predisposition, customized preventative care, and improved extrapolation of drugs from adults to children. Pharmacogenomics could also be applied to nonhuman organisms, for example, to bacteria for identifying the correct antibiotic development [11].

13.2 CASE STUDIES

Numerous examples show how pharmacogenomics can be applied in current medicine. A few selected examples are presented next. This list is far from comprehensive.

13.2.1 P450 Family of Enzymes

Cytochrome P450 (CYP) enzymes represent a family of xenobiotic metabolizing proteins and are found in endoplasmic reticulum of cells in a variety of human tissues, predominantly in liver and intestine. These enzymes are thought to be responsible for the metabolism of approximately 75% of currently available drugs. The subfamily CYP3A by itself is responsible for nearly half of this activity [12]. For most drugs, activation of CYP enzymes determines how long and how much of a drug remains in the body [4]. Poor metabolizers would be more likely to experience toxicity from drugs metabolized by the affected enzymes. The percentage of people classified as poor metabolizers varies by enzyme and population group. As an example, approximately 7% of Caucasians and only about 1% of Asians appear to be CYP2D6 poor metabolizers [12]. Pharmacogenomics tests such as those for the CYP2D6 family of enzymes are currently being used in clinical research [9].

13.2.2 Heart Arrhythmia

An example of a gene with variations that can result in heart problems is SCN5A. Certain mutations in this gene have been associated with congenital long QT syndromes, a rare hereditary heart arrhythmia [13]. When appropriately stimulated, normal SCN5A protein forms a sodium channel that opens to allow the flow of sodium ions into heart muscle cells, thus triggering the cells to undergo contraction. It has been reported that variants of SCN5A could result in generation of sodium channels that reopen during the time that they should be closed, a change that could result in developing an arrhythmia [14].

13.2.3 Breast Cancer

One of the first treatments based on pharmacogenomics is the drug Herceptin—a monoclonal antibody that targets the protein product of the HER2 oncogene [6]. Breast cancer patients can be divided into two groups—HER2-positive and HER2-

negative. Herceptin is a candidate treatment drug for HER2-positive patients. This drug can bind to HER2 products, slowing tumor growth [3].

In the context of breast cancer, the genes BRCA1 and BRCA2 are also of particular interest. These genes are believed to be tumor suppressors, and their functional variation is believed to depend on their sequence variation. Women carriers of germline BRCA1 and BRCA2 mutations have more than an 80% lifetime risk of developing breast cancer. Approximately 0.5% of women carry one of these mutations, although this percentage may be higher in certain ethnic groups [12].

13.2.4 Thiopurine Methyltransferase

The inactivation of mercaptopurine requires metabolism by the enzyme thiopurine methyltransferase (TPMT). TPMT polymorphisms have been reported to be considerably overrepresented in patients with ADRs to mercaptopurine [15]. A deficiency in TPMT is inherited as an autosomal recessive trait [12]. Patients with two copies of one of the polymorphisms of this gene that decreases activity are at risk for serious and potentially life-threatening dose-related side effects (e.g., bone marrow suppression). Even the heterozygotic patients are at risk for severe side effects [3]. Approximately 10% of people are heterozygous in this gene (i.e., carry one bad copy), which makes them poor metabolizers of mercaptopurine and necessitates a reduction in dose. A considerably smaller percentage of people carry two bad copies of the gene, which renders them extremely sensitive to the drug and requires up to a 95% reduction in the dosage [3].

13.2.5 Alzheimer's Disease

Tacrine is used to treat Alzheimer's disease and is in a class of drugs known as acetylcholinesterase inhibitors. Unfortunately, only one quarter of patients benefit from Tacrine and roughly the same fraction suffer severe side effects such as liver toxicity. This is believed to be linked with a polymorphism in APOE gene (apolipoprotein E) [5]. APOE has three common alleles—APOE2, APOE3, and APOE4—resulting from nonsynonymous coding single nucleotide polymorphisms (SNPs). APOE is involved in modulation of cholesterol and transport of lipids in plasma and within the brain. APOE and its associated receptors are highly expressed in the brain. The APOE4 allele is associated with sporadic and late-onset familial Alzheimer's disease. It has been shown that APOE4 correlates with the risk of developing Alzheimer's disease, more specifically the age of onset, accumulations of plaques, and reduction of choline acetyltransferase activity in the hippocampus [12]. APOE4 allele copy number has an inverse relationship with residual brain choline acetyltransferase activity and nicotinic receptor binding sites in both the hippocampal formations and the temporal cortex of patients with Alzheimer's disease [12].

13.3 RELATED TECHNOLOGIES AND THEIR ISSUES

13.3.1 SNP Genotyping

The general idea of pharmacogenomics hinges on the belief that genetics is a main component of the drug efficacy. This belief provides motivation for deconstructing

the genetic blueprint. The Human Genome project provides the common (99.9% similarity) DNA template among individuals. What remains is the 0.1% that is different among individuals. These differences are mostly manifested in the form of SNPs. On average, one SNP is expected to exist for every 1 Kb. Therefore, given the 3.1 billion base pairs of the genome, one would expect to have approximately 3.1 million SNPs.

Although sequencing the genome is a great effort, it has been done once for a small number of individuals and is considered valid for the whole human population. The SNP profile, however, is expected to be different for every individual. After all, it is this 0.1% that makes each of us different from one another. Therefore, although it may seem a small proportion (0.1%), SNP genotyping is a great effort, as it needs to be done for all the individuals that participate in a particular study (e.g., a clinical trial study). The large number of SNPs and the huge potential for their use have been the driving forces for a plethora of methods for multiplexed SNP discovery, and ultimately the whole genome SNP microarrays.

Once the SNPs are identified for a population of individuals, they can be studied to understand phenotype/genotype correlations. This study is done in two modes. In the first mode, the assumption is that the SNP is causal, that is, it directly causes the phenotype. For this to happen, the hypothesis is that the SNP should (a) be in the coding region or (b) cause a nonsynonymous amino-acid change, and (c) the change should result in a significant modification of the expressed protein. Alternatively, the SNP could be in a regulatory region and cause a notable expression change in the observed phenotype. In another mode, the SNPs are used purely as landmarks for susceptibility genes. This use is empowered by the belief that there has been a limited amount of shuffling in the DNA, which results in coinheritance of disease-causing genes and nearby SNPs. This coinheritance is discussed later in the context of linkage disequilibrium (LD) analysis. The SNPs that can be used as informative markers are the ones in a high LD with the susceptibility genes.

Despite their popularity, there are a few challenges in using SNPs as genetic markers. The main factor has been the cost of SNP genotyping. The other factor is the quality of purported SNPs. Many areas of genome are difficult to sequence (e.g., the centromeres), and thus there are not many SNPs identified in those regions. Even for the areas that do contain valid SNPs, given the state of most SNP discovery assays, it may be difficult to obtain a confident identification of polymorphism because of the existence of genome repeats and other low-complexity regions in the vicinity of the SNP. Nevertheless, such problems are not unique to SNPs. Indeed many such concerns are also shared by other types of markers.

13.3.2 Haplotyping

There is an ongoing debate about the number of SNPs that is necessary for genome-wide association studies. The objective of haplotype mapping is to find a subset of SNPs that contain the information within the complete set. The idea underlying this concept is the fact that in certain areas of the genome, SNPs on a small region of one chromosome are correlated. This correlation is due to the fact that these areas have not undergone drastic genomic shuffling via recombination through evolution.

In other words, it is believed that a set of haplotype blocks exist, where SNPs within such block boundaries change together, that is, knowing one would reveal the identity of the others. The existence of such boundaries has been the subject of much debate. The haplotype blocks in older populations (e.g., Africans) range from less than 1 Kb to 100 Kb, and in younger populations (e.g., Finns) from more than 50 Kb to more than 1 Mb [7]. The International Haplotype Map project (HapMap) was funded a few years ago and is moving quickly toward completion. The hope is that with the aid of haplotype blocks, one may be able to scale down from analysis of 3 million SNPs to approximately 300,000 SNPs.

With respect to pharmacogenomics, whether this hypothetical 10:1 reduction of number of SNPs makes a difference or not is open for debate. Assuming the HapMap delivers 300,000 SNPs that well represent the complete SNP set, the added cost of three million dollars for a clinical trial of 1,000 individuals still remains. This added cost must be balanced against the cost savings that is expected to occur as a result of starting with a prescreened set of patients for whom the drug would have a greater potential to be effective.

13.3.3 Linkage Disequilibrium

Association studies are expected to find areas of the genome that may harbor susceptibility genes, without any prior assumption about the position or composition of such genes [16]. LD is the cornerstone of association studies. (A list of LD-related software packages is presented in table 13.1.) Nevertheless, LD has remained a controversial methodology [4]. In fact, theoretical estimates of the average extent of LD in the human genome range from less than 100 Kb to less than 3 Kb [16]. This value is a function of many parameters, including population admixture, population bottlenecks, heterogeneous recombination, genetic drift, mutation, and natural selection [7]. The concerns about the extent of the LD, along with the other concerns such as required population sample sizes, the number of SNPs needed in a map, the cost of genotyping, and the interpretation of results, are some of the challenges that surround this technique [16]. Sample size is of particular interest for pharmacogenomics applications, as it is linearly correlated with the cost of the clinical trial. The weaker the LD between the marker and the susceptibility genes, the more difficult the association is to detect, unless the sample size is increased proportionately. The required sample size is also affected by the match between the marker allele frequency and the susceptibility allele frequency. If marker allele frequencies are substantially different from susceptibility allele frequencies, the sample size, the number of markers, or both, will need to be dramatically increased [16]. Nevertheless, LD is still considered as a method worth pursuing, because the traditional alternatives (e.g., linkage analysis) remain costly and laborious.

13.3.4 Gene Expression

In the last decade, with the advent of microarrays, the interest for high-throughput gene expression has escalated significantly. Since protein generation is derived from the mRNA, and since assays for mRNA profiling are simpler than for protein profiling

TABLE 13.1
Examples of LD-Related Software

Software Name	Software Description	URL
MERLIN (Multipoint Engine for Rapid Likelihood Inference)	General tool for estimation of haplotypes using maximum likelihood method	http://www.sph.umich.edu/csg/abecasis/Merlin/
QTDT (Quantitative Trait Transmission Disequilibrium Test)	Tool for fitting linkage and association models to pedigrees	http://www.sph.umich.edu/csg/abecasis/QTDT/
GOLD (Graphical Display of Linkage Disequilibrium)	Calculation of LD-related parameters (e.g., D', r^2, and color coding of LD-coefficient matrices)	http://www.well.ox.ac.uk/asthma/GOLD/
SNPtagger	Tool for selection of tag SNPs	http://www.well.ox.ac.uk/~xiayi/haplotype/index.html
PHASE	Estimating multimarker haplotypes in unrelated individuals	http://www.stats.ox.ac.uk/mathgen/software.html
SNPhap	Tool for EM-based haplotype frequency estimation in unrelated individuals	http://www-gene.cimr.cam.ac.uk/clayton/software/
BLADE (Bayesian Linkage Disequilibrium Analysis Mapping)	Robust tool for estimation of LD using Markov Chain Monte Carlo algorithm	http://www.people.fas.harvard.edu/~junliu/index1.html
DHSMAP (Decay of Haplotype Sharing Mapping)	Tool for fine-mapping of qualitative traits by LD, by estimating the location of the trait-associated variant by maximum likelihood	http://galton.uchicago.edu/~mcpeek/software/dhsmap/
DMLE (Disease Mapping using Linkage Disequilibrium)	Tool for high-resolution mapping of the position of a disease mutation relative to a set of genetic markers using population LD	http://www.dmle.org/
EMLD (EM estimation for LD)	Tool for EM estimation of haplotype frequencies and LD calculation	http://request.mdacc.tmc.edu/~qhuang/Software/pub.htm
FBAT (Family Based Association Test)	Tool for testing association and linkage between disease phenotypes and haplotypes using family-based controls	http://www.biostat.harvard.edu/~fbat/fbat.htm

Note: LD = linkage disequilibrium.

(mRNA-based), gene expression has been in vogue as a reasonable proxy for protein expression. However, the technical difficulties of gene-expression microarrays are numerous, among which one can enumerate the following:

1. Gene expression is *in vitro* and does not necessarily reflect the *in vivo* state of the genes of interest, although in certain circumstances, with proper handling of samples (e.g., immediate freezing following the biopsy from a live tissue), one could have a higher assurance in retaining the *in vivo* state of the genes.
2. Most of the microarray data are not quantitative (e.g., as compared to quantitative-PCR). Part of this anomaly is due to the fact that most microarray technologies are hybridization-based, and the hybridization signal is not a linear function of the molecular concentration.
3. The gene expression probes often interrogate a small portion of the mRNA molecule. Since mRNA can undergo fractionation, whereas the probe only views one of the fractions, the translation from the microarray numbers to the mRNA molecules is not simple. Most manufacturers of microarrays try to minimize this anomaly by biasing the probes towards the 3' end (i.e., the poly-A tail of the mRNA molecule).
4. The fold-change in mRNA does not exactly translate to the fold-change in the final protein.
5. The significance of the scale of expression for different genes is variable and depends on the role of the gene in its associated pathway. For some genes, small changes in expression level could result in significant physiological changes, while for other genes, large changes in expression level may indicate an insignificant physiological change.
6. Most genes in complex diseases play a susceptibility role, and thus result in a probabilistic nature for regulation. This is a major hurdle in integrating the pieces of information collected from a series of genes and making claims about the underlying phenotypes of interest.

With respect to pharmacogenomics, in addition to the aforementioned points, there are certain particular concerns that limit the applicability of gene-expression analysis for such application. First, gene expression requires tissue samples. It would be hard to justify the need for a biopsy as a precursor to prescribing the effective medicine. Second, gene-expression information is dynamic, so if a patient is to be tested over time, the original test is not applicable for the follow-up visits.

13.3.5 Methylation

Epigenetics is the study of heritable changes in gene function that occur without a change in the DNA sequence. Methylation patterns are epigenetic changes that can modify the extent of gene expression by affecting the regulatory regions of genes. Methylation sites are best known as epigenetic signals residing in genomic DNA [17]. A testimony to the importance of the methylation is the destiny of cloned lambs, which died soon after birth due to a lack of imprinting, which resulted in overexpression of

genes due to biallelic expression [18]. It is also believed that a methylation footprint on chromatin pertains to genome stability and overall chromatin packaging [17]. The challenge in methylation is that such patterns are dynamic and may undergo somatic mutations in different tissues. Therefore, for early detection or monitoring of a disease such as cancer, methylation patterns are of paramount importance. These dynamic patterns make the landscape more difficult for pharmacogenomics, in the sense that it increases the space of search in genetic variations. Even if we had the complete genome sequence of a certain individual, we might still not have the clear picture of the genomic landscape of that individual because methylation changes are not detected using ordinary sequencing and SNP genotyping assays.

13.3.6 Proteomics

Since, ultimately, proteins are responsible for performing bodily functions, there is a great interest in monitoring them directly. However, there are several challenges that have prevented this field from reaching its potential. One of the main reasons is the difficulty in making the detection molecules for proteins. This is in contrast to nucleic acids, where a complementary DNA molecule can be easily synthesized for detection of mRNA. The second challenge is specificity, that is, making detection molecules that attach uniquely to a protein motif. The third challenge is a byproduct of the second. Since it is nearly impossible to put the whole proteome on a chip, people resort to making a small subset of the complete proteome on chips (e.g., on the order of 100 or less). In this case, the challenge is the selection of the content; the success of such chips for pharmacogenomics applications depends highly on this content selection.

13.4 EXPECTATIONS AND FUTURE POSSIBILITIES

13.4.1 Resurrecting Previously Failed Drugs

It has been reported that 10% of drugs are withdrawn in the years following FDA approval [6]. This statistic provides a great deal of motivation for resurrecting such drugs using pharmacogenomic knowledge. This view is the essence of the Lazarou's program, which focuses on resurrecting previously failed drugs [11]. Most of these drugs are expected to be the ones that failed during clinical trials due to toxicity or lack of efficacy. It is known that the level of toxicity is a critical value for most drugs, that is, beyond a certain point, many drugs could have toxic effects. Since the level of toxicity of a drug is confounded by the level of drug metabolism, there is a chance that by matching the drug dose to the genetic information, one can control the bounds on the toxicity and thus use such drugs for genetically selected responders. Therefore, for drugs that failed during clinical trial or at the discovery stage because of ADRs, pharmacogenomics provides hope for gaining a balance between the generality of a drug and its efficacy. In other words, one could obtain an effective drug (i.e., less prone to causing ADR) by narrowing the scope of a drug to certain genetic groups [6].

One point of caution in setting expectation for Lazarou's program is the dilemma of intellectual property. Many patents for abandoned drugs either have expired or

are near expiration, thus removing the competitive advantage granted to the patent owners. Many of the intermediate compounds or technologies associated with such drugs lose their patentability because of time limits that are built into the patenting process. Thus, such unpursued drugs or compounds may not have lucrative returns for the associated pharmaceutical company [6].

13.4.2 Balancing Efficacy and Toxicity of Drugs

For a drug to be effective, it must be exposed to the tissue of interest at a critical concentration for a given period of time. Below this critical concentration, the drug is not expected to be effective. Above this critical concentration, there is a margin above which the drug could be toxic. This critical concentration and the associated margin (for effectiveness vs. toxicity) are functions of the drug dose and drug metabolism. Drug metabolism has been linked to genetic variation (e.g., the polymorphic cytochrome P450 enzyme). People with certain CYP2D6 polymorphisms have been correlated with having fast drug metabolism [6]. For such individuals, given a typical dosage, the drug concentration in the tissue of interest drops too rapidly to be clinically effective. Thus, to counteract this effect, one could increase the dosage of the drug for such individuals. A clearer example is the drug Omeprazole. In some Asian populations, 15 to 23% are reported to be poor metabolizers of this gastrointestinal drug, because of polymorphism in CYP2C19. This figure is in contrast with 2.5 to 6% in Caucasians [19]. Pharmacogenomics could use such information on such polymorphisms to predict the correct dose for effectiveness of a drug.

The role of pharmacogenomics in drug action (pharmacokinetics) is seen as the lowest-hanging fruit for pharmaceutical companies in an attempt to achieve commercially meaningful results within the constraints of a clinical trial, in contrast with the impact of this field on the genes involved in the pathogenesis of disease [20]. In fact, many of the large companies are already considering pharmacokinetic variations [5] with the particular interest of drug effectiveness and toxicity [23].

In summary, there is great interest in fine-tuning the effective drug concentration to obtain a maximal effect and minimal toxicity. Cancer is considered to be an ideal condition for which to apply this approach, as subtle differences could account for notable differences between a particular dose of chemotherapy being toxic or effective [24]. The margins of toxicity and effectiveness vary widely between drugs and individuals. To further complicate the issue, one cannot expect to always find drugs that are effective and not toxic at a given dosage. This combination depends on the way in which the toxicity curve (vs. drug dose) and the effectiveness curve (vs. drug dose) are shaped. An illustration of a hypothetical case for drug effectiveness and toxicity as functions of drug dosage is provided in figure 13.1. Such curves are expected to vary for different individuals. One aim of pharmacogenomics is to characterize such curves for different individuals.

Normally, one would fix the level of tolerable toxicity and maximize the effectiveness under that constraint. However, setting the tolerable level of toxicity is related to the individual's current state of health. Under certain conditions, setting such thresholds could be exceptionally challenging for the health practitioners.

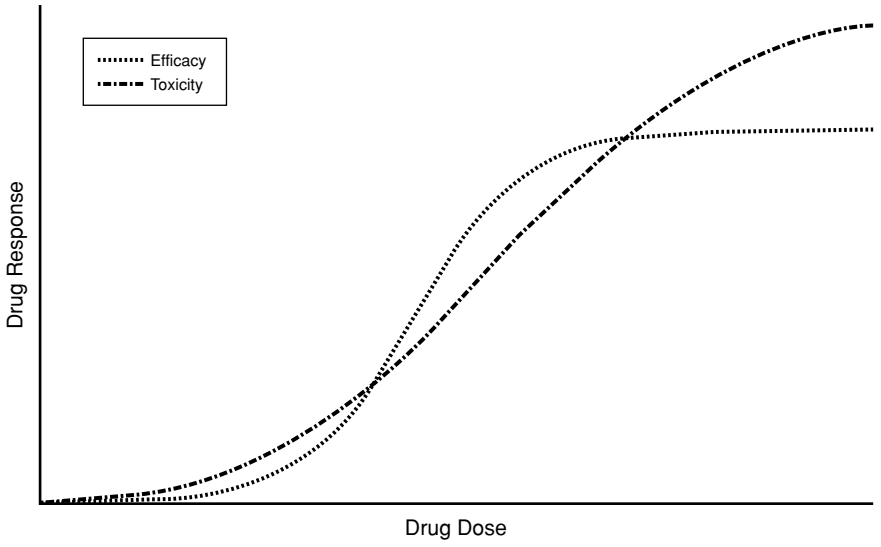


FIGURE 13.1 Hypothetical curves for drug efficacy and drug toxicity as a function of dosage.

13.4.3 Improved Generalization

Japanese pharmaceutical authorities require clinical trials on the Japanese population [19]. This bias toward a certain population could create a gap in the applicability of such products for other populations. Such biases are not always evident and overtly stated. For instance, similar biases do exist in drug development in the United States, where a majority of the drugs are tested on the Caucasian population. Such biases provide the basis for inefficacy of the drugs on other populations (e.g., the untested groups).

Pharmacogenomics can provide guidance to those drugs being of use to other populations in two ways. The most obvious way would be to design drugs for different ethnic groups based on their genetic composition. However, this method has serious flaws, some of which are discussed next. The more elegant way would be to bypass the dependency of the drug to population composition by screening for compounds that bind to all expressed variants of a target (if possible), thus eliminating the need for such a genetic test [19].

13.5 TECHNICAL CHALLENGES AND CONCERNS

13.5.1 Stationary and Global Genetic Information

So far, the foundation of pharmacogenomics has been the dependency of drugs to genetic variations, while the genetic variations have been considered to be stationary and global. This stationary condition implies that the genetic information is constant across time. In other words, one does not expect the genetic constituents of an individual, as derived from the parents' germ lines, to change over time. Likewise,

the global condition implies that the genetic information in all cells is similar. In the context of diseases, the existence of both these conditions can be challenged if the genetic change is caused by a somatic mutation. For instance, consider a cancer caused by a somatic mutation in liver. Such a change invalidates both of the aforementioned conditions, that is, it happens at a certain point in time and it is localized to a certain organ.

Now consider the implications of the invalidation of these stationary and global conditions. Pharmacogenomics adds at least one significant layer to the current diagnostic process—a genetic test. Ideally, such a genetic test is comprehensive, that is, it includes information on SNPs, methylation, gene expression, proteomics, and so on. Currently we cannot encompass all such tests in an affordable manner. We could perform this rather expensive test for each patient, once the genetic composition of every human being could be extracted and stored in a databank. If such information were to be useful for detection of liver cancer in a patient at some point, then the genetic information must be conserved across both time and tissue. In other words, the genetic composition of the blood (taken for the genetic test) and liver (tissue of interest) must be the same, and the composition of liver at the time the test was performed must be the same after the onset of the disease. If the cancer is caused by a sudden somatic mutation in liver at some point after taking the blood sample for genetic test, both of the aforementioned conditions will be violated. In other words, the data that are stored in the databank might not be useful for the detection of such a cancer.

Ordering a new test after the onset of the disease is then required. This solution, even if affordable (multiple genetic tests), has the underlying limitation that the physician must know what organ is the suspect in the disease, which implies a knowledge of the affected tissue or a bias toward the affected tissue. One must know that something is wrong with the liver before discovering that something is wrong with the liver! This does not completely invalidate pharmacogenomics and its required genetic tests, it simply places limits on how far expectations should grow before becoming unreasonable. For example, the following scenario may be typical. A patient complains of certain symptoms. The treating physician suspects liver cancer. The patient's liver is subjected to genetic test. The test reveals a chromosomal deletion and hyperexpression of certain cancer-related genes. The doctor diagnoses the disease as a certain type of liver cancer and recommends a certain treatment for the disease. If the treatment is to be guided by pharmacogenomics, a further and separate genetic test may be necessary.

13.5.2 Disease Complexity and Mendelian Assumption

Phenotype refers to any physiological, morphological, or biochemical characteristic of an organism [7]. Phenotypes may or may not have genotypic causes, and even when they do, they could be the result of interplay between multiple genetic and/or environmental factors. In fact, the taxonomy of a disease is often defined by multiple factors, including somatic mutations, epigenetic modifications, gene expression, protein expression, and protein modification [25]. Despite the fact that many phenotypes have genotypic origin, the connection between the two in drug metabolism

and disease manifestation remains vague and complex [9]. Very few common diseases have Mendelian genetic sources, that is, are caused by mutations in a single gene. In fact, it has been claimed that there are virtually no examples where a single DNA variant has always been associated with a particular phenotype in all subjects of human population [7].

Unfortunately, most diseases of economical and societal interest (e.g., heart disease, cancer, diabetes and asthma) are considered to be such complex diseases, that is, they cannot be attributed to single genes. In this context, the related genes are referred to as susceptibility genes. Such genes may or may not be related functionally; they may be unrelated but belong to intrinsic biochemical pathways [8].

13.5.3 Nature Versus Nurture

It is not yet known how much of human variation is caused by genetic makeup versus other factors such as environment, age, diet, lifestyle, and state of health. Based on the studies performed on identical twins, it is claimed that 8% of their differences are from the result of their environment and the remaining differences are associated with their genetic makeup. However, if the identical twins are not separated at birth, such estimations could be highly biased, since they would grow in similar environments and have similar lifestyles, because of a common family. Therefore, knowing the exact extent of the nature versus nurture influence remains a challenging problem. However, it is evident that a considerable proportion of our differences are due to genetic variations. Such variations are the motivation for pharmacogenomics.

13.5.4 The Cost of Medicine

The cost of medicine is governed by the cost of drug discovery and the cost of drug development and evaluation through clinical trials. For drug discovery, the effect of pharmacogenomics, on one hand, is a linear increase in the type of drug targets, as compared to the classical model of "one drug fits all." On the other hand, the identification of drugs for each disease may be simplified, as each class addresses a simplified, more direct group of diseases. In essence, pharmacogenomics provides a classic type of divide-and-conquer model for problem solving, in which a complex problem (i.e., a disease in population) is tackled by solving a multiplicity of simpler problems (i.e., the disease in subpopulations). One may argue that the problem has retained the original complexity, because each subproblem is simpler to solve, but the number of problems has increased. The reason behind the efficacy of divide-and-conquer algorithms is the fact that the complexity often grows linearly (by having a multiplicity of subproblems), but each subproblem is simpler than the original problem in a stronger-than-linear fashion (e.g., exponentially). Therefore, such an approach is expected to improve the solution.

So far, we discussed the efficacy of the divide-and-conquer method in the abstract sense. With regards to pharmacogenomics, such efficacy may manifest itself in the form of better quality, faster convergence to a solution, or better cost. Whether it actually accomplishes one or more of the aforementioned objectives depends on the

type of the problem and the implementation of the solution. In general, in the absence of the specific problem parameters and constraints, one cannot predict any of these conjectures. Similar arguments hold for clinical trials, that is, the true improvements, if any, are a function of the specific drug and its genetic link. The market size for a certain drug is one factor among others that determines the overall cost of drugs [6]. If the market is small prior to stratification via pharmacogenomics, it may not be economical to deviate from the classical drug-development approach. Such market evaluations may also be subject to ethical issues, if one considers the buying power of the individuals within each ethnically divided subclass.

13.5.5 Clinical Trials

Drug discovery and development are laborious processes, taking an average of 15 years from the identification of targets to marketing of the product. Between identification and marketing there are preclinical and three phases of clinical trials. In the preclinical stage, the genomic predictors of human toxicity are studied. Phase I trials are usually performed on healthy volunteers and are designed to identify the early tolerability of the drug, thus forming the base of knowledge on how the medicine should be dosed. Phase II trials are usually conducted with several hundred to a thousand individuals with the disease to be treated. Phase III involves focused trials for efficacy and submission of safety markers [3].

It has been reported that 80% of drugs currently fail in clinical trials [5], such that the clinical trial process has a yield of 20%. Given the lucrative profit margins upon the approval of drugs, such a yield is not necessarily low. However, the 80% gap is a great impetus for seeking improvements. Here is where pharmacogenomics may have value. The idea is (if we only consider pharmacokinetics) that prominent reasons for drug failure are toxicity issues. If such toxicities are linked to genetic variations, one must be able to stratify the population, based on their genetic makeup, and devise a different dose for each subgroup. The caveat is that by such division, we will have to increase the size of overall clinical trial, as each subgroup would still require a considerable population size for clinical trial to come up with statistically and clinically significant results. It is true that each subpopulation does not require as many individuals as would the classical clinical trial. However, the sum of individuals in subpopulations for clinical trial is expected to be larger than the original sum. This is particularly true for traits that appear with low frequencies in certain populations. The other caveat is the fact that by dividing the population into subpopulations, the problem changes to a multiple comparison. While statistical corrections, Bonferroni, and so on, can be used to address such issues [20], such methods result in a necessity for having a larger number of individuals, which in turn results in increased time and cost for trials.

For the United States, the role of the Food and Drug Administration (FDA) in pharmacogenomics is worthy of note. In November 2003, the FDA issued a draft guidance that encourages drug developers to conduct pharmacogenomic tests during their development process. In this guideline, submission of pharmacogenomics-related data is deemed voluntary and welcome. The FDA claims not to use such information for regulatory decision making for new or investigational drug applications, as it considers the pharmacogenomics data to be of exploratory/research nature. Such

submissions are meant to prepare and train FDA scientists for appropriately evaluating the anticipated future submissions.

Whether the voluntary submission of pharmacogenomics data will be adopted by pharmaceutical companies is subject to debate. If the bottom-line profit of these companies is jeopardized by pharmacogenomics, it would be hard to imagine that these institutions would provide such data (even if applicable and available) to the FDA. If they do, they may be creating incentives for undesired future regulations based on pharmacogenomics. Such companies may collect this information, internally, so as to be prepared, just in case pharmacogenomics becomes a requirement by FDA in the future. On the other hand, if pharmacogenomics opens up opportunities for new markets, then one would expect the pharmaceutical companies to welcome regulations in this area, as it would impede their competition.

13.6 INFORMATICS CHALLENGES

It has become a common practice in biology (e.g., in highly parallel gene-expression studies) to generate a massive amount of data (often of low quality and repeatability) and leave the rest to a general claim that bioinformatics, often referred to in the context of postgenomic era, would be developed to cope with that glut of information. There are two inherent assumptions in such a hypothesis. The first assumption is that the complex genomic problem of interest does have a solution. The second assumption is that such a solution can be obtained using ordinary or complex (possibly futuristic) techniques. Both of these assumptions could be challenged when it comes to the complex drug–body interactions. In fact, it is not obvious that the current mathematical tools are even suitable for such studies. The larger hindering factor is the many-to-one mapping nature of drug–body and body–disease interactions, which results in rank deficient sets of equations (linear or nonlinear), that is, nonexistence of a unique solution to the problem.

Next, some popular methods that are used for pharmacogenomics are listed.

13.6.1 Genotype/Phenotype Correlation

Discovering the hidden relationships between genotypes and phenotypes is one of the main enabling steps for pharmacogenomics. If such relationships are found, one would be able to make an inference system, by which the response of a patient to a specific drug (or its dosage) could be predicted. Discovery of such relationships, however, is a complex task and requires sophisticated pattern recognition systems. The following methods are some of the popular methods that could be used for this purpose: artificial neural networks, support vector machines, projection pursuit, genetic algorithms, fuzzy logic, Bayesian methods. Often, a hybrid of these systems is the method of choice.

13.6.2 Differential Gene Expression

The methods used for this type of analysis include fold-change, statistical tests of hypothesis, and Bayesian methods. In all these methods, two populations are

interrogated—the population of the case-specific samples and the population of the control samples. For pharmacogenomics applications, the case and control populations correspond to the individuals who received the drug or the placebo, respectively. Alternatively, it could correspond to different individuals who received the drug at two different doses. The objective of differential gene expression is to discover whether there is a significant difference between the case and control samples, as viewed by the drug response.

The fold-change uses the minimal information (i.e., the mean values) for two populations and sets a flag if the ratio of the mean of the measurements for the two populations exceeds a certain threshold (e.g., 2). Obviously, this method is weak as it only uses the sample mean and does not take the higher-order statistical measures into account. An improvement to the fold-change method is a statistical test that employs the standard deviation information besides the mean. *T*-test is the most popular test for differential expression. It assumes Gaussian distributions with different means and the same standard deviations for case and control populations. *T*-test is stronger than fold-change but suffers from the assumption it makes that the standard deviations are invariant. In other words, two gene populations that have similar intensities but widely different variances will end up being undetected by a *T*-test. The other problem with the *T*-test is its reliance on the estimate of the standard deviations. This estimate is known to have wide confidence bounds if the number of samples is low. Bayesian methods are improvements to *T*-test, particularly in the case of small sample size, as they have built-in regularization machinery.

13.6.3 Differential Quantitative Genotyping

Similar to gene expression, differential quantitative genotyping has a great value for pharmacogenomics. Here, the signal is the minor-allele frequency (MAF). The objective is to detect a significant difference between the MAF of the case and control populations. All the methods that were previously mentioned are applicable to differential quantitative genotyping. In addition, since MAF is limited to the range of [0, 0.5], the performance of the aforementioned methods can be boosted by tailoring the algorithms to use such prior information.

13.6.4 Haplotype Map

It was mentioned that representative SNPs of haplotype blocks could play a major role in pharmacogenomics by allowing researchers to use a smaller set of SNPs in association and linkage studies. The three popular ways by which haplotype boundaries can be detected are Confidence Intervals [21], Four Gamete Rule [22], and Solid Spine of LD.

In an attempt to systematically identify the haplotype blocks and assign their corresponding tag SNPs, the International Haplotype Map (HapMap) Project was established in October 2002. This project is a collaboration of scientists from the United States, Canada, China, Japan, the United Kingdom, and Nigeria. Table 13.2 shows the breakdown of the chromosomes and percent coverage on genome for HapMap participants (http://www.hapmap.org).

TABLE 13.2
Breakdown of the Chromosomes and Genome Coverage for the HapMap

Country	Genome Coverage	Chromosomes
United States	32.4%	4q, 7, 8q, 9, 12, 18, 22, X, Y
Canada	10.1%	2, 4p
China	9.6%	3, 8p, 21
Japan	24.3%	5, 11, 14, 15, 16, 17, 19
United Kingdom	23.7%	1, 6, 10, 13, 20

13.7 DISCUSSION: ETHICAL ISSUES AND ALTERNATIVE RESEARCH

13.7.1 Biases for Racial and Ethnic Groups

Race is a vaguely defined concept. Often, ethnicity is used instead to describe changes developed by segregation during the evolutionary process. There are reasons to believe that pharmacogenomics may result in development of different drugs for large subpopulations (e.g., Caucasians, Africans, and Asians). Such reasons include the difference in allele-frequencies as well as lifestyle and societal aspects. The ethical issues that would arise for such stratification include the selection of candidates for clinical trials and the financial impetus for the development efforts, as the main thrust of the pharmaceutical companies is the bottom-line profit, and not all subpopulations have the same level of financial power. The other concern is the issue of other ethnic groups that are either excluded from the above or the overlap of some. The latter is particularly of interest in recent years, where cross-cultural marriages have increased.

13.7.2 Insurance

The issue with insurance companies is worth investigating from two perspectives. First, patients with genotypes that reduce the probability of the appropriate therapy could be at risk of being denied coverage [6] or losing an existing coverage as well as becoming targets of stigmatization [9]. Second, the pharmacogenomics drugs and pharmacogenomics-based treatments may be more expensive than the corresponding conventional ones [3]. In such cases, the insurance companies might be more inclined to approve the conventional drugs, even if the benefits of the pharmacogenomics counterparts are higher.

13.7.3 Gender Differences

Differentiating drugs based on gender is not a new development prompted by pharmacogenomics. Some drugs are already presented in different packages, although essentially with similar active ingredients, for men and women (e.g., Rogaine). Since

the cornerstone of pharmacogenomics is genetic differences, and since by definition, there are large genetic differences between men and women (via the X/Y and methylation patterns on X/X), there are reasons to believe that drug development for males and females could result in concerns similar to what was stated for different racial groups.

13.7.4 Security of Genome Data Banks

Genetic tests demand data banks for storage and coordination. The security of these data banks is of paramount importance. In the wrong hands, genetic information can be used to obtain immoral advantages (on a population or individual basis). Since no security system is "hackerproof," data bank security is a serious concern that arises with pharmacogenomics, as it relies heavily on data obtained from genomic tests.

13.7.5 Biopsy from Healthy Tissues

Case-control studies are among the most popular ways of associating phenotypes with genetic causes. For example, in a case-control gene-expression study, expression levels of many genes are interrogated identify genes that best correlate, as viewed by their differential expression level in malignant versus benign tissues. Thus, such studies depend strongly on the existence of biopsy samples from benign tissues. Such tissues do exist in historic repositories, (e.g., paraffin-embedded samples), and their availability makes these samples unique and highly useful for performing biological case-control studies. The only complication while dealing with paraffin-embedded samples is the need for a sensitive assay [10], as the RNA molecules in the mentioned samples are subject to large degradations. Healthy tissue samples can also be acquired if, at the time of surgery (if such procedure is inevitable), a sample of healthy tissue is taken along with the diseased sample. Aside from cases such as these, taking biopsy samples from known benign tissues, just for the sake of making a predictive model, would be unreasonable and unethical [7].

13.7.6 Diverting Attention from Alternative Research

The overexcitement factor could be dangerous for pharmacogenomics. This factor can be criticized from two different angles: insufficient grounds and impeding alternatives. With respect to insufficient grounds, the problem is setting high expectations and not being able to deliver. Such expectations may result in raising suspicions of the field and create negative public opinion about the appropriateness of such techniques. This is analogous to what happened to neural networks in the 1960s and gene therapy in the 1990s. In the case of neural networks, it did return with more limited expectations and a more established backbone after the passage of nearly two decades. However, in the case of gene therapy, no signs of a return are yet evident. With regard to the second problem (i.e., impeding alternatives), the argument is that given a fixed industrial or governmental budget by diverting a large portion of the funds to what sounds promising (although its support is merely based on a vision and not much data), for example, a panacea version of pharmacogenomics,

the funding for alternative technologies in medicine may be more limited and thus delay or prevent the evolution of such methods.

13.8 CONCLUSION

There are numerous controversies concerning the utility of pharmacogenomics. While at a small scale and for a limited number of drugs, it may be possible to use genomic information to provide drugs that are more potent and have fewer side effects for certain individuals, generalizing this idea to the whole genre of medicine and treating pharmacogenomics as a panacea is the subject of much speculation and debate. Aside from the technical and ethical arguments, large pharmaceutical companies have economic disincentives to adopt pharmacogenomics if that approach proves to be less profitable for them than conventional drugs, unless they undergo a major paradigm shift. These economic disincentives suggest that the initial attempts to launch pharmacogenomics could be limited to small or new enterprises. Nonetheless, the amount of data for and efforts toward pharmacogenomics is growing. All of this will aid in establishing realistic boundaries on the expectations from this field.

ACKNOWLEDGMENTS

I extend my gratitude to Drs. David Barker, Michael Barnes, Darryl León, and Scott Markel for their invaluable discussions and comments.

REFERENCES

1. Roses, A. D. 2002. Genome-based pharmacogenetics and the pharmaceutical industry. *Nat Rev Drug Discov* 1:541–9.
2. Hackett, J., and L. Lesko. Microarray data—the US FDA, industry and academia. *Nature Biotechnol* 21:742–3.
3. Rothstein, M. A. 2003. *Pharmacogenomics: Social, ethical, and clinical dimensions*. New York: Wiley-Liss, Inc.
4. Marshall, A. 1998. Laying the foundations for personalized medicines. *Nature Biotechnol* 16:6–8.
5. Marshall, A. 1998. Getting the right drug into the right patient. *Nature Biotechnol* 16, Suppl. no. 2:9–12.
6. Shah, J. 2003. Enoconic and regulatory considerations in pharmacogenomics for drug licensing and healthcare. *Nature Biotechnol* 21:747–53.
7. Nebert, D., L. Jorge-Nebert, and E. Vesell. 2003. Pharmacogenomics and individualized drug therapy. *Am J Pharmacogenomics* 3:361–70.
8. Genetic variation in drug development. 1998. *Nature Biotechnol* 16, Suppl. no. 2:16–9.
9. Williams-Jones, B., and O. P. Corrigan. 2003. Rhetoric and hype, where's the ethics in pharmacogenomics? *Am J Pharmacogenomics* 3:375–83.
10. Fan, J. B., J. M. Yeakley, M. Bibikova, E. Chudin, E. Wickham, J. Chen, D. Doucet, et al. 2004. A versatile assay for high-throughput gene expression profiling on universal array matrices. *Genome Res* 14:878–85.

11. Marshall, A. 1998. One drug does not fit all. *Nature Biotechnol* 16:1.
12. Humma, L. M., V. L. Ellingrod, and J. M. Kolesar. 2003. *Pharmacogenomics handbook*. Hudson, OH: Lexi-Comp.
13. Sesti, F., G. W. Abbott, J. Wei, K. T. Murray, S. Saksena, P. J. Schwartz, S. G. Priori, D. M. Roden, A. L. George, Jr., and S. A. Goldstein. 2000. A common polymorphism associated with antibiotic-induced cardiac arrhythmia. *Proceedings of the National Academy of Science* 97:10613–18.
14. Splawski, I., K. Timothy, M. Tateyama, C. Clancy, A. Malhorta, A. Beggs, F. Cappucino, G. Sagnella, R. Kass, and M. Keating. 2002. Variant of SCN5A sodium channel implicated in risk of cardiac arrhythmia. *Science* 297:1333–6.
15. Evans, W. E., Y. Y. Hon, L. Bomgaars, S. Coutre, M. Holdsworth, R. Janco, D. Kalwinsky, et al. 2001. Preponderance of thiopurine S-methyltransferase deficiency and heterozygosity among patients intolerant to mecaptopurine and azathiopurine. *J Clin Oncol* 19:2293–301.
16. McCarthy, J., and R. Hilfiker. 2000. The use of single nucleotide polymorphism maps in pharmacogenomics. *Nat Biotechnol* 18:505–8.
17. Beck, S. F., A. Olek, and J. Walter. 1999. From genomics to epigenomics: A loftier view of life. *Nat Biotechnol* 17:1144.
18. Rhind, S., T. King, L. Harkness, C. Bellamy, W. Wallace, P. DeSousa, and I. Wilmut. 2003. Cloned lambs—lessons from pathology. *Nat Biotechnol* 21:744–5.
19. Hodgson, J., and A. Marshall. 1998. Pharmacogenomics: Will the regulators approve? *Nat Biotechnol* 16, Suppl. no. 2:13–21.
20. Ledley, F. 1999. Can pharmacogenomics make a difference in drug development? *Nat Biotechnol* 17:731.
21. Gabriel, S. B., S. F. Schaffner, H. Nguyen, J. M. Moore, J. Roy, B. Blumenstiel, J. Higgins, et al. 2002. The structure of haplotype blocks in the human genome. *Science* 21:2962225–9.
22. Wang, N., J. M. Akey, K. Zhang, R. Chakraborty, and L. Jin. 2002. Distribution of recombination crossovers and the origin of haplotype blocks: The interplay of population history, recombination, and mutation. *Am J Hum Genet* 71:1227–34.
23. Kling, J. 2003. US FDA contemplates collection of pharmacogenomic data. *Nat Biotechnol* 21:590.
24. Smaglik, P. 2001. Pharmacogenetics initiative galvanizes public and private sectors. *Nature* 410:393–4.
25. Collins, F., E. Green, A. Guttmacher, and M. Guyer. 2003. A vision for the future of genomics research. *Nature* 422:835–47.

14 Target Identification and Validation Using Human Simulation Models

Seth Michelson, Didier Scherrer, and Alex L. Bangs
Entelos, Inc.

CONTENTS

14.1 INTRODUCTION

The principles of systems biology, as recently described by Dr. Leroy Hood of the Institute of Systems Biology, define a process of research that requires us to rethink our now obsolete definition of systems biology: systems biology is more than the mere application of informatic technologies to biological research. According to Dr. Hood, the process that defines systems biology must be hypothesis driven, quantitative, integrative, dynamic, and global (in the sense that it employs all relevant data in a unified and coherent theoretical structure). One discipline that has emerged as a primary contributor to these efforts is the mathematical modeling of dynamic biological systems, especially in the context of human disease. As with any new and novel research strategy, the ultimate utility of systems biology will be measured by its ability to solve real-world problems in medical research. One such problem is the identification and validation of drug targets in pharmaceutical research.

Identifying potential drug targets is a primary goal for the pharmaceutical researcher. In the age of high-throughput technologies, one success marker of this quest has been the number of data points one can collect and store. Hampered by a lack of pathophysiological context in which to interpret one's results, searching those databases and deriving from them adequate information regarding a potential target's

role in a given disease is a daunting task. Thus, validating a target with any degree of predictive confidence remains a major bottleneck for the industry. To understand, at the most fundamental level, the physiologic implications of a potential target's function, one must know what is known about that target in the context of the disease and its progression. However, and more importantly, one must also know what is not known about it. Returning to Dr. Hood's definition of systems biology, one must be able to integrate, into an intellectually coherent and rigorous structure (i.e., a model), all the relevant pieces of information regarding a presumptive target's role in a disease and identify from that theoretical structure what is still to be understood about that molecule and its activity in the disease state. Based on a systematic analysis of that infrastructure, one can decide what pieces of information are most vital to the validation process and pursue them directly. For this process to work optimally, the scope and breadth of a model must be determined by the decisions it is intended to support and should, at the very least, be able to represent disease progression as a dynamic evolution of pathology. Ideally, this will take place in a system that represents human physiology and can replicate human clinical outcomes, giving a unique window into how manipulating a target would behave in the clinic well before a drug reaches that stage.

14.1.1 Modeling: A Methodology for Idealizing a System

To understand how mathematical modeling supports the processes defined by Dr. Hood's systems biology, one must first understand what mathematical modeling is. First and foremost, mathematical modeling is a means to an end. It explicitly idealizes and characterizes a system in sufficient detail so that the research scientist can directly apply it to his or her research question. For example, every biochemistry laboratory in every pharmaceutical company has, hanging on its wall, a poster showing the interaction of biomolecular entities in a given cell under a given condition. These posters are models and as such are idealizations of a complex dynamic that underlies a biological process. Based on this type of theoretical infrastructure, one can identify the key players and interactions driving a biological dynamic and from that, as a mental exercise, project the hypothetical behavior of the system under a given condition. The problem with the poster model is that it is, by its very nature, a snapshot of one particular state of the system and thus not dynamic. It is also not particularly quantitative, thus violating two of Dr. Hood's key criteria for inclusion in the systems biology process.

Taking the poster-modeling paradigm to the next level, some investigators are using statistical modeling to identify potential correlations in datasets of interest, thus establishing potential relationships between biomolecular entities in a given system. Others are using sophisticated pattern-recognition techniques to infer pathway dependencies from these same data [1–3]. Being able to identify the dynamics of the system and their subsequent quantification is wholly dependent on the conditions under which these data are acquired—the conditions of the cell culture/tissue samples and the assays designed to sample them.

Alternatively, one can construct a mathematical model of a given system by rigorously characterizing the underlying biological structure and its related dynamics into a set of well-defined equations. By representing the underlying biological processes as a

collection of time-dependent mathematical structures, the resulting system of equations also idealizes the disease biology and its pathophysiological evolution.

A number of organizations, mainly academic, are presently taking a first principles or bottom-up/data-driven approach to developing a mathematical infrastructure for complex biological systems (an abbreviated sampling of the modeling space is provided in table 14.1). These groups model the components of a system one by one and then try to integrate them into a coherent biological whole. The major directive of this effort is to understand and uncover all the component parts of a system, model each one discretely, and then build the system's control circuitry to link them. Using this approach, a detailed representation of each component and its function is created.

TABLE 14.1
A Sample of Organizations Using Bottom-Up Mathematical Modeling and Biosimulation-Based Approaches to Address Problems in Pharmaceutical Research

Data Driven	Mission	Model Systems or Application Areas
Alliance for Cellular Signaling (http://cellularsignaling.org)	Identify and integrate proteins contributing to cellular signaling	B lymphocytes; cardiac myocytes
Cell Systems Initiative (http://csi.washington.edu)	Establish a comprehensive theory of the cell; create predictive models	Dendritic cells; T cells
Caltech ERATO Kitano (http://www.cds.caltech.edu/erato/)	Provide software infrastructure that enables model and resource sharing	Software platform: System Biology Workbench and System Biology Mark-Up Language (SBML)
E-Cell Project (http://www.e-cell.org/)	Provide framework and software to simulate cellular behaviors	*E. coli*; erythrocytes; neurons
Institute for Systems Biology (http://www.systemsbiology.org/)	Study genes and proteins simultaneously by perturbing model organisms	Microbes; immune, cancer, and stem cells
Molecular Sciences Institute (http://www.molsci.org/)	Focus on predictive biology; create tools for the analysis, design, and engineering of biological systems	*E. coli*; yeast
Gene Network Sciences (http://www.gnsbiotech.com/)	Build models of cell function and human biology to support pharmaceutical research and development	Human cancer cell; canine ventricle
Genomatica (http://www.genomatica.com/)	Provide software and models of cell metabolism to enhance bioproduction and antimicrobial discovery	Metabolic network models for a variety of microbes
Virtual Cell Project (http://www.nrcam.uchc.edu/)	Provide modeling and simulation framework and software	Intracellular processes such as calcium dynamics and nuclear envelope breakdown

Two organizations, however, have taken a top-down/hypothesis-driven approach to modeling these kinds of complex biological systems and their dynamics (table 14.2). This approach starts by defining a general set of behaviors indicative of the disease state. Then, within these constraints, one defines the set of nested subsystems whose control and context are required to reproduce those particular behaviors. Each subsystem is then deconstructed in greater detail from the top down, going from whole-body dynamics to the molecular level. The depth of the modeling (i.e., its detail) is determined either by the limits of our knowledge or by the depth necessary to replicate a given biological behavior.

TABLE 14.2
A Sample of Organizations Using Top-Down Mathematical Modeling and Biosimulation-Based Approaches to Address Problems in Pharmaceutical Research

Hypothesis Driven	Mission	Model Systems or Application Areas
Entelos, Inc. (http://www.entelos.com/)	Develop effective new disease treatments and reduce the time and cost needed to develop them	Adipocytes; asthma; diabetes; obesity; rheumatoid arthritis
Kenna Technologies, Inc. (http://www.kennatechnologies.com/)	Improve decision-making based on complex biological and medical data	Osteoporosis; otitis media; oxidative stress; periodontal disease

Once a model has been developed, one can analyze it in a number of ways. If it is mathematically tractable, that is, small enough and describing a system of linear dynamics, one can apply the well-developed practices of theoretical mathematics to derive a closed form solution of the system. If that is possible, the product of the analysis will be a set of fully parameterized equations, which can, for any given parameter vector and with the aid of a simple calculator, establish the dynamic state of the system at any time, *t*. However, even if deriving a closed-form solution for the entire system is impractical, one can analyze the basic dynamics of the system by assuming steady-state conditions, and deriving expressions for fixed-point behaviors (e.g., see [4]) around the steady state. Identifying the dependence of those dynamics on the parameter vector will give the modeler an insight into the forces driving the system and its behavior.

Complex biological behavior is controlled by a network of interacting components (e.g., tissue systems, cells, cytokines, receptors, transcription factors, etc.). Their physiologies and dynamics determine the makeup of the mathematical model. If one ignores the issues of dynamics and time dependency in the model, thus violating one of Dr. Hood's criteria for the new systems biology, one can still analyze the basic structure of the network using graph theory [5–8]. For example, Anderson and Hunt [8] analyzed the degree of connectedness and closeness (in a graph theoretic measure of distance) of a complex mathematical model of human metabolism and obesity (the Entelos® Obesity PhysioLab® Platform). Their stated aim was to use

this methodology to determine the "pressure points" in the system that are most likely to lend themselves to manipulation and thus yield potential drug targets. They found that while the degree of connectedness of the individual elements could be quite misleading due to nonspecific activities of highly integrated control structures, there do exist "gatekeepers" in the system that oversee access to the dynamic links that control each subsystem's behavior. These gatekeepers actually provide pressure points for manipulating the system and do suggest that a viable target may reside in their midst.

14.1.2 Biosimulation: A Means of Characterizing the Solution Set of the Model

Typically, for human disease, the systems one encounters are much too large and far too complex for the modeler to derive simple closed-form solutions and/or to perform simple steady-state analyses. As just noted, these systems usually include significant nonlinear dynamics, feedback controller systems, and time dependencies that are not easily modeled with pencil and paper. However, one can, by applying numerical differential equation solvers, still attack the problem by numerically estimating, under a variety of conditions, a set of systemwide solutions. Numerically solving the equations that make up a model of human disease and projecting these solutions forward in time is termed *predictive biosimulation* [9,10].

By being hypothesis driven, quantitative, dynamic, and inherently integrative—and depending on the scope of the model—global, predictive biosimulation is able to fulfill all of Dr. Hood's criteria for the new systems biology paradigm. In addition, when taking advantage of all the recent advances in computer systems development, especially processor speed and cluster computing, it is fast.

More formally, predictive biosimulation is a process that solves, by numerical approximation (e.g., Runge-Kutta numerical integration algorithms), the complex equations that describe a system's behavior. Since it is so fast and allows for maximum flexibility in specifying the hypothetical conditions under which a system will evolve, predictive biosimulation allows researchers to rapidly develop and plumb *What if* scenarios surrounding the potential influences of a presumptive target on disease progression. In addition, the researcher can easily identify any inconsistencies in the applicable data and, more importantly, any misconceptions derived therefrom that could lead to flawed decision making. Once these gaps and inconsistencies are uncovered, hypothetical exploration *in silico* allows the researcher the luxury of predetermining the impact of a particular physiological hypothesis on the ultimate course of the disease. Thus, data collection efforts can be better focused, assays can be better designed, and the resultant data can be more efficiently interpreted. By iteratively focusing modeling and data collection efforts on subsystems with the greatest impact on the key aspects of a disease's phenotypes, predictive biosimulation can help to clarify a system's complexity and, through that effort, distinguish causal factors from mere correlates in underlying pathophysiology.

By providing an environment that not only manages data and information but also helps characterize and manage the inherent unknowns associated with such a complex knowledge space, modeling and biosimulation provide a systematic

approach to research that facilitates a hypothesis-driven strategy to experimental design and exploration. Managing these hypotheses under the aegis of the systems biology approach thus fulfills the first of Dr. Hood's criteria.

14.2 THE CHALLENGE OF IDENTIFYING AND VALIDATING A DRUG TARGET

While the resources required to successfully bring a new drug to market are enormous, averaging 14 years and 802 million dollars, these costs continue to rise [11–13]. A significant contributor to this cost comes from the large number of drugs that fail in the clinic (e.g., ~53% of compounds fail during Phase 2). Of these failures, a significant proportion are due to unanticipated systemwide effects. Biological redundancies, unexpected control mechanisms, and multiple physiological timescales (e.g., a mismatch between the timescale of a drug's effects and the timescales that characterize disease pathophysiology) can contribute to these unexpected clinical effects.

In the process of drug discovery, target selection consists of two main steps: target identification and target validation; the ultimate success of a new drug relies almost entirely on the quality of the target that it modulates. Thus the key to successful target selection becomes the predictability of ultimate clinical outcome.

In today's drug-discovery environment, researchers have access to an enormous amount of data, most of it coming from genomic, proteomic, *in vitro,* and *in vivo* animal experiments. The challenge for the pharmaceutical researcher is how to most efficiently use all these data to close "the predictability gap" (i.e., how does one translate *in vitro* and *in vivo* data into a clinically predictive context?). The key to meeting this challenge falls on how one goes about curating these data and extracting from them the bits of biological insight that will ultimately help to identify a novel, clinically relevant target, that is, a molecular entity that both plays an important role in a particular disease's pathophysiology and is accessible to exogenous modulation (i.e., is druggable). Using traditional approaches, targets are identified and evaluated using *in vitro* systems such as cell lines or primary cell culture. These *in vitro* systems suffer because they remove relevant cells from potentially important regulatory controls. In other cases, *in vivo* systems, such as animal models (e.g., knockouts and/or transgenics), are often used for this exploratory research. *In vivo* animal models present significant limitations as well. Typically they represent an artificially induced state meant to mimic the disease they intend to represent. Furthermore, even if some of the pathways involved in the disease process are conserved between species, their regulation and equilibria are most likely not [9].

In trying to understand the underlying biology of human disease, the pharmaceutical research community has come to increasingly depend on technologies that characterize the genome and proteome, that is, they are taking a bottom-up approach to disease. These automated and high-throughput technologies have produced a deluge of sometimes incoherent biological datasets. While these data, depending on the context of the sampling procedures (i.e., which tissues or cells, taken from which samples, under which conditions?), can potentially correlate changes in gene and

protein expression with a particular disease state, they are typically incapable of independently and directly identifying causal relationships. In other words, these data cannot distinguish between changes caused by the disease and those that cause the disease. The data are also not able to predict how these changes, which are usually observed in isolated tissue samples or cells, may affect, or be affected by, the system as a whole. The second of Dr. Hood's criteria is missing: these approaches are not integrative. To integrate these data into the proper perspective, one must first embed them in a coherent human biological context—one that includes a dynamic description of all the relevant regulatory mechanisms surrounding the disease, its onset, and its progression. Only within this larger, human context can researchers efficiently interpret the massive amounts of data at hand and gain the in-depth knowledge needed to successfully develop a drug.

For example, consider the modeling efforts surrounding IL-5 inhibition as a treatment strategy for chronic asthma. The original assumption underlying these clinical efforts was that by reducing IL-5 levels, one can reduce airway eosinophilia, which in turn should yield an efficacious effect in asthma. Modeling studies predicted instead that although this therapeutic strategy would indeed reduce eosinophilia, it would produce little improvement in lung function during an acute asthma attack because of other redundant inflammatory pathways [14,15]. The lung function result was later born out by clinical data [16].

14.2.1 Identifying the Key Biomolecular Entities Involved in a Disease's Pathophysiology

By way of example, consider a cell that is inappropriately activated by an endogenous signal, and, in response, upregulates and secretes a particular cytokine. Suppose further that this cytokine stimulates a second cell that then upregulates the expression of another cytokine and its autocrine receptor. And suppose further that this secondary cytokine signals another cell in the disease cascade, which goes on to alter its physiology in a deleterious way. While a genomic snapshot of this process may yield a set of differentially expressed genes that correlate strongly with the disease state, do these genes necessarily contain the optimal therapeutic target, or is it possible that the most influential target actually lies upstream (e.g., the receptor on the cell receiving the initial signal) or downstream of the cascade (e.g., the cytokine receptor on the third cell)? Without this larger, more integrated physiologic context, one may believe that he or she has identified many novel candidate "targets" but has little guidance as to which ones will be the most effective in treating a given disease. Consequently, pharmaceutical researchers must sift through thousands of potential failures before achieving likely clinical success. The earlier one can drop ineffective targets, the greater the savings in time and money will be.

14.2.2 The Context of the Biology—The Pathway

While it is vital that the evaluation of a new target determines how modulating that target may impact clinical outcome, it is equally important to understand which pathways are driving this effect. Predictive biosimulation provides researchers with

a unique environment for studying both of these questions. For example, consider the scenario previously outlined. How much does this particular pathway contribute to the conglomerate physiology of the disease? Is there a feedback mechanism in place that amplifies or damps this signal? Are there redundant or backup pathways that will mitigate inhibition of this particular cascade? By systematically modulating a presumptive target's *in silico* activity within the integrated context of whole-body human biology, the impact of each pathway on clinical outcome can be more completely assessed.

14.2.3 The Logic of the Biology—The Dynamic Control Circuitry

Once a target has been selected, the focus of the discovery process shifts to chemistry. The role of the chemist is to develop a molecule that interacts with that target in the most appropriate manner. But what criteria should one use to optimize candidate selection? In other words, what does "appropriate" mean in this context? Currently, the objective criteria for advancing compounds into the early drug-development process (e.g., pharmacokinetics, safety studies, animal studies, etc.) are incomplete and poorly reflect the compound's clinical effect. We know this because only 1 of 5,000 potential drug candidates is likely to be approved for therapeutic use [17,18]. To overcome this hurdle, the pharmaceutical researcher must be able to establish, quantitatively, the nature of the presumptive target in the context of the disease and its dynamic regulation. Only then can he or she determine its impact on disease progression and pathophysiology. In addition, to understand the impact of a potential therapy on clinical outcome, one must also determine the ultimate effects of a target's dynamic modulation on disease evolution and reversal. Using *in silico* modeling and predictive biosimulation, one can, *a priori,* develop, mathematically express, and explicitly test hypotheses regarding a target's activity in the context of a clinical response. However, since these findings typically are based on the assumptions of the hypotheses at hand, these *in silico* results must be considered as conditional answers such as, "If it is true that Molecule X in Pathway Y alters the physiology of Cell Z in thus and such a way, then Molecule X is a potentially valuable target to pursue." These conditional solutions thus establish the necessary conditions for the target validation assays in the lab.

14.2.4 The Pressure Points of the System—Regulation of the Control Circuitry

While building a model, and during any *post hoc* analysis, one can explicitly identify and explore gaps in the extant data for potential new target opportunities. For the purposes of this chapter, we term these gaps *knowledge gaps*. By clearly and completely mapping the contextual knowledge space surrounding disease pathophysiology, mathematical modeling can help the research scientist focus on and resolve the most important of these, that is, those that yield the greatest impact on the clinical outcome. By focusing wet-lab experimentation on these key pathways, that is, those most likely to drive disease pathogenesis and progression, and by ensuring that the

right experiment is performed in the right context, modeling and biosimulation help to make the target validation process more predictive. In addition, because a model tells the pharmaceutical researcher not only *what* is happening but *why* it is happening, these types of multidisciplinary efforts (*in silico* studies linked to experimental research) cannot help but aid in the understanding and interpretation of key experimental results.

14.2.5 Patient Variability and Target Validation

If a molecule is on the cause–effect pathway of the disease, it may be considered a presumptive target. However to be a commercially viable target, it must be both druggable and expressed in a reasonably large proportion of the patient population. When dealing with the effects of a biochemically active compound, it is not sufficient that the molecule merely modulate the activity of the target, but it must exhibit all the characteristics of a medicine, that is, that it is safe, effective, and able to gain access to the target site in sufficient quantity that it is able to exert its therapeutic effects. Mathematically speaking, this is a multivariate optimization problem. The problem is that patient variability can creep into any of these aspects of medicinal character and must be accounted for when selecting a target and its modulating medicine.

How does one account for this patient variability in the drug-discovery process? The obvious answer is through the execution of predictive bioassays. However, one unfortunate consequence of modern drug discovery is that the bulk of the research and development is not performed in human individuals but rather in cellular systems and test animals. The point of clinical trials is to prove that what we see in test animals and in *in vitro* cellular preparations is, in fact, generalizable to an outbred and variable population called humans. Since lab rats are typically derived from inbred strains, we can control in our animal studies, to some extent, the effects of genetic variation. From a scientific point of view this is a good thing to do, as it controls experimental variance and helps to separate the signal from the noise. However, our ability to generalize these scientific findings to humans and their inherent variability is severely hampered. In addition, we control the environmental variability each test animal experiences by keeping it in a controlled laboratory animal care facility. Again, from a scientific point of view this is a good thing to do. However, this is surely not the case one observes with the human population.

The fact that the human population is not composed of genetically controlled, environmentally sequestered subjects but is an outbred, highly variable composite of distinctly unique individuals is a constant difficulty for the clinical research scientist. To better understand the sources and theoretical constructs of this variability, one must capture, in explicit, quantifiable hypotheses, the effects of genetic and environmental variations on the underlying physiology and account for them in disease expression and progression. Mathematically, this combinatoric set of interdependent hypotheses can be represented as specific parameter vectors in any underlying mathematical model. How one can use predictive biosimulation to both represent and explore the impact of this variability on target selection and validation is presented in greater detail in the case studies next.

14.3 THE ROLE OF PREDICTIVE BIOSIMULATION IN TARGET IDENTIFICATION AND VALIDATION

Once a model has been specified, built, and validated, one can use biosimulation as a means of solving the model equations in the context of a directed research effort. Given the focus of this book, we center our discussion on the application of biosimulation to the identification and validation of disease targets and their dynamic control. As previously noted, the results generated by such an *in silico* study are necessarily conditioned upon the underlying hypotheses being tested and are, in and of themselves, necessary but insufficient to the decision-making processes surrounding target identification, validation, and prioritization. To adhere even more firmly to Dr. Hood's definition of the processes defining systems biology, it is incumbent on the researcher who is using this particular hypothesis-driven approach to explicitly test his or her assumptions in the appropriate physiological context. Correct use of biosimulation can provide sufficient guidance and scientific explanation to the researcher to ensure that he or she can test these hypotheses directly in the lab. By explicitly defining the biology of interest in the context of the disease, the biosimulation produces a set of recommendations for focused laboratory experiments that are sufficiently predictive to validate the role of a specific target in the pathology of the disease.

14.3.1 Capturing Patient Variability in the Biosimulation Milieu—The Virtual Patient

To identify and validate a target, as in all medical research, it is essential that one be able to explicitly formulate, represent, and test multiple hypotheses that underlie the variability observed in a particular patient population. For example, variations in the expression levels of cell-surface receptors; metabolism rates of drugs; patient-specific absorption, distribution, metabolization, and excretion; pharmacodynamic variations in dose–response relationships; and different phenotypic behaviors can all affect the efficacy of a therapeutic intervention. The question then arises as to how one explicitly defines the underlying hypotheses one makes concerning the pathophysiology of the disease. Given the existence of a mathematical model, one can characterize each hypothetical alternative as a vector of explicit model parameter values. In human system models, these vector constructs are termed "virtual patients." However, before one can explore the impact of these hypotheses on the viability of a putative target, it is necessary that each virtual patient represent a valid piece of biology. To be considered a valid virtual patient, the underlying biological attributes of the virtual patient must first fall within known biological ranges and its simulated behaviors must match known human, *in vitro*, and *in vivo* behaviors. For example, the response to an oral glucose tolerance test in a virtual diabetic must both correspond to observed responses in the human population and exhibit biologically reasonable dynamics in the underlying physiologies of the liver, adipose, gut, and pancreas.

These constructs can then be used to investigate different hypotheses regarding the impact of a proposed physiologic modulator (e.g., 100% inhibition of a presumptive target in a particular cell under a particular condition) in patients of different character (e.g., a virtual diabetic who has accelerated gluconeogenic pathways vs. one who has

normal glucose production but has impaired insulin production). By using multiple virtual patients to represent specific realizations from among all possible biological variations, researchers can begin to understand the types of variability that they may encounter in the clinic. Given the transparency of the mathematical model, researchers can also begin to understand how to distinguish patient types that respond differently to proposed therapeutics (e.g., well vs. marginal vs. not at all).

To adequately represent the highly variable nature of the human patient population, one must, in addition to representing variations on an individual basis, extend his or her theoretical methodologies to capture and characterize population variability across an entire set of virtual patients. These sets of virtual patients are termed *virtual patient cohorts*, and it is here that the gap between possibility and probability must be bridged. To meet this challenge, Entelos has developed a series of methodologies for bridging the gap between the existence of a phenotype (i.e., the virtual patient) and its prevalence in an epidemiological profile of the clinical population.

An alternative approach would be to generate a large number of virtual patients to create a population that is epidemiologically comparable to the real population of patients in size and characteristics [19]. This approach, however, is more computationally intensive and does not give any real advantages over a prevalence weighted population scheme.

14.3.2 Applying Predictive Biosimulation to Target Identification

Applying *in silico*–based research to assess a target's relevance in a particular disease involves the following series of well-defined steps (see fig. 14.1 and fig. 14.2). Detailed case studies of these processes as implemented on the Entelos PhysioLab platforms are given next.

14.3.2.1 Step 1: Define Target Functions and Assess Their Potential Clinical Impact

To evaluate a specific target, the pharmaceutical researcher must first conduct a systematic analysis of any and all relevant data regarding the presumptive target's activities, that is, where it operates, under what conditions it operates, how it is controlled, and how it alters the physiology of the cells in question. If the target is truly novel and not fully characterized, then, based on any indirect evidence regarding it and its activity, a set of hypotheses can be generated and mathematically characterized. Once characterized, one must quantify the impact of modulating the target on each function or hypothesis individually. The outcome of this initial analysis is an inventory of detailed descriptions for each target function, including the scientific rationale describing the underlying assumptions, observations, and findings regarding those characterizations.

14.3.2.2 Step 2: Modify the Model and Simulate Target Modulation

Once a target has been fully specified, one must modify the mathematics in the model to explicitly represent the physiological impact of those relevant target functions. If

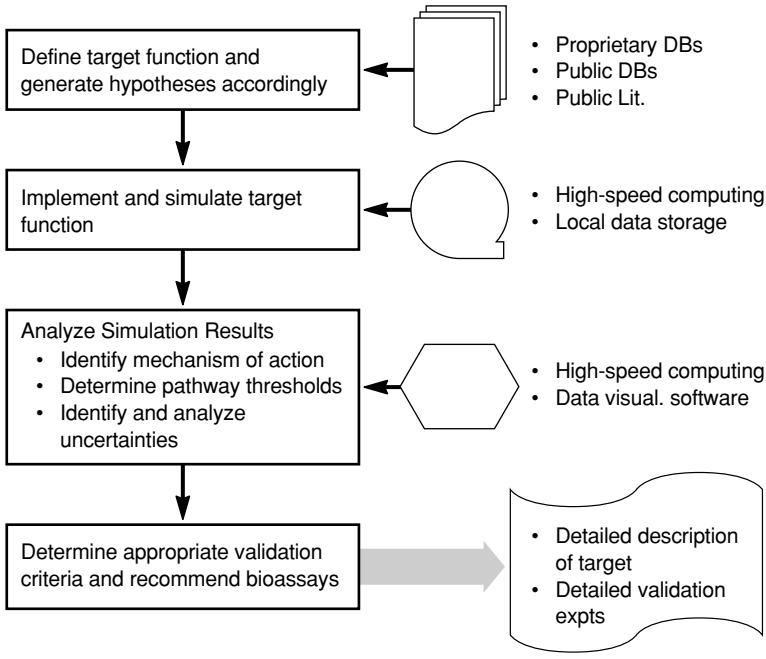


FIGURE 14.1 Flowchart of data/information requirements driving target identification and validation *in silico*.

the dynamics are not explicitly represented in the model, then appropriate surrogates must be identified and modified appropriately. In conjunction with adding target-specific characterizations to the model, one simultaneously develops a family of *in silico* experimental protocols that characterize target modulation in the context and condition of the disease for each hypothesized virtual patient. The impact of target modulation, on each pathway individually and in combination, can then be assessed and, based on these results, its clinical impact predicted.

The outcome of this effort is a structured collection of *in silico* experiments that explicitly characterize target modulation in the context of individualized disease. At Entelos, the results of each simulated protocol for each virtual patient are stored in a database, which allows the research scientist to recall and analyze every model variable at any point in the simulated experiment. This transparency of information forms the basis of a subsequent pathway analysis.

14.3.2.3 Step 3: Analyze and Evaluate Biosimulation Results

14.3.2.3.1 Identify the Target's Mechanism of Action

The goal of analyzing the simulation results is to identify causal linkages between a target and relevant clinical endpoints. By identifying the pathways driving the target's effect on the clinical outcome, this analysis also provides the required information needed for defining minimal thresholds for achieving a desired clinical

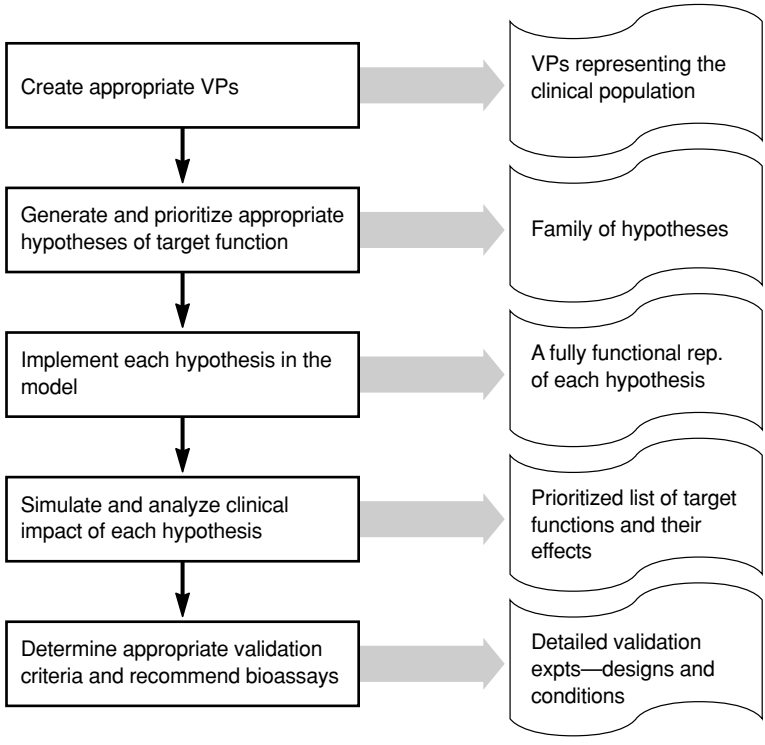


FIGURE 14.2 Flowchart highlighting the hypothesis generation and explicit characterization for target identification and validation *in silico*.

result. As a first step, one can use the biosimulation results to analyze and then prioritize the hypothetical contribution of each individual pathway on the ultimate clinical outcome. To take full advantage of these results in the validation of the target, one can exploit the transparency of the system to characterize its detailed mechanism of action in each cell type in each pathway.

14.3.2.3.2 Determine Pathway Thresholds/Criteria

Once the pathways through which the target affects the clinical outcome are identified, one can directly address the question of how much that target must be modulated to make it a worthwhile candidate for further development. This process provides the research scientist with an objective measure of a target's druggability. These mathematical expressions explicitly define minimum thresholds or constraints for a set of pertinent variables in the model. These thresholds allow instant assessment of predicted clinical efficacy for a given target modulation experiment. Once established for all the pathways in tandem, one can analyze the relative contribution of each pathway separately. Given the complexity of the biological interdependencies at hand, it is highly likely that individual threshold values may vary from pathway to pathway. By dissecting these dynamics *in silico*, one can define the different combinations of pathway thresholds required to achieve clinical efficacy.

These are the criteria and constraints that define the context for the making quantitative recommendations for *in vitro* or *in vivo* target validation efforts.

14.3.2.3.3 Analyze Any Uncertainties Surrounding Target Activity

While evaluating a novel target, it is very likely that one cannot, with a high degree of confidence, quantify its behavior in a particular pathway under a particular condition. When this occurs, it is incumbent on the researcher to explicitly identify this knowledge gap and test the robustness of the assumptions he or she has made to fill it. For example, if no data are available to quantify a target's involvement on a specific pathway yet there is compelling evidence that the target has a role in that pathway, the researcher can use the power of the biosimulation platform to evaluate an appropriate range of values representing the suspected biomolecular dynamics of the target.

Once explicitly defined and characterized, one can explore the impact of these assumptions on the eventual decision as to whether a target is worth pursuing. The decision itself, however, may not be as clear cut as a simple "yes" or "no" and may lead to the development of additional experiments to help resolve all the pertinent knowledge gaps. The transparency of the model is fundamental to the efficient development of these highly focused assays. An example of how this process has been applied at Entelos is outlined in the following case studies.

14.4 CASE STUDIES

The case studies presented next are illustrative examples of how mathematical modeling and predictive biosimulation can be used to identify and help validate particular targets. These studies were carried out at Entelos on the Entelos PhysioLab platforms. The PhysioLab platform provides a framework for building a mathematical model of disease composed of large sets (> 1,000) of ordinary nonlinear differential equations. The platform also includes a differential-equation-solving engine and user interface that allows for the solution of multiple experiments over multiple conditions simultaneously. The key to the applicability of the PhysioLab platform is that it is a functional representation of human pathophysiology. The underlying mathematical model is developed based on the most current understanding of human physiology, and the quantification of the model is based on the latest genomic, proteomic, *in vitro*, *in vivo,* and *ex vivo* data. The mathematical models developed by Entelos are designed and built using a top-down, disease-specific, systems biology approach that relates clinical outcomes to human biology and physiology. That is, scientists identify and focus on those components considered most relevant to a disease and its progression, typically starting with the major organ systems involved in its expression (e.g., circulatory, immune, metabolic) and then modeling "downward" to the most relevant pathways, that is, those that drive the behaviors of those organs deemed most responsible for the disease state, detailing, in turn, the dynamic interrelationships between relevant tissues, cells, proteins, and genes. In this manner, feedback, redundancy, and other compensatory mechanisms are incorporated into the system. The result is a functional representation of human health and disease as

represented in a deterministic model system. Once calibrated and validated, the PhysioLab model is capable of predicting human clinical response to therapeutic intervention. Solving the underlying equations numerically creates a predictive biosimulation platform of human pathophysiology.

14.4.1 Evaluating Novel Genes

In this example, we describe the *in silico* target validation of a novel gene as a potential drug target for the treatment of asthma. The information supplied by Entelos's collaboration partner, Bayer, included the gene's cDNA sequence and limited mRNA expression data. The methodology Entelos employed in pursuit of this potential target consisted of six steps (see fig. 14.2):

1. Create virtual patients who represent a fairly representative sampling of the disease population.
2. Generate and prioritize hypotheses of gene functions.
3. Represent those functions in each virtual patient.
4. Test the hypotheses through the simulation of the human response.
5. Compare and prioritize the data and results of the simulated clinical outcomes for a hypothetical therapeutic intervention.
6. Validate the underlying hypothetical assumptions through directed research (*in vitro* and/or *in vivo* experiments).

14.4.1.1 Creating Virtual Patients

As just described, virtual patients are explicit representations of the underlying hypotheses developed to characterize patient pathophysiology. At Entelos, the virtual patients are represented within the PhysioLab platform as explicit mathematical vectors of model parameters. To account for the underlying patient variability just described, scientists create as many virtual patients as are required to understand potential patient variability, especially as it relates to the novel target. The *in silico* scientist creates a virtual patient based on known or hypothesized factors (genetic, lifestyle, and/or environmental) that give rise to a specific disease phenotype. In this case, three virtual patients were created representing mild, moderate, and severe asthmatics. In other words, certain pathways within the single PhysioLab model of human asthma were modulated to represent three levels of disease severity. The patients were then tested by simulated antigen challenge, and each was validated against measures of known forced expiratory volume in one second (FEV_1), the responses used in the clinic to classify an asthmatic.

14.4.1.2 Hypothesis Generation and Prioritization of Gene Function

Once the virtual patients were created and validated, the research team needed to develop specific hypotheses regarding the specific functional role of the gene in the relevant pathways underlying the asthmatic response to antigen insult. Through the use of expression and homology data, multiple functional roles for the gene were

identified and represented in each virtual patient. A battery of predefined *in silico* experiments was applied to the virtual patients to ensure that the behavior of each virtual patient remained consistent with multiple clinical data sets. Each virtual patient (mild, moderate, and severe) was required to reproduce not 1 but 60 separate clinical responses simultaneously as demonstrated previously in humans.

To understand the role this novel gene may play in asthma and derive the hypotheses for its function, its cDNA sequence was scanned against the GenBank database to see if it had any relationship to any known genes. If significant homology was uncovered, it was hypothesized that the gene would function similarly. This analysis revealed that the novel gene exhibited a fairly high degree of homology with the serine proteases (e.g., tryptase). As most serine protease functions in the airways are well defined, hypotheses involving the novel gene's activity were generated based on these known functions; the mRNA expression data provided to Entelos by Bayer were then used to define the cell types in which each potential function might be active. For each hypothesis, the potential involvement of the new gene in pathways or biological phenomena was quantified using available data.

The possible roles this hypothetical serine protease might play in asthma were then prioritized based on their potential impact on predicted clinical outcomes and a confidence in the data used to generate them.

14.4.1.3 Representation of Gene Function in Each Virtual Patient

Since the Asthma PhysioLab platform reproduces observed physiological behaviors in the airway, the contribution of every gene to the disease state is already *implicitly* represented. The goal of this step was to *explicitly* represent the hypothesized functions of the new gene in the appropriate cells and pathways underlying the asthmatic response. To do this, each hypothesized function in each pathway was added to the Asthma PhysioLab platform. The effects were quantified as ranges of activity, and the robustness of these assumptions were tested (see the following section) over the entire range of hypothetical effect.

14.4.1.4 Hypothesis Testing through Simulation of Human Response

At this point, the research team created an *in silico* experiment to measure the effects of regulating the target gene on the overall clinical response. Typically, these experiments are designed to look at the incremental up or down regulation of the target over time. The incremental regulation of the target can be represented as a percentage of activity from 0 to 100 and over a period ranging from minutes to years. In addition, the experiment can be designed to simulate the additive or multiplicative effects of genes that regulate a variety of pathways. Once designed, the experiment is simulated in the PhysioLab platform.

The experiment set was applied to each virtual patient individually to identify potential phenotypic variability in these patients' responses to pathway modulation. The goals of these biosimulations were to predict the clinical impact of modulating

the gene's functions and then to identify the main biological mechanisms/hypotheses driving this effect.

14.4.1.5 Compare and Prioritize Data and Results

The clinical outcomes for each hypothesis in each virtual patient were analyzed to determine if the novel gene had a causative or correlative role in the disease process. The objective of this analysis is to identify those hypotheses, which might lead to an effective therapeutic intervention. Evaluation criteria are developed to assist in this analysis. The minimal clinical effect was set so that the patient's FEV_1 improved by at least 10% over baseline. Likewise, limits were put on the incremental regulation of the target gene, in that it was assumed that the gene did not that require more than a 25% change in pathway activity to elicit the minimal clinical effect. The results of these biosimulation studies provided a prioritized list of hypotheses consistent with our understanding of (a) the pathophysiology of the disease, (b) the phenotypic patient(s) at whom a therapy would be directed, (c) the functional role of the target, and (d) the predicted clinical results of a therapy directed at that target.

For the novel serine protease, the biosimulation results showed that modulating the new gene's activity on specific hypothesized functions improved clinical outcome, and thus, this gene might be a good target for asthma. The outcome of the analyses (see table 14.3) showed that this potential benefit was driven by one main function and revealed synergistic effects on other functions.

- Reducing the target's basement membrane degrading activity, which restricted the transendothelial migration (TEM) of inflammatory cells into the airways, produced the greatest clinical improvement.
- Certain combinations of hypotheses may produce clinically relevant synergies in improvement of the FEV1 (for examples, see table 14.3, "sensory nerves" and "BK production").

14.4.1.6 Hypothesis Validation through Directed *In Vitro*/*In Vivo* Experiments

In silico analysis, while necessary, is not sufficient to validate a presumptive target. It is necessary that one finalize this process by examining, *in vitro* and/or *in vivo*, the assumptions underlying the hypotheses captured in the biosimulation runs. The obvious experiments were recommended based on the *in silico* predictions captured in table 14.3. Since the PhysioLab platform is transparent and can illustrate the physiological effects and conditions underlying these hypothetical functions, a research plan was specifically designed that described, in detail, the laboratory experiments required to confirm the hypotheses developed above.

The particular assay recommendations provided by Entelos to Bayer included a suite of experiments and a detailed workflow necessary to fully validate the target and confirm the *in silico* predictions. Typically, these recommendations include the detailed rationale required to perform validation and screening assays necessary for advancing this particular target to high-throughput screening. In this case, biologists

TABLE 14.3
Effect of 60% New Gene Inhibition Late Phase FEV_1 Response

	FEV_1 (Late Phase Minima, Day 6)			
			Improvement Over Control	
ESP-1 on…	(–) TEM	(+) TEM	(–) TEM	(+) TEM
RANTES	0.420	0.489	–4.5	3.4
EOS degranulation	0.437	0.517	–0.7	9.3
ASM contraction	0.449	0.520	2.0	9.9
Sensory nerves	0.467	0.556	6.1	17.5
BK production	0.466	0.554	5.9	17.1
All non TEM effects together	0.477	0.604	8.4	27.7

Note: Highlighted area shows synergies. EOS = eosiniphil; ASM = airway smooth muscle; BK = bradykinin; TEM = transendothlial migration.

using standard *in vitro* (e.g., recombinant protease) and cell-based assay systems, performed laboratory experiments revealing

- Overexpression of the novel serine protease increases polymorphonucleocyte cell polarization and adhesion.
- Cells overexpressing the protease show a pronounced up-regulation of proinflammatory cytokine and chemokine mRNA, suggesting various mechanisms that might account for the synergies predicted by the biosimulation results.

As a result of these studies, Bayer decided to pursue this target and validate its role in TEM.

14.4.2 Evaluating PDE4 as a Target for Asthma

Cyclic AMP (cAMP) is an important intracellular mediator regulating the activation of inflammatory cells. Particularly important with respect to asthma, cAMP also plays a role in regulating airway and vascular smooth muscle contractibility, inflammatory cell proliferation, and pulmonary neuronal responsiveness. Cyclic phosphodiesterases (PDEs) comprise a protein superfamily, whose function is to inactivate cyclic AMP and cyclic GMP. The fact that certain PDE inhibitors suppress immune cell functions *in vitro* and pulmonary inflammation *in vivo* may represent an opportunity for the development of novel anti-inflammatory drugs.

Currently, eight PDE gene families have been identified, each exhibiting a unique tissue and cellular distribution pattern, substrate affinity, and specificity. Within the airways, PDE4 appears to be playing an important role in regulating airway smooth muscle and inflammatory cells responses. In addition, expression studies suggest that PDE4 is exerting this effect on numerous biological pathways, and many studies have focused on PDE4 as a potential anti-inflammatory drug target due to its expression

in inflammation-associated cell types. While it was widely thought that selective PDE4 inhibitors would enable targeted suppression of inflammation, clinical trial results have been disappointing. Thus, defining PDE4's mechanism of action and the potential clinical impact of its inhibition is critical for the successful validation of this particular target. A major difficulty in this process has been accounting for the complexities underlying asthma.

The goal of this case study was, using an *in silico* approach with the Entelos Asthma PhysioLab Platform, to evaluate the potential of PDE4 as a drug target for moderate asthma, by evaluating the clinical impact of PDE4 inhibition in moderate asthmatic patients.

14.4.2.1 Characterization of PDE4 Roles in the Airways

Even though PDE4 has been known for many years, its activities and functions are still not fully characterized. By analyzing the public literature, the known effects of PDE4 in the human pathophysiology of asthma were identified. In addition, a set of hypothetical functions were generated pertaining to any potential roles for PDE4 in the airways. This family of hypotheticals was derived based on any nondirect evidence for PDE4 activity in the airway (e.g., evidence of PDE4 expression combined with cAMP elevation in a particular cell type in a particular control environment). Each hypothetical function was then quantified to define the specific contribution of PDE4 to the specific biology involved. At the end of this process, more than 50 potential functions for PDE4 in the airways were identified.

14.4.2.2 Virtual Patients

An important factor to consider when evaluating a target is the diversity of the patient population for which the therapeutic is being developed. Are there subpopulations of patients more or less likely to respond to its modulation than not? The variability of the population can lie in its phenotypic presentation of the disease (i.e, mild vs. moderate asthmatics) as well as in the underlying pathophysiology that gives rise to the specific phenotype. For instance, two moderate asthmatic patients may have an equivalent clinical presentation but very different underlying pathophysiologies.

To explore this scenario, five virtual patients were developed to explore the uncertainty surrounding the pathophysiology of moderate asthma, that is, the simulations were designed to plumb the space of how one became a moderate asthmatic in the first place. While the array of inflammatory mediators underlying the pathophysiology of asthma have been identified, for the most part, and are fairly well understood, the contribution of each to the disease process is not yet known in detail. Though all five virtual patients have similar clinical behaviors consistent with moderate asthma (i.e., a forced expiratory volume in one second [FEV1] between 65 and 80%), each virtual patient was created to explore a different combination of these mediators.

14.4.2.3 Evaluating the Impact of PDE4 Inhibition on Clinical Outcome and Delineating Its Mechanism of Action

A key aspect of target validation is not only the impact of target modulation on the clinical outcome of a therapeutic intervention but also the identification of the laboratory experiments most relevant to validating the assumptions underlying the hypotheses of a target's mechanism of action.

PDE4 inhibition was simulated for a period of 28 days, during which time the baseline FEV1 was monitored. Within the PhysioLab platform, we were able to modulate each of the individual pathways, either singly or in combination. In so doing, we were able to identify those pathways that were driving the impact of PDE4 inhibition on the clinical outcome. This analysis determined that not only could PDE4 inhibition significantly improve FEV1, but while PDE4 is involved in numerous pathways (50+) in the airways, only four yielded a critical impact on the eventual efficacy of a potential therapeutic. Identifying these pathways led to the recommendation of a small set of laboratory experiments sufficient to fully validate the target *in vitro*. In addition, all five virtual patients demonstrated very similar responses to simulated PDE4 inhibition, suggesting that the PDE4 inhibition was relatively insensitive to the pattern of the mediators underlying the pathophysiology of moderate asthma.

14.4.3 Identifying Novel Targets in Rheumatoid Arthritis

How one uses a mathematical model to maximum advantage depends entirely on the questions one wants to answer. Once a calibrated and validated model is in place, one may choose to solve the equation systems in a variety of ways: deriving closed form solutions, analyzing the steady state behaviors of the system based on the equation structure, analyzing the topology of the system (e.g., the graph theoretic strategy described by Anderson and Hunt [8]), or solving the systems equations numerically using computational differential equation integrators (i.e., via biosimulation). While these approaches are not mutually exclusive, we at Entelos have used biosimulation studies to identify and quantify the relative and absolute contributions of individual pathways to disease state and pathophysiology. Studying each pathway and its components individually and in combination is termed a sensitivity analysis. Assuming a particular pathway has been identified as a major contributor to the clinical expression of the disease, one can then exploit the top-down structure of the model to tease apart the individual cellular physiologies that contribute most to the pathway's activities. Once these physiologies have been identified in the context of the top-down superstructure, the control circuitry that manages them can be quantified and the key molecular pressure points of the system determined. Those biomolecular entities that exert the greatest control over these pressure points are then identified as presumptive *in silico* targets, and the effects of their modulation are directly evaluated by simulation. From this point forward, the *in silico* validation process is very similar to that carried out for the gene screen previously described. First, virtual patients that fairly represent the clinical population are created. Then,

the activities of the presumptive molecular targets are quantitatively represented in each virtual patient. Then, each hypothetical is simulated and quantified as to its contribution to the effect on the clinical outcome. Then, the simulated results are tested as to the robustness of any underlying parameter estimates. Finally, the results are validated through directed research (*in vitro* and/or *in vivo* experiments).

14.4.3.1 Sensitivity Analysis, Target Identification, and Quantification

In this case study, we used the Rheumatoid Arthritis (RA) PhysioLab Platform to identify a set of novel targets in the pathophysiology and progression of RA. The results of an initial sensitivity analysis suggested that four main physiologies were responsible for driving the bulk of the disease condition. The results for the quantification of one molecule in particular are presented in table 14.4.

TABLE 14.4
Summary of Effects Quantified for Simulation of Clinical Effects from Molecular Blockade

PhysioLab Function	% Inhibition in Most Likely Scenario	Range of % Inhibitions from Best-Case to Worst-Case Scenarios
Pathway 1	88	88 to 67
Pathway 2	40	88 to 20
Pathway 3	0	84 to 0
Pathway 4	0	40 to 0

Based on these results, the team began a detailed search of the literature to identify those molecules most responsible for controlling these functions. The search was not restricted to RA but was expanded to include all inflammatory processes, including those surrounding tumor growth and progression in cancer. From these studies, five potentially novel targets were identified for RA.

For each of the functional pathways identified above, the team estimated the physiologic effect of inhibiting the target molecule with 100% efficiency. To derive these estimates, a "best case/worst case/most likely case" strategy was employed. The data themselves were derived from the literature and are documented in the online reference architecture of the Physiolab Platform. By way of example, consider Pathway 2 in table 14.4. In this case, an experiment was reported in the literature, showing that when all other signaling molecules had been neutralized *in vitro*, inhibiting this particular molecule alone could block that pathway by 88% (the best-case scenario). Other *in vitro* data were observed wherein the impact of the other (redundant) physiologic controls was explicitly measured, suggesting that in the worst-case scenario (i.e., in the case where significant overlap and redundancy were active *in vivo*), the best one could hope for was a 20% inhibition. Based on all the observed data, the most likely case scenario was estimated to yield 40% inhibition when this particular molecule was blocked.

TABLE 14.5
Cellular Expression Profile or Presumptive *In Silico* Molecular Target

	Relative Expression
Leukocytes: thymus, blood, and lymph nodes	
Monocytes	+++
Mature T cell (CD4 and CD8):	+++
CD4 Naïve (CD45RA+RO-, CD69-)	++
CD4 Memory (CD45RA-RO+, CD69+)	+++
Thymocytes	++++
Mature B cell	++
Naïve (IgM+, IgD+)	+
Memory (IgG+)	++
Other cell types	
Endothelial cells	++
Langerhans cells	++
Fibroblasts	ND
Pancreatic islet cells	ND
Granulosa cells of the ovary	ND
Sertoli cells of the testis	ND
Cancer cells	
Ewing sarcoma cells, peripheral primitive neuroectodermal tumors	++++
Breast cancer	+++
Hodgkin's disease	↓↓ (down regulation)

Note: ND = not determined.

Tissue-level expression patterns for the target molecule were then identified (see table 14.5), and based on the level of expression, individual model components were identified; based on the numbers derived in table 14.4, each pathway in each virtual patient was modified appropriately.

14.4.3.2 Simulation Results

Blockade of the presumptive target *in silico* was simulated in two virtual patients: the RA PhysioLab platform's reference patient and a Methotrexate-reduced responder (MTX-RR) patient. Presently, Methotrexate is considered one of the "gold standard" therapies for RA, and methotrexate resistance occurs in approximately 30% of the patient population. The reference virtual patient represents an average RA patient with progressive disease that responds appropriately to the common RA therapies. The MTX-RR patient was developed to represent the subpopulation of patients who do not respond to MTX. In simulations, this patient's response to MTX is less than half that of the reference patient. The therapeutic responses of these patients were assessed by measuring relative changes in synovial cell density, the rate of cartilage degradation, and the levels of synovial IL-6. All three of these

parameters were measured in the two virtual patients after six months of simulated treatment. This simulated length of time allowed the virtual patients to stabilize in response to the treatment.

14.4.3.3 Reference Patient

14.4.3.3.1 Synovial Cell Density Response

To determine the level of molecular blockade needed to obtain significant clinical improvement, we first measured the decrease in synovial cell density over a spectrum of increasing levels of blockade. Figure 14.3 shows the simulated effects on synovial cell density after six months of molecular blockade. A decrease in synovial cell density greater than 33%, the typical response observed in a MTX gold standard, was determined to be the desired threshold for clinical efficacy. This level of inhibition was achieved in all three hypothetical scenarios. These results suggest that if one can achieve 80% blockade of this particular molecule, the clinical outcome of this therapy, at least in terms of synovial cell density, should in any case be better than the standard MTX therapy.

To assess the length of time needed to achieve a clinically relevant therapeutic effect, the decrease in synovial cell density was measured at various points across the six-month treatment regime. Figure 14.4 shows that the effect on synovial cell density is near maximal at 28 days and plateaus after 90 days. This result suggests that the effect of molecular blockade on synovial cell density should give clinically measurable responses after one month of treatment.

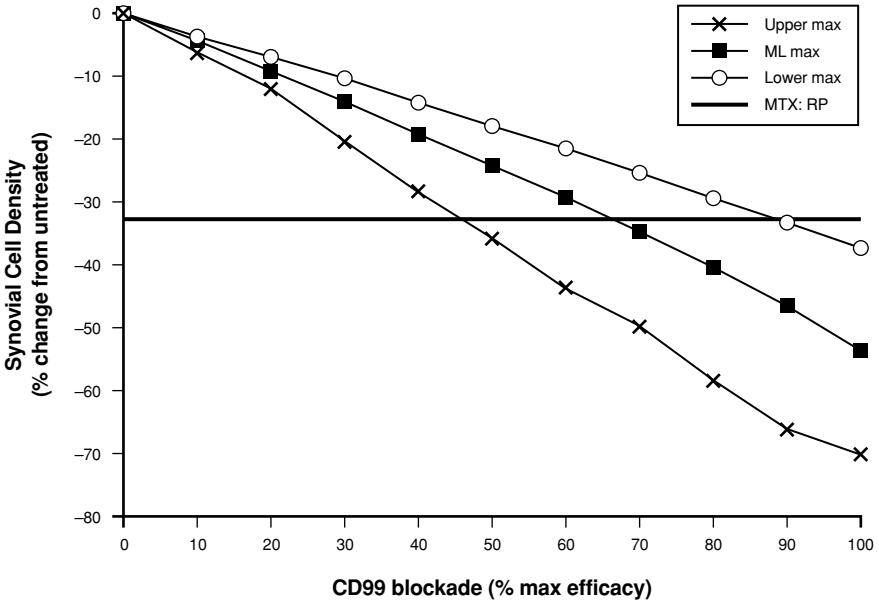


FIGURE 14.3 Impact of molecular antagonism on synovial cell density in the reference patient.

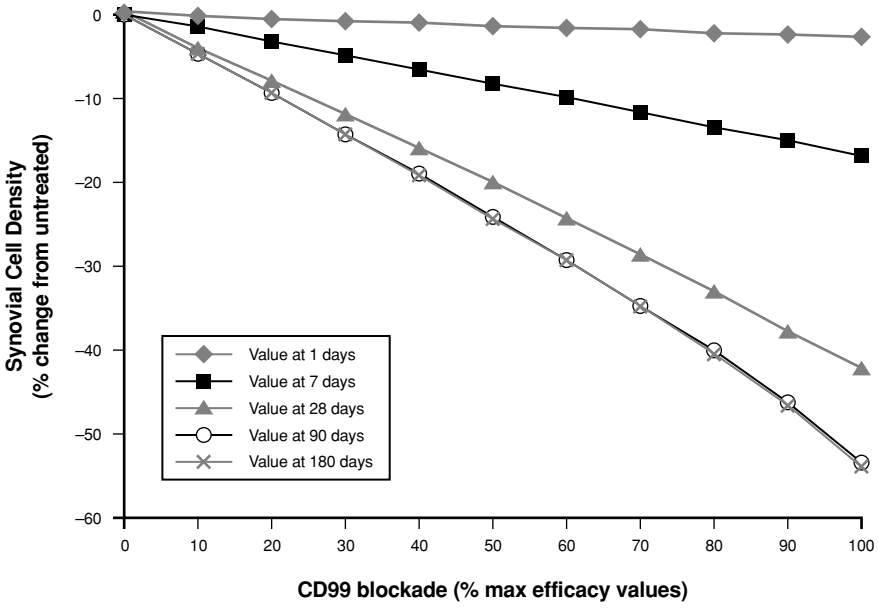


FIGURE 14.4 Impact of molecular antagonism on synovial cell density at different times during treatment.

14.4.3.3.2 Cartilage Degradation Rate Response

Next, the effect of molecular blockade needed to significantly decrease the rate of cartilage degradation in the reference patient was assessed. Figure 14.5 shows the simulated effects on cartilage degradation after six months of simulated therapy. A decrease in cartilage degradation greater than that achieved in the clinic by MTX (17%) was again achieved in all three hypothesized scenarios. Figure 14.6 shows that the effect of molecular blockade on cartilage degradation rate is also near maximal at 28 days and again plateaus after 90 days. This again suggests that the effect of molecular blockade should give clinically measurable responses relatively rapidly.

14.4.3.3.3 IL-6 Response

Finally, the impact of molecular blockade on the levels of synovial IL-6 (an indirect indicator of the effect of proinflammatory cytokines levels in the patient's joint) was measured. Figure 14.7 shows the simulated effects on synovial IL-6 levels after six months of simulated molecular blockade. When 50% blockade is achieved, the levels in synovial IL-6 decrease significantly in all three hypothetical scenarios. Very similar results were found for the MTX-RR patient as well, suggesting that a therapeutic aimed at this target may provide a reasonable alternative to methotrexate in the resistant population.

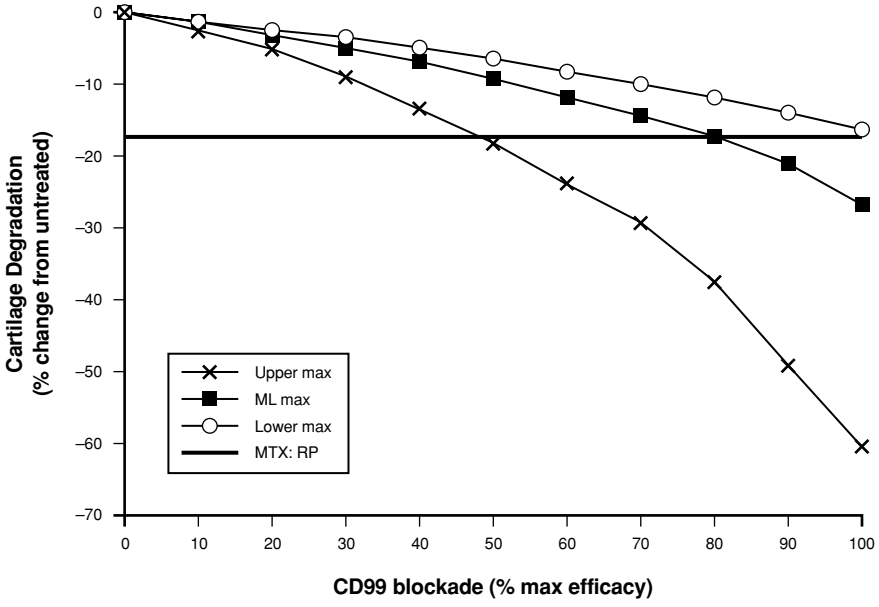


FIGURE 14.5 Impact of molecular antagonism on cartilage degradation rate in the reference patient.

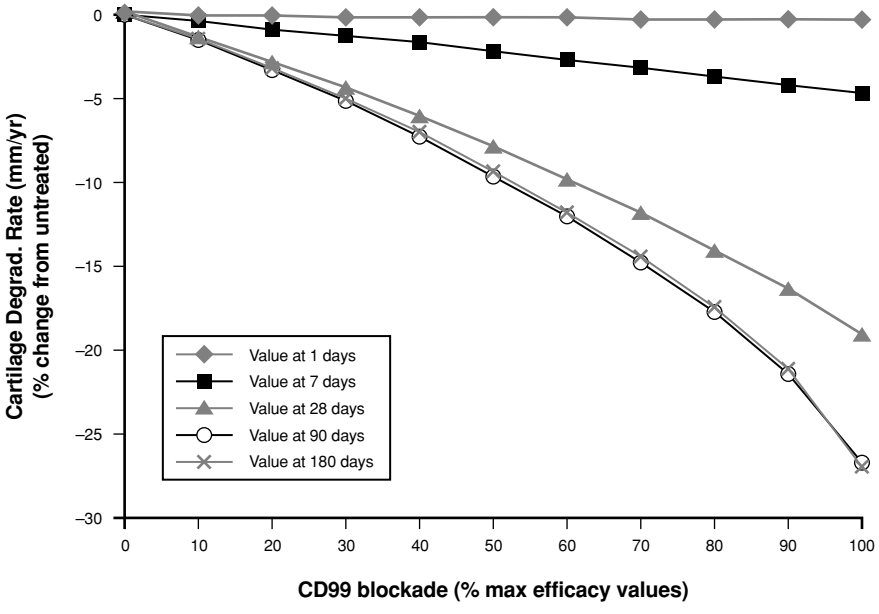


FIGURE 14.6 Impact of molecular antagonism on cartilage degradation at different times during treatment.

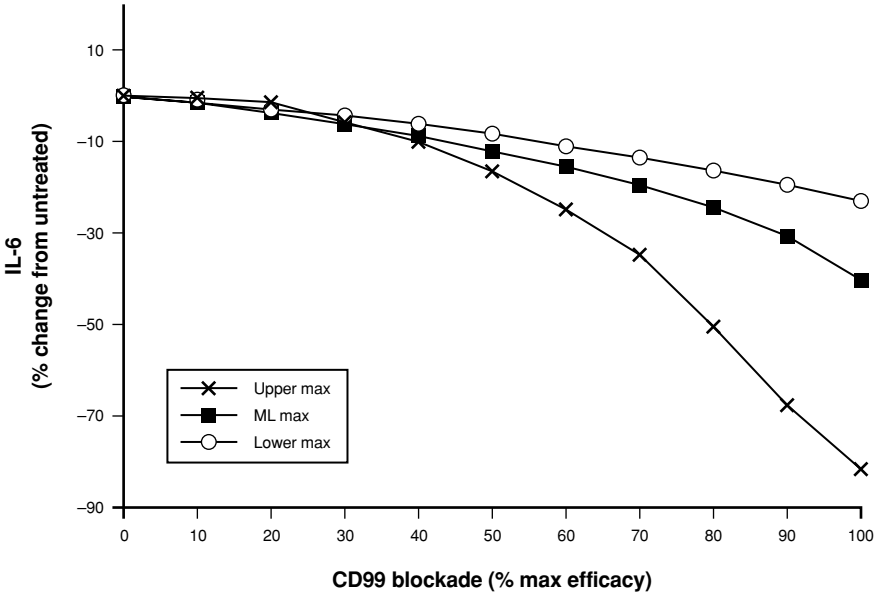


FIGURE 14.7 Impact of molecular antagonism on synovial IL-6 concentration in the reference patient.

14.4.3.4 Mechanism of Action

To identify the critical laboratory experiments necessary for validating this molecule as a drug target for RA, we identified the main pathways driving the impact of molecular blockade on the clinical outcome of RA. To define the likely mechanism of action, the most likely scenario and the best-case scenario were analyzed in the reference patient on a pathway by pathway basis.

14.4.3.4.1 Most Likely Case Scenario

To identify the main pathway(s) driving the impact of molecular blockade on the clinical outcome, the consequences of activating each hypothesis in turn (assuming 100% efficacy) were examined. Figure 14.8 shows that, in the most likely case scenario, Pathway 1 is the primary pathway driving the impact of molecular blockade on the clinical outcome in the reference patient. Conversely, by accepting as realistic all hypothetical effects simultaneously and then turning each effect off individually, one can begin to assess potential synergies across the different minor on clinical outcome (fig. 14.9). For this scenario, no synergistic effects of any pathways other than Pathway 1 were observed.

14.4.3.4.2 Best-Case Scenario

When using the best-case scenario quantifications, three different pathways appear to be contributing to the global effects of this particular target (fig. 14.10). Pathway 1 is the major driver, followed closely by Pathway 4 and Pathway 2. Figure 14.11 shows clearly the functional redundancy of the combination of effects that allow

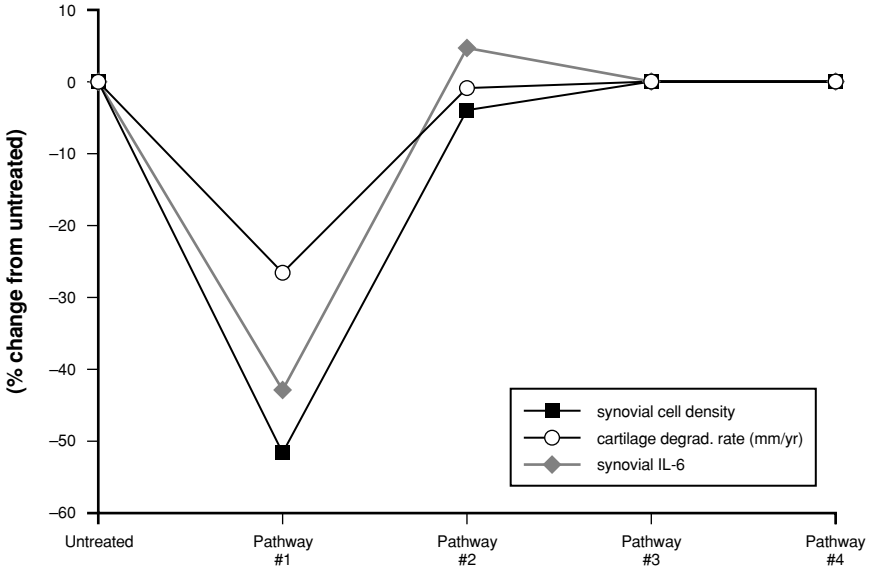


FIGURE 14.8 Effects of turning on each hypothetical effect individually in the most likely case scenario.

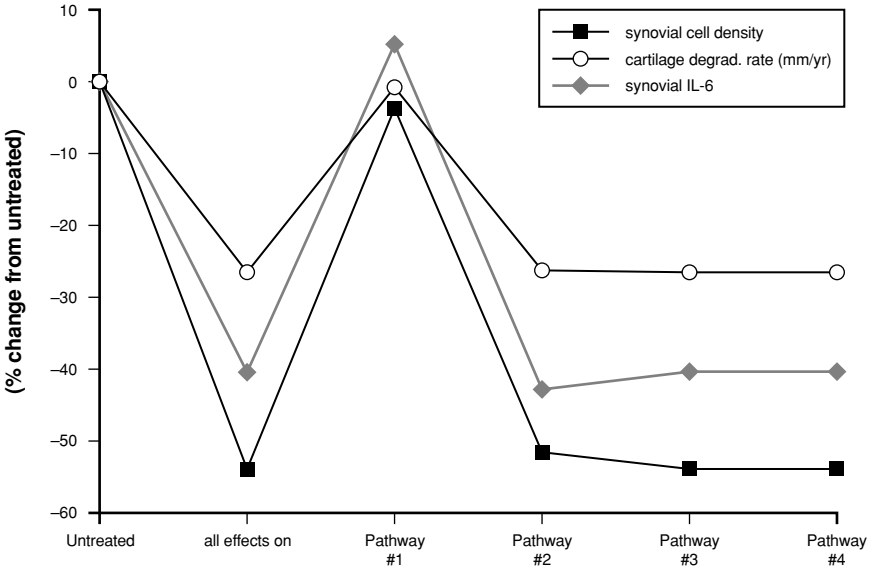


FIGURE 14.9 Effects of turning off each effect individually, assuming all other effects remain on for the most likely maximum effects on the reference patient.

compensation for turning off one effect at a time. By systematically delineating the mechanism of action for this particular target, we identified Pathway 1 as one of the main pathways driving the impact of molecular blockade on the clinical outcome.

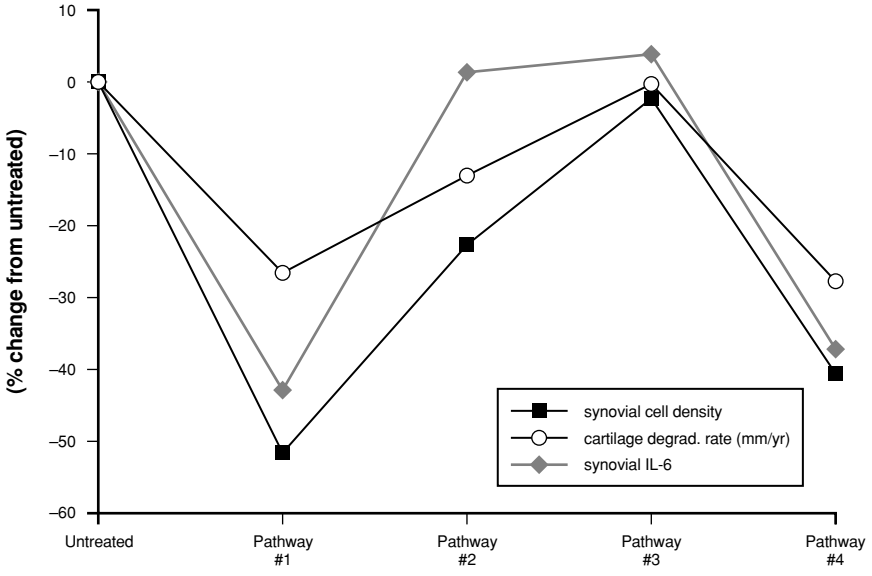


FIGURE 14.10 Effects of turning on each effect individually for the upper maximum effects on the reference patient.

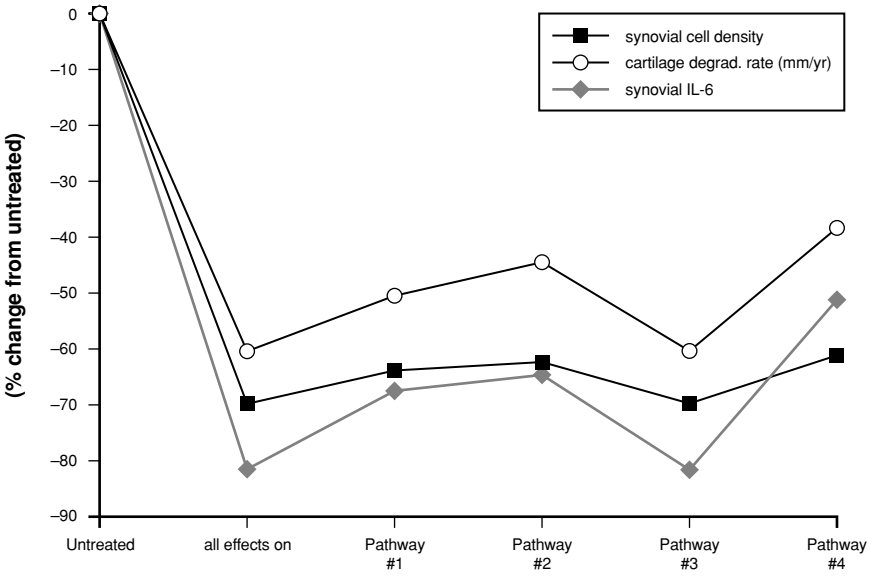


FIGURE 14.11 Effects of turning off each effect individually, assuming all other effects remain on for the upper maximum effects in the reference patient.

However, a detailed analysis of the best-case-scenario quantifications suggested that there maybe other, compensating, mechanisms present. Any target validation plan should thus focus on developing assays to reconcile these underlying assumptions.

14.5 CONCLUSIONS

Mathematical modeling and predictive biosimulation play a vital role in the process of target identification and validation. Using a formalized structure of mathematical rigor and applying to it the power of high-speed computing, the pharmaceutical researcher can rapidly identify sensitive pathways underlying disease, characterize the molecular entities within those pathways that are most druggable, and assess the effects of manipulating those molecules within the context of whole-body physiology. However, most putative targets molecules are likely to impact numerous pathways or underlie multiple biological phenomena in the physiological context of a disease. Therefore, it is vital that these *in silico* results be linked to an active and directed wet lab validation effort.

To truly validate a target, one must be able to understand, at its most fundamental level, the biology controlling the disease and its dynamic. By explicitly identifying the control circuits that govern the behaviors of those pathways underlying disease pathophysiology, one can explicitly define the biological context of the disease and, based on these insights, develop the most predictive assays possible for validating a putative target.

Furthermore, mathematical modeling and *in silico* predictive biosimulation can help us survey the knowledge landscape by demanding an explicit characterization of what we know and an inventory of what we do not. Systematically exploring what we do not know allows us to evaluate the impact of these knowledge gaps on the eventual efficacy of a candidate therapy, leading to a set of prioritized experimental recommendations aimed at resolving the most important. These experiments, in all likelihood, will provide significant insight into a target's mechanism of action.

REFERENCES

1. Kightley, D. A., N. Chandra, and K. Elliston. 2004. Inferring gene regulatory networks from raw data: A molecular epistemics approach. *Pac Symp Biocomput* 9:498–509.
2. King, R. D., K. E. Whelan, F. M. Jones, P. G. K. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver. (2004). Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* 427:247–52.
3. Chakraborty, A. K., M. L. Dustin, and A. S. Shaw. 2003. In silico models for cellular and molecular immunology: Successes, promises and challenges *Nat. Immun* 4:933–6.
4. Chaves, M., E. D. Sontag, and R. J. Dinerstein. 2004. Steady-states of receptor–ligand dynamics: A theoretical framework. *J Theor Biol* 227:413–28.
5. Fox, J. J., and C. C. Hill. 2001. From topology to dynamics in biochemical networks. *Chaos* 11:809–15.
6. Sen, A. K. 1990. Metabolic control theory: A graph theoretic approach. *Biomed Biochim Acta* 48:817–27.
7. Sen, A. K. 1990. Tological analysis of metabolic control. *Math Biosci* 102:191–223.
8. Anderson, A. A. 2002. Elucidating essential targets in pharmacologically relevant systems models. Ph.D. dissertation, University of California–San Francisco.
9. Hall, K., R. Baillie, and S. Michelson. 2002. Biosimulation: Dynamic modeling of biological systems. *Annu Rep Medicinal Chem* 37:279–88.

10. Musante, C. J., A. K. Lewis, and K. Hall. 2002. Small- and large-scale biosimulation applied to drug discovery and development. *Drug Discov Today* 7, Suppl.:S192–6.
11. Dimasi, J. A. 2001. New drug development in the United States from 1963 to 1999. *Clin Pharmaco. Ther* 69:286–96.
12. Dimasi, J. A. 2001. Risks in new drug development: Approval success rates for investigational drugs. *Clin Pharmacol Ther* 69:297–307.
13. Dimasi, J. A. 2001. *The economics of pharaceutical innovation: New estimates of drug development costs*. Tufts Center for the Study of Drug Development 25th Anniversary Forum (Philadelphia, PA, USA).
14. Stoke, C. L. 2001. A computer model of chronic asthma with application to clinical studies: Example of treatment of exercise-induced asthma. *J Allergy Clin Immunol* 107:933.
15. Lewis, A. K., T. Paterson, C. Leong, N. Defranoux, S. Holgate, and C. Stokes. 2001. The roles of cells and mediators in a computer model of chronic asthma. *Int Arch Allergy Immunol* 124:282–6.
16. Leckie, M. J., A. ten Brinke, J. Khan, Z. Diamant, B. J. O'Connor, C. M. Walls, A. K. Mathur, et al. 2000. Effects of an interleukin-5 blocking monoclonal antibody on eosinophils, airway hyper-responsiveness, and the late asthmatic response. *Lancet* 356:2144–8.
17. Barrett, A. 2003. Feeding the pipeline. *BusinessWeek*, May 12.
18. PhRMA: Pharmaceutical Research and Manufacturers of America. 2000. *Pharmaceutical industry profile 2000*. Retrieved from http://www.phrma.org/publications/publications/profile00/
19. Eddy, D. M., and L. Schlessinger. 2003. Validation of the Archimedes Diabetes model. *Diabetes Care* 26:3102–10.

15 Using Protein Targets for *In Silico* Structure-Based Drug Discovery

Tad Hurst
ChemNavigator, Inc.

CONTENTS

15.1 INTRODUCTION

Drug discovery traditionally has been understood in two steps: (a) target identification and screen development and (b) drug lead identification and optimization. The first step involves understanding the biological basis for a disease, including the natural pathways and biochemical reactions, along with the deviations that cause the disease. A target protein or receptor is often identified, and a screening method is devised to determine when the activity of that target is modulated by the presence of a potential drug molecule. The fields of genomics and proteomics have contributed greatly to this process over the last several years, allowing a much greater understanding of the biological entities involved in many healthy and diseased pathways.

Once a screening method has been developed and verified, the process of drug lead discovery can be followed much as it has over the last several decades. In the traditional process, a large number of small molecules are tested with the screening method to determine their efficacies as potential drugs or drug lead compounds.

Once one or more lead compounds have been identified and their activity has been verified, the process of medical chemistry or lead optimization follows. In this process, synthetic changes are made to the compound, and the change in the activity is determined. If the compound is more active than the original lead compound, then subsequent changes are made to its structure. This iterative process continues until the desired activity profile is obtained. An activity profile may include, in addition to the direct biological activity of the compound, the selectivity it has for the desired result over competing undesirable effects.

After the desired activity profile has been obtained, the new drug candidate is subjected to tests for toxicity, then clinical trials to demonstrate its overall efficacy against the disease of interest. These latter processes are not discussed here.

The processes as just described do not require any information coordination between the genomics and proteomics work and the subsequent lead discovery and optimization. In the last decade, there has been a growing effort to transition the information from the proteomics step into a computer-aided drug discovery process—to use the information about the protein itself to help choose appropriate compounds to screen for the desired efficacy. The use of computers to assist in drug discovery neither is new nor was always tied to the use of protein structures. The recent developments in proteomics have focused the work on protein structure-based computer-aided drug design.

15.2 TWO-DIMENSIONAL COMPUTER-AIDED DRUG DISCOVERY

Early methods of rational lead optimization sought to find a set of potential leads from a database of small molecules chemically similar to a known lead. The earliest of these techniques was performed by the synthetic chemist in the process of producing the next substance to test and was performed without the aid of a computer. The chemist would make a set of small changes to the structure and determine if those changes had a beneficial or detrimental effect on the efficacy. This process, called analog synthesis, is quite effective. Analog synthesis is the basis for medicinal chemistry and remains an important part of drug discovery today.

Early attempts at using computers to make the lead optimization process more efficient involved determining chemical similarity between a potential lead and a known lead using graph-theoretical treatments. Methods of this type include searching a database of compounds for those that contain the same core structure as the lead compound—called substructure searching or two-dimensional (2D) searching—and searching for compounds that are generally similar based on the presence of a large number of common fragments between the potential lead expansion compound and the lead compound—called 2D similarity searching. Examples of programs that perform these 2D searching techniques include ISIS, UNITY, Merlin, and many others (see table 15.1). These techniques are effective but are limited to finding compounds that are obviously similar to the lead, thus affording the medicinal chemist few new insights for directing the synthesis project.

TABLE 15.1
Some Prominent Drug-Discovery Programs and Their Uses

Program	Agent	2D Searching	QSAR	Pharmacophore Perception	3D Coorinate Generation	3D Searching	Docking	Database Searching/Docking	Binding Site Analysis	URL
ISIS	MDL	X				X				http://www.mdli.com
Merlin	Daylight Chemical Information Systems	X								http://www.daylight.com
Unity	Tripos, Inc	X				X				http://www.tripos.com
CoMFA	Tripos, Inc		X							http://www.tripos.com
C2·QSAR+	Accelrys		X							http://www.accelrys.com
CONCORD	Tripos, Inc				X					http://www.tripos.com
Corina	Molecular Networks				X					http://www.mol-net.de
DISCO	Tripos, Inc			X						http://www.tripos.com
GASP	Tripos, Inc			X						http://www.tripos.com
Hypogen	Accelrys			X						http://www.accelrys.com
Catalyst	Accelrys	X				X				http://www.accelrys.com
AutoDock	Scripps Institute						X			http://www.scripps.edu/mb/olson/
Dock	Univ. of California, San Francisco						X			http://dock.compbio.ucsf.edu/
Gold	Cambridge Crystallographic Data Centre						X			http://www.ccdc.cam.ac.uk/
FlexX	BioSolveIT						X			http://www.biosolveit.de/
SiteId	Trpios								X	http://www.tripos.com
3DPL	ChemNavigator				X			X	X	http://chemnavigator.com

15.3 QUANTITATIVE STRUCTURE ACTIVITY RELATIONSHIPS

Another field for assisting in the selection of potential lead compounds or optimized leads is called Quantitative Structure Activity Relationships (QSARs). These methods determine correlations between certain descriptors of the molecular structures and the measured biological activity to produce a predictive model. That model can then be used to predict the activity of other structures prior to actually testing.

The descriptors of the molecular structures that are used in QSAR include physical and electronic properties, fragment compositions, as well as calculated properties of the three-dimensional (3D) structures of the compounds. The 3D properties include scalar parameters like solvent-accessible surface area, or hydrophobic surface area. They also include field-type reductions of the structure that represent steric interactions, electrostatic potentials, hydrogen-bonding potential, hydrophobic interactions, and so on.

The early use of 2D parameters and scalar values was described by Leo, Hansch, and Elkins [1] and has been predominately replaced with more rigorous fragment-based techniques and 3D-QSAR techniques. The fragment-based techniques are exemplified by HQSAR [2] while the most successful of the 3D-QSAR techniques is Comparative Molecular Field Analysis (CoMFA) [3].

Many predictive, validated models have been developed using these QSAR techniques and have often assisted in the selection of structures for lead optimization. Often the QSAR results are not available until after the process of lead optimization has already progressed, and these models represent retrospective analysis of the lead optimization process rather than a direct influence on the design of the lead optimization compounds.

15.4 3D SEARCHING TECHNIQUES

Other techniques of computer-aided drug discovery use knowledge of how the potential drug molecules bind to a protein target (e.g., a receptor). There are two general classes of such techniques: those techniques for which the 3D structure of the target is known, and those techniques that do not require direct knowledge of the target 3D structure. Theses two classes of techniques are called structure-based [4] and ligand-based methods, respectively. (It should noted that the word structure is overloaded and can cause confusion. In bioinformatics and genomics, this word is used to refer to the 3D structure of the protein or receptor, in contrast to the linear sequence of the receptor. In medicinal chemistry, the word structure refers to the connectivity pattern of the atoms of a small molecule and not to the 3D coordinates of the atoms of that compound.)

Ligand-based techniques often characterize the groups in the known ligands that are responsible for much of the stabilization energy of the protein–ligand complex. These important groups are called pharmacophore groups, and the spatial relationship between the pharmacophore groups that is required for activity is called the pharmacophore model.

The pharmacophore models are often predicted by pharmacophore-perception software programs such as DISCO [4] and GASP [5]. These programs analyze the various pharmacophore-group arrangements for each of the active ligands and detect the geometric arrangements that are common among them. These common arrangements are presumed to include one that represents the geometry required by the unknown receptor for binding.

The pharmacophore models produced are then used as the query for 3D searching techniques, which try to find from a large database of potential lead compounds those that meet the geometrical requirements of the model. Compounds that are found to contain the correct arrangement are called hits and are candidates for screening. These hits differ from the substructure or 2D similarity hits in that the backbone of the structure may be quite different from that of the original lead compound and often represents an important new area of chemistry to be explored.

The simplest of the 3D searching techniques compares the position and arrangement of the pharmacophore groups of the candidate structures, as they are stored in the database. This is referred to as static 3D searching. Static 3D searching is limited, because it does not explore the multitude of conformations that are available to most druglike compounds. Most compounds have a large number of accessible conformations formed by rotating the molecular framework of a molecule about its rotatable bonds. Small molecules in pharmaceutical databases typically contain an average of six to eight rotatable bonds per molecule.

This can easily afford a set of accessible conformations that number in the millions. Searching just one static conformation from among millions of possible conformations for the correct arrangement of groups will cause many compounds (that could be good leads) to be missed.

To consider energetically accessible conformations, some systems populate the small molecule databases with a small subset of the accessible conformations of each small molecule. These are called multiconformational searching techniques. These techniques are still limited by the conformations stored. It is impractical to store the millions of accessible conformations for each molecule that are required for rigorous investigation of conformational space.

A more advanced approach in 3D search technology involves investigation of the accessible conformational space of the potential hits as part of the searching process. These techniques, called conformationally flexible 3D searching techniques, adjust the conformation of the potential hit according to the requirements of the 3D pharmacophore query. One such technique that has been found to be effective uses a method called Directed Tweak [6]. This method is very effective for finding molecules of interest when the geometry of the binding site of the large molecule is not known. Directed Tweak adjusts the conformation of the small molecule by changing the angle values of the rotatable bonds, which includes essentially all of the accessible conformational flexibility of a small molecule.

15.5 PROTEIN-DOCKING TECHNIQUES

Today there are a growing number of proteins for which the 3D structures are either known or estimated. The 3D structures of proteins are usually the result of either

experimental determination, such as X-ray crystallography or nuclear magnetic resonance studies, or modeling systems like de novo folding programs or homology modeling systems. When the 3D structure of the protein target is known, computers may be used to find potential leads by "docking" small molecules into the protein. The potential for a small molecule to bind to the protein is evaluated according to a set of rules and equations that model the physical interactions between the receptor and potential ligand. Many systems adjust the location and orientation of the small molecule with respect to the receptor, and many also investigate different conformations of the molecules. As in 3D searching, there are systems that store or generate a small number of accessible conformations, while other systems deal with the molecular flexibility of the structures "on-the-fly" as the search progresses.

15.5.1 Docking Systems

There are a number of well-known docking systems, and these vary in the method of evaluating and optimizing the predicted binding affinity. Many docking systems use molecular mechanics force field methodologies to estimate the binding affinity. These methodologies attempt to model the short-range and long-range forces between a target and a small molecule using field representations. The types of interactions often considered in docking systems include electrostatic contributions, steric interactions, hydrogen-bonding stabilization, and hydrophobic effects. Other important considerations for docking include the effect of solvation and the flexibility of the protein itself.

One commonly used system is AutoDock [7]. This program uses a grid-based technique in which the interaction energies for the atoms in the small molecule are precalculated on the points of a grid. This process simplifies the calculation of the energy estimate of the small molecule in a particular position. The grids are determined by standard molecular mechanics force-field methods. The position and conformations of the small molecules are adjusted using a hybrid genetic algorithm to sample over the feasible conformations and positions of the ligand relative to the protein. It takes about one minute for AutoDock to dock a structure when the flexibility of the ligand is considered.

Another well-known docking tool is DOCK [8]. This program reduces the information from the 3D receptor to produce a negative image of the binding site. This image consists of spheres that fill the binding site of the protein. During the search, subsets of ligand atoms are matched to spheres, based on the distances between ligand atoms. Once the molecule is placed, the full estimation of the binding affinity is computed using standard molecular mechanics techniques. DOCK can take several minutes to dock a ligand structure.

Another docking program, FlexX [9–12], uses a fragmentation approach. The ligand is fragmented and incrementally reconstructed in the binding site and matched to template points in the receptor. Bond torsional flexibility is adjusted, and a tree-search algorithm is used to keep the most promising partially constructed ligand conformations during the search. FlexX typically takes a minute or more to dock a ligand structure.

Hammerhead [13] uses up to 300 hydrogen-bond donor and acceptor and steric points to define a template, and the ligand is incrementally reconstructed, as in FlexX. A fragment is docked based on matching ligand atoms and template points with compatible internal distances. If a new fragment is positioned closely enough to the partially constructed ligand, the two parts are merged and the most promising placements kept.

Another successful docking program, GOLD [14], uses a genetic algorithm to sample over possible matches of conformationally flexible ligands to the template. GOLD uses a template based on hydrogen-bond donors and acceptors of the protein and applies a genetic algorithm to sample over all possible combinations of intermolecular hydrogen bonds and ligand conformations. A drawback of genetic algorithm approaches is the high computation time, especially in comparison to fragment-based docking approaches.

The UNITY 3D Searching System (Tripos Inc., St. Louis, MO) has been extended to provide what is essentially a docking tool. In this approach, six parameters corresponding to the six rotational/translational degrees of freedom are added to the rotatable bond list, and these parameters are adjusted to place pharmacophoric groups at the positions giving favorable interactions with the receptor. This method produces acceptable accuracy but is time-consuming because the derivatives needed for the minimization must be calculated numerically.

There are many other docking systems available [15–19]. For the most part, they follow the precepts outlined in the methods specifically discussed above.

15.5.2 Accuracy

The effectiveness of these systems is measured in different ways. Often docking systems are graded on how well they reproduce the conformation and position of a cocrystallized ligand-receptor structure. There are many standard test sets and much information on how well various docking systems do at placing a ligand that is known to bind into the known binding configuration. The accuracy is normally reported as the root-mean-square (RMS) of the atomic positions as docked compared to the positions from the cocrystallized X-ray structure. The best placements have RMS values of 1.0 or slightly less. The ability of a docking system to correctly place a known ligand is an important factor in its use in drug discovery, but it is not sufficient.

Another common measure of accuracy for docking systems is the comparison of the predicted binding energy to the measured binding energy. This is not as widely used as the RMS score and is not a sufficient indication of a docking system's applicability for drug discovery.

Most docking systems are designed to dock known ligands accurately but not to discriminate between good and bad potential ligands. A good measure of the applicability of docking systems to their use in drug discovery is the enhancement ratio. This value is the ratio of the number of active molecules found, as determined by actual screening of the compounds, to the number of active molecules expected from random screening. The most common docking systems achieve an enhancement ratio of between 1.0 and 25.0. An enhancement ratio of 25 indicates that 25 times more

active compounds are found when the results of the docking run are tested than would be found if the same number of randomly selected compounds was screened.

15.5.3 Speed of Docking

Among commercially available chemical suppliers, more than 13 million different structures are available. When screening databases of this size, the computational efficiency of the search process becomes a significant concern. If it takes one minute on average to dock a structure, a database of 11 million structures will take 21 years if done on a single computer.

Most researchers who wish to use docking systems to enhance their ability to discover new drugs have applied one more approache to overcoming the speed issue of standard docking programs. The most common technique is to reduce the size of the structure database to be investigated. This technique is typically done by applying druglike filters and further reducing the structures to be docked to those that are similar to known lead compounds. This approach, of course, eliminates the possibility of finding new types of active compounds.

A widely used technique for improving the docking efficiency is to parallelize the computation. Many research facilities have constructed large arrays of computers, often numbering in the hundreds, to dock the large databases. It can still take a week or more to process the structures from a large database.

A recent innovative approach involves a marriage of docking and 3D searching. This technique, called TweakDock [20], is the basis for the 3DPL Database Searching System (ChemNavigator, San Diego, CA). This technique precalculates the energy and the *derivatives* of energy on a grid and uses a 3D searching method to adjust the position and configuration of the ligand molecules relative to the protein structure (fig. 15.1). 3DPL produces accurate docking poses and is much faster than conventional docking systems. It often docks molecules at a rate of 15 structures/second on a single CPU. When run on eight CPUs, 11 million compound structures can be docked in less than three days, and enhancement ratios of 15 to 25 are often observed. This speed allows rapid turnaround of tests that can directly influence the compounds that are purchased or synthesized, making *in silico* drug discovery based on the growing number of protein structures easily feasible.

15.5.4 Binding Site Determination

Most of the docking programs require specification of the specific portion of the protein where binding occurs. Protein targets are generally too big for exhaustive examination of the surfaces and internal cavities for possible binding. Often, the site of a cocrystallized ligand is used to define the active site for the protein target. Docking at this site is guaranteed to be important, but the results will not include possible interactions at other important sites. If there is no crystal structure with a cocrystallized ligand, then the binding site must be determined through other means.

Several programs are available to identify potential binding sites of a protein. SiteID from Tripos helps the researcher identify the potential binding sites by producing an image that encodes depth, hydrogen-bonding ability, and other factors on the surface of

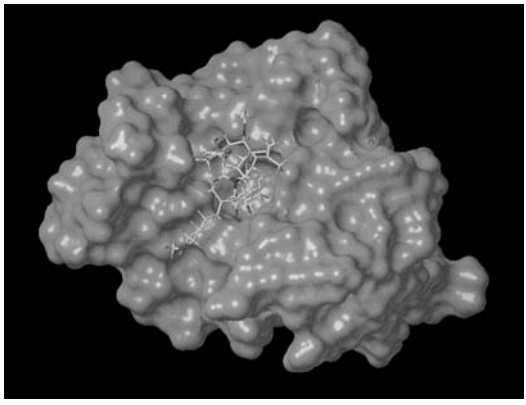


FIGURE 15.1 A potential drug molecule found by 3DPL for calcineurin inhibition.

the protein. The Binding Site Analysis function from Accelrys (San Diego, CA). analyzes the variation in related proteins to estimate the binding site of a protein. It also can identify crevices that are big enough for binding a ligand. Locus Discovery [21] has software that predicts the active binding sites by filling potential locations with fragments. The actual binding site is identified as the sites with higher affinity. The ChemNavigator [22] 3DPL software identifies all possible binding sites on a protein. The 3DPL software can perform the docking fast enough to investigate all possible binding sites.

15.6 CONCLUSION

The recent advances in genomics and proteomics are leading to a great increase in the knowledge of disease and in the generation of effective biological screens to help find important new drug therapies for those diseases. Computer-based approaches, especially docking and 3D database searching, are evolving to make direct use of the more abundant 3D structures of the protein and enzymes associated with disease and are increasingly improving the efficiency of new drug discovery.

REFERENCES

1. Leo, A., C. Hansch, and I. Elkins. 1971. Partition coefficients and their uses. *Chem Rev* 71:525–616.
2. Hurst, T., and T. Heritage. 1997. HQSAR—A highly predictive QSAR technique based on molecular holograms. Paper presented at the 213th ACS National Meeting, San Francisco, CA.
3. Cramer, R. D., III, D. E. Patterson, and J. D. Bunce. 1988. Comparative Molecular Field Analysis (CoMFA) 1. Effect of shape on binding of steroids to carrier proteins. *J Am Chem Soc* 110:5959.
4. Martin, Y. C., M. G. Bures, E. A. Danaher, J. DeLazzer, I. Lico, and P. A. Pavlik 1993. A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. *J Comput Aided Mol Des* 7:83.

5. Jones, G., P. Willett, and R. C. Glen. 1995. A genetic algorithm for flexible molecular overlay and pharmacophore elucidation. *J Comput Aided Mol Des* 9:532–49.
6. Hurst, T. 1994. Flexible 3D searching: The Directed Tweak technique. *J Chem Inf Comput Sci* 34:190–6.
7. Morris, G. M., D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, and A. J. Olson. 1998. Automated docking using a lamarckian genetic algorithm and empirical bnding free energy function. *J Comput Chem* 19:1639–62.
8. Shoichet, B. K., I. D. Kuntz, and D. L. Bodian. 1992. Molecular docking using shape descriptors. *J Comput Chem* 13:380–97.
9. Rarey, M., B. Kramer, and T. Lengauer. 1999. Docking of hydrophobic ligands with interaction-based matching algorithms. *Bioinformatics* 15:243–50.
10. Rarey, M., S. Wefing, and T. Lengauer. 1996. Placement of medium-sized molecular fragments into active sites of proteins. *J Comput Aided Mol Des* 10:41–54.
11. Kramer, B., M. Rarey, and T. Lengauer. 1999. Evaluation of the FlexX incremental construction algorithm for protein-ligand docking. *Proteins* 37:228–41.
12. Rarey, M., B. Kramer, T. Lengauer, and G. Klebe. 1996. A fast flexible docking method using an incremental construction algorithm. *J Mol Biol* 261:470–89.
13. Welch, W., J. Ruppert, and A. N. Jain. 1996. Hammerhead: Fast, fully automated docking of flexible ligands into protein binding sites. *J Chem Biol* 3:449–62.
14. Jones, G., P. Willett, R. C. Glen A. R. Leach, and R. J. Taylor. 1997. Development and validation of a genetic algorithm for flexible docking. *J Mol Biol* 267:727–48.
15. Schnecke, V., C. A. Swanson, E. D. Getzoff, J. A. Tainer, and L. A. Kuhn. 1998. Screening a peptidyl database for potential ligands to proteins with side-chain flexibility. *Proteins* 33:74–87.
16. Dixon, J. S. 1997. Evaluation of the CASP2 docking. *Proteins* 29, Suppl. no. 1:198–204.
17. Leach, A. R. 1994. Ligand docking to proteins with discrete side-chain flexibility. *J Mol Biol* 235:345–56.
18. Jackson, R. M., H. A. Gabb, and M. J. E. Sternberg. 1998. Rapid refinement of protein interfaces incorporating solvation: Application to the docking problem. *J Mol Biol* 276:265–85.
19. Wasserman, Z. R., and C. N. Hodge. 1996. Fitting an inhibitor into the active site of thermolysin: A molecular dynamics study. *Proteins* 24:227–37.
20. Hurst, J. Methods for identifying a molecule that may bind to a target molecule. US Patent 6,671,628, December 30, 2003.
21. Locus Pharmaceuticals, Inc. Locus Pharmaceuticals. Retrieved August 2004 from http://www.locusdiscovery.com/
22. ChemNavigator. ChemNavigator. Retrieved August 2004 from https://www.chemnavigator.com/

Part IV

Computational Infrastructure

16 Database Management

Arek Kasprzyk and Damian Smedley
European Bioinformatics Institute

CONTENTS

16.1 INTRODUCTION

Certain distinct features make biological data management a challenging undertaking. Biological sciences, as a domain, are described using complex and fuzzy concepts. Consequently, biological databases require complex representations and ongoing alterations of the existing models. Recent advances in high-throughput technologies such as genome sequencing and microarrays have significantly increased the amount of data being stored, thus creating scalability issues. Repositories of biological data contain different subsets of biological knowledge. They are

maintained independently in different locations and made available on different release schedules.

At the same time, the true benefit of interacting with different datasets can only be attained if the data are integrated, allowing the user to use criteria from one dataset to query another. Consequently, biological research requires tailor-made datasets, which are compiled from unique combinations of individual databases and creates a need for technologies that provide such functionality. However, it must be stressed that even the most sophisticated technology on its own will not produce truly integrated data. The failure to adhere to common data standards and the lack of common semantics of stored data hinder data integration approaches. Therefore, the constant control of data quality; the use of globally unique identifiers, which facilitate data aggregation; and the use of common controlled vocabularies, in particular ontologies, greatly facilitates integration efforts.

In addition to the requirements for data storage and integration, there are also varying requirements for interfaces to the data to support the community. Bench scientists generally require intuitive Web interfaces or click-and-install graphical user interfaces. Application developers within the bioinformatics sector require stable application programming interfaces (APIs) in their preferred language (e.g., Java, Perl, C, C++, or Python).

Along with technologies to store and access data, there is a requirement for data manipulation in analysis and format conversion tasks. There is a variety of packages, free and commercial, that provide uniformed means of carrying out much of this analysis. Data resources and data manipulation software available on the Web can also be linked using Web services or workflow technologies.

16.2 BIOLOGICAL DATABASES

Biological databases are a heterogeneous collection of datasets, which exist in a variety of formats. Historically, they have originated as formatted text files, each representing a single, atomic database entry. This particular solution was well fitted with old sequence-oriented databases, and there are still some, such as UniProt [1], that are predominantly stored in this format. Modern biological databases have taken full advantage of relational technology, allowing for much more flexible data modeling and data structures as well as constraints on data integrity. Databases such as ArrayExpress [2] and dbSNP [3] use complex relational representations to fully express the complexity of data.

Biological databases also differ in their focus. They can represent the horizontal view of a particular domain involving a collection of organisms (e.g., Ensembl [4] or UCSC [5] Genome Browser). They can also be focused on a particular organism, offering a vertical view over a variety of datasets, which is typical for databases grouped in the GMOD [6] project such as Rat Genome Database [7], FlyBase [8], and WormBase [9]. A number of biological databases simply model a particular abstraction, such as ontologies (GO) [10] or complex biological processes (Reactome) [11]. Consequently, designing a particular data-management strategy requires careful planning involving both the data content and the technology used for data access and storage.

16.3 DATA INTEGRATION

Although there is a plethora of existing solutions for data integration, they usually fall into one of two basic categories: centralized or federated. The immediate result of using one or the other can sometimes be very similar. However, there are some important implications of the chosen architecture that are critical for long-term management and maintenance of the system.

Centralized architectures are based on a single data model and by definition require that all the data are stored locally, typically in a tailor-made data warehouse. In this approach, the integration effort is data-model driven. Both the data model and the middleware are designed for a particular knowledge domain and operate on a fixed set of data. The quality and granularity of the internal representation of the data depends on the domain expertise of the system designer. Such solutions tend not to be easily extensible and normally require a considerable programmatic effort to bring in new types of data. However, they do provide a good fit in the situation where all the data needed for a given task can be easily acquired and brought in-house. The usage pattern is well defined and unlikely to evolve in the future, and there are enough resources for maintenance of the data traffic involved.

In contrast, federated architectures tend to be more flexible and are more generally applicable. Typically they either leave data in its native format or require that data be put in a format common to all the datasets. They do not rely on any domain-specific abstractions but instead model the generic features of data and employ some kind of query-based logic for their API abstractions. These solutions tend to be much more extensible and require configuration rather than a programmatic effort when bringing new data types into the system. These federated architectures can be either local or distributed.

The distributed type of the federated architecture benefits from the local domain expertise of scientists who maintain the data. It tends to be more practical, obviating the necessity to move large datasets around and immune to the synchronization problems affecting local solutions. The drawback of this solution is the dependency on the network quality, bandwidth, and uncontrolled network congestion. It is therefore critical that distributed systems are designed with particular care with respect to their scalability for large datasets. An additional benefit of the distributed system is its deployment flexibility. It can be configured to be used in either a distributed or local fashion, or in any combination of the two.

It is important to note that existing technologies such as Web services can turn a local system into a distributed one by making it either a consumer or producer of a particular flavor of Web services. However, these constitute extensions to the existing architecture (rather than a part of the original design), are dependent on third-party interface definitions, and are unable to evolve on their own. Therefore, they are discussed separately for the purpose of this chapter.

Finally, it is worth noting that systems will differ in the availability of different types of user interfaces, APIs, and their support for third-party applications and/or programming protocols. Although this may not be a critical issue from the system architecture perspective, it can be an important factor when deciding on the final choice of a system when there is a heavy dependence on the existing user base

and/or systems already in place. In addition, the amount of effort needed for the maintenance, initial deployment effort, and the projected flexibility mustbe taken into account.

In the review of the existing integration solutions presented in this chapter, I do not aspire to be comprehensive, because this would be a rather hopeless task, given the speed with which this area of bioinformatics evolves. Instead, I try to exemplify the differences and similarities between the systems and trends in their design.

16.3.1 Centralized Architecture

Centralized systems are typically based on data-warehousing ideas [12]. In this approach, all data are gathered in one physical location, and a single data model encompassing the data properties is used. Typically these systems come with fixed datasets available as a part of the distribution, and the API abstractions include the representation of the modeled data (e.g., genes, proteins, pathways). The centralized system architecture is not easily extensible to other types of data; therefore, these systems frequently use technologies such as Web services to federate other types of data. However, it needs to be noted that the data federated in such a way rarely share the same functionality and scalability as the core system. These solutions are typically designed for a particular environment and make a good fit for users, assuming a well-defined and similar usage pattern.

16.3.1.1 Grand Unified Schema

The Grand Unified Schema (GUS) project [13] (http://www.gusdb.org) developed at the University of Pennsylvania is one of the pioneers of using the data-warehousing ideas to deal with the wealth of biological data. The GUS system is based on an extensive strongly typed relational schema. The system includes the GUS Application Framework, which assists in the development of data acquisition and analysis programs, and the GUS Web Development Kit, which assists in query-based Web site development. The GUS platform integrates the genome, transcriptome, and proteome of one or more organisms; gene regulation and networks; ontologies and controlled vocabularies; gene expression; and interorganism comparisons.

GUS uses the central dogma of biology as its organizational principle. Sequence centric entries from the external databases are mirrored within GUS and transformed into gene-centric entities. Thus, GUS tables hold the conceptual entities that the sequences and their annotation ultimately represent (i.e., genes), the RNA derived from those genes, and the proteins derived from those RNAs. The GUS database is quite large, containing over 400 tables and views that are divided into five schemas: Database of Transcribed Sequences, RNA Abundance Database, Transcription Element Search System, Shared Resources, and Nonbiological Tracking.

The GUS application framework consists of a Perl and Java object layer complemented by a pipeline, GUI applications, and a Web development kit. The object layer includes one-to-one mappings between object and tables/views.

16.3.1.2 SeqHound

SeqHound [14] (http://www.blueprint.org/seqhound/) is a freely accessible bioinformatics database warehouse made available to the public through a rudimentary Web interface but mainly using a remote API. SeqHound is available from the "public good" Blueprint Initiative research program of the Samuel Lunenfeld Research Institute at Mount Sinai Hospital, affiliated with the University of Toronto. The project arose out of a requirement to integrate information from a variety of sequence, structure, and annotation databases for the production of the Biomolecular Interaction Network Database (BIND) as well as to support Blueprint's systems-biology-oriented research program.

The data are collected daily from a number of sources, including the National Center for Biotechnology Information (NCBI) and Gene Ontology (GO) Consortium. The warehouse includes sequences, structures, and complete genomes from GenBank as well as annotation links such as NCBI taxon database terms, redundant sequences, sequence neighbors, conserved domains, Online Mendelian Inheritance in Man identifiers, GO terms, LocusLink identifiers, and PubMed links.

The API is available for Perl, Java, C, and C++. A BioPerl API, which can be used in conjuction with BioPerl, is planned that SeqHound calls will eventually become part of the main BioPerl source code tree. The API is well documented with an extensive manual as well as tutorials and help guides available from the main Web site.

SeqHound is the basis for many of Blueprint's applications. The BIND Interaction Viewer 3.0 queries SeqHound to recover sequence annotation and interaction information. SeqHound has been used as a back end to replace all previous calls to the NCBI for retrieval of data such as GenBank records, taxonomy traversal, DNA and protein sequence by taxonomy ID or organism, 3D neighbors, and GO traversal. Recently the SeqHound API has been incorporated into Taverna [15], which allows biologists to assemble Web services into a pipeline. The current Web interface allows access only to the sequence aspects of the database, as the primary aim of SeqHound was to be a resource for programmers. However, a more fully featured interface is under development, which will make most of the API's functionality available to nonprogrammers.

16.3.1.3 Cancer Bioinformatics Infrastructure Objects

The Cancer Bioinformatics Infrastructure Objects (caBIO) [16] (http://ncicb.nci.nih.gov/core/caBIO) is the primary architecture for data integration at the National Cancer Institute (NCI). It is based on a data warehouse that is used to integrate several of the NCI data sources. The content of the caBIO data warehouse is refreshed approximately biweekly and consists of variety of National Institutes of Health datasets, including genomic, expression, pathway, and clinical trials data.

Four APIs to caBIO are available, each suitable for different client-programming environments: Java, Perl, SOAP, and HTTP-XML. Domain objects represent

biological, laboratory, and clinical entities. The presentation layer is built on a Web server infrastructure that includes Apache HTTP Server, Tomcat, Zope, and Apache SOAP. Java servlets and server pages deliver content to client applications. All caBIO objects can be transformed into serialized XML representations.

The caCORE component forms the foundation for a number of scientific and clinical applications. One application is CMAP, a work in progress that can be regarded as a prototypical caCORE-powered application. The availability of the caCORE-enabled CMAP is to be prototyped in a relatively short time. Cancer data and data relationships are presented in CMAP with rich graphics, and the application leverages caBIO APIs to provide a straightforward interface to quite complex underlying queries.

16.3.1.4 Atlas

Atlas [17] (http://bioinformatics.ubc.ca/atlas/) is a biological data warehouse, developed at the UBC Bioinformatics Centre, University of British Columbia, that locally stores and integrates biological sequences, molecular interactions, homology information, functional annotations of genes, and biological ontologies. Atlas has five main parts: the source data, the relational data models, the ontology system, the APIs, and the applications. Data sources fall into four main groups: sequence, molecular interactions, gene function, and ontology.

There are two classes of APIs in Atlas: loader and retrieval. The loader APIs are used to build the loading applications and populate instances of the relational models in the Atlas databases. The retrieval APIs are required for developing custom user-retrieval applications such as the Atlas toolbox applications. The loader API for Biological Sequences has been implemented in C++ as it relies heavily on the NCBI C++ Toolkit to parse the ASN.1 data. The Biological Sequence retrieval API, on the other hand, is provided in three languages: C++, Java, and Perl. Finally, the loader and retrieval APIs for Molecular Interactions are provided in Java.

A publicly available Web interface to the Atlas databases is available. This interface provides basic access to GenBank, RefSeq, NCBI Taxonomy, Atlas Ontologies, BIND, HPRD, MINT, and DIP. Web interfaces to the Atlas toolbox applications ac2gi, ac2seq, ac2tax, feat2seq, gi2ac, gi2feat, gi2seq, gi2tax, tax2seq, techtax2seq, and tech2seq are available. In addition, interacting partners for proteins identified by accession numbers or GI numbers can be retrieved from any of the four interaction databases stored in Atlas. These Web tools can be found at http://bioinformatics.ubc.ca/atlas/webtools/.

16.3.2 Federated Architecture

Database middleware systems employed by federated systems offer users the ability to combine data from multiple sources in a single query without creating a physical warehouse. Federated systems take advantage of the availability of shared identifiers. These identifiers allow for the given database to be related to another without the requirement that both datasets need to be part of the same data model. Although the actual implementations range from systems based on flat-file indexing (e.g.,

Sequence Retrieval System [SRS]) through simple systems creating query optimized views on existing data sources (e.g., BioMart) to complex systems using a full database engine to drive a virtual database (e.g., K2, DiscoveryLink), the basic idea remains the same. Data sources are presented in a unified format to the query-building middleware. The typical feature of those solutions is that they tend to be viewed as data-handling tools rather than the data repository. They usually offer some sort of generalized query language (e.g., ODD, OQL, SQL, or MQL) in addition to standard graphical user interface (GUI) applications.

16.3.2.1 SRS

SRS [18] is an indexing system for flat-file libraries such as EMBL or UniProt. Originally developed at EMBL, SRS was later acquired by LION Bioscience AG and released as a licensed product. It remains freely available for academics.

SRS supports the data structure of individual databases in flat-file format by providing special indexes for implementing list of subentities such as feature tables. SRS has the ability to define indexed links between databases. Once indexed, the links become bidirectional and operate in multistep fashion. They operate on sets of entries and can be weighted and combined with logical operators (AND, OR, and NOT).

SRS uses metadata to define a class for a database entry object and uses rules for text-parsing methods, which are coupled with database entry attributes. For object definitions and recursive text-parsing rules, SRS uses its own scripting language called Icarus. For library specification and organization and for the representation of individual data fields within a system, SRS uses a Object Design and Definition (ODD) language.

Recently, two improvements have been made to bring SRS closer to the relational world. First, the SRS Relational module permits the user to extend the SRS query capabilities to include relational databases; second, SRS Gateway for Oracle permits querying of Oracle databases from within SRS. In addition, SRS supports various sequence analysis tools such as FASTA, CLUSTALW, and selected programs from EMBOSS.

16.3.2.2 K2

K2 [13] is a distributed query system that has been developed at the University of Pennsylvania. K2 relies on a set of data drivers, each of which handles the low-level details of communicating with a single class of underlying data sources (e.g., Sybase** relational databases, Perl/shell scripts, the BLAST family of similarity search programs, etc.). A data driver accepts queries expressed in the query language of its underlying data source. It transmits each such query to the source for evaluation and then converts the query result into K2's internal complex value representation. Data drivers are also responsible for providing K2 with data source metadata (i.e., types and schemas), which are used to type check queries.

K2 decomposes the user's OQL query into subqueries that can be answered by the underlying data sources. Furthermore, it must rewrite the OQL query fragments,

where necessary, into queries that the data sources can understand. Both tasks are handled by the system's query optimization module, which is an extensible rule-based optimizer.

K2 is implemented as a multithreaded server that can handle multiple client connections. Clients communicate with the server using either Remote Method Invocation, Internet Inter-Orb Protocol, or an ad hoc socket protocol. In addition to a set of client libraries that simplify accessing K2 from any Web application that uses Java servlets, a client that provides interactive command line access to the system has been implemented.

16.3.2.3 DiscoveryLink

IBM's DiscoveryLink uses database middleware technology to provide integrated access to biological data sources. DiscoveryLink provides users with a virtual database to which they can pose arbitrarily complex queries in the high-level, nonprocedural query language SQL.

DiscoveryLink employs DB2's query optimizer and a complete query-execution engine The main features of its architecture are the so-called *wrappers*, software modules that act as intermediaries between data sources and the DiscoveryLink server. The DiscoveryLink server uses information supplied by wrappers to develop execution plans for application queries.

The overall architecture of DiscoveryLink is common to many heterogeneous database systems. Applications connect to the DiscoveryLink server using a variety of standard database client interfaces, such as Open Database Connectivity or Java Database Connectivity, and submit queries to DiscoveryLink in standard SQL. The information required to answer the query comes from one or more data sources, which have been identified to DiscoveryLink through a process called *registration*.

Data sources of interest to the life sciences range from simple data files to complex domain-specific systems that not only store data but also incorporate specialized algorithms for searching or manipulating data.

16.3.2.4 BioMart

BioMart is a system focused on a large-scale aspect of distributed data integration. This technology was originally developed for Ensembl database [19] and subsequently extended to support a federated data-mart architecture. To achieve scalability it uses relational, query-optimized views on existing data sources rather than data sources themselves. The federated marts are based on a common data model and can be automatically configured using dataset configuration software, making it possible to build a sophisticated interface capturing the semantics of the data content and incorporating the "links" to other datasets. In addition, the configuration software automatically detects new tables added to the system and supports updates of the data content. Once the data marts have been created, they can be accessed by a set of stand-alone and Web-based query interfaces. BioMart presents a lightweight solution for organizing in-house data and integrating it with publicly available data sources.

The BioMart data model is a simple modular schema composed of a central table, linked to its satellite tables by primary/foreign key relations. The schema can be normalized but typically includes denormalizations to achieve maximum query response optimizations. All the BioMart metadata is stored in XML configuration files on the database servers. The metadata files can be readily created and modified using MartEditor, a Java-based configuration editor.

The BioMart software suite consists of two APIs, one written in Perl and the other written in Java. There are three applications: MartView, a Web site query wizard available as a stand-alone Web site installation; MartExplorer, a stand-alone GUI; and MartShell, a command line interface. MartShell uses a Mart Query Language (MQL)—a simple structured query language. BioMart API has been incorporated into the Taverna workflow system [15]. The "biomaRt" annotation package, which enables direct access to BioMart databases, has become part of BioConductor—an open source and open development software project that provides a wide range of powerful statistical and graphical tools.

16.4 DATA MANIPULATION SOFTWARE

Data manipulation software is an important component of data management systems. As well as a vital component of the build process of integrated data resources, they are used in their APIs and user interfaces. In addition, Web services and the workflow software provide a means of linking applications with data resources to create complex workflows. These workflows may be used by biologists as stand-alone analysis packages. Again, we present here a review of the most currently used packages rather than a comprehensive list.

16.4.1 The Bio* Family

The BioPerl, BioJava, BioPython, BioRuby, BioCorba, and BioPipe [20] family of packages are facilitated by the Open Bioinformatics Foundation to provide open source modules and scripts for life science research in the major programming languages of bioinformatics.

BioPerl [21] has been running officially for nearly 10 years now and forms a stable package of useful data-manipulation tools. Applications include sequence file manipulation, parsing of data files, retrieving annotation from major databases, and pipeline automation. The modules are object oriented and, hence, have some dependencies on each other. BioPerl is divided into repositories: bioperl-live contains the core functionality of BioPerl. Auxiliary repositories of modules are available and are capable of generating GUIs (bioperl-gui); persisting storage of objects using databases (bioperl-db); and running and parsing of results from hundreds of bioinformatics applications such as BLAST, ClustalW, EMBOSS, Genscan, and HMMER (bioperl-run), and other modules to automate analyses (bioperl-pipeline). For performance reasons, BioPerl also contains extensions for several C programs for sequence alignment and local BLAST searching. BioPerl is useful for both everyday Perl scripting tasks as well as a component of larger bioinformatics applications such as the Ensembl project.

BioJava is a newer project but has been stable for some time now. The Java framework provides extensive support for sequence manipulation, file parsing, DAS clients and servers, access to Ensembl and BioSQL databases, and analysis and statistical routines, including a dynamic programming toolkit.

BioPython provides similar support for bioinformaticians using the popular Python language. Files in formats such as BLAST, FASTA, and GenBank can be parsed into Python-utilizable data structures. Access to well-known bioinformatics destinations such as NCBI BLAST, Entrez, and PubMed services are supported. Interfaces to BLAST, the ClustalW alignment program, and tools for the usual sequence manipulation tasks of translation and transcription are provided. Finally code to simplify parallelization of tasks and GUI-based programs for basic sequence manipulation tasks are packaged with BioPython. Integration with other languages, including the aforementioned BioPerl and BioJava projects, is possible using the BioCorba interface standard (biopython-corba).

BioRuby performs the usual bioinformatic tasks in a similar manner to the previous Bio* packages but uses the Ruby language. BioPipe is a recent attempt to release some generic code for automation of bioinformatics workflows on large compute-farms and has arisen from the Ensembl project's pipeline code. All the Bio* projects contain extensive documentation, tutorials, and useful mailing lists on their Web sites and are under active development.

16.4.2 GCG

The GCG Wisconsin Package [22] is an integrated package of nucleotide and protein manipulation and analysis tools. The Wisconsin package arose from the Genetics Computer Group (GCG) at the Department of Genetics, University of Wisconsin. Eventually GCG was commercialized and is now available from Accelrys (http://www.accelrys.com/products/gcg_wisconsin_package). The 130-plus programs in the Wisconsin package can be grouped into the following main categories:

- Pairwise and multiple sequence comparisons: creation, editing, display, and analysis; searching of nucleic acid and protein sequence databases for similar sequences to an input sequence or pattern using BLAST and FASTA in their various forms
- DNA/RNA secondary-structure prediction and display
- Editing and display of sequences for publication
- Phylogenetic tree generation and display
- Assembly of nucleotide sequence fragments
- Gene finding and pattern recognition in sequences (protein coding regions, terminators, repeats, etc.)
- Hidden Markov model (HMM) generation from a set of related sequences and use of this model to search databases, align sequences, and generate new sequences (HMMER suite)
- Restriction digest and RNA fingerprinting prediction and display
- Prediction of optimal polymerase chain reaction primers

- Protein sequence motif identification
- Translate nucleic acid sequence into protein and vice versa

The Wisconsin package provides three interface options. SeqWeb is the Web interface to a core set of the programs. SeqLab provides access to all the packages via an X Windows GUI. Finally, the programs can be used via the command line with the option to do batched procedures via scriptable command lines.

16.4.3 EMBOSS

EMBOSS, the European Molecular Biology Open Software Suite [23], was largely initiated in response to the commercialization of GCG (see the previous section). Originally the source code of the GCG libraries was available, allowing new programs to be developed and distributed freely. However, when the source code was made unavailable, such development was no longer possible, and distribution of source code of programs using the GCG libraries was also not possible.

EMBOSS handles all sequence and many alignment and structure formats. The extensive libraries support the development of further open-source software for the community. In addition, EMBOSS integrates existing packages and tools to provide a single suite for sequence analysis. EMBOSS is maintained and developed at the Rosalind Franklin Centre for Genomic Research, Wellcome Trust Genome Campus, Cambridge, United Kingdom (http://www.rfcgr.mrc.ac.uk/Software/EMBOSS). EMBOSS contains about 100 applications, covering areas such as

- Sequence alignment
- Searching databases with a sequence pattern
- Protein motif identification
- Nucleotide sequence pattern analysis
- Codon usage analysis
- Identification of sequence patterns in large sequence sets
- Presentation tools for publication

The EMBOSS applications are under general public license (GPL), although the libraries are under the Lesser GPL. Third-party programs with a Lesser-GPL-compatible license (e.g., PHYLIP) are packaged with EMBOSS under the EMBASSY grouping, allowing linking to the rest of the libraries and to the user, looking exactly like any other EMBOSS application.

A variety of graphical and script modules have been developed. Jemboss, a Java-based graphical interface, is the main supported interface and forms part of the EMBOSS distribution. Applications can be run interactively or in batch mode monitored by a job manager with sensible presentation of applications and defaults generated on the fly. The Jemboss Alignment Editor can be accessed to view and edit any sequence alignments produced.

At many sites shell, Perl or Python scripts are used to run EMBOSS applications. BioPerl and BioPython both provide support for accessing EMBOSS programs, while the Bio::Emboss module provides specific access for Perl programmers.

Finally, various specific Web interfaces to EMBOSS exist, such as wEMBOSS (http://www.wemboss.org). In addition, EMBOSS has been incorporated into the SRS interface via generation of external-application definitions in SRS's Icarus language.

REFERENCES

1. Bairoch, A., R. Apweiler, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, et al. 2005. The Universal Protein Resource (UniProt). *Nucleic Acids Res* 33, Database Issue:D154–9.
2. Parkinson, H., U. Sarkans, M. Shojatalab, N. Abeygunawardena, S. Contrino, R. Coulson, A. Farne, et al. 2005. ArrayExpress—A public repository for microarray gene expression data at the EBI. *Nucleic Acids Res* 33, Database Issue:D553–5.
3. Sherry, S. T., M. H. Ward, M. Kholodov, J. Baker, L. Phan, E. M. Smigielski, and K. Sirotkin. 2001. dbSNP: The NCBI database of genetic variation. *Nucleic Acids Res* 29:308–11.
4. Hubbard, T., D. Andrews, M. Caccamo, G. Cameron, Y. Chen, M. Clamp, L. Clarke, et al. 2005. Ensembl. *Nucleic Acids Res* 33, Database Issue:D447–53.
5. Karolchik, D., R. Baertsch, M. Diekhans, T. S. Furey, A. Hinrichs, Y. T. Lu, K. M. Roskin, et al. 2003. The UCSC Genome Browser Database. *Nucleic Acids Res* 31:51–4.
6. Stein, L. D., C. Mungall, S. Shu, M. Caudy, M. Mangone, A. Day, E. Nickerson, et al. 2002. The generic genome browser: A building block for a model organism system database. *Genome Res* 12:1599–610.
7. Cruz, N. de la, S. Bromberg, D. Pasko, M. Shimoyama, S. Twigger, J. Chen, C. F. Chen, et al. 2005. The Rat Genome Database (RGD): Developments towards a phenome database. *Nucleic Acids Res* 33, Database Issue:D485–91.
8. Drysdale, R. A., M. A. Crosby, W. Gelbart, K. Campbell, D. Emmert, B. Matthews, S. Russo, et al. 2005. FlyBase: Genes and gene models. *Nucleic Acids Res* 33, Database Issue:D390–5.
9. Chen, N., T. W. Harris, I. Antoshechkin, C. Bastiani, T. Bieri, D. Blasiar, K. Bradnam, et al. 2005. WormBase: A comprehensive data resource for Caenorhabditis biology and genomics. *Nucleic Acids Res* 33, Database Issue:D383–9.
10. Harris, M. A., J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, et al. 2004. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res*32, Database Issue:D258–61.
11. Joshi-Tope, G., M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, et al. 2005. Reactome: A knowledgebase of biological pathways. *Nucleic Acids Res* 33, Database Issue:D428–32.
12. Kimball, R., L. Reeves, M. Ross, and W. Thornthwaite. 1998. *The data warehouse lifecycle toolkit*. New York: Wiley.
13. Davidson, S. B., J. Crabtree, B. P. Brunk, J. Schug, V. Tannen, G. C. Overton, and C. J. Stoeckert Jr. 2001. K2/Kleisli and GUS: Experiments in integrated access to genomic data sources. *IBM Systems Journal* 40:512–31.
14. Michalickova, K., G. D. Bader, M. Dumontier, H. Lieu, D. Betel, R. Isserlin, and C. W. Hogue. 2002. SeqHound: Biological sequence and structure database as a platform for bioinformatics research. *BMC Bioinformatics* 3:32.

15. Oinn, T., M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, et al. 2004. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 20:3045–54.
16. Covitz, P. A., F. Hartel, C. Schaefer, C. S. De, G. Fragoso, H. Sahni, S. Gustafson, and K. H. Buetow. 2003. caCORE: A common infrastructure for cancer informatics. *Bioinformatics* 19:2404–12.
17. Shah, S. P., Y. Huang, T. Xu, M. M. Yuen, J. Ling, and B. F. Ouellette. 2005. Atlas—A data warehouse for integrative bioinformatics. *BMC Bioinformatics* 6:34.
18. Etzold, T., A. Ulyanov, and P. Argos. 1996. SRS: Information retrieval system for molecular biology data banks. *Methods Enzymol* 266:114–28
19. Kasprzyk, A., D. Keefe, D. Smedley, D. London, W. Spooner, C. Melsopp, M. Hammond, P. Rocca-Serra, T. Cox, and E. Birney. 2004. EnsMart: A generic system for fast and flexible access to biological data. *Genome Res* 14:160–9.
20. Mangalam, H. 2002. The Bio* toolkits—A brief overview. *Brief Bioinform* 3:296–302.
21. Stajich, J. E., D. Block, K. Boulez, S. E. Brenner, S. A. Chervitz, C. Dagdigian, G. Fuellen, et al. 2002. The Bioperl toolkit: Perl modules for the life sciences. *Genome Res* 10:1611–8.
22. Womble, D. D. 2000. GCG: The Wisconsin Package of sequence analysis programs. *Methods Mol Biol* 132:3–22.
23. Olson, S. A. 2002. EMBOSS opens up sequence analysis. European Molecular Biology Open Software Suite. *Brief Bioinform* 3:87–91.

17 BioIT Hardware Configuration

Philip Miller
University of California, San Diego

CONTENTS

17.1 INTRODUCTION

The application of BioIT computational hardware is an interdisciplinary field growing out of molecular genetics, mathematics, and computer science. This area is a result of the explosive growth of sequence databases and computational and digital communication capacity. Early sequencing projects generated demand for storage, retrieval, and comparison of newly generated sequence, while computational requirements focused on homology matching, sequence assembly, and taxonomy. The foundation for these analyses lay in information theory, statistical models, and pattern recognition algorithms. Methods of statistical testing, distributions, and models were required for an objective matching and quality assessment scoring. Performance issues arose from large-scale whole-genome sequencing and systems analysis, and database development became necessary for project development.

The postgenomics scenario is shifting emphasis in the field. In the past, much of bioinformatics revolved around BLAST and associated sequence and data management. The problem was to test the hypothesis that a query sequence would not occur by chance within the database (e.g., GenBank). Now it is more unusual to find novel sequences, and the emphasis is shifting to categorizing and collating functional and structural information. In this chapter, I cover the impact of continuing explosive data growth, followed by the hardware configurations to support heterogeneous computing,

and I conclude with issues related to distributed forms of laboratory information management.

17.2 COMPUTER HARDWARE SYSTEMS

17.2.1 BioIT Systems Design

Computer systems and communications networks compose the bioinformatics infrastructure. I discuss the planning and design of the infrastructure to meet the needs of bioinformatics and BioIT. There are three components of the systems: computational resources, storage facilities, and communications networks. BioIT systems can best be understood by developing a schematic design. A typical schematic is shown in figure 17.1.

Figure 17.1 shows the elements of the system and the connectivity of the elements using standard iconography. The computational resources are a central server and desktop analytic workstations. The disk storage and backup systems are

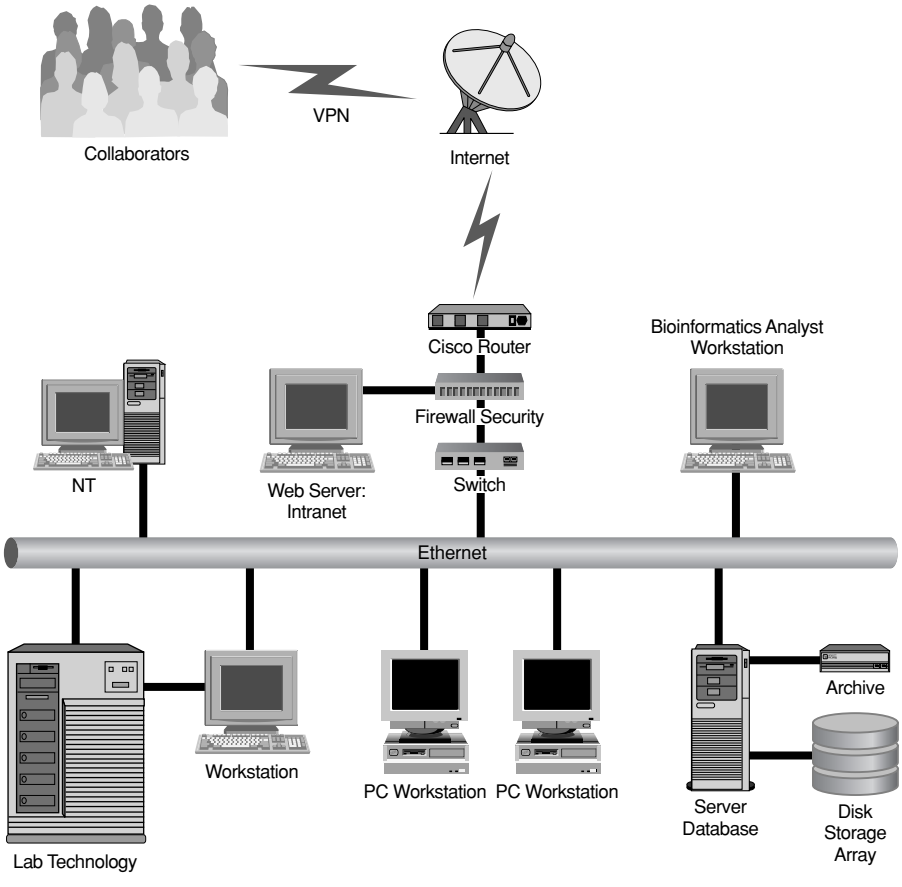


FIGURE 17.1 BioIT schematic.

indicated. Additional desktop systems are denoted for office and mail facilities. A network backbone interconnects the components and routes messages to the Internet portal. Finally, a security system is shown for Internet security functions. Characteristic of bioinformatics systems are an emphasis on large-disk storage and computational capacity in desktop workstations. In the following sections, these aspects of performance and capacity in the architecture of BioIT systems are emphasized.

GenBank doubled in less than one year; this rate exceeds the growth rate in computing capacity, which has been roughly modeled by Moore's law as a doubling rate of one and a half years. The cumulative effect has been a large increase in the cost of computing for the main computational loads, such as homology searching or assembly. Figure 17.2 shows the relative rates of increase over the past decade.

Sizing of the systems is based on capacity planning. To estimate the computational load and the performance of database searching and retrieval, samples specifications can be developed. Trial runs and *a priori* calculations are two common methods. BLAST, for example, is a standard homology searching algorithm. Hardware sizing for BLAST operations is a function of processing speed, memory size, disk retrieval, and software optimization.

In sizing the computational hardware for a large human-genome-sequencing lab, the following combined approach was used. The requirement was 4,500 base pair queries in four hours. Based on the performance analysis conducted in the year 2000, a SUN 4500 server was unable to meet this requirement, even in a 16-processor configuration. In addition to hardware configuration planning, the option of gaining performance through software optimization was explored. Profiling of the National Center for Biotechnology Information BLAST program identified four lines of C-code on which 75% of the computing time was spent. These lines carried out data-fetching operations. Software optimization [1] was done by rewriting the code in lower-level assembly language and resulted in a 100% speedup of performance, meeting the specification of the genome laboratory. Figure 17.3 shows the benefit of software optimization, comparing pre- and postoptimization performance for BLAST. Optimizing software is often cost-effective when designing systems for sequence searching with BLAST, BLAT, or any other search algorithm.

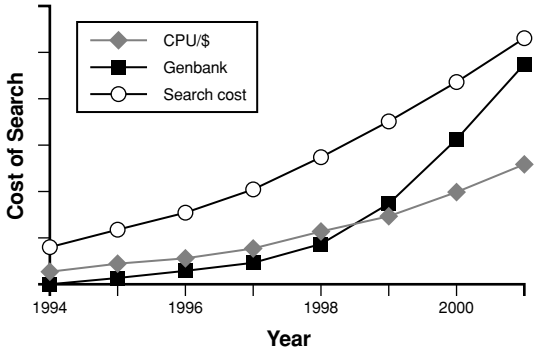


FIGURE 17.2 Comparison of relative growth of GenBank, computing performance, and the cost of executing a search.

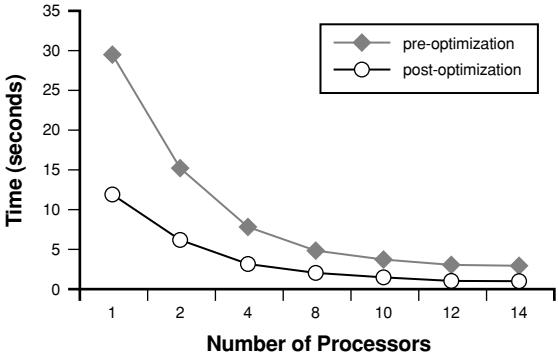


FIGURE 17.3 BLAST performance, time in seconds to process a 500 base pair query against the NR database, and pre- and postsoftware optimization.

In hardware sizing and capacity planning, two sources of data are used. The first is the server capacity, often given as a SPEC mark, or the time in which a standard set of procedures is executed. A less-accurate measure of computing capacity is processor speed in cycles per second. Memory size may play a role in some calculations. BLAST queries, for example, are limited by the fetches from the disk; the rule is, the more memory the better. Optimally, the entire nonredundant GenBank database can be stored in memory rather than on disk. The second component of the capacity analysis is the computation load, modeled as a typical workload factors.

Sizing of hardware is a cost–benefit problem. The results of benchmarks can be applied to various complements of hardware, and the best configuration can then be selected. More complex analyses are appropriate for communications loading problems. In these cases, the analysis can become nonlinear.

17.2.2 Cluster Computing

Alternative server architectures are considered in this section on cluster computing. Cluster computing, or arrays, is a trend in computing for bioinformatics because of superior price/performance characteristics. Many search algorithms can be easily implemented simultaneously by dividing the data into smaller sets and then recombining the computed results. Based on comparative studies [2], LINUX clusters using low-cost, standard PCs may offer as much as a 5:1 price–performance advantage over mainframe servers.

Systems configuration of clusters may involve unique considerations. The nature of clusters of computers necessarily requires additional factors involved in the systems management. To demonstrate one of these issues, I present a brief case study. A 100-node LINUX cluster was configured with standard PCs and provided heterogeneous information services and search and retrieval functions for a large external user base. The organization used the cluster to deliver general, informational content from multiple databases. In managing this cluster, an unusual failure syndrome was observed.

Figure 17.4 shows the daily log of one node of the system from the 100-node LINUX cluster. This graph shows the number of transactions per second on the

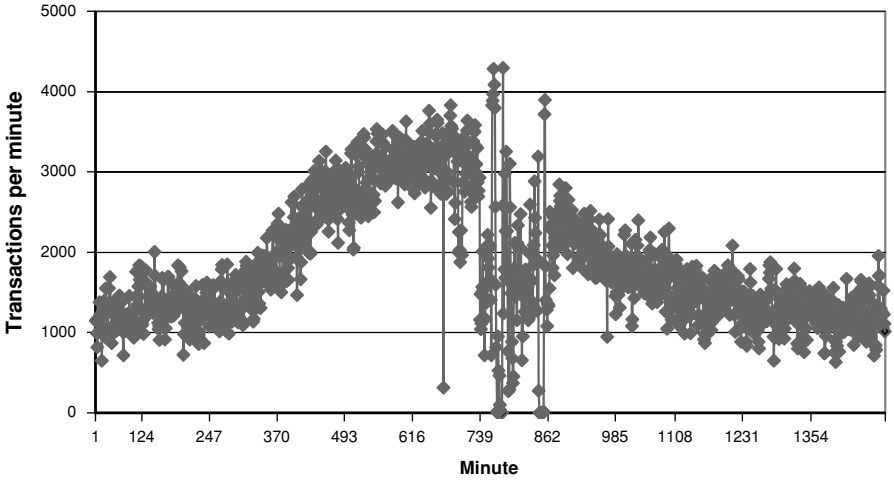


FIGURE 17.4 LINUX cluster performance for 24 hours. The log of transactions per minute for single element of 100-node cluster.

vertical axis and time as a 24-hour cycle, in minutes, on the horizontal. The system showed oscillating failure, beginning at approximately 12.00 with damping after about 1 hour.

The LINUX cluster had been managed under continually increasing load for a period of months. Systems administrators had responded by incremental addition of PCs, but the performance had shown continued decline. The system was loaded near capacity, which is often far below theoretical, 70% of maximum being a common benchmark. Another feature of the system, which is especially relevant to clusters, was the impact of component failure. For computers, this is likely the disk drive. Drives have a mean time between failure of one to two years and follow a "bathtub" curve of burn-in failure, followed by reliable performance and then high failure near the end of the lifecycle. Cluster design must take into account the need for graceful degradation.

The cause of the failure was localized to load balancing. The process of allocating load to the PCs resided in a device called a web switch. The effect of the load balancer under high load conditions and impacted by sudden disk failure was unplanned. When a disk failed, the load balancer was programmed to shift all current sessions from that PC to its neighbor. As a result, under high load conditions, failures cascaded throughout the system, demonstrating the ringing shown in figure 17.4. The problem was fixed by reprogramming the web switch to distribute the load of a single node failure to all of the systems. This scenario illustrates some of the system management implications of cluster computing in bioinformatics. In fact, because cluster computing has become so important in BioIT systems, many of the major computer manufacturers such as SUN, IBM, and HP now offer cluster solutions. There are also specialized providers, and a public domain system software—Beowulf—is available. The major systems management implications of clusters are for the optimization and maintenance of highly redundant systems. Load balancing, fall-back systems, and traffic management are important areas to consider.

17.2.3 Communications Network and Security

A well-designed communication network is the backbone of the BioIT system. It makes possible the interconnection of diverse systems, ties together geographically separated workgroups, and is especially important to the long and complex process of drug development. The network consists of internal and external communications systems. The internal network consists of (a) the physical wiring and the interfaces and (b) the switching and routing hardware. The layout of the wiring is done in the planning stages of the facility or by incrementally adding additional wiring as needed. Alternatively, wireless systems can be used without the need for extensive installation.

Communications systems performance is often nonlinear. *Queuing theory* [3,4] provides the best view of this behavior. Simple rules of thumb can be used to reflect the nonlinearity of communications systems. For example, the loading capability of communications channels is often 60 to 70% of the theoretical maximum capacity, because overhead becomes proportionately greater in the total communications as the load increases. A direct analogy to highway systems is apt and a close analogy. Highway traffic behavior will slow to a crawl in a traffic jam but then will suddenly come to a halt. Likewise, digital communications systems at near capacity can often crash precipitously. For this reason, appropriate nonlinear models should be used for planning the capacity of the communications channel and hardware.

External communications to the Internet are designed based on the requisite communications load. Typical transmission rates and load packages can be used to estimate the size of the communications channel. The junction of the external communications channel is also the position of the Internet firewall. Security requirements for external communications are implemented in the firewall policy. The policy determines which communications may be received and transmitted from the facility.

In general, security risks consist of external and internal security threats. Protection of proprietary information and recent government legislation, such as HIPPA and CFR 11 [5] has heightened the importance of security in biotech. The scale of risk, types of threats, and potential economic impact of security breaches should be considered in devising a security policy. Internal security for the company is often neglected. Technical solutions for ensuring a high level of internal security, appropriate to biotechnology, should be considered. For example, a system implementation of trusted operating system will provide a higher level of control over access to documents, files, and output devices such as printers and disk drives. The various roles and levels of security appropriate to a biotechnology company should also be incorporated into a well-designed security policy.

17.3 LIMS, MATERIAL TRACKING, AND RFID

LIMS, or laboratory information management system, involves all of the technologies in the acquisition, transmission, and storage of laboratory measurements. LIMSs typically consist of measurement instruments, network, and data storage hardware and software.

LIMS is also involved with material tracking. Material transfer tagging is of utmost importance in genomic research because of the long and complex screening process. In brief, these processes include field collection, molecular characterization, genetic cloning, chemical screening, and analytic chemistry. Completely reliable registration is required for this lengthy procedure. Completion time may be many years and require material transfer between several laboratory sites. Thousands of compounds can be derived from these assays.

One new technology that is well suited to material tracking is radio frequency identification devices (RFID). RFIDs are passive components that store and transmit a digital bar code [6]. The tag is interrogated wirelessly by the reader, usually at power densities far less than cellular phones. Line of sight reading, as in visual bar codes, is not necessary. The current generation of RFID tags also provides storage of up to 2KB of data on the chip. The chip is typically the size of a postage stamp and the thickness of the mylar or other tape on which it is layered.

To illustrate the application of RFID in LIMS, consider the following example. This system was designed for an agricultural genomics research laboratory. There were three major hardware components:

1. An RFID system consisting of transceiver, personal digital assistant (PDA) platform, and RFID tags at 13.54 MHz. (The PDA was field portable and used in the collection of field samples and in the laboratory.)
2. A global positioning system for acquiring position in the field.
3. A relational database to store and retrieve RFID information stored in the PDA.

An RFID tag could be affixed to a microtiter plate (figure 17.5) or to standard laboratory tubes. The storage capacity of the tag was used to record geographical

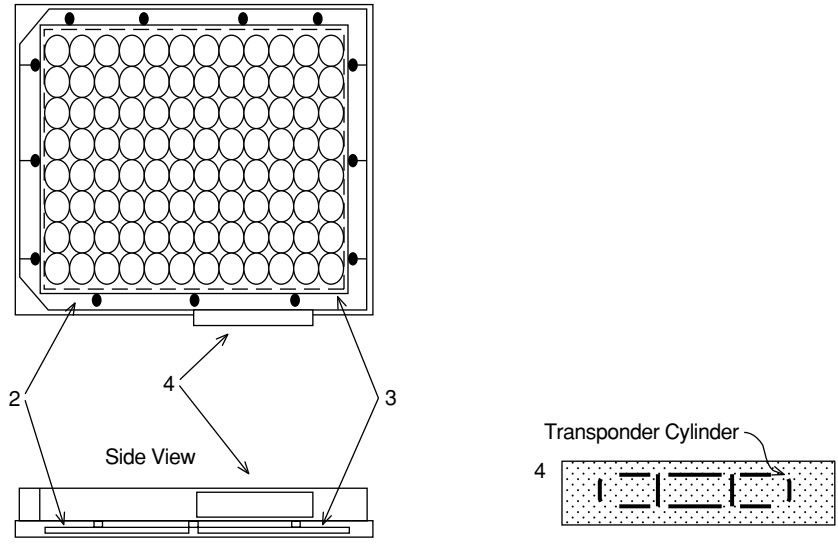


FIGURE 17.5 RFID tag and microtiter plate.

position information, processing steps, and a permanent label code. These data were stored at the time of capture in the field on the tag. This RFID system was beneficial because the information travels with the material. With RFID, the plate or tube becomes a distributed data source. In the highly diverse and lengthy process of pharmaceutical development, RFID offers a more reliable and efficient method for material handling and tracking.

17.4 CONCLUSIONS

Hardware design is a function of sizing, component selection, and system integration. A knowledge of the software applications and the research process is also essential. Systems for BioIT are composed of server, storage, communication, security, LIMS, and material handling. Cluster computing is an important trend for meeting future computational demands. New directions in material handling are appearing in the RFID technology. Mandated RFID systems for large corporations and government are pushing this technology toward speedy implementation. The integration of other wireless technologies such as PDAs and WiFi is an important direction, providing increased flexibility and mobility. These technologies are also of interest given a heightened awareness of the need for tracking and security in biotechnology.

REFERENCES

1. Crawford, I., and K. Wadleigh. 2000. *Software optimization for high performance computing*. Upper Saddle River, NJ: Prentice-Hall.
2. Wayne, J. S., and A. Mink. 1999. LINUX clusters at NIST. *LINUX Journal* 62:105–109.
3. Kleinrock, L. 1976. *Queuing systems, Volume 1: Theory.* New York: Wiley Interscience.
4. Robertazzi, T. 1994. *Computer networks and systems: Queuing theory and performance evaluation*. New York: Springer-Verlag.
5. Bass, S., L. Miller, and B. Nylin. 2002. *HIPAA compliance solutions*. Redmond, WA: Microsoft Press.
6. Fenkenzeller, K. 2003. *RFID handbook*. New York: Wiley.

18 BioIT Architecture: Software Architecture for Bioinformatics Research

Michael Dickson
RLX Technologies

CONTENTS

18.1 A DEFINITION OF BIOIT ARCHITECTURE

Software architecture used in discovery informatics is characterized by its diversity. It is rare to find an environment used in target identification and validation that consists solely of commercial off-the-shelf software. In most cases the research environment consists of a variety of commercial, open-source, and locally developed software packages. HTML and Web technologies are often used as a mechanism to integrate these disparate environments. One aspect of research IT is its datacentric nature, often involving a wide variety and types of data in unstructured and semistructured textual forms and in structured forms as relational and object-oriented

databases. This heterogeneous collection of software and data often seen in drug-discovery informatics can pose some significant challenges to administer and manage. The research process itself is one of experimentation, iteration, and hypothesis testing. In this environment the analysis tools utilized can be changed or augmented frequently. This process requires that the applied software architecture be flexible enough to accommodate rapid and frequent changes. Ultimately the requirements for the integration of informatics software and data are driven by the research process.

18.2 REQUIREMENTS THAT DRIVE BIOIT ARCHITECTURE

The research process imposes a variety of requirements on any software architecture used in target identification and validation.

18.2.1 Integration of Public Versus Proprietary Data

The informatics process requires the integration and analysis of datasets in the public domain as well as from proprietary data sources that represent the intellectual property of the research organization. Public flat-file biology data are often curated and delivered from a number of public ftp sites. This information is released periodically, and any private storage of this information must be kept up to date with the public data as they are released. In addition, these data may be available in a variety of physical formats, such as flat file, Extensible Markup Language (XML), and relational. Software systems must support the ability to keep proprietary information physically separate to facilitate both the periodic update of public data and the integrity and security of private intellectual property.

18.2.2 Compute-Intensive Analytical Algorithms

Many of the computational algorithms used in target identification and validation are compute intensive and can benefit from a high-performance cluster (HPC) or high-availability grid-computing environment. Scheduling a computation on a cluster and coordinating the flow of information between steps in the computation can complicate the execution of analyses. The computations used in informatics analyses can also consume and produce significant datasets. The computational infrastucture must therefore maximize access and bandwidth to the data. The research software systems must be able to discover, monitor, and manage resources within the system; schedule jobs on the cluster; and manage the flow of information between services within the cluster. All these tasks require monitoring and provisioning of systems to balance the utilization of the cluster across the variety of workloads being scheduled at any given time.

18.2.3 Annotation of Knowledge onto Existing Data

Software systems must provide for regular update of public data sources while ensuring the security for proprietary intellectual property. Knowledge gained while

analyzing experimental results from the research process must also be captured and related to the public and private data. The relationships between public, proprietary, and analytical datasets can be captured in a "gene index," which provides a focal point for linking information across the various information sources. The gene index serves as a map of the genome, with entries on the map that record relationships within the genome and to data stored separately. The ability to store relationships between data elements allows the gene index to serve as a navigation point for this genomic data without the need to build an integrated dataset that combines proprietary data with the public data sets. Ideally this index is a separate optimized database that uses stable identifiers to refer to the linked data. The gene index and "knowledge" stored within it represent a critical information source to document research progress and to provide decision support capabilities as the research program progresses.

18.2.4 Information Sharing Across Project and Geographic Boundaries

Research is often performed by groups distributed across multiple projects, organizations, and geographical boundaries. The research system must support these interactions while ensuring the security and integrity of research information. Controlled access to information through secure channels must be supported by the research system. This access may need to be provided across multiple tiers in the system: access to presentation components may be different than the access provided by services within the system.

18.2.5 Ability to Quickly Adopt New Research Methods

During the course of the research process, new methods will be applied to analyze and collect information in support of this process. Software systems used in this research must be flexible enough to incorporate new methods into the data and processing models built into the system. The system must support efficient repurposing of compute hardware and flexible description and discovery of computational and data resources within the system.

18.2.6 Manageability Built into the Infrastructure

The construction and management of complex compute IT infrastructures, often utilized when constructing a bioinformatics research system, introduce some important site management issues that must be addressed sufficiently to successfully control the complexity and cost to adminster such an environment. Complex compute clusters introduce power and cooling considerations that must be carefully planned for. Resource allocation of compute nodes, allocation, and mapping of storage connections and administration of network connectivity and bandwidth must be addressed throughout the lifetime of the compute infrastructure. In extreme cases, failure to manage the infrastructure can impact the ability to realize the expected return on investment from the installation of computational resources. The infrastructure must provide tools to configure and tune the deployment of services to fully realize the benefits of the architecture.

18.3 AN ARCHITECTURE THAT REALIZES THE REQUIREMENTS

18.3.1 High-Performance Computing and Computing on Demand

Because the process of scientific inquiry involves hypothesis testing and a variety of compute-intensive analytical methods, a flexible compute architecture must be able to be redeployed and repurposed quickly to support the needs of the various research activities and research efforts. While it might seem attractive to build dedicated compute facilities to support the various programs, this is often not practical because of the cost of the software and hardware assets and the associated maintenance cost. In the past, a large multiprocessor "supercomputer" would be deployed to meet the computational and data-handling demands. Unfortunately such a resource was fiscally very expensive, and reconfiguring to handle new types of workloads would result in costly downtime and further restrict the ability to share it. For the past few years, a more common configuration is to utilize commodity-priced hardware and gigabit (or faster) network connections to construct compute clusters that can be scaled and reconfigured to meet the changing demands. High-speed network connectivity can also be utilized to support network-attached storage, which can often be flexibly mapped to the appropriate compute nodes to support the repurposing of the homogenous compute hardware.

Development of a shared, managed compute infrastructure is often the most practical approach when supporting reconfiguration of the compute infrastructure to address changing analytical needs. This approach to computing is being utilized in both high-performance clusters and more loosely coupled computational grids. While the single supercomputer can be inflexible to use, it is often easier to manage because the design of the system is realized in a static hardware configuration. HPC computers or loosely coupled grids require more management to configure them for the workloads run on the cluster. This management cost is a small price to pay, however, for the return on investment realized from the ability to flexibly use the cluster for a variety of computational tasks.

The software architecture used to deploy and manage these infrastructures must support the identification and location of services dynamically and support the flexibility to quickly repurpose compute nodes and map data resources to these nodes to meet application-processing requirements. It should be relatively easy to reallocate resources or add new compute nodes or data sources to the cluster. The capabilities used to manage and monitor such an environment represent an active and developing segment of the software market; for example, a variety of solutions exist commercially and within the open source community. One project that is bringing together some of the best-of-breed elements utilized in clusters is the SourceForge OpenSSI project.[1] There are also a number of commercial hardware and software technologies

[1] The OpenSSI project at http://www.openssii.org is attempting to build a "supercomputer"-style environment representing a Single System Image out of homogenous clustered nodes of Linux systems. There is more information about SSI-style clusters and other clustered systems at http://www.linuxhpc.org.

available that can help to manage a clustered compute environment. Choices in this area can have a significant impact on the cost to maintain and operate the computing infrastructure, and they can affect the subsequent return on the investment the compute platform represents.

18.3.2 Service-Oriented Architecture

Service-oriented architecture (SOA) is an architecture that defines a loosely coupled multitier platform organized into logical layers that separate the delivery of data and software into components organized and presented as services. This architecture allows evolution and replacement of components in a scalable, stable environment. A services approach also avoids the drawbacks of more traditional "software silo" approaches (which are characterized by tightly coupled, specialized, single-purpose software elements) by supporting evolution of the environment through the incorporation of new types and instances of services. An SOA can support multiple user presentations, including batch access, Web clients, and desktop clients. One can organize the research informatics system into sets of cooperating services by grouping common functional elements and defining service interfaces that export functionality to cooperating service tiers. In this approach one can replace any individual component with another that implements a service interface. One can also duplicate and distribute or replicate services across compute nodes to scale the capabilities of the system to support distributed or greater numbers of clients. Figure 18.1 shows groupings of services that collectively can be utilized to construct a research informatics system. A discussion of each service grouping with examples for life sciences applications is presented next.

The physical/logical hosting environment represents the physical compute infrastructure used to make up the research system. This infrastructure includes compute

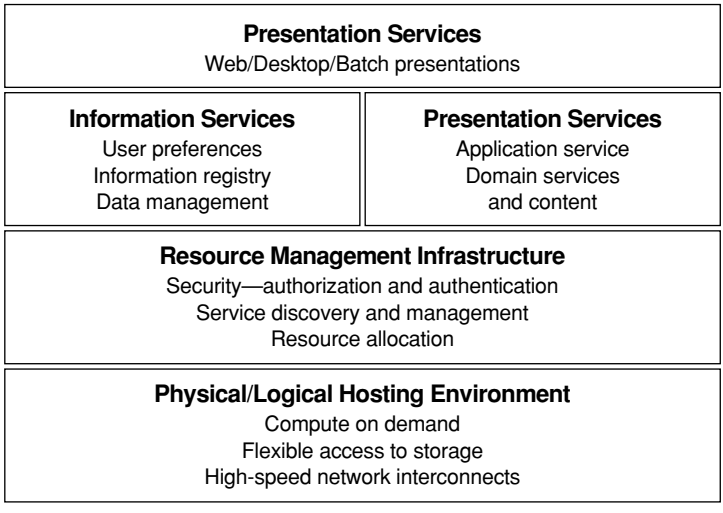


FIGURE 18.1 Service-Oriented Architecture

nodes, network connectivity, direct or network-attached external storage, and systems software configurations. Ideally the physical hardware infrastructure should consist of homogenous, often commodity priced, compute nodes. Each node should be configured with high-speed network interconnects such as 2 gigabit (or better) Ethernet interfaces, FibreChannel or Infiniband, or similar technologies. Local storage for booting is convenient, although network booting is also possible and may simplify configuration of the compute cluster. In data-intensive applications such as those in informatics, access to a storage area network (SAN) for data access is also desirable. A SAN is a high-speed subnetwork of shared hard-disk devices. In conjunction with a clustered file system, a SAN can be used to provide shared high-speed access to large datasets.

There is a growing trend toward hosting applications and services in an environment where compute capacity can be quickly allocated and reallocated to solve problems. In a traditional system, a large n-way multiprocessor machine might have been constructed to meet the specialized computational demands of a specific application. This machine would likely have directly attached storage to meet the performance needs of such a server. With the advent of network-attached storage (NAS) and high-speed interconnects like FibreChannel and Infiniband to provide switched connectivity to the SAN disk farms, it is possible to construct a compute engine made up of commodity-priced processors that will support such an application and allow for the addition of compute nodes as needed.

By utilizing homogenous compute nodes, it is relatively easy to add capacity or to reallocate capacity to applications on the cluster as needed. The "blade server" marketplace is an example of a hardware configuration specifically suited to the construction of "utility" compute infrastructures. In most such environments, many aspects of the cluster can be managed without the need to ever physically enter a data center to touch the hardware. Blade servers provide cost-effective high-performance computing with lower maintenance and management costs. For example, the Sanger Institute has deployed a high-performance cluster made of clustered blade servers, resulting in increased compute capacity while lowering the overall cost to manage and maintain the cluster.[2]

The resource management layer provides services to manage the compute infrastructure and map resources to compute tasks. Within this layer, access control policies that describe permitted/disallowed access to compute nodes, disk and memory resources, and applications are applied by services that accept jobs from the

[2] RLX provides server management software for rack dense and blade server environments. RLX has years of experience in developing and managing scale-out environments. Originally the inventor of the blade server form factor, RLX is now focused on the software that manages and automates these environments. The RLX ControlTower™ management application provides tools to provision an operating environment on the blades. Cluster Manager, a value-added ControlTower component, supports the definition and monitoring of compute clusters and the configuration, including the addition or removal of compute resources to the managed cluster environment. More information about the RLX product line can be found at http://www.rlx.com. The Sanger installation was described in an article in *BioIT World*. For more information see http://www.bio-itworld.com/archive/031704/computing_sidebar_4637.html

aforementioned application layer. A common approach to this problem is to deploy a distributed resource manager[3] to accept jobs into queues that are configured to provide various levels of service. Jobs are then scheduled to nodes on the cluster based on attributes attached to the queue and the job. Job scheduling in such a resource-managed environment can be configured to react to dependencies between tasks, the time a job is queued, how compute or data intensive a job is, or other related factors.

In addition to scheduling against compute nodes, the Resource Management layer should provide tools to locate and advertise services. The service location protocol[4] (or SLP) provides a capability to dynamically locate services that meet desired criteria. Since a computational environment involves not only scheduling across compute nodes but also potentially the need to dynamically connect to data providers and other service elements, the SLP protocol provides an important interface to flexibly locate and connect to distributed network services, limiting explicit coupling within the system.

Management and allocation of storage devices is also often a requirement. In particular, mapping NAS volumes to compute nodes is important in fully utilizing a cluster. Unfortunately, storage management services traditionally have been provided by storage vendors themselves, and the solutions they provide tend to be proprietary. The Storage Network Industry Association[5] (SNIA) is attempting to change this and has published a standard based on the Device Management Task Force (DMTF) Common Information Model (CIM). The DMTF CIM model provides a mechanism to model and describe management information services in a class and objects approach. Information about the model is stored in a CIM Object Manager and accessed via Web-based network protocols. Adoption of SNIA's CIM-based storage management model has been slow but progressing. When designing a research system where logical mapping of storage is required, working closely with vendors to produce an optimal and interoperable solution is still required.

The information services layer provides general-purpose infrastructure services such as authentication and authorization services, access to user preferences, metadata information, and data management and integration. The information services layer can also aggregate lower-level services across loosely coupled compute "grids." For example, resource management across a grid-compute environment may require negotiating with differing job-scheduling resource managers within tightly coupled compute clusters. These common services are accessible to the Presentation tier and

---

[3] There are a variety of available commercial and open-source resource managers available. A few common ones are Platform Computing LSF (http://www.platform.com), The Sun Grid Engine (http://gridengine.sunsource.net), and OpenPBS (http://www.openpbs.org). When scheduling across distributed environments, the Globus Toolkit DRM provides resource scheduling capabilities that can leverage the aforementioned tools (http://www.globus.org).

[4] The SLP is defined by the Internet Engineering Task Force. For the protocol specification see http://www.ietf.org/html.charters/svrloc-charter.html. An implementation of the SLP specification is available via OpenSLP (http://www.openslp.org).

[5] The SNIA is a standards body that produces specifications targeted at increasing the interoperability across storage vendors and storage management products. More information can be found at http://www.snia.org.

are usable by the domain components that are hosted inside the Application/Domain Integration tier.

The Globus Project[6] is an alliance of commercial software vendors and research organizations producing software that addresses many of the service requirements for information services. The project is a collaborative effort across organizations from the United States, Europe, and the Asia/Pacific. The project is basing their current work around Web-services standards including extensions, such as the WS–Resource Framework that extends Web services to support stateful interactions. The Globus toolkit contains software elements to handle security, data management, grid resource scheduling, monitoring, and management, and a development toolkit for extending and building applications using the supported services. Given the datacentric nature of research informatics, the data-management elements of the Globus toolkit provide some interesting tools to manage and query metadata and to manage and replicate datasets across the grid infrastructure.

Another noteworthy project with similar goals is the GridLab. GridLab is a research project focused primarily within Europe but with effective collaborations throughout the world. With similar goals to the Globus project, the GridLab project will produce a grid-enabled toolkit for producing grid–computing-aware applications, grid services, and a grid portal layer that can be used to implement the presentation services layer discussed next. The project's stated goal was to produce a completed production grid computing capability by December 2004. To date the GridLab project has released components for all of the major work areas addressed by the project. Updates to components continue and a number of major commercial and academic partners have adopted the infrastructure.

The GridLab project provides a number of important information services that support a layered services structure. The security services provide identification- and role-based authorization services. The data management infrastructure includes support for replica management, data movement, metadata management, and data structuring and organization. Services for resource management provide capabilities to schedule workload across grid-service elements and monitoring tools to manage quality of service across the grid. There are also services for event management that provide a monitoring infrastructure and a mechanism for services to source and synchronize events. The GridLab adaptive services components provide metrics for other service elements to use to make decisions about how to modify behavior based on changes to the grid-computing environment. Two other unique areas being addressed by the project are utilization of mobile devices to interact with the grid and tools for data visualization.

The domain services layer provides life-sciences-specific services, such as bioinformatics and cheminformatics applications and data, and tools that can be utilized to build and maintain the life sciences data and analytical models. There are a number of notable projects that can be utilized to provide components to build out an

[6] The Globus Project has produced a grid-computing toolkit largely based on proprietary protocols. The project's current development efforts are focused on grid services using Web-services standards to deliver services. For more information on the project see http://www.globus.org.

informatics capability. A group of projects developed under the auspices of the Open Bioinformatics Foundation (OBF) are particularly noteworthy. The most well known of these projects are the BioPerl and BioJava efforts, each of which provides language-centric classes and object definitions for the access and manipulation of biological data. In addition to the biological toolkits, there are projects that provide tools to access and query a biological model. The Distributed Annotation Service[7] (DAS) provides a capability to browse and query a reference server for genomic data. One or more "tracks" providing genome annotation information can be layered on top of this reference data. DAS provides a simple but powerful model for the presentation of genomic information and both public and proprietary annotation data. Similarly, the OBF MOBY[8] project is attempting to provide a more generally useful integration layer through the definition of Web services (MOBY–S) and the use of XML-based semantic Web technologies (S–MOBY) like RDF and OWL.

Another project affiliated loosely with the OBF is the European Molecular Biology Open Software Suite (EMBOSS)[9]. EMBOSS is a software toolkit and collection of applications specifically focused on molecular biology. There are hundreds of applications covering areas such as sequence alignment, database searching including pattern searching, protein motif and domain analysis, DNA sequence pattern analysis, sequence characterization tools, identification of sequence patterns in large sets of sequences, and tools to format biological data for publication. All the EMBOSS programs are built upon a library-based toolkit that may be used to support the development of new types of analyses.

The Interoperable Informatics Infrastructure Consortium (I3C) has sponsored the definition of a specification for an identifier that can be used to reference and locate biological objects. The Life Sciences Identifiers Definition[10] defines the syntax of a stable identifier and a mechanism to locate an authoritative server and resolve the identifier to a biological object. There are a number of reference implementations of this specification, and it has generated some interest in the bioinformatics community. It has yet to achieve widespread adoption but deserves a look if a research system requires the ability to specify stable identifiers for biological objects.

Combining domain services that define the object and service models for the biological content and analyses with other application services (like data-integration services, information searching, etc.) produces a model system for the definition and execution of bioinformatics research. By layering the application and domain model on top of a flexible service-oriented architecture, one can scale the infrastructure to

---

[7] The Open Bioinformatics Foundation efforts focused on the distributed annotation service are described at http://biodas.org.

[8] The MOBY project is focused on provided data integration through the use of services and metadata. More information can be found at http://www.biomoby.org.

[9] The EMBOSS project provides a toolkit and analysis tools for bioinformatics research. In addition to providing analyses to characterize DNA and protein sequences, EMBOSS also supports simple searching and retrieval of sequence information. For more info see http://emboss.sourceforge.net.

[10] The I3C has specificied a Uniform Resource Name syntax and semantics for a life-sciences identifier. Submissions of proposed implementations of these technologies were also made to the Life Sciences Research Domain Task Force at the OMG. More information on the I3C specification can be found at http://www.i3c.org/wgr/ta/resources/lsid/docs/index.asp.

handle large, compute-intensive problems. The architecture also supports the ability to flexibly reprovision the compute environment for new problems or to handle simultaneous computational activities. An example in bioinformatics might include access to an index of genes. A DAS reference server could provide the application service interface for access to genomic content by genetic locus. A DAS annotation server would then be accessed to retrieve annotations from the public information sources as well as locally created, proprietary annotations. A distributed resource manager could be used to coordinate the execution of analyses that generate annotations into the gene index that the DAS tools are serving. A cheminformatics example might include the ability to build a collection of compounds and apply a set of descriptor calculations implemented in the domain services layer to perform compound prioritization calculations on the collection. Again, the scheduling of these calculations can leverage clustered computational resources through the grid-aware resource-management services.

The Taverna Project[11] is an example of a domain-aware workflow toolkit that helps to define and perform these biological analyses and transformations. Taverna can handle both proprietary and XML-based data formats and uses XML itself to describe the workflows and data format transformations. Taverna also supports Web Service Definition Language (WSDL) and Web Services integration to exploit a services-oriented architecture when executing workflows, like the one described here. The "services" orientation that Taverna is built around works well with the services-based architecture presented here. Because Taverna interfaces with services (both informatics services and grid services like resource managers and security services), it is well suited for construction of the kinds of complex research pipelines common in bioinformatics research.

The presentation services layer provides an appropriate visualization mechanism to the end user. Because the architecture is organized as layered services, the platform can support a desktop client presentation as well as Web-based information retrieval presentation. A Web-based presentation is common choice to facilitate the often-distributed nature of the research process. Presentation services utilize the services provided by the layers below them for security and authentication, computational analysis, data management and information searching, and so on. There are a number of interesting presentation approaches that could be utilized at this level. Information "portals" and content management systems provide useful tools to organize and present research informatics data. A portal provides common elements such as menu structure and navigational tools, content transformation, and formatting capabilities, and it supports development and presentation of customized application content within the portal framework. This can also be a convenient mechanism to integrate third-party applications content that is not directly controlled by the research informatics staff.

---

[11] The Taverna project is hosted at SourceForge and provides Web-services-based computational elements; a description language for workflow definition; and a workbench GUI to visualize, define, and control workflows. For more information, see http://taverna.sourceforge.net.

In addition to the portal or content management system itself, the actual presentation that makes up the content of the research system will be defined and delivered using the presentation services. Selection of a flexible presentation infrastructure will allow separation of navigation and content from the actual screen layout and formatting. Improvements to the interactive nature of the Web-based presentation environment have continued to make browser-based HTML presentations for services and content the preferred mechanism for building and delivering informatics systems. Web-based portals provide an effective publishing framework for scientific content while managing many details of the presentation.

Macromedia Flash[12] has also emerged as a tool for developing rich presentation interfaces. Macromedia Flex is an example of a presentation services framework and application delivery framework that provides traditional application widgets not unlike those found in a desktop application while supporting their use in a browser-based environment. Development of applications in Flex is enhanced using tools to visually construct the user interface rapidly through prototyping. Utilizing technologies like Macromedia Flex enables the development of highly interactive user interfaces while using the Web to deliver applications and content in a way familiar to the research scientist. Macromedia Flash is also widely available on most operating platforms so deployment of applications without concern for the operating environment (Microsoft Windows, Apple OSX, and Linux) is possible. A project similar to Macromedia Flex has also recently been open-sourced. Lazlo and OpenLazlo also use Macromedia Flash for rendering and provide a presentation server and XML-based description format for construction of user interfaces. The resulting presentation provides a rich interactive client interface and connectivity into data and application services for business logic and content.

18.4 MODELING THE RESEARCH DOMAIN

Over the last few years, a number of attempts have been made to build distributed computing reference models for biological and chemical data. These efforts were designed to define a model for definition and interaction with biological data that could be extended across the network. These reference models would provide semantic integration through normalization of various back-end implementation and storage approaches through a behavioral model that standardizes the access to the implementations through methods on the objects. While many of these efforts contributed to a better understanding of how to model and access such systems, they did not produce the "standards" that would serve to define a reference model for research informatics.

---

[12] Macromedia Flex consists of a language definition, compilers, and interactive interface-building tools to quickly construct rich client presentations using the Web and Macromedia Flash for rendering. More info on Flex can be found at http://www.macromedia.com. The Lazlo system also defines an interface-description language and uses Macromedia Flash as its rendering engine. Lazlo has recently been open sourced under the Common Public License; see http://www.openlazlo.org.

Many of the previous modeling efforts operated from an implementation language approach. That is, the end goal was the description of a specific physical implementation technology. An alternative to starting with a physical implementation model is to utilize a model-driven approach that provides an implementation-neutral model of the domain and multiple models for the implementation tiers derived from the implementation-neutral model through rules describing the transformation. This model-driven approach is being employed in the Object Management Group (OMG) as the Model Driven Architecture[13] and has been used in life sciences to produce the Microarray and Gene Expression (MAGE)[14] specification for gene-expression analysis. MAGE-OM is a conceptual model for data exchange of data from expression experiments. MAGE-XML is derived from the MAGE-OM model and describes an XML-based interchange format. A related toolkit, MAGEstk, provides implementations of the conceptual model in various languages and code to import and export MAGE-ML formatted XML streams. Ideally a tool that automates the process would produce these mappings. Currently there are a few vendors delivering early versions of such tools, but current common practice is to derive these models manually. For example, Unified Markup Language (UML) and the XML-based XMI, a portable interchange format for UML, can be used to describe and define an abstract model. From that model, a variety of concrete repesentations could be defined. A physical storage representation via database table definitions, XML schema definitions to support interchange, and WSDL definitions describing a service layer that mediates access to the model are examples.

XML is becoming increasingly more important as a mechanism to exchange data between applications[15]. Because many types of bioinformatics analyses require iterative application of tools in a pipeline, a mechanism to transfer information between stages of a pipeline is required. In the past these were often proprietary formats, and it was necessary to write specialized tools to transform the output of one tool into the input of the next. XML and the use of style sheets to drive transformations promises to at least make the parsing and transformation of data metadata driven as opposed to customized for each new application pipeline.

While the adoption of XML is a positive step toward simplifying the interachange and processing of data in informatics pipelines, there is still a tendancy to invent a new XML-based format for each new solution to a problem. Utilizing an XML Document Type Definition (DTD) or schema-based approach is an improvement on flat files because of the variety of pre-existing tools to parse XML and the ability to apply a DTD or schema to drive the parsing. There is still much room for improvement if efforts to standardize XML-based interachange formats can be more broadly adopted. With the possible exception of the MAGE-ML gene-expression interchange effort, which succesfully exploited a standards body to produce a data

[13] Infomation about the OMG and Model Driven Architecture is available at the OMG Web site (http://www.omg.org).

[14] MAGE is an OMG standardized interchange format defined by the Microarray Gene Expression Data Society. For more information see http://www.mged.org/Workgroups/MAGE/mage.htm).

[15] The XML specification is maintained by the World Wide Web Consortium and is described at http://www.w3.org/XML.

model; XML interchange format; and language-specific processing models that have seen wide adoption, the development of "standard" formats for the manipulation of bioinformatics data has been limited. This is not for lack of trying. I and numerous others have participated in standards bodies to try to drive the adoption of such standards. Perhaps the iterative and more speculative nature of the research process tends to work against such efforts.

18.5 SUMMARY

Bioinformatics software architecture is unique in both its evolving nature and fast-paced development. BioIT must track the changing and advancing scientific research process while facilitating access to large amounts of data and new types of inquiries. In this chapter, I presented an architecture that addresses some of these fast-paced, changeable requirements through modularity and layered services. The foundation is a flexible physical architecture that emphasizes "compute on demand" through the ability to flexibly manage and reallocate compute resources. Management tools that facilitate this utility computing approach are mandatory to realize the benefits of a flexible, scalable computational capability.

On top of this flexible, managed computation environment, layered subsystems and services have been organized that allow one to replicate and scale services to meet the changing computational demands. This process can be achieved through the use of resource-management tools and clusters within a local environment and a grid-computing model when scaling and distributing across physical sites and potentially logical research organizations. The compute architecture should ideally extend and build on established capabilities and standards so the research scientist can focus on extension in areas where real value is added: within the bioinformatics arena directly. Thankfully, there are myriad excellent choices for bioinformatics research software (including value-added locally created and maintained software elements) that will work within the SOA that was described earlier.

There is a rich and expanding universe of available software and hardware tools to implement a research IT environment such as the one described here. If there is any critical take-away message, it would be to be willing to adopt and leverage the excellent existing capabilities within the research landscape. If one must build, build where you add the most value, and, in all other areas, walk in the excellent shoes of those who have explored this landscape before you. Well-researched and carefully selected service components will pay back in the future in the form of a flexible architecture that can be retooled and redeployed with maximal reuse to address the changing needs of research informatics.

19 Workflows and Data Pipelines

Michael Peeler
SciTegic, Inc.

CONTENTS

19.1 INTRODUCTION

Many practices in computational research environments are iterative in nature. A researcher might apply a series of operations to a dataset and then repeat this series a number of times, making slight modifications to the process on each occasion until some desired result is achieved. At a later date, the researcher might have new data for which the initial process has to be repeated. Automation of the whole process offers the significant advantage of reducing inconsistencies and errors, and the time savings are obvious. It is for this type of use that researchers increasingly prefer using workflow or data-pipelining tools over traditional software applications. Although these latter applications allow for automation in some cases (e.g., through a scripting engine), this process is less flexible and requires special expertise.

19.1.1 WORKFLOWS

The term *workflow* is applied loosely to various automated procedures and to flowcharts of human activities. However, for the purposes of this chapter, we use the following definition:

> A workflow is a network of well-described computational tasks that together accomplish a specific goal. The network defines the sequence of tasks in addition to decision points and resultant alternatives. Each task is described in terms of its input, output, and software dependencies.

Therefore, a workflow process has an explicit specification. This differentiates it from collaborative tools that allow people to communicate and share information but with no formal structure—in this case, the overall process is not well defined.

Most processes related to drug-target screening, identification, and validation do not consist of a single step but progress through a sequence of tasks directed toward the ultimate goal of the process. Indeed, this is true for most computational and business activities today, with the exception of some single-shot desktop activities such as writing a document or sending an e-mail. As a result, there has been much interest in this area over recent years, focused on software tools and on the interoperability between these tools.

A computational workflow is an attempt to capture a process that experience tells us is a "best-practice" approach, or a standard operating procedure. Once

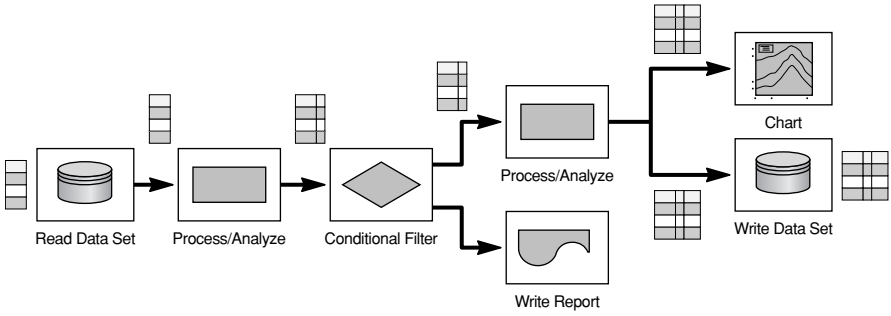


FIGURE 19.1 An example workflow, beginning with a data source, leading to a set of branching tasks and result datasets.

captured, the automation of the workflow can provide both high throughput and consistency in the way an organization runs the procedure. The consistency factor can be vital in the drug research and development field to ensure repeatability and to provide an electronic paper trail to meet corporate and governmental standards.

Workflow engines are software systems that (a) provide a framework for defining the set of tasks that constitutes a workflow and the links between the tasks, and (b) run a predefined workflow. Interest and investment in this area has grown as organizations have recognized the advantages in making consistent and optimized use of the software tools they have at their disposal.

A typical scientific workflow takes one or more source datasets as input and passes them through a sequence of branching tasks, each of which operates on one or more input datasets to generate one or more output datasets, until finally the result data are output from the system (see fig. 19.1).

Some authors [WF1] have drawn a contrast between data-centric "scientific" workflows (or dataflows) and task-oriented "business" workflows. In a scientific workflow, the flow of control is tied to the flow of data. In a business workflow, control flows between tasks that operate upon workflow objects, but the flow of control is independent of the data flow. Consider the difference between a workflow designed to extract novel compounds from a candidate data set and one designed to help a customer order a concert ticket.

In this chapter, we are primarily discussing the data-centric, scientific workflows.

19.1.2 Data Pipelines

A *data pipeline* has similar goals to a computational workflow. However, whereas a traditional workflow describes tasks that operate on an entire dataset, a data pipeline focuses on the management of data records. Each task in a pipeline is involved with the processing of a single, independent data record and passing it down the pipeline as early and rapidly as possible, with a primary goal of minimizing the memory footprint and maximizing data-throughput rates.

In an ideal data-pipelining case, the first data record of a dataset has passed entirely through the branching network of tasks in a data pipeline even before the reading of

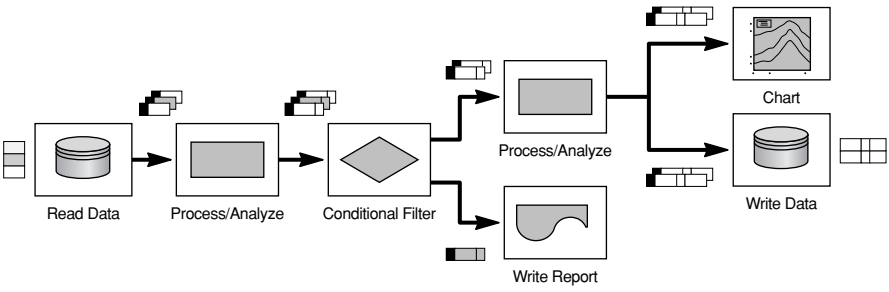


FIGURE 19.2 An example data pipeline, beginning with a data source, leading to a set of branching tasks, with data records flowing to fill the result datasets.

the second data record. Contrast this with a workflow implementation, where the first task processes the entire dataset before the invocation of the second task.

Data pipelines introduce a workflow paradigm at a more fine-grained level of analysis, without descending to the complexity of visual programming. In a traditional workflow system, each task is a black box that operates on the input dataset. In a data pipeline, the operational specification is at the data record level and therefore facilitates customization on a per-record basis. As an example, you can filter or edit individual records within the workflow itself (see fig. 19.2).

Data pipelines can introduce higher throughput for a number of reasons:

- The processing of individual data records minimizes memory overhead and therefore requires less data caching to disk between tasks or because of RAM limitations.
- The reduced memory overhead allows a greater number of workflows or workflow threads to execute simultaneously.
- Each task can get started as soon as a data record is available, without waiting for the entire input dataset to be complete. This flexibility makes for a more nimble system that can maximize the use of available compute resources.
- Optimized methods can focus on the real-time analysis of individual data records to achieve high throughput of entire data collections.
- Components can treat a data record flowing in a pipeline in the same manner regardless of its source location or format. Data from disparate sources (databases, flat files, or the Web) are handled in an identical manner, avoiding cumbersome data migration or data integration operations.

In reality, the workflow and data pipeline approaches are two ends of a spectrum, and in practice, most software systems adopt some elements of both.

- Some workflow systems can break a dataset up into smaller chunks for parallel processing or for other advantages of flexibility derived from the pipeline model.

- A data-pipeline system cannot always follow the ideal pipelining paradigm that treats each data record as independent from the others. For example, a component that has the task of reordering data records must cache all the data and treat them as an integral dataset.
- Integration with a legacy system imposes its own individual constraints, since each software program requires data input of a certain batch size for functional or performance reasons. A workflow management system has to be adaptable to pass data to a legacy system in whatever volume or format that it expects.

19.1.3 Workflow Management Challenges

Whether a workflow or data-pipeline strategy is used to orchestrate tasks in a process, all workflow engines face three broad management issues:

1. *Application management*. The tasks employed by a workflow must be invoked appropriately, taking into account the details of data input and result data retrieval. The applications may be run locally or on some other location on the network.
2. *Data management*. Data are what "flow" in a workflow. A workflow is generally initiated by reading data from a data source, and the final result of the workflow is often data that are written to a data destination. Different applications require data in different formats and employ different object models to represent similar data objects. The final formatting of the data may be performed to make that data readable by a human. The workflow engine needs the capability to manage this sort of complexity and flexibility when flowing data.
3. *Resource management*. This covers the computational and network realities that the workflow must deal with. Data reside in databases and on file servers with their individual networking and security considerations. Similarly, the invocation of an application that is needed to complete a workflow may have specific network requirements. To be useful in a typical organizational network environment, the workflow engine has to maintain information about network locations and network users, plus have the ability to utilize these data in the way that is expected by users and administrators.

19.2 BACKGROUND

The concept of a workflow is not new. A useful procedure generally attains its goals through the completion of a number of individual tasks organized in some appropriate manner. With respect to life sciences, the individual tasks are typically computational applications related to the management or analysis of structural or experimental data. The applications may be run sequentially or may be organized into a complex

network including decision points, iterative operations, parallel tasks, and synchronization, according to the needs of the problem to be solved. The term *workflow* relates to the concept of activity (or perhaps, more specifically, data) flowing in a flexible, liquid fashion between applications, represented as nodes in the network.

19.2.1 Manual Workflows

From a historical perspective, most workflow procedures required human mediation. Indeed, that is still true today in many instances, where the computational tasks in a workflow represent islands of automation surrounded by procedures that are essentially manual. A workflow can be represented graphically as a set of application nodes with lines between them representing the directed flow of activity. However, if we are to see these lines as "pipes" that channel the flow of data between applications, the graphical representation is often a distortion of the manual drudgery involved in making the "flow" a reality.

The tasks that constitute a workflow are often very heterogeneous and may include importing and exporting data in disparate formats, running individual programs on different hardware and system platforms that exist in mutually remote locations. Unless proximate tasks in a workflow happen to be sourced from a common vendor, chances are they have no way to communicate directly with each other. In discovery, research groups frequently use different databases from different vendors (e.g., one for biology and one for chemistry). Hence, database-centric solutions do not generally provide the flexibility to access data from another vendor's data schema.

Microsoft Excel is often the tool of choice for researchers to process and manage their data. However, as those datasets grow, Excel's scalability limitations become apparent, since it cannot work with datasets above a certain length or width and the automation of Excel is nontrivial to implement or support.

For these sorts of reasons, a human is necessarily involved in filling the gaps between the various tasks. Typically, the overseer of a workflow is responsible for the reformatting of data between tasks; the transfer of data over a network; and the parameterization, launch, and monitoring of software programs to work on the data.

19.2.2 Simple Automation of Static Workflows

One solution to the problem of access to heterogeneous data sources is the Extract, Transform and Load (ETL) approach. ETL is a system designed to move data from multiple sources, to clean it up for consistency and uniqueness, and then to store it in a central data warehouse for application access. The ETL process involves multiple custom steps and needs to be scalable to deal with large datasets and execute on a regular basis.

Such scenarios are obvious candidates for automation, and scripts or other custom-written software frequently supplement or supplant the purely manual procedures. Automation helps to overcome problems of human error, lack of consistency, lack of repeatability, and plain tediousness. In addition, a workflow that can run with minimal human intervention typically completes earlier, since it does not wait for anyone to type in the next command or to come back from lunch (or to be freed up from monitoring another workflow).

A programmer might compose a script to automate a static workflow, which can be useful in some predictable business applications that make use of equally unchanging back-end data infrastructure, such as payroll or order fulfillment. However, static workflow systems do not adapt well to the discovery process where the back-end infrastructure is diverse, the workflow itself needs to be flexible, and the outcomes are unpredictable.

19.2.3 Automated Workflow Engines

The lack of flexibility in static workflow orchestrations drives the need for more adaptable systems. However, the software challenge is significant to enhance and grow a customized, specific workflow system to one that supports requirements, such as the following:

- The system is extensible to different workflows.
- It supports elemental task definitions that are parameterized, modular, and reusable without dependencies on other tasks.
- The system is extensible to support the integration of new software tools and scripts.
- It is portable to different hardware, operating systems, and network locations.
- The system is upgradeable with bug fixes or new application capabilities.
- It facilitates the deployment of workflow execution to multiple users at different locations with little knowledge about the workflow itself.
- Standard error handling informs the user of problems in a consistent manner regardless of the underlying software tools in use.
- Administrative tools facilitate global configurations such as logging, access security, prioritization, and load balancing.

Expensive software-engineering skills are required to evolve a custom solution into a system of more general utility, which is also costly to maintain and adapt.

For these reasons and others, a market has developed for off-the-shelf workflow engines that orchestrate abstract tasks unknown to the developers of the workflow engine itself. This allows domain experts to focus on the development of such tasks and on the construction of workflows appropriate to the problems at hand.

19.2.4 Parallel Workflows

Increasing data volumes in the field of genomics and target discovery drive the demand for ever-increasing performance characteristics from software solutions. In addition to the continual improvements in raw processor speed, the principal historical solution has been to parallelize an individual software application, so that certain elements of its functionality can be run simultaneously with different data on multiple processors. This is a costly and nonuniversal solution, and the workflow context can provide an alternative perspective. Parallelization within a workflow can afford the end user a significant advantage in data throughput, without the need to find and purchase specialized parallel application software. Data-pipelining concepts have

the potential to take this a step further with the management of parallelization at the data-record level.

As a result, the demand for greater data throughput is driving an increase in the use of dedicated hardware clusters and grid-computing architectures that require scalable middleware and distributed management tools. The difficulty in adapting ad hoc script-based solutions to such technologies only helps to accentuate the requirement for robust workflow solutions that can leverage the hardware and systems investment in modern network architectures.

19.2.5 Workflow Theory

A background discussion of workflow systems would not be complete without a reference to some of the concepts that have supported the development of workflow management systems as a way to provide an organizational framework for tasks.

19.2.5.1 Petri Nets

Those interested in the analysis and modeling of workflow processes have often turned to a process-modeling technique known as the Petri net. This is a popular choice since much work exists that formalizes and enhances the technique introduced by Carl Petri in the 1960s [PN1]. A Petri net is a directed graph that describes the relationship between transitions and the conditions that trigger those transitions to take place. The net executes as transitions consume enabling tokens and produce new tokens that in turn enable other transitions.

In this chapter I do not detail the formal constructs that surround Petri nets (see [PN2, PN3] for some pointers), but it is interesting to look at some of the enhancements that have been required to reflect real-world workflows.

1. *Data values*. A transition not only moves tokens through the net but also may change its value, because in many modeled processes, the *content of the data* is what is important. The suitability of a workflow system for a given purpose depends on its ability to represent the complexity of the data that must flow though the system in addition to providing access to the operations that need to work on that data.
2. *Time*. Temporal issues are often an important element in useful workflows. The system should be able to represent delays, time limits, and time-based milestones.
3. *Hierachical encapsulation.* Real-world workflows can become very complex. It is useful to represent subflows of the workflow as a single simple entity, which can reveal its full detail only when required. This encapsulating behavior also encourages the clean reuse of these subflows and avoids tedious repetition of the same set of tasks in multiple locations in the workflow.

19.2.5.2 Workflow Patterns

In general, one can categorize workflow tasks as one of a number of common patterns. These workflow patterns can be useful in providing an abstract description of the sorts

of behavior that a workflow engine might support. An analysis of workflow patterns helps to provide a common vocabulary for these behaviors; it also facilitates a direct comparison between workflow products and languages. In choosing a workflow engine, there are many factors to consider, such as usability, extensibility, domain coverage, platform support, and price. It is also important to consider whether you will be able to describe and store the complexity of the workflows that are important to you.

The simplest workflow pattern is the *sequence*. (Start with Task A; then when A is finished, do B; and when B is finished, do C.) The customary visual representation of a sequence is a series of task blocks with a single link between successive tasks (see fig. 19.3A).

You will often see workflows in which the simple sequence pattern runs into a task that has multiple lines downstream. This is an example where a clear understanding of workflow patterns is useful. This visual description might represent a *parallel split*, in which data clone themselves and pass to two or more parallel or independent sets of processes (fig. 19.3B). Alternatively, it might represent an *exclusive choice*, in which the data pass to just one of a number of downstream options (fig. 19.3C). Clearly, these are very different behaviors, and indeed, they are described by different patterns.

Patterns exist for workflow elements such as merging, synchronization points, loops, and cancellation. At the more complex end of the spectrum, there are patterns that deal with multiple instances of a task, either specified at design time or generated on demand at run time.

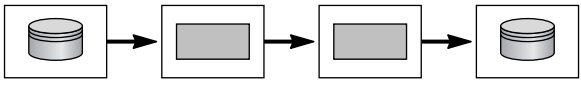


FIGURE 19.3A A simple sequence workflow pattern.

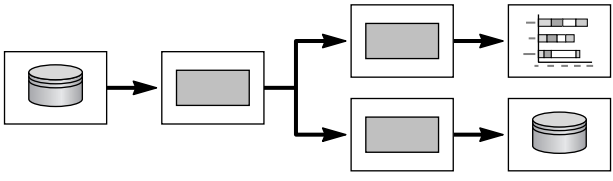


FIGURE 19.3B A parallel split workflow pattern.

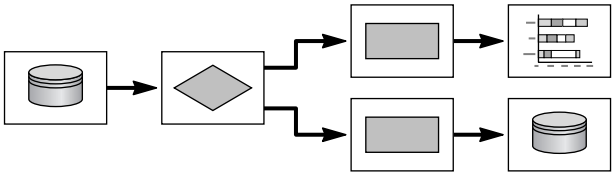


FIGURE 19.3C An exclusive choice workflow pattern.

Space does not permit a comprehensive description of possible workflow patterns in this chapter. A good place for more focused information is (unsurprisingly) http://www.workflowpatterns.com. At this site, you will find animations that illustrate the behavior of the 20 different workflow patterns that it catalogs [WP2]. The site also provides an analysis of the expressiveness of some workflow representation standards described in the Standards section of this chapter.

19.3 TOOLS

19.3.1 Tools Overview

This section contains descriptions of a cross-section of the workflow-based tools designed to manage computations in the discovery arena. This is not an exhaustive list of such products, or even of just those with a life science focus. In addition, although there are interesting workflow tools focused on modeling and statistical analysis, they are beyond the scope of this chapter (e.g., the Clementine workbench from SPSS [DM1] and Insightful Miner from Insightful Corp. [DM2]).

The section divides along commercial and noncommercial lines. Each tool description includes

- A descriptive overview
- A description of the product line and its application in life sciences
- A summary of specific or remarkable features

19.3.2 Commercial Workflow Tools

Information about the tools listed is limited to publicly available sources of data. Since workflow is an area of active interest and software development, there are frequent new releases of products; the best advice is to check the relevant Web site for up-to-date information on product specifics.

Although there are open-source initiatives related to workflow engines (see the next section), commercial organizations are typically able to provide higher-quality product support. They focus on the needs of larger life sciences organizations that are prepared to pay for high-performance, cross-platform, and scalable deployments, with supporting professional services if required.

Some vendor companies have Independent Software Vendor (ISV) or partner programs to encourage third parties to develop components for their platform. Since customers all have their preferred tools for a certain task, ISV relationships allow vendors to offer the required flexibility to their customers.

The listing is alphabetical by vendor name.

Company	Product	Headquarters	URL
Incogen	VIBE	Williamsburg, VA	www.incogen.com
InforSense	KDE	London, UK	www.inforsense.com
SciTegic	Pipeline Pilot	San Diego, CA	www.scitegic.com
TurboWorx	TurboWorx	Shelton, CT	www.turboworx.com
White Carbon	Pathways	Melbourn, Herts., UK	www.white-carbon.com

19.3.2.1 Incogen

19.3.2.1.1 Overview

Incogen is a life sciences software and services company. The Visual Integrated Bioinformatics Environment (VIBE), a visual programming and workflow platform, is one of their software offerings. The user can create workflows to import data from a number of sources and pass the data through common bioinformatics algorithms.

19.3.2.1.2 Product Line

VIBE is Incogen's bioinformatics workflow product. It is deployable in a client/server mode for workgroup access or stand-alone on a desktop machine.

1. The client software user interface (UI) features a canvas where the user creates and edits pipelines of modules, where each module represents a task in the workflow. Each module can have a set of parameter values that the user can edit. A module defines the types of data that it can accept as input and the types of data that it outputs. This information governs which modules can be connected.

   Pipelines start with an input module that sources an initial dataset. The remainder of the pipeline consists of analysis, transformation, or conditional filter modules. The designer can insert visualization modules to view and filter the data content at any point. The dataset passes to successive modules in the workflow and may be cloned to multiple outputs or conditionally branched as it passes through the workflow.
2. VIBE modules are divided into the following categories:
   a. Input modules that can read in source data as input to a pipeline. Data types supported are DNA and protein sequences, HMM data, and Trace chromatograms.
   b. Search modules that encapsulate common search algorithms for DNA and protein databases, such as BLAST, Smith–Waterman, and HMMER.
   c. Seals modules supporting some of the NCBI SEALS utilities for sequence analysis.
   d. Visualization modules for viewing the types of data supported by the software.
   e. Utility modules for various other tasks such as conditional filtering and sequence alignment.
3. The VIBE Software Development Kit (SDK) provides a way for users to add their own or third-party tools and data types to their VIBE installation. A combination of XML and application program interface (API) compliant Java classes can be used to add new modules and, if necessary, define new data types. An installation can then be hand-edited to include the custom extensions.

19.3.2.1.3 Product Summary

- VIBE focuses on bioinformatics workflows with support for industry-standard formats and tools.

- Data typing validation controls the workflow design.
- Extensibility for a local installation requires Java coding and XML files.

19.3.2.2 InforSense

19.3.2.2.1 Overview

Founded in 1999, InforSense markets a workflow engine known as Open Discovery Workflow, originating from Imperial College in London. The company's flagship software product is the Knowledge Discovery Environment (KDE) that supports a number of horizontal and vertical applications for life sciences discovery in addition to a graphical tool for the design and analysis of workflows.

19.3.2.2.2 Product Line

1. The KDE workflow engine is implemented on a J2EE server, which also provides centralized project management and data storage.
2. The user employs a graphical workflow builder to compose an analytic workflow built from a core database of statistical tasks or from the domain-specific modules.
3. The BioScience Module targets the needs of a user who perform data analyses across a number of life science application areas. It supports the data formats and content required for sequence analysis, gene expression analysis, SNP analysis, and proteomics.

   The module includes analysis tools for clustering and classification and statistical operations. It also integrates standard tools such as BLAST, ClustalW, and EMBOSS utilities. Interactive visualization software is available for viewing data structures relevant to the bioinformatics domain.

   An end user can extend the BioScience module by creating components to encapsulate the command lines of bioinformatics tools.
4. InforSense's ChemScience module targets cheminformatics solutions with a set of chemistry-specific components that a user can employ to construct a workflow for experimentation, data analysis, and visualization. Component capabilities include data import and export; structure-, descriptor-, and fingerprint-based processing; and library enumeration.

   Interactive tools are provided for structure rendering and other domain-specific visualizations. The ChemScience module incorporates chemistry algorithmic and visualization software from Chemaxon (http://www.chemaxon.com).
5. TextSense is InforSense's module for free text searching and analysis. These tools can generate raw results that may then act as input to the generic statistical and data-analysis tools.
6. The Oracle Edition module includes components that employ the native Oracle 10g data mining capabilities for data stored in that database. These components perform data analysis on the data without moving it out of the database.

7. The Discovery Portal product supports the publishing of a workflow as a Web application that can be accessed as a custom application via a browser or programmatically as a Web service.
8. Support for extensibility comes in the form of an SDK environment known as the Developer Pack. The Developer Pack allows a developer to use the published Java API to write a new local component that is deployable within an InforSense installation. In this way, select third-party or in-house functionality can be integrated into the workflows built by end users.

19.3.2.2.3 Product Summary

- The core Knowledge Discovery Environment is a workflow platform targeted at the needs of analysts handling large data volumes from disparate sources.
- Domain-specific modules for biology, chemistry, and text analysis supplement the core workflow technology.
- The platform supports a Web-services model for Web browser or application access to a validated workflow by a broader set of users.
- An SDK is available to support the creation of new components to integrate third-party or in-house tools.

19.3.2.3 SciTegic

19.3.2.3.1 Overview

SciTegic's software product is named Pipeline Pilot. It orchestrates tasks in a workflow using the concept of data pipelining as outlined earlier in this chapter. Individual data records flow through a pipeline of components to minimize memory footprint and disk access, with the overall goal of high throughput. SciTegic has supplied components for cheminformatics data analysis and modeling for some years and more recently has added bioinformatics capabilities.

19.3.2.3.2 Product Line

1. Pipeline Pilot is deployed as a client/server system, with desktop PCs connecting to a central service-oriented architecture deployed on an Apache Web server. The server maintains a repository of components and protocols, in addition to its role as a computational engine. Jobs are spawned to run protocols and provide feedback to the client. These facilities are exposed as standard Web services.
2. SciTegic provides a graphical client program that allows a user to build and edit new protocols and components. A protocol consists of any number of components organized into one or more separate pipelines. Pipelines can branch and merge. Filtering components in the pipeline can route data based on a defined condition. The user can also run protocols from this interface, monitor progress, and view data result with a number of viewer components.
3. A nested set of components can be captured in a subprotocol, which facilitates reuse of useful pipeline networks, helps to simplify workflow designs, and can be used to support iterative operations.

4. A Pipeline Pilot deployment includes a set of standard components for reading, writing, filtering, processing, and viewing generic data. There are horizontal and vertical domain component collections for chemistry, sequence analysis, text analysis, reporting, statistics, and data modeling.
5. Protocols stored in the central repository are also executable through a standard Web browser interface, providing network-wide access to the stored protocols without installing any client software.
6. Pipeline Pilot supports extensibility in two ways.
   a. Due to its Web service architecture, customers can write custom client programs, scripts, or Web applications to run protocols and process results. Software libraries are provided to assist with this task.
   b. The second approach is the user's ability to create new components. A Pipeline Pilot deployment includes a number of integration components that provide facilities to create a new component to capture a simple command line, or by coding up new functionality in Perl or Python scripts or in Java code, using a component API. To utilize network resources, the user can also configure telnet, ftp, or SOAP components. SOAP components can execute in a multithreaded fashion to parallelize the use of remote servers.

      The user can publish any custom components and protocols for broader use by all users of the server, either from a Web browser or from the standard design and execution client.
7. An administrator uses a browser-based Web portal to perform configuration and monitoring tasks on the Pipeline Pilot server from any network location.

19.3.2.3.3 Product Summary

- Data pipelining is designed to support high throughput of large volumes of data from disparate sources.
- The product is deployable with a selection of domain-specific component packages for chemistry and biology in addition to generic statistical and reporting capabilities.
- The Web-services architecture supports the publication of protocols that are then accessible to a variety of clients, including Web browser access for organization-wide deployment.
- Various integration techniques allow the user to create new components to incorporate third-party or in-house scripts and programs.

19.3.2.4 TurboWorx

19.3.2.4.1 Overview

TurboWorx sells workflow software into several vertical markets including life sciences. Their product line targets the bottleneck that arises when an organization has large volumes of distributed information that it cannot process as efficiently and effectively as it would like. The proposed solution to such a situation is a workflow system that facilitates the definition of a sequence of computational applications that

operate on high volumes of data and captures this sequence as a reusable procedure. The software can provide hooks into the necessary computational tools and invoke processing across distributed computational resources.

19.3.2.4.2 Product Line

1. The heart of the TurboWorx software is its TurboWorx Hub (aka SmartGrid) technology. This includes
   - A Component Library that stores information about application programs and workflows in the system. Each reusable program is stored as a component. The program might be a compute-intensive calculation or a data-retrieval operation, for example. Each component includes information on the inputs and outputs of the program, and instructions on running the program.
   - A Data Repository that stores data or references to data. The repository provides input to workflows and manages result data.
   - A Master program that orchestrates the execution of a workflow. Computational components are invoked when their data inputs are ready. The Master can invoke each program on a heterogeneous set of networked Worker computers, allowing for a degree of parallelism to aid workflow performance. Maximal data throughput is achievable when there is a Worker available for each program so that multiple datasets may be processed at the same time, each at a different component in the workflow.

     Components that split the data can leverage this parallelism by breaking the data into smaller data sets that are processed independently.

     To ease bottlenecks created by longer-running components, the Hub can create replications of components for increased parallelism at key points in the workflow.
2. The TurboWorx Builder is the graphical environment where a user can create new components that wrap existing commercial, open-source, or proprietary software tools. A user can create a workflow from new or preexisting components and execute it on the TurboWorx Enterprise runtime environment (see next), monitor progress, and view results.

   The TurboWorx Builder software provides prepackaged components for data access, visualization, and algorithms for its target vertical markets including industry-standard sequence analysis utilities for searching, alignment, and HMM algorithms.

   Extensibility comes in the form of graphical wizards that help the user to create command line, Java, or Jython components. The wizards generate the necessary wrappers to integrate the new components into a TurboWorx workflow.
3. TurboWorx Enterprise is the execution engine for workflows and can be enabled to utilize distributed computational and data resources. During the orchestration of a workflow, the workflow engine uploads component programs to Worker machines, as outlined in the previous Hub description. This approach also provides for a degree of fault tolerance, where failed

component executions can try again with a different Worker. TurboWorx Enterprise provides a Web portal as a lightweight client for the execution of published workflows.

4. The TurboWorx Cluster Manager is a tool to manage and monitor computational resources and data resources within a cluster, with the goal of tuning application configurations and resource allocations to provide optimal throughput.

19.3.2.4.3 Product Summary

- The software includes a generic workflow management system for the distributed execution of tasks across networked computational resources. Various forms of parallelization can be employed to maximize data throughput. These forms include the simultaneous operation of components within the workflow, the distribution of Worker computational machines, the splitting of data sets for independent processing, and the replication of bottleneck components.
- Life-science-specific components can be installed, with wizards to help create custom components.
- A centralized data repository is included for data input to workflows and result storage.
- A shared component and workflow repository stores collaborative work.

19.3.2.5 White Carbon

19.3.2.5.1 Overview

White Carbon offers a software platform named Pathways that targets the development of workflows for discovery procedures in the laboratory (not to be confused with the more recent application of the term to biochemical pathways in the systems biology sense). A user of the software develops and refines a model that captures a best practice and may then deploy that model for the benefit of a wider discovery community. The software also provides a basis for the use of lab equipment to automate appropriate tasks within the workflow. The procedures that the software models generally relate to laboratory activities, but may also include interaction with other software tools or data repositories.

19.3.2.5.2 Product Line

White Carbon characterizes the Pathways software platform as two separate suites:

1. Pathways Laboratory Automation focuses on the development of a workflow to capture and automate best practices for laboratory procedures. To achieve this, the product consists of four distinct tools.
   a. The Workshop Configuration Editor is used to define and organize the procedures that the laboratory supports. This model of lab capabilities details a set of unit processes with inputs and outputs in terms of

physical materials and data content. These unit processes act as the raw material for subsequent steps of workflow creation.

b. The Workbench tool is an environment that allows the user to describe an experiment in functional terms by creating a workflow from the defined unit processes.

c. The Agent Tier is a number of applications, each of which is specific to a particular task. It allows the user to map the logical design of the experiment to actual processes that achieve each logical task. This process may require a human to work with a task-specific UI that the software helps to develop. Alternatively, an agent may operate in a more automated fashion to control laboratory equipment or to perform data-processing operations. The idea is that as the workflow evolves and is validated, some of the manual steps will become automated to run a script or drive equipment. This may involve a different set of skills from the logical design of the workflow itself.

d. The Operations Manager tool actually runs a workflow, by controlling the sequence of operations and by invoking the appropriate agent for each task.

2. The Pathways Services product is used to deploy a validated workflow beyond its initial development environment to a wider set of users and to integrate it into broader processes within an organization. This publication model allows new users to utilize the workflow with custom parameters, without altering the logic or processes of the workflow. The goal is to support projects that span multiple labs in different locations by sharing and orchestrating common methodologies in an organized fashion.

   The Pathways Services software uses a Web-services model for workflow publication, providing flexibility for incorporation into other applications. White Carbon offers a set of baseline Pathways Services applications to meet some common requirements, with a professional services model for tailoring to the specific requirements of an end-user organization.

19.3.2.5.3 Product Summary

- The Pathways user designs a workflow model to capture best practice and standardize data recording in the laboratory environment.
- The software encourages an iterative approach to workflow design to refine a model and increase the proportion of automated tasks.
- It supports a Web-services model for Web browser or application access to a validated workflow by a broader set of users.

19.3.3 Open Source Workflow Tools

The use of open source tools is one way to investigate how you might use a workflow system without having to make any cash investment. They can provide a workable system for an individual researcher or a small group, if you are able to supply the software development and IT skill set (with the necessary time) to configure and extend the system and to handle low-level administration.

This section references the following tools:

Product	URL
Taverna	taverna.sourceforge.net
Biopipe	biopipe.org
Enhydra Shark	shark.objectweb.org
Perl Workflow	www.cpan.org
Apache Agila	incubator.apache.org/projects/agila

19.3.3.1 Taverna

19.3.3.1.1 Overview

Taverna is a design environment and workflow engine offered as an element of the publicly funded myGrid project in the United Kingdom. Its emphasis on workflow solutions for bioinformaticians performing *in silico* experiments has also attracted collaboration from the European Bioinformatics Institute and other UK-based research centers. The program emphasizes the use and orchestration of Web resources and services, allowing users to leverage the significant data sources and computational capabilities that are available to them on the Web without a great investment in in-house facilities [OS1].

19.3.3.1.2 Software Details

1. Scufl (also termed XScufl) is an acronym for the "Simple conceptual unified flow language" and is a nondomain-specific XML representation for a workflow in the Taverna system. A workflow consists of a set of components with links between them. Within a workflow, a link defines the transfer of data between components or defines a control relationship that sets conditions for component execution.
2. The Taverna workbench is a graphical toolset that runs on a client machine to allow the user to construct and edit workflows and submit the resulting Scufl representation as a workflow for execution by an enactment engine. The workbench also provides browsing facilities for data results associated with a workflow instance.
3. The Freefluo enactment engine runs in process with the client for out-of-the-box convenience, but the software is also designed to act as a centralized enactor service if necessary. The enactment engine holds information about specific components that can be invoked by a given installation of that engine.
4. Components can execute tasks on the local machine or on a remote computational resource (known as "services"). Components encode the logic to fulfill their prescribed function, including the details of communications with remote services, if needed. In this way, the workflow description does not need to include information on the mechanics of each task. A software developer may create new components for the system using published Java object models.
5. Taverna comes prepackaged with components to access a number of relevant Web services, such as SoapLab (exposing EMBOSS tools from

EBI), Moby (an international registry of bioinformatics services), and KEGG (Kyoto Encyclopedia of Genes and Genomes).

6. Because of its focus on remote services, Taverna has evolved a fault tolerance subsystem that allows the user to fine-tune the required behavior when a particular service fails for some reason. The user can configure the number of retries, delays, and alternate servers. Multiple threads may be used to make a number of parallel requests to remote service where, for example, the service is supported on a cluster architecture.

19.3.3.1.3 Summary

- Taverna supports the creation of workflows that utilize the large number of databases and computational tools that are publicly available to bioinformatics users on the Internet.
- Java programmers can script new components for custom tasks not available in the prepackaged set of services.

19.3.3.2 Biopipe

19.3.3.2.1 Overview

Biopipe [OS2] is a framework for running bioinformatics workflows and is deployable as an extension to the Bioperl project library [OS4]. It borrows heavily from the Ensembl pipeline project [OS7]. Potential users of the system should be comfortable coding Perl scripts and actually reconfiguring a Perl installation to add the necessary prerequisites. The user also needs to deal with coding raw XML to compose and configure a workflow and setting up the mySQL database that is required for the system to function.

19.3.3.2.2 Software Details

1. A Biopipe protocol represents a series of analyses. Each unit of analysis consists of specifications for input, analysis, and output. The input layer consists of a number of adaptors for various common database formats or for remote fetching from Web sources like GenBank. The role of the input layer is to retrieve data into a common format for a subsequent analysis. The complementary output layer contains adaptors to push the analysis result out to the desired database or format. The analysis layer functions through the action of wrapper Biopipe Perl modules that make standard Bioperl runnable binaries accessible to the Biopipe system. An explicit design goal of Biopipe is to reuse the encapsulations of binary tools, importers, and exporters that Bioperl already includes, with thin wrappers that specify the inputs that the input layer must provide in a workflow context.
2. The XML for a Biopipe protocol supports the specification of simple rules that dictate the sequence of analyses and the data dependencies between them. For example, a rule can specify that all data must be processed by analysis A before starting B. This implies a workflow rather than a pipeline strategy.
3. Each combination of an analysis and a set of input data is managed as a job by the Biopipe job management system. It stores state information for all jobs

in the mySQL database. Its goal is to facilitate the restart of a workflow even after system failure in a compute-intensive environment by tracking the progress of all jobs at all times. The job management system can run jobs sequentially on the local machine or else utilize modules to delegate jobs to a third-party load-balancing system for asynchronous execution. Modules are available for Load Sharing Facility from Platform Computing and Portable Batch System (PBS), which has a free version known as Open PBS.

19.3.3.2.3 Summary

- Biopipe offers a way to add workflow concepts and scalability to scripts that employ Bioperl tools.
- Protocol authoring and system extensibility comes in the form of Perl coding and handcrafting XML.

19.3.3.3 Other Open Source Workflow Tools

The following tools are not specific to the life sciences community but might facilitate the development of a custom workflow solution for a single user or a small group. A simple workflow system offers more flexibility than a basic scripting approach, dependant on the availability of the software and IT skills required to configure and maintain the system.

19.3.3.3.1 Enhydra Shark

One interesting aspect of this software is its adoption of workflow standards from the Workflow Management Coalition (WfMC), namely, XPDL to represent a workflow and a WfMC tool agent API for workflow extensions. The next section of this chapter provides an overview these standards. Enhydra JaWE is a graphical editor for the XPDL workflows. The software is deployable in a J2EE container, as a CORBA object service, or as a Web service.

19.3.3.3.2 Perl Workflow Module

This relatively new addition to the vast Perl module repository might be of interest to a proficient Perl software developer [OS3]. It supports XML definitions for the following workflow elements:

- A *workflow* defines a set of states. Each state defines actions that can be executed against it.
- An *action* invokes a custom Perl object; it defines input data and result data in a state transition.
- A *condition* provides execution control.
- A *validator* invokes a custom Perl object for data validation.

19.3.3.3.3 Agila

This is still an Apache incubation project. It is an open source workflow engine for embedding in Java environments. It defines an XML document format for workflow specification, basic administration, user management, task lists, and notification services. Agila will become an element of the Apache Jakarta product suite.

19.4 STANDARDS

Standards related to workflows tend to focus on one of the following aspects:

1. A common *workflow definition* facilitates the reuse of an individual workflow with different engines. A workflow constructed on one workflow software platform could then execute on a platform from another vendor.
2. Component services included in the workflow could expose a common *application specification*. Integration of arbitrary application tools is an important characteristic of workflow engines. If all application tools could publish the configuration information required by any workflow engine, then a workflow plug-and-play technology would result. This configuration information needs to cover the variability concerned with the invocation of a task on a local or networked system, providing it with parameters and input data, retrieving results, and detecting and handling error situations.
3. Standards related to *interoperability between workflow subsystems* hope to enable the plug and play of new subsystems into a workflow system, such as a new graphical design program or an administration interface. This would only be possible where all components communicate with a common language and compatible technology.

Because there are different aspects to interoperability standards and because there has been standardization interest in this area for over a decade, the result is a plethora of APIs, data formats, and standards bodies targeted at a number of technologies, including C, Java, CORBA, XML, and SOAP. The more recent interest in modeling and standardizing business-to-business processes has made the situation even more confusing, with considerable debate around the overlap between workflow and business processes and whether the pre-existing workflow standards are even appropriate for peer-to-peer business processes [WS6].

19.4.1 Organizations

This section references the following organizations:

Name	AKA	URL
Workflow Management Coalition	WfMC	www.wfmc.org
Business Process Management Initiative	BPMI	www.bpmi.org
Object Management Group	OMG	www.omg.org
Organization for the Advancement of Structured Information Standards	OASIS	www.oasis-open.org

19.4.1.1 Workflow Management Coalition

The venerable institution for workflow standards, the WfMC, has been around since 1993, founded by workflow vendors and users to increase awareness of workflow

systems and to promote standards for interoperability. The basis of this organization's output is the Workflow Reference Model, which defines five discrete interfaces for which interoperability standards appear to be useful [WS4].

19.4.1.2 Business Process Management Initiative

The BPMI organization was founded in 2000 to promote common XML-based standards for Business Process Management.

19.4.1.3 Object Management Group

The OMG developed a workflow management specification in the late 1990s that defined a set of CORBA services [WS7]. Some vendors have extended this for their own use, making a Java version of the CORBA specification, for example.

19.4.1.4 Organization for the Advancement of Structured Information Standards

The OASIS consortium is an umbrella organization for the generation and promotion of many electronic business standards across a number of market sectors.

19.4.2 A Sampling of Workflow-Related Standards

1. XML Process Definition Language (XPDL). This XML specification from the WfMC specifies an interchange format for a workflow process. This allows a workflow defined with one product to execute on the management system of another vendor. It is a graph-structured language where the nodes represent activities within a process, with transitions as the directed links between the activities. It has some block-structured and scoping representation, although it does not handle nested process definitions [WS1].
2. Business Process Modeling Language (BPML). The BPML standard [WS8] is considered by some to be somewhat competitive with XPDL. The WfMC and BPMI have at times sought a convergence of standards (they see the terms workflow and Business Process Management as interchangeable), but that is still an ongoing effort [WS5].
3. Web Services Business Process Execution Language (WSBPEL). In 2002, Microsoft and IBM combined their respective XLANG and Web Services Flow Language as BPEL4WS (or BPEL for short, and more recently WSBPEL) [WS9]. Processes in WSBPEL export and import functionality by using Web service interfaces exclusively. It is now under the auspices of OASIS.
4. Wf-XML. Unlike the XML workflow specifications listed previously, this is a Web services API that defines process automation across heterogeneous implementation environments [WS2]. It has application to a workflow client (such as a design tool or another workflow engine) to communicate with a workflow engine about what protocols are available and to upload and download process definitions. It extends the Asynchronous Service Access Protocol API from OASIS [WS3].

19.5 FUTURE TRENDS AND CHALLENGES

Some observers see an analogy between the workflow software sector today and the early days of database management systems (DBMS) and UI tools. If the DBMS and UI analogies are correct, then standards emerging from consortia or from market leaders will promote large-scale adoption. In turn, business opportunities will emerge from the ability to provide a common workflow platform to application writers and tool vendors. However, despite efforts to develop workflow standards, the picture is still too unclear for commercial vendors in life sciences to feel pressure to fall into line, or to benefit from doing so. In addition, life science vendors and users will not necessarily follow the crowd if they feel they can get better solutions from non-standard-compliant products (in the past, molecular biology has provided an enclave of Macintosh usage in a world of Windows PCs).

Drug-research groups need to process efficiently the large volumes of data available to them. The cost of hardware to run computational facilities is not generally a limiting factor compared to the cost of administration and the cost of software licenses and maintenance. Therefore, a successful workflow system will be able to maximize the return on the investment made in computational systems, by distributing workloads appropriately across available resources, and to achieve this with a minimal IT burden by providing powerful administrative tools and intelligent load balancing between jobs and between parallel tasks within jobs.

Large numbers of cheap processors arranged in grids can be an effective way to achieve massive parallelism and high throughput, provided the workflow system is able to orchestrate tasks appropriately between these processors.

Sophisticated distribution of tasks can also provide higher throughput in another way. In large research organizations, data sources often exist in multiple locations and are stored in multiple formats. An optimal workflow system would invoke agents to perform first-level filtering or analysis tasks in close proximity to the data rather than pulling all data to a central location.

A successful product cannot afford to overlook the usability factor in the desire to achieve these levels of workflow sophistication. In the cost–benefit equation, complexity for end users or administrators adds greatly to the expenses of owning and running a software system, whatever its technical merits.

History shows us that open systems are a key to success as a software platform. In the case of workflow management systems, this means that end users should be able to extend the system to incorporate their favored programs and scripts. In addition, third-party vendors should have facilities to build products on the platform to extend its capabilities and increase its deployment, making it an even more desirable target for other vendors.

19.6 CONCLUSION

When you are considering a workflow management system for your own needs, consider the range of software that you will need to automate, including in-house scripts and utilities. Analyze your data sources and consider how you will successfully plug together applications that have different data-format requirements. The

right solution for you is sufficiently extensible to include your current resources in addition to those that you might deploy in the future.

Scalability should also be a factor in the assessment. Judge how easy it will be to deploy the chosen solution to a larger number of users, some of whom will construct new workflows and others who will only be executing workflows. Web-browser-based solutions may be essential for providing execution access on a cross-site or global basis.

In a fast-changing discipline such as the life sciences, we can anticipate continuing diversity in the application software tools and data sources that researchers use to build software solutions. In addition to competition between larger vendors, new innovative software for researchers will continue to emerge to fill new niches from both commercial and noncommercial sources. Some tools and data will exist on the local machine, while others may be located on the local network or on the Web. The diversity of software that must work together to achieve a larger research goal will drive the growth in usage of workflow solutions, and the necessary flexibility will dictate the requirements for any successful automated workflow system.

REFERENCES

(In addition, please note the Web links within the text for vendor information on the listed software tools.)

Petri Nets

[PN1] Petri, C. A. 1962. *Kommunikation mit Automaten*. Bonn, Germany: Institut für instrumentelle Mathematik.

[PN2] van der Aalst, W. M. P. 1998. The application of Petri Nets to workflow management. *J Circuits Systems Comput* 8:21–66.

[PN3] Rölke, H., and F. Heitmann. Petri Nets World. http://www.informatik.uni-hamburg.de/TGI/PetriNets/

Workflows

[WF1] Altintas, I., C. Berkley, E. Jaeger, M. Jones, B. Ludäscher, and S. Mock. 2004. Kepler: Towards a grid-enabled system for scientific workflows. Workflow in Grid Systems Workshop in GGF10-10th Global Grid Forum, Berlin, Germany, March 2004.

Workflow Patterns

[WP1] van der Aalst, W. M. P., and A. H. M. ter Hofstede. 2002. Workflow patterns: On the expressive power of workflow languages. In *Proceedings of the Fourth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools*.

[WP2] van der Aalst, W. M. P., A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. http://www.workflowpatterns.com/

Data-Mining Tools

[DM1] SPSS Inc. Clementine. http://www.spss.com/clementine/

[DM2] Insightful Corporation. Insightful Miner. http://www.insightful.com/products/iminer/default.asp

Open Source Tools

[OS1] Oinn, T., M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, et al. 2004. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 20:3045–54.

[OS2] Hoon, S., K. Kumar Ratnapu, J. Chia, B. Kumarasamy, X. Juguang, M. Clamp, A. Stabenau, S. Potter, L. Clarke, and E. Stupka. 2003. Biopipe: A flexible framework for protocol-based bioinformatics analysis. *Genome Res* 13:1904–15.

[OS3] Winters, C. 2004. *Workflows in Perl*. Pittsburgh, PA: Optiron Corporation (2004). Available from http://www.cwinters.com/pdf/workflow_lt.pdf

[OS4] BioPerl. BioPerl. http://www.bioperl.org/wiki/Main_Page

[OS5] Shah, S. P., D. Y. M. He, J. N. Sawkins, J. C. Druce, G. Quon, D. Lett, G. X. Y. Zheng, T. Xu, and B. F. F. Ouellette. 2004. Pegasys: Software for executing and integrating analyses of biological sequences. *BMC Bioinformatics* 5:40.

[OS6] IBM alphaWorks. Bioinformatic Workflow Builder Interface. https://secure.alphaworks.ibm.com/tech/biowbi

[OS7] Potter, S. C., L. Clarke, V. Curwen, S. Keenan, E. Mongin, S. M. J. Searle, A. Stabenau, R. Storey, and M. Clamp. 2004. The Ensembl analysis pipeline. *Genome Res* 14:934–41

Standards-Related Publications

[WS1] WfMC. 2002. *Workflow Process Definition Interface—XML Process Definition Language (XPDL)*, Document WFMC-TC-1025 FINAL.

[WS2] WfMC. *Wf-XML 2.0,* Current Draft. http://www.wfmc.org/standards/docs.htm

[WS3] WfMC. 2003. *Wf-XML 2.0/ASAP.* http://www.wfmc.org/standards/docs.htm

[WS4] Hollingsworth, D. 2004. *The Workflow reference model: 10 years on.* WfMC. Available from http://www.wfmc.org/standards/docs/Ref_Model_10_years_on_Hollingsworth.pdf

[WS5] Shapiro, R. 2003. *A comparison of XPDL, BPML and BPEL4WS.* http://xml.coverpages.org/Shapiro-XPDL.pdf

[WS6] Dubray, J-J. *XPDL* (Discusses its suitability for describing business processes in contrast to workflow processes). http://www.ebpml.org/xpdl.htm

[WS7] OMG. 2000. *Workflow management facility specification.* http://www.omg.org/cgi-bin/doc?formal/2000-05-02

[WS8] BPMI. 2002. *BPML 1.0 specification.* http://www.bpmi.org/bpml-spec.htm

[WS9] OASIS. 2005. *Web Services Business Process Execution Language Version 2.0.* http://www.oasis-open.org/committees/documents.php?wg_abbrev=wsbpel

20 Ontologies

Robin A. McEntire
GlaxoSmithKline

Robert Stevens
The University of Manchester

CONTENTS

20.1 INTRODUCTION

This chapter gives an overview of the application of ontologies within the bioinformatics domain, specifically for drug-target identification and validation. We provide a background of the ontology field and its technical underpinnings as well as the current state of the art within the field. We then link the way that biological research, assisted by bioinformatics tools, can be carried out. This underscores the need for

ontologies. Having done this, we provide some case studies pertinent to bioinformatics in general, and target identification in particular.

Molecular biology currently lacks the mathematical support prevalent in disciplines such as physics and chemistry. Biology has Darwin as a scientific grounding, and Darwin described principles with descriptive evidence. In physics, however, we have laws based in mathematics that allow us to predict planetary orbits, behavior of waves and particles, and so on, from first principles. We cannot yet take a protein sequence and use the amino acid residues present to calculate the structure, molecular function, biological role, or location of that protein. This is exemplified at the core of bioinformatics: sequence is related to molecular function and structure. Taking this "law," a biologist can compare the protein sequence to others that are already well characterized. If the uncharacterized sequence is sufficiently similar to a characterized sequence, then it is inferred that the characteristics of one can be transferred to the uncharacterized protein—hence the sequence similarity search. The characterization of single sequences lies at the heart of most bioinformatics, even the new high-throughput techniques that investigate the modes of action of thousands of proteins per experiment and the bioinformatics of drug-target identification.

When performing a sequence similarity search, it is not simply the similarity statistics that determine biological insight into the uncharacterized protein. The bioinformatician uses the knowledge about the proteins already characterized in order to arrive at any insights. Thus it has been said that biology is a knowledge-based discipline [1].

Much of a community's knowledge is contained within its data resources. In a database such as UniProt/Swiss-Prot, the protein sequence data are a relatively small part of the entry. Most of the entry is taken up by "annotation," which can be considered the knowledge component of the database. The knowledge is usually captured as stylized natural language. The same biological knowledge can be represented in many different ways, which leads to the same concept having different terms in each resource and different concepts having the same terms [2,3]. This *semantic heterogeneity* is a perennial problem in integrating bioinformatics resources. Although this style of representation is suitable for human readers, the current representation of the knowledge component is difficult to process by machine.

As well as the knowledge component within resources, biological data are characterized in the following ways:

1. *Large quantity of data:* The genome sequencing projects now mean that data are being produced at increasingly fast rates; a new sequence is deposited in the public genome database EMBL every 10 seconds.[1] Microarray experiments measuring gene expression and other high-throughput techniques now mean that other data are also being produced in vast quantity, at petabytes per year [4].
2. *Complexity of data:* It is difficult to represent most biological data directly in numeric form. The basic data representation and the many relationships

[1] http://ebi.ac.uk/

held by each entity are characteristics of biology's data. For instance, any one protein has a sequence, a function, a process in which it acts; a location, a structure, physical interactions it makes; diseases in which it may be implicated; and many more relationships. Bioinformatics resources need the ability to represent this complex knowledge in a computationally processable form.

3. *Volatility of data:* Once gathered, biological data are not static. As knowledge about biological entities changes and increases, so the knowledge annotations within data resources change.
4. *Heterogeneity of data:* Much biological data are both syntactically and semantically heterogeneous [2,3]. Individual concepts, such as that of a gene, have many different but equally valid interpretations. There is a widespread and deep issue of synonymy and homonymy in the labels used for concepts within biology as well as those used for the names of individuals.
5. *Distribution of data:* Bioinformatics uses over 500 data resources and analysis tools found all over the Internet [5]. They often have Web interfaces through which biologists enter data for analysis, cut and paste results to new Web resources, or explore results through rich annotations with cross-links [2].

This scene leaves both the curators of bioinformatics resources and their users with considerable difficulties. A typical user, as well as a bioinformatics tool builder, is left trying to deal with the following knowledge-based problems to attempt the following tasks:

- Knowing which resources to use
- Discovering instances of those resources
- Knowing how to use each of those resources and how to link their content
- Understanding the content of the resources and interpreting results
- Recording all that occurred during the *in silico* experiment

Taking these steps requires knowledge on the part of the biologists. It is no longer tenable for an individual biologist to acquire and retain this range and complexity of knowledge. This means that bioinformatics practice needs computational support for storing, exploring, representing, and exploiting this knowledge.

20.1.1 What Is an Ontology?

An ontology attempts to capture a community's understanding of a domain as a structured collection of vocabulary terms and definitions [6]. An ontology describes what a community understands about its domain of interest—in this case molecular biology, the interaction of drugs and targets, signal transduction, and bioinformatics. It describes, in a conceptual form, the things that exist in the domain, both concrete and abstract, such as DNA, nucleic acids, signaling, protein, enzyme, alpha-helix, species, function, process, location, disease, and so on. It also describes

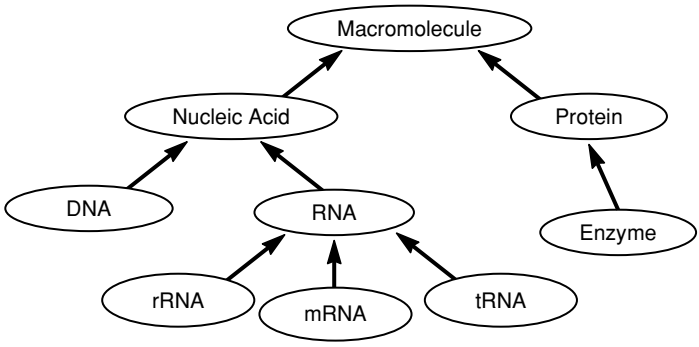


FIGURE 20.1 A naïve ontology of important macromolecules. Ovals represent concepts, which are classes of instances; the arrows represent *is-a* relationships, such that all members of a child class are also members of a parent class.

the relationships between these concepts. For example, an ontology can describe knowledge such as the fact that all messenger RNAs comprise nucleic acids, and that Uracil is a part of RNA. Figure 20.1 shows a simple ontology of some of the basic components of molecular biology.

The discipline of ontology has its origins with Aristotle and in the philosophical domain is the art of describing things that exist in the world. Computer science has taken this term and altered it. In computer science, an ontology is a conceptualization of a domain of interest rather than a description of reality.[2] Concepts are units of thought that refer to things in the world—protein, gene, drug, target [7]. Words are symbols that we use to communicate about things in the world, and, in an ontology, terms are used to label concepts. It is these terms that are used by a community to talk about the domain of interest. If the conceptual model of the world (ontology) and the terms for those concepts (lexicon) can be agreed upon by the community, then ambiguity in communication can be avoided. This shared or common understanding of a domain is one of the primary aims of an ontology. A goal of computer science research into ontologies is to make these conceptualizations of human knowledge processable by computers in a manner that enables inferences to be made about knowledge stored in a computational form.

So, the main components of an ontology are

- The concepts representing entities that exist in the domain. A concept can be either a class that represents a set of instances or a particular instance itself.
- The terms or symbols that label those concepts and allow humans to communicate about those concept or entities in the world.

[2] In philosophy, an ontology is a description of reality; a description of a conceptualization of reality is a description of a description of reality.

- The relationships between those concepts. The principle among these is the *is-a* relationship that describes a parent–child relationship, or class subclass, where the child concept is also a kind of the parent concept. The second major relationship is the *part-of* relationship that describes parts and their wholes [8], such as parts of proteins (active site, alpha-helix, amino acid residue) and their relationship to the whole protein. Other associative relationships are used: causative, nominative, and so on.
- Other statements about the concepts and relationships. In logic systems, for example, it is possible to say that sibling concepts are disjoint; it is not possible for an individual to be a member of both classes. Other statements might include equivalence between classes or that a set of child concepts is a complete covering of the domain.

An ontology may also be defined as a "specification of a conceptualization," a definition from Tom Gruber [9], one of the early pioneers in the ontology field. This definition takes a little unpacking but leads neatly onto a key aspect of ontologies. A conceptualization is how a community thinks about its domain of interest. This forms a conceptual model in terms of the components just given—the specification is how this conceptual model is encoded so that it can be used by humans, but especially by a computer. This is the role of knowledge representation languages (see Section 20.1.2.2).

Although biologists may not have used the term *ontology,* the use of classification and description as a technique for collecting, representing, and using biological knowledge has a long history in the field. For example, the Linnaean classification of species is ubiquitous,[3] and the Enzyme Commission has a classification of enzymes by the reaction that they catalyze [10]. Families of proteins are also classified along axes such as function and structural architecture [11]. Over the years there has been a surge of interest in using ontologies to describe and share biological data reflecting the surge in size, range, and diversity of data and the need to assemble it from a broad constituency of sources.

20.1.2 Ontologies from Knowledge Representation

The ontologies field is a direct descendant of the Knowledge Representation and Reasoning work done within the Artificial Intelligence community over the last several decades. There is not sufficient space in this chapter to provide a complete history of knowledge representation; however, some key points will give context to the current state of ontologies.

Early work in the field of knowledge representation explored a range of representational formalisms, which included graphical representations such as semantic networks (of various styles), conceptual graphs [32], predicate logics (First Order Logic [FOL]), frame-based systems (KL-ONE, CLASSIC, NIKL), and description logics. Each of these knowledge representation formalisms has differing degrees and

[3] http://www.ncbi.nlm.nih.gov/Taxonomy

types of expressivity, computability, and satisfiability, and each has applicability within a particular problem space. For example, FOL is highly expressive in its ability to describe the world; however, there is no guarantee that the language is computable in real time. Of these formalisms, frame-based formalisms description logics have particular relevance to the current work in the ontology field, so we provide a brief description.

Frame-based systems are most like object-oriented systems and provide a high degree of structure. They are centered on the idea of a frame, or a class, where each frame represents a set of instances of that frame or class. Each frame has associated slots that represent attributes of the frame. Slots are filled by specific values or by other frames. Slots may be of various kinds. So, for example, frames may have an associated *is-a* slot, mentioned earlier, which is used to create a taxonomy within the frame system. The *part-of* slot, another highly important slot or relationship, may also be represented in a frame-based system. The frame-based representation system is the most widely used of the KR formalisms and has been used extensively within the life sciences community, for example, in EcoCyc, using SRI's Ocelot frame system.

Description logics (DLs) are built in a very different way from frame-based systems. So, rather than building a taxonomy explicitly, a DL provides reasoning capabilities in the form of a classifier that will build the ontology from smaller conceptual units. These smaller conceptual units provide sufficient description, a concept with one or more associated relationships, so that the DL reasoner can classify the new concept in the proper place within the ontology. DLs have been of significant interest in the last several years and provide the underlying representation for the Web Ontology Language, which we discuss in more detail later.

In the late 1990s the Bio-Ontology Consortium delivered a recommendation for a language for the exchange of bioinformatics information [33]. This recommendation resulted in the development of the XML-based Ontology-Exchange Language (XOL)[4] at SRI. XOL provided a frame-based representational system in an XML-based syntax. This language broke ground with respect to developing XML-based representation languages; however, XOL was quickly overtaken by OIL (Ontology Inference Language) from University of Manchester.

Over the next several years a DARPA-sponsored project, the DARPA Agent Markup Language (DAML), incorporated OIL in its efforts. The DAML effort fostered a significant collaboration among many of the top researchers in the ontology field, which resulted in the creation of a new description logic language, DAML+OIL, closely, though not completely, modeled after the OIL language. DAML+OIL has since become the framework for the W3C's effort to create a language for the semantic Web. This culminated in February 2004 with W3C approval of the Web Ontology Language (OWL). OWL[5] is a knowledge representation and transfer language for building ontologies that delivers vocabularies with a formal semantics. OWL has three increasingly expressive sublanguages: OWL Lite, OWL DL, and OWL Full. It also

[4] http://www.ai.sri.com/pkarp/xol/

[5] http://www.w3.org/TR/2003/PR-owl-features-20031215/

has a rules language under development for capturing knowledge that cannot be contained in an ontology [23]. The OWL sublanguages are described here:

- OWL Lite provides the capability to describe simple taxonomic classifications and lacks the expressivity to make rich descriptions of classes of instances. It provides a migration path for thesauri and simple taxonomies, such as those commonly seen in bio-ontologies such as GO.
- OWL DL is an expressive language that is a fragment of FOL. This means that it is amenable to machine reasoning. Ontologies described in OWL DL can be checked for logical consistency and subsumption hierarchies (the lattice of *is-a* links) inferred from the descriptions of classes formed from the links made between classes [20,24,25]. This form of OWL is the focus of this section.
- OWL Full is more expressive than OWL DL but is not amenable to machine reasoning.

In OWL, classes describe sets of instances or individuals that belong together because they have properties in common (Gene is the class of all genes—TripA, ADH, Bra1, etc.). Classes may be arranged into subsumption hierarchies using the subclass relationship. By stating that Gene is a subclass of Nucleic Acid Sequence Region, we are stating that all instances of Gene are also instances of Nucleic Acid Sequence Region. Properties can be used to state relationships between classes and individuals or from individuals to data values. For example, we can say that instances of the class Gene express instances of the class Protein. In OWL DL, we can place restrictions on how properties form relationships that make what that relationship means explicit. We can use *existential quantification* (has some values from) and say that all genes express some protein (but might express something else), or we can use *universal quantification* (has only values from) and say that all instances of the class Protein express only Protein (but might not do so). We can form more complex expressions by saying that all instances of the class Gene express some (Protein or RNA) and express only (Protein or RNA).

OWL DL is much more expressive than this fragment indicates. For instance, we can describe properties of properties such as transitivity, range, and domain constraints and form hierarchies of properties. It is also possible to say that the instances of two siblings do not overlap by using a disjointness axiom—the classes Protein and NucleicAcid are both kinds of macromolecule, but being disjoint it is not possible for an individual protein to also be a nucleic acid.

We can also describe a class as being *partial* or *complete*. When the properties of a class are partial, those properties are a *necessary* condition of class membership. That is, an instance must have those properties. Describing a class's properties as complete means that an instance having those properties is *sufficient* to recognize it as a member of a class. So, labeling our class Gene as complete would mean that any instance that expresses both Protein and RNA would have to be a Gene.

Figure 20.2 shows the OWL abstract syntax and English paraphrasing for a selection of classes from the Gene Ontology (GO) biological process ontology. The formal semantics of OWL, partially described earlier, mean that it is possible for a

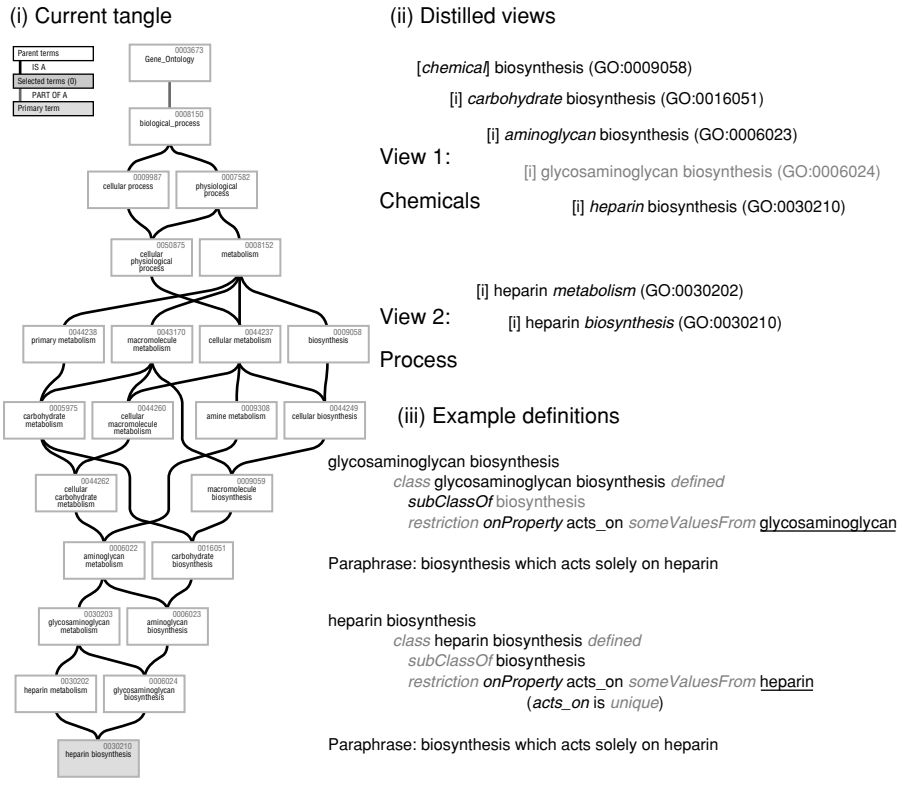


FIGURE 20.2 A portion of the Biological Process section of the Gene Ontology with associated OWL abstract syntax.

machine to process this representation. The precise descriptions also reduce the ambiguity when read by humans: we know exactly what the statements mean.

20.1.3 The Value of Ontologies

Ontologies provide a clear framework for representing classes of objects and attributes of objects in a domain of interest. This representational structure can be used in a number of ways to provide solutions to very pragmatic issues and problems in the area of drug-target identification and validation. Ontologies are used in a wide range of biology application scenarios [12,13]:

- As a mechanism for defining database schema or knowledge bases. Examples include RiboWeb [14,15], EcoCyc [15], and PharmGKB [16]. In this case, the ontology provides a structure for recording data, and the expressivity of the language used (often Frames, but now also OWL) offers a description that has high fidelity to the domain of interest.

- As a common vocabulary for describing, sharing, linking, classifying, querying, and indexing database annotation. This is currently the most popular use of ontologies in bioinformatics, and among many examples we can count are the GO [4], Microarray Gene Expression Data (MGED) [17], and those vocabularies that originate from the medical community, such as UMLS.[6]
- As a means of interoperating between multiple resources. A number of forms appear, for example, indexing across databases by shared vocabularies of their content (domain maps in BIRN [18], interdatabase navigation in Amigo using the GO[7]); a global ontology as a virtual schema over a federation of databases and applications (TAMBIS [19]); and a description of bioinformatics services inputs, outputs, and purposes used to classify and find appropriate resources, such as bioinformatics Web Services, and to control the workflows linking them together (myGrid [20]). As a scaffold for intelligent search over databases (e.g., TAMBIS [19]) or classifying results. For example, when searching databases for "mitochondrial double stranded DNA binding proteins," all and only those proteins, as well as those kind of proteins, will be found, as the exact terms for searching can be used. Queries can be refined by following relationships within the ontologies, in particular the taxonomic relationships. Similarly, Fridman Noy and Hafner [21] used an ontology of experimental design in molecular biology to describe and generate forms to query a repository of papers containing experimental methods. The extensions to a typical frame-based representation allow them to accurately describe the transformations that take place and the complexes that form within an experiment, and then make queries about those features.
- As a method for understanding database annotation and technical literature. The ontologies are designed to support natural language processing that link domain knowledge and linguistic structures.
- As a community reference, where the ontology is neutrally authored in a single language and converted into different forms for use in multiple target systems. Generally, ontologies have been developed to serve one of the previous categories of use and then adopted by others for new uses. For example, the GO was developed solely for database annotation but is now used for all the purposes just outlined.

20.2 THE CURRENT ENVIRONMENT FOR ONTOLOGIES

Significant work has been performed over the last five to six years in developing ontologies for the life sciences. In this section we describe some of the major work

[6] http://www.nlm.nih.gov/research/umls/
[7] http://www.godatabase.org

that has been done, the ontologies that have been developed, and the organizations that support and perform the work in the field.

20.2.1 Current Life Sciences Ontologies

Foremost among the ontologies in the life sciences are the GO and MGED. However, there are a number of efforts underway that have varying degrees of support. The GO initiative has grown to include the Open Biological Ontologies (OBO)[8] effort, which develops ontologies in many additional biological domains. There is also work on the development of ontologies for tissues, anatomy, pathways, and other subdomains of the field. The GO [4] is the central bio-ontology effort. It grew from a recognition that in the postgenomic era biologists would increasingly wish to make cross-species comparisons and queries. As already mentioned, the species-centric terminologies used by each community are a barrier to such investigations. By describing a common understanding of the principle attributes of gene products, the GO is attempting a *de facto* integration of genomic databases. GO has three orthogonal attributes describing molecular function, biological processes, and cellular components. A fragment of the molecular function ontology is shown in figure 20.3.

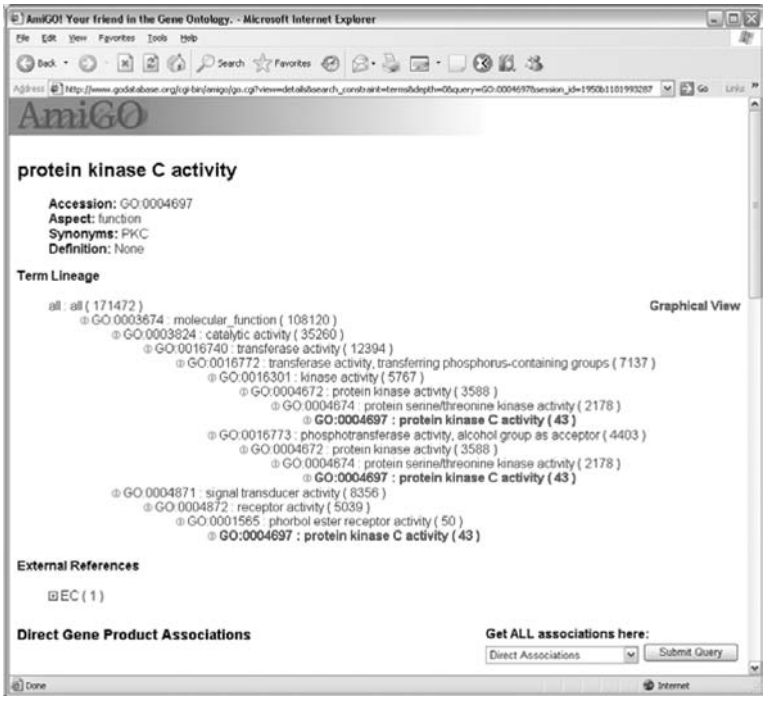


FIGURE 20.3 The Gene Ontology DAG for protein kinase C (GO:0004697) taken from the Amigo browser at http://www.godatabase.org/

[8] http://obo.sourceforge.net

The three GO ontologies now contain some 18,000 terms. They are represented as a directed acyclic graph or DAG. The nodes of this graph represent the biological concepts and the labels of these nodes provide the terminology. The arcs of this graph represent therelationships *is-a* and *part-of* between the biological concepts.

As discussed earlier, the primary use of GO is for the annotation of gene products within databases. Each of the 16 species databases use the same terms to represent the same attributes within their respective databases. As well as the species databases, community-wide resources such as UniProt/Swiss-Prot and INTERPRO also annotate their entries with GO terms. Retrieval tools such as AmiGo[9] and SRS can then use the ontology itself or the terms it provides to query gene products with increased recall and precision. As the use of such resources lies at the center of the bioinformatics of drug-target identification, improved querying through the use of ontologies can be seen to be a benefit to the field.

The use of the GO has now moved well beyond its original purpose of retrieval. It is widely used in analyzing microarray data by placing clusters of up or down regulated genes into categories according to GO terms. It is easy to see that if a collection of genes in a disease condition are all related to a particular metabolic process this process lies at the heart of the disease itself. This kind of ontological analysis has obvious consequences in the hunt for drug targets.

There is much use of ontologies in the domain of microarray analysis itself. As the technology developed, those holding repositories of microarray data were determined to avoid some of the problems found in the natural language representation of knowledge in older resources. Both expressive schema and the vocabularies for populating them have been developed to aid querying and analysis of microarray data. The MGED Society (see next) have played a guiding role in developing standards for storing and describing these data. The MGED ontology describes [17] the biological samples, treatments, and experimental conditions in microarray experiments. The MGED ontology is formulated so that it is easy to refer to other ontologies that provide vocabulary necessary for descriptions—anatomies, species, disease, and so forth. An example of such complementary work is eVoke™,[10] which offers an Expression Ontology Toolkit that links experimental data with standardized terms that provide insight into phenotype. The Evoke ontologies provide vocabularies for describing biological samples and are now a part of the Minimal Information for the Annotation of Microarray Experiments.

The growth of ontologies within bioinformatics has been rapid over the past few years. It is futile within a book chapter such as this to attempt a comprehensive coverage of bio-ontologies. The references and URLs within the chapter are a good start in providing coverage of those ontologies, and use organization Web sites (see next) give access to further ontology efforts.

---

[9] http://www.godatabase.org/cgi-bin/amigo/go.cgi
[10] http://www.egenetics.com/evoke.html

20.2.2 Ontology Tools

There are now a number of commercial and open source tools for the development, maintenance, merging, and visualization of ontologies. A comprehensive survey of ontology tools was conducted in July 2004 by XML.com.[11] We do not attempt to reproduce that survey here, but it is worth mentioning some tools and organizations that are notable within the ontology field.

Undoubtedly, the best-known ontology authoring tool is Protégé, from the Stanford Medical Informatics group at Stanford University. This tool has been in use for more than 10 years, and one could argue that it has been around longer, since it is an outgrowth of the Knowledge Representation initiatives at Stanford University, which have been in existence since the 1970s.[12] Protégé has a large and active user community with a number of commercial and academic projects and is open source, so it can be downloaded at no cost and is easily installed. There are several mailing lists to which developers and new users may subscribe, and the tool comes with example ontologies to help new users get started on a project. Protégé has a core engine, which is extended with plug-ins. There are a number of plug-ins, and the user community is actively involved with creating new ones. One of the most recent plug-ins is for OWL. Although the plug-in does not yet allow the user to code all of OWL's features using just the Protégé tool, it does provide support for the basics, including classes, slots, facets, and instances.

Another widely used tool in the life sciences community is DAG-Edit. DAG-Edit was developed at the Berkeley Drosophila Genome Project to be used as a part of the knowledge acquisition effort of the Gene Ontology Consortium. DAG-Edit is limited in its representational capabilities and is primarily used to represent simple *is-a* and *part-of* relations; however, it is simple to use and has been a very effective tool in the development of the very successful GO effort and others. The success of DAG-Edit is in its ability to rapidly assimilate new content for the GO. There are a number of other efforts that have leveraged the GO content and have been successful in transcribing, semiautomatically, the GO content into a more formal ontology, such as DAML-OIL [20].

In addition to these two tools, there are well over 50 other ontology authoring tools, of varying degrees of sophistication and ease of use, from universities and research organizations around the world. A large number of commercial organizations offer commercial ontology authoring tools. Some of the more prominent of these commercial tools are LinkFactory Workbench (Language and Computing), Integrated Ontology Development Environment (Ontology Works), OntoEdit (Ontoprise GmbH), OpenCyc Knowledge Server (Cycorp, Inc.), and Construct (Network Inference). Although none of these tools, commercial or academic, yet has complete support for all of the representational capabilities of ontology languages such as OWL or DAML+OIL, or the reasoning capabilities of description logics, there are a number of sophisticated tools and a lively marketplace for developing the next breed of tools.

---

[11] http://www.xml.com/pub/a/2004/07/14/onto.html?page=1

[12] Stanford Knowledge Systems Laboratory, http://www-ksl.stanford.edu/

20.2.3 Organizations Promoting Ontology Development

In this section we discuss the organizations that are promoting the development and the adoption of ontologies in the life sciences field. GO and MGED are the most prominent examples, but other groups are active in the field as well, including the Bio-Ontologies Consortium, the Bio-Pathways Consortium, and others.

As mentioned earlier, the Gene Ontology Consortium brings together 16 model organism databases (at the time of this writing) to develop the GO. As each new organization joins, they commit to using the GO terms to describe the functionality of gene products in their databases. As a consequence, each new group drives the development of the GO to make available terms needed for that species.

The Sequence Ontology[13] is also part of the Gene Ontology Consortium. It is a grouping of genome annotation centers, including WormBase, the Berkeley Drosophila Genome Project, FlyBase, the Mouse Genome Informatics group, and the Sanger Institute. The aim is to provide a shared vocabulary for the features described on nucleotide and protein sequences. It is intended to range from the basic features seen on a sequence, through interpretations such as "pseudogene" to mutations.

The Open Bio-Ontologies effort acts as an umbrella under which bio-ontologies may be developed and disseminated. OBO has a set of principles that govern inclusion:

- Submitted ontologies must be open but cannot be altered and redistributed under the same name.
- Use cannot be restricted—ontologies are for sharing.
- A common representation should be used, either the form accepted by DAG-Edit or the OWL.
- Ontologies should be nonoverlapping.
- Namespace identifiers should be used so that the source of any entity within an ontology can be identified.
- All terms should have a textual definition to prevent ambiguity in interpretation by human readers.

OBO offers access to a wide range of ontologies. Several ontologies of anatomy for various species are prominent. These specific ontologies are of particular interest to the community, as they can be used to identify the biological source of material in experiments (e.g.,this microarray experiment used mouse lung, etc.). In addition to anatomies, there are several ontologies of development within species. Finally, there are a growing number of phenotype ontologies available, including traits, disease, and behavior.

BioPAX[14] is a consortium of pathway databases that aims to develop an exchange language for biological pathways (BioPAX). Pathways include the metabolic, regulatory, and signal pathways. The BioPAX initiative aims to overcome the heterogeneity

[13] http://song.sourceforge.net/so.shtml
[14] http://www.biopax.org

of formats and conceptualizations in the many pathway databases. Initially, BioPAX has used an ontology, written in OWL, to develop a schema for describing the entities and their attributes to be exchanged. Further levels of BioPAX will be developed to provide ontologies for vocabularies to facilitate descriptions of pathway data.

The MGED Society,[15] which we discussed earlier, has a similar goal to that of BioPAX in that it aims to develop both schema and the vocabularies that fill attributes of that schema for the description of microarray experiments. The MGED has been in existence longer than BioPAX and has a developed ontology for providing vocabularies for the description of biological samples, their treatments, and the experimental conditions pertaining during hybridizations [17]. This ontology is now moving away from the world of model organisms to include toxicology and environmental genomics experiments. As proteomics experimentation develops, there are efforts being made to share descriptions of experiments across broader communities.

While GO and MGED are the most prominent and mature life sciences ontologies, they are not the only efforts within the life sciences community to develop open source ontologies. There are also active efforts to develop ontologies in a number of other areas, including the following:

- Foundational Model of Anatomy [13,14] is an ontology of human anatomy.
- Tissue Ontology [18] offers a controlled vocabulary for describing tissues across a range of contributing databases.
- Chemical Entities of Biological Interest[16] is a dictionary of small molecular entities that are either products of nature or synthetic products used to intervene in the processes of living organisms.

These organizations have common goals. There is a recognition that this is a community effort and that inclusion will make an ontology work [22]. Ontologies are meant for sharing and for capturing a community's knowledge of a domain, so community inclusion is crucial. A multiplicity of ontologies will not solve the inherent problems of bioinformatics resources described earlier.

20.3 LEVERAGING ONTOLOGIES FOR DRUG-TARGET IDENTIFICATION AND VALIDATION

The target identification and validation process in the pharmaceutical industry is characterized by a set of steps, or processes, with key decision points to determine whether a target should be moved forward in the drug-discovery process. Each step of the overall target identification and validation process is characterized by a set of experiments, data, and information gathering and execution of best practice that enlists the expertise of a number of scientific teams. Each team brings expertise in a particular scientific domain, and the results from each of the teams must be brought

[15] http://www.mged.org
[16] http://www.ebi.ac.uk/chebi/

together to ensure that a complete picture of the molecule being examined is produced. Identification and validation of targets relies heavily on the accumulated evidence, both scientific and business related, that has been generated inside the pharmaceutical company and/or the biotech as well as the information available from databases, external information stores, and documents (both internal, such as monthly reports, and external, such as full-text patents).

Over the last 10 years, there has been a huge leap in the ability of research organizations to generate new data and information, and the rate of accumulating new data and information is accelerating. High-throughput screening (HTS), combinatorial chemistry, genomics, transcriptomics, proteomics, and metabolomics offer opportunities for more efficient and effective identification of targets and therefore assist in the overall process of developing new drugs. These new technologies and knowledge domains have become a standard part of the early-discovery process. Systems biology is also of great interest, as it offers the promise of bringing together an aggregated picture of all the science relevant to identification and validation. Furthermore, just as we have seen these new domains of knowledge (e.g., the X-omics) find a place in the early drug-discovery process, the expectation is that the target identification and validation process will change as other areas of scientific exploration and new tools become available. Therefore, it is important that the early-discovery process be flexible to efficiently and effectively integrate new information and knowledge into the current knowledge base of the organization as well as existing processes.

As has been pointed out, scientists are not able to absorb all of the information that results from high-throughput testing and the resulting volume of information. We explore the way in which ontologies can provide critical assistance to the current challenges in the early-discovery process. Specifically, we show that ontologies can provide benefit in the following areas:

- Consistent representation of scientific data and information
- Organization and integration of information
- Visualization of data and information in context to provide meaningful abstractions
- Identification and extraction of key information in text
- Workflow and best practice, or knowledge management

Next, we explore each of these areas in more detail and look at the role that ontologies play.

20.3.1 Representing the Scientific Data and Information

As stated earlier, well-defined ontologies provide the community with a set of common, shared understandings of the data and information elements that are used by scientists and also offer, in those cases where a more formal ontology language is used to represent the ontology content, a structured and computable resource for managing the data and information. Immediate benefits of shared, common understandings of

terms in the field are that this enhances communication among scientists and provides clarity to a general understanding of the field.

It is clear that a great deal of laboratory work is performed during the target identification and validation process. Sophisticated technology supports the use of HTS and microarray assays, which results in volumes of data. The scientist's favorite, and most ubiquitous, tool for viewing and analyzing experimental results is the spreadsheet, typically Excel. While an Excel spreadsheet is a very able tool, with much to recommend it, it has serious shortcomings when used to view large volumes of data. It is not hard to imagine that scientists' eyes grow weary after long periods of examining their results in spreadsheets. The more serious problem is that it is all too easy to miss information, sometimes key information, when it is buried in such a form.

It is no longer feasible for scientists to use only Excel spreadsheets to analyze their results. Results must be captured and aggregated in a systematic manner to allow scientists to get a broad view of the information and knowledge that derives from their experiments. The large quantities of data can be used as support for the higher-level view of the information, and the scientist should be allowed to drill down into this supporting data from the high-level view. This will be an iterative process of viewing information, reviewing and manipulating the supporting data, and analyzing the results. The interim and final analyses created by the scientists are also key pieces of information. These annotations provide a layer of knowledge above the raw data and information, and these annotations are very valuable to the organization, which should capture and maintain them. However, this analysis is typically captured in unstructured, free-text files in various file formats, using various document styles and kept on local hard drives or embedded within different document systems or e-mail. Clearly this is an informatics problem, and, as with most problems in informatics, it has many solutions. Frequently, informaticians (Bio-, Chem-, and others) approach this problem by coding one-off solutions to these problems, using such valuable, and available, tools as Perl, Python, Oracle DB, and others. However, these tools do not address the problem of data/information heterogeneity. For example, to reproduce and interpret DNA microarray assay results and to effectively predict protein sequence and structure from DNA sequence, it is necessary to know the semantics of the underlying data elements. If the semantics are captured in procedural code, then the semantics are a single-point solution that provide little or no help to solutions developed subsequently.

It quickly becomes clear that the most efficient method for defining the semantics and leveraging these semantics for later work is to define the semantics in a separate layer, an ontology that provides a commonly held, shared understanding of the data and information. With the advent of XML and XML-based technologies it becomes simple and easy to develop a Document Type Definition, a Resource Description Framework (RDF), or an RDF Schema (RFDS) representation of the data. Bioinformaticians have found that this is a quick and relatively painless solution to their problem. However, the ease with which these representations can be built can present another problem, namely, the propogation of many heterogeneous representations of data, which are perhaps well suited to an individual experiment or set of experiments by a small team but do not provide a structure that can be leveraged for other

work. "Quick and dirty" representations, even when done by skilled bioinformaticians, can often result in representations that are flawed or incomplete. The work of creating well-structured representations of the data generated by a pharmaceutical company or a biotech is therefore a nontrivial undertaking. It requires expertise; a broad understanding of the forms in which the information may be expressed (as opposed to an understanding of the immediate needs of one or a small number of scientific teams); and, perhaps most important of all, significant resources, in that people must dedicate their time to completing the task. The life sciences industry is highly competitive, and most work performed in an organization is done under considerable time pressure. Critical deliverables and near-term milestones are not conducive to dedicating project team resources to produce a computable layer of abstraction above the data elements being used in a project. Any resources dedicated to building this kind of abstraction introduce the risk of slipping deadlines. It is a difficult task to convince management that one of their projects should provide additional resources to develop these abstractions. However, as we have seen, the benefit to the business as a whole is significant. Overall, the business will be able to reduce the time and resources spent in coding access to and aggregation of datasets, and it will reduce the time it takes to produce meaningful results to the leadership of project teams and to key decision makers within the business.

It is clear that there is a business case in the life sciences industry for the use of ontologies for datasets produced in the organization; however, it is not clear that a pharmaceutical company or biotech has the goal, or the proper skill set, to do all of this work by itself. The better approach is to build bio-ontologies in a consortial fashion that allows the life sciences industry and academic organizations to drive the requirements for ontology design but to share the burden of the work in creating these ontologies. The MGED Society[17] and the Gene Ontology Consortium that we mentioned previously are good examples of the value of consortial work.

The research process for target identification and validation generates a significant number of results for expression experiments. Another benefit that ontologies can provide is the ability to view experimental data, such as expression results, in context. Providing experimental data in context gives the scientist an immediate understanding of the meaning of this data and, one can argue, is the appropriate way for scientists to view their data and information, from a general context with the ability to drill down into the supporting data. Ontologies, and the GO in particular, provide such a context. HTS results, which use Affymetric Identifiers (AffyIDs), can be mapped into one or more of the three GO taxonomies in a straightforward manner because of the mapping from AffyIDs to GO Identifiers (GOIDs), and is a good example of the uptake of GO within the life sciences.[18] This mapping allows experimental results to be mapped to each of the three GO ontologies, which, with the proper visualization of the GO, allows scientists to immediately see their significant results mapped into

[17] MGED Society, http://www.mged.org/, http://mged.sourceforge.net/

[18] GOIDs are also referenced in a number of other bioinformatics databases, such as Reactome Biological Process, PFAM, TIGR, and Enzyme Classification numbers. A full list of the associations and their mappings can be found at http://www.geneontology.org/GO.indices.html/

a biological context. More hits in a particular area of each of the ontologies provide evidence about the underlying biological processes that are affected.

Each of these ontologies offers the same opportunity to provide benefit to the life sciences industry. It is easily seen that even this small, unrepresentative set of ontologies covers many major areas of interest in target discovery: anatomy, tissue, pathways, proteins, and chemicals. Bioinformatics resources are at the point where much of the knowledge they hold might be open to computational processes across organizations.

20.3.2 Integrating Information

In addition to visualizing experimental results in a biological context, it is increasingly important to be able to see the same results in multiple contexts. For example, the scientist is interested in seeing results in the context of a metabolic pathway to gain an understanding of the pathway, or pathways, that a particular target will affect. However, the scientist will also need to understand these results in the context of biological process, cellular component, and molecular function. In addition, it is valuable to understand which tissues are involved. The task of mapping results into each of these contexts is often slow and requires a significant amount of resource within the organization. This delays analysis work and ties up people in the organization who could be dedicated to other tasks. Clearly, there is an incentive to reduce the time and resources it takes to connect from one context to another. Once again, ontologies can provide assistance with this problem. The organization will leverage a number of ontologies, where each provides a valuable context. The key is to allow the scientist to bridge from one context, or ontology, to another in a seamless manner. The GO is again a leading example of how ontologies are being built to handle this problem. The GO, specifically its identifiers (GOIDs), are cross-linked to many other ontologies and vocabularies, which facilitates bridging from one ontology to another. GOIDs are now included in such bioinformatics databases and identifiers as UniProt/SWISS-PROT, LocusLink, Affymetrix Identifiers, Enzyme Classification numbers, and others.

A simple mapping from one identifier to another is quite powerful and can go a long way toward integrating ontologies and contexts so that research scientists can see their experimental results in one context and then move to another. However, this simple method also has its pitfalls. Mapping from one system's identifier to another is not always a straight one-to-one mapping. The mapping is often many-to-one, and perhaps one-to-many. In addition, mappings do not always exist from one ontology's set of identifiers to another's. It is often the case that mappings from Ontology A to Ontology B must be made through one or two intermediate ontologies. Finally, these mappings are not always complete or correct. Although still highly useful, it is easy to see that this method of integration relies on mappings at a relatively superficial level and that any errors in the mappings are compounded when additional, intermediate ontology mappings are used. There are many efforts within the ontology community to bridge across, or merge, ontologies. In addition, there

are a number of efforts that are specifically geared toward bridging the gaps between life science ontologies [18] or providing focused entry points to multiple ontologies representing the same domain.[19]

In addition to the work conducted in the ontology community, there is also related work being performed in commercial IT organizations, such as IBM, GeneticXchange, and BioWisdom, in the field of intelligent information integration. These companies have developed integration products that allow federated querying of multiple life sciences databases, both structured and unstructured. Each organization uses varied techniques in their current generation of products; however, many of the organizations show interest in leveraging the work being conducted by the ontology community.

20.3.3 Workflow and Sharing Information within a Virtual Organization

Typically, project teams are formed in each phase of the drug-development process. The team is usually led by a champion, usually someone from the biological research community, who believes in the potential of the compound to become a drug. This champion acts as project lead and forms a team to conduct research necessary to move the molecule forward in the drug-discovery process, or, in this case, the target identification and validation process. In a typical industry organization, this project team is a matrix of several expert members. In other words, in the organizational structure, there are scientific teams focused on a particular area of scientific expertise (e.g., chemistry, toxicology, genetics, biology). However, the researchers in these scientific teams serve as team members of multiple projects that focus on the identification and validation of a specific substance to determine whether it acts as a drug. The scientist who leads the target identification and validation team often creates her or his team from members of a number of the scientifically focused teams. Over and above the science, then, there are issues that must be resolved to allow this matrix of team members to operate effectively and to respond quickly to results and information as they become available. Teams must be focused around both team and project boundaries, and there is a significant effort needed to coordinate the many teams that compose a project. Each team member will need to balance the obligations to the team in which he or she is a direct report with the responsibilities to the (possibly multiple) teams for which that member provides scientific expertise. This problem is particularly difficult for the leader of the team that owns the substance under investigation, as this person must organize and coordinate the efforts of many people over whom this lead has no direct control. A typical team might be composed of

- A biologist subteam to investigate the potency and selectivity of the substance
- A subteam to perform screening of potentials

[19] SOFG Anatomy Entry List (SAEL), at http://www.sofg.org/sael/

- A chemist subteam to analyze structure activity relationships and tractability
- A pharmacokinetics group
- Information analysts to review the competitive landscape and to provide the team with relevant scientific documents and information from other groups within the organization that may have a bearing on the project team's effort
- Patent experts to judge the current landscape and leverage any patentable outcomes from the team
- Subteams in transcriptomics, proteomics, and other scientific disciplines (other X-omics)

The team lead is usually a scientist, well versed in her or his area of expertise but not necessarily knowledgeable or experienced with the formal process for bringing the drug through the drug-discovery process. This process often involves well-defined decision points that must be supported by formal or semiformal proposals and presentations to committees, all of which can or should be guided by best practice for scientific investigation.

Clearly, information integration work, supported by ontologies, will be a key criterion for success in this area, but even with substantial scientific evidence, a team will not succeed if there is not sufficient coordination of team members and of the information that they find and/or produce. The problem here is to abstract and formalize the workflow, including decision points and timelines for deliverables for each member of the matrix, as well as coordination and dissemination of the information produced by each member of the team. This complex task is usually left to the team leader to solve; however, it is an area in which ontologies can provide a level of abstraction and coordination that will help bind the team together. Work in this area is focused primarily in the Semantic Web or Semantic Grid space, which we discuss separately in this chapter.

The ability to manage and leverage the combined knowledge of a team or organization is typically referred to as Knowledge Management. Therefore, Knowledge Management is closely aligned with workflow and information sharing. Capturing knowledge and best practices, and subsequently finding that knowledge so that it may be leveraged in appropriate situations, is a key objective for any knowledge-based industry. This is a critical problem in the pharmaceutical and biotech industries, and it will become increasingly so as project teams become larger and more complex as they work to assemble the information needed to move targets through the drug-discovery pipeline. Management of these "virtual" teams is a complex problem in itself.

Though there are a number of groups that are working in this area, one of the most prominent research efforts is the myGrid project.[20] myGrid is a UK e-Science project funded by the EPSRC involving five UK universities, the European Bioinformatics Institute, and many industrial collaborators [19]. This collaborative effort

[20] myGrid Project, http://www.mygrid.org.uk/

has been at the center of many other life science initiatives and has built an infrastructure currently being leveraged by a number of other funded research projects. Although the infrastructure is not a commercial product, several commercial organizations, large and small, have been involved in the project. myGrid uses workflows to create *in silico* experiments in bioinformatics. In composing services to form these workflows, ontological descriptions can be used to facilitate discovery, overcoming problems of inadequate keyword-based searches or reliance on community knowledge. When myGrid workflows run, large amounts of provenance data are generated that provide a context of use. These can include a knowledge-level provenance, provided by ontology terms, which describes results, as entities; relationships between those entities; and, for instance, the topics under investigation. myGrid provides a good example of using ontologies to facilitate interoperation of data and its subsequent exploitation.

20.3.4 Text Mining and Ontologies

A key source for scientific and competitive information is free-text documents. Structured information, such as that found in relational databases, is highly important to the target identification and validation process; however, most information supporting this endeavor is locked inside scientific journal articles, Medline abstracts, patents, competitive pipeline reports, news articles, organizations' internal documents, and many other sources. Scientific journal articles are read by discovery scientists themselves; however, typically scientists will read a subset of all relevant articles, choosing articles from the journals they consider to be of highest quality or to which the organization subscribes. Scientists are limited by the time they can devote to this process, and often a significant portion of their time is spent searching for articles of interest. Other full-text sources, as previously mentioned, are usually reviewed and/or read by a small group of students hired for this purpose or are done by professionals, either internal departments dedicated to mining the competitive landscape or by external organizations that are paid a substantial fee to perform an analysis. Again, money and resources are precious, so only a subset of all the documents that may be of interest can be reviewed. Text mining offers a huge benefit to organizations in that it may reduce the time taken to search for documents of interest. It can also be leveraged to extract from documents the information directly relevant to the current interests of the scientists and the organization.

Text mining is described in detail in a separate chapter of this book (see chap. 6), so we do not discuss the technology further. Here we discuss how ontologies may benefit text-mining systems. We can view ontologies as providing assistance to text mining in two distinct ways: through the use of vocabularies that may be used in text-mining systems and as a formal structure that can be combined with text mining.

In the first case, as chapter 6 points out, ontologies can be an important source of vocabularies, or terminologies, for text mining. Ontologies, specifically in the form of a thesaurus, are able to supply terminology that has been organized into a formal system. Simple word lists can be useful to text mining; however, they are typically *ad hoc,* so are often incomplete, are sometimes incorrect, and rarely include synonymous terms. An ontology provides the following:

- Synonymous terms so that the text-mining system can identify a more complete set of terms that represent entities of interest, such as genes, proteins, and chemical names. This is a significant benefit, as there is substantial variation in the names given to entities and concepts in the life sciences domain.
- A set of terms that is the result of a community effort, as described earlier in this chapter. The advantage that ontologies provide to a community in providing a common language to which everyone in that community may refer is directly translated as a benefit to text-mining systems and enhances the completeness of its results.
- Relationships in addition to the basic *is-a* and *part-of* relationships. These relationships are described in a formal manner in the ontology and provide guidance to text-mining systems (either through manual creation of those relationships in the text-mining system or through an automatic or semi-automatic method) for identifying those same relationships in full-text documents.

As the text-mining chapter points out, ontologies are not always useful for providing vocabulary for text-mining systems. For example, the GO, which we described as one of the most successful of the ontologies in the life sciences, is not useful for the terminology needs of a text-mining system for two reasons, namely, GO is high level so the terminology is general and not useful for identifying specific entities in text such as gene or protein names, and the GO is not a complete thesaurus. So, even though GO provides some synonymous and homonymous terms, the purpose of GO is not to be a complete thesaurus. Other ontologies have similar limitations, though some ontologies, such as UMLS, can be useful. Ontologies can also be made more useful through formal incorporation of appropriate word lists.

Ontologies can also benefit text mining by providing a formal structure to be used by the text-mining system, as chapter 6 recognizes. Specifically, the formal structure of an ontology can provide text mining with the following:

- Formal organization into categories so that words are well defined with respect to the concepts they represent. A thesaurus will categorize terms within an appropriate taxonomic structure. For example, a thesaurus will define terms that are kinases and others that are proteases, all of which are defined within the category of enzymes, which is again categorized within the concept proteins. This separation allows text-mining systems to be very targeted in regard to the terminologies included in the system.
- Taxonomies into which documents may be categorized or classified. Chapter 6 discusses the methods used by text-mining systems to perform this task; it is clear that there are significant benefits to the scientist in having a large set of documents categorized into a well-defined taxonomic structure. It is easy to see that the amount of time scientists spend searching for documents of interest can be significantly reduced if the documents have been organized into meaningful subsets.

- A browsable structure of concepts and relationships into which information extraction results may be deposited. Since a thesaurus can provide terminology associated with concepts within an ontology, the results of text-mining information extraction can be systematically used to enhance an ontology with *facts*. Ontology browsers and visualization tools that sit on top of ontologies can be used to allow the scientist or information analyst to browse a large corpus of documents in a highly structured fashion. The obvious benefit is that this can surface specific information locked in full-text documents, allowing scientists to spend less time gathering needed information and enabling them to find information that they would not otherwise access, due to resource constraints.

In this case, the GO is highly useful as it supplies an organization of biological function, molecular process, and cellular location as a context for the results of the text-mining process.

20.4 FUTURE WORK

20.4.1 Ontologies and Text Mining

There is significant overlap between the work being done in the fields of ontologies and text mining. We have discussed what ontologies currently bring to the text-mining field. As we look out another five years, there is reason to believe that these two disciplines will become much more integrated.

The text-mining field takes a very rich, but highly unstructured, representation of information and knowledge (i.e., natural language) and attempts to impose a structure over it. This structure can take a number of different forms, as the text-mining chapter points out. Text mining can cluster documents with respect to their similarity of content, categorize documents into a predefined taxonomic structure, and extract entities and relationships from documents and place them into a separate, persistent, structured form. The ontology field approaches the problem from the other side, in that ontologists build models to represent concepts and relationships in the real world (e.g., which proteins are expressed for a particular gene) and attempt to populate those concepts and relations with facts. The representational capabilities of ontologies are poor compared to the richness and subtlety that natural language allows, and clearly reasoning systems are far from the capabilities of human beings. However, there is work being done in the ontologies field to give ontologies richer expressiveness. As discussed earlier, ontologies typically use *is-a* and *part-of* as their primary relationships. However, these relationships do not represent the detail that we find in the real world, and they set limits on the utility of ontologies in their ability to integrate with other ontologies and to support sophisticated reasoning. There is currently significant work in the ontologies field that attempts to provide a richer representation for relationships. This work will provide ontologies that can distinguish between, for example, *part-of* relationships in which an entity is a part of another entity but distinct by itself (the kidney is a part of the human body) and a *part-of* relationship in which the part is the same as the whole (where a part of an

apple pie is still apple pie). There is also work that allows ontologies to distinguish between various layers of entities, to see the levels of granularity in objects. For example, human anatomy is one level of granularity, tissues another, and proteins/enzymes yet another. As reasoning systems become more ubiquitous and a more intrinsic part of ontologies, these granularity distinctions will become critical.

As the representational structure within ontologies becomes richer, we can expect this richness to inform text-mining techniques. One can imagine a highly sophisticated knowledge base containing a thesaurus within it, which can be used to produce matching criteria for natural language expressions. If the ontology, or knowledge base, contains a large lexicon, including synonyms, homonyms, and the like, and includes fine-grained expressions of relationships, this material comes closer to approximating some of the structure of language itself.

As ontologies move toward the richness of natural language, we can envision systems that interface with these knowledge bases in a language that is very close to natural language but perhaps a bit less *natural* and more stylized.

20.4.2 Ontology Standards

As ontologies become incorporated into commercial systems, there will be a strong push to provide standards for ontology representation. The standards will come in the form of efforts to agree on a language for the communication of (and sometimes representation of) ontologies and efforts to agree on content. We see that the W3C has produced the major effort in the former case in the development of the OWL language(s). As to content, we next describe some of the organizations involved in ontology development and cite many ontology efforts throughout this chapter.

20.4.3 Ontologies and Reasoning Systems

Where does all this expressivity bring us? Obviously, we have precise descriptions of our domain of interest. The formality also brings support for machine reasoning. Some of the most advanced work in the ontologies field uses Description Logics as the formal representational scheme for the ontologies. Description Logics provide reasoning capabilities, classification, and subsumption specifically, which can be leveraged to validate the information (terms, taxonomic relationships, and other relationships) within a taxonomy; it can also be used to arrange new information into appropriate areas within an ontology.

The descriptions in figure 20.3 of the transport functions in GO can be submitted to a reasoner such as FaCT [34] or Racer [35]. The OWL is mapped down to its underlying logical form, and the reasoner attempts to infer whether all the statements in the ontology are satisfiable. As a consequence, logical inconsistencies can be found. One of the most powerful effects of this is the possibility of inferring the subsumption relationships implied by the descriptions. Taking the descriptions in figure 20.3, we can see by inspection of the descriptions that X and Y are children of Z and that the description of P implies that it is a child of both X and Y. This power has been used in migrating the GO from its DAG representation, which is built by hand and consequently prone to structural errors [26,27]. Patterns in the

lexical phrases representing the terms in GO [20,28] can be used to map GO terms to a property-based representation in OWL. After submission to a reasoner, the OWL descriptions imply that changes should be made to the lattice. In an early experiment on the metabolism region of the GO, some 250 classes were processed in this way, and 22 changes were made. Seventeen of these were directly accepted by the GO editors, but the remaining suggestions prompted more significant changes in modeling than the reasoner's change itself [27]. Further unpublished experiments have revealed a similar pattern.

The power of a representation such as OWL comes from its formal semantics. The formal semantics allow it to be used to make precise, explicit descriptions of the domain of interest. This provides unambiguous representations for humans but also permits machine processing. In addition, one of the real attributes of the OWL style is that simple, handcrafted ontologies can be described in OWL Lite and migrated to the more powerful, expressive OWL DL for machine processing. We examine the wider implications of this machine processing in the next section.

20.4.4 Semantic Web

As we have observed, biologists are major users of the Web for accessing tools and acquiring information. We have also observed that bioinformatics and biology are knowledge-based disciplines. Yet the Web, with its reliance on query by navigation and human understandable natural language in an unstructured form, is not really amenable to computational analysis. The growing amounts of biological information are only manageable at all using search tools such as Google. Any user of a Bioinformatics database, however, knows the unreliability of the keyword search in terms of recall and precision, and this is generalizable to the whole Web. In addition, such searches are sensitive to lexical variants—a well-known problem of semantic heterogeneity in Bioinformatics resources. These are the problems that have driven the use of ontologies within Bioinformatics, and there have been parallel developments within the Web community.

The Semantic Web is envisioned as the next generation for the Web [29]. At present, the Web is a human-understandable collection of data and services. In his vision for the Web, Tim Berners-Lee sees ontologies being used to mark up the semantic content on the Web so that machines or agents can work more autonomously upon those data and services. The consequence will be that instead of "searching the Web," users or their agents will be able to query the Web using semantic descriptions provided by ontologies that are actually attached to parts of pages rather than to the whole page. This process will enable pages to be processed to optimize presentation. In addition, inference over semantic descriptions will provide flexibility in querying, such as specialization or generalization that is not available in keyword searches.

We have seen that markup of resources with terms from ontologies can improve recall and precision; this could also apply to the Web in general. We have also discussed how we can use machine reasoning to make inferences over data using such markup. OWL is envisioned to be the semantic driving force of the Semantic Web. OWL, however, is only one of a range of technologies supported by the W3C

that will support this Semantic Web. These technologies provide a series of layers, moving from syntactic descriptions of document structure using XML to rich semantic description of content using OWL ontologies.

- XML provides a syntax for document structure but offers no semantic description of content.
- XML Schema is a language for restricting the structure of XML documents.
- RDF[21] is a data model for describing facts about *resources* and their relationships.
- RDFS is a vocabulary for describing classes and relationships in an ontologically compatible manner.
- OWL (see earlier) is a more expressive ontology language than RDFS and can be used for building terminologies that can then be represented in RDFS.

These layers of technology form the building blocks for well-structured, semantically described resources on the Web and other application contexts. XML and XML Schema are themselves used to form the syntax for the languages in higher layers. RDF provides a simple, graph-based model for describing resources. A statement in RDF is a triple, comprising a subject, a predicate or verb, and an object. The subjects and objects form the nodes of a graph, and the properties/predicates/verbs form the arcs. So a triple forming the RDF statement "Gene expresses Gene Product" has the subject Gene, the object Gene Product, and the predicate expresses.

RDF itself can be used to create vocabularies, and the most prominent of these is RDFS. RDFS allows descriptions of classes, subclasses, and arbitrary relationships between classes. This means that ontologies described using OWL can also be represented in RDFS and that tools built to support OWL will also support RDFS.

Each element of an RDF triple is a resource with a Universal Resource Identifier (URI) that can point to a resource item itself, a string, or a term from a vocabulary. So an RDF triple could point to a protein sequence with a URI and name that protein via a "has Name" predicate, with an object resource as another URI that points to a name. The same protein resource might also have a predicate pointing to a resource holding a GO term or a database entry describing a particular feature of that protein. Obviously, a URI can also point to another RDF triple. In this way, a large, complex graph of facts or assertions can be created. In addition, the URI mechanism means that more than one autonomously created RDF graph can be linked by predicates. When this happens, graphs become aggregated to form collections of descriptions of distributed resources. It is thus possible to capture increasingly large amounts of semantically described data that are machine processable. The semantics of the RDF language mean that inferences can be made over RDF statements, and the use of controlled terminologies, developed from ontologies, means that rich, expressive queries can be used.

---

[21] http://www.w3.org/TR/rdf-primer/

There is already much activity in producing Semantic Web technology and applications built on those technologies.[22] These include RDF stores, RDF query languages and tools, RDF editors, RDF annotation plug-ins for browsers, and RDF visualization tools. These technologies can be brought together to form Semantic Web-style applications [30].

Biology, with its large Web presence and growing use of ontologies, is already well placed to create a Semantic Web for life sciences. These Semantic Web technologies are already being used within bioinformatics to provide computational access to the knowledge content of bioinformatics resources and data analyses. October 2004 saw the first meeting of the Semantic Web for Life Sciences[23] hosted by the W3C.

20.5 SUMMARY

Significant progress has been made in the last six years in the creation and adoption of ontologies in the life sciences. There now exist two very prominent efforts in the field: the Gene Ontology Consortium and the MGED Society. These two organizations have successfully defined a number of key ontologies in the life sciences and are extending their reach to ontologies in other subdomains within the life sciences. There is strong evidence of the adoption of these ontologies within the community: we have seen GOIDs used by a number of prominent bioinformatics databases, and many of the key conferences in the bioinformatics domain are including tracks on life science ontologies.

In addition to the adoption of ontologies within the life sciences community, there is also significant work going on in the computer science field to develop richer and more sophisticated ontologies. The use of an ontology language for the semantic web, OWL, by the W3C is a significant step in making ontology languages a key building block of IT infrastructure components. We have seen that there is also work being done to identify the limits of current ontology languages and representational schemes and to address those limitations.

While progress has been substantial, there are still challenges for the broad adoption of biological ontologies in the life sciences community. There are some primary obstacles that remain:

1. Adoption of a common language for the representation and exchange of life sciences ontologies
2. Creation of standard ontologies that represent the various key domains of knowledge within the bioinformatics field
3. Adoption of life sciences ontologies by commercial organizations such as the pharmaceutical industry, biotechs, and IT companies producing commercial software systems in support of bioinformatics and life sciences

[22] http://www.ilrt.bris.ac.uk/discovery/rdf/resources/
[23] http://www.w3.org/2004/07/swls-ws.html

With the recent adoption by the W3C of the OWL languages, the field is well on its way to adopting a common, standard language for representing and exchanging life science ontologies. As a W3C recommendation, OWL has strong support from the Semantic Web community. OWL also has good prospects for continued development, as the working group that supports it is very active and comprises some of the top researchers in the field. In addition, as discussed earlier, OWL is a language that will form the basis of much of the Semantic Web community and so will likely be used actively.

With respect to standards for ontologies, it is unrealistic to expect that there will be a single standard ontology for every domain within the life sciences. However, even convergence on a small number of ontologies within each community is a positive step forward and shows great progress in the field. The expectation is that leading ontologies such as GO, MGED, and others will dominate their field and will be adopted by the user community as a useful tool.

The adoption of ontologies by the commercial community is not guaranteed; however, we have already seen that the GO has been successfully incorporated into a number of the key bioinformatics databases and has been adopted by tool vendors, including Spotfire. In addition, most of the key pharmaceutical companies are now leveraging ontologies in their internal bioinformatics efforts. Perhaps the strongest indicator of the incorporation of ontologies into the mainstream of Information Technology and life sciences is the report by the market research firm Gartner, which identified ontologies and taxonomies as one of the leading IT technologies, putting ontologies as third in its list of the top 10 technologies in 2005.

It is heartening to note that bio-ontologies are having an impact within the life sciences community. Serious work is being performed, with good results. Active research underway in the computer science field is planting seeds for the next generation. We can expect bio-ontologies to grow significantly in the years to come.

REFERENCES

1. Baker, P. G., C. A. Goble, S. Bechhofer, N. W. Paton, R. Stevens, and A. Brass. 1999. An ontology for bioinformatics applications. *Bioinformatics* 15:510–20.
2. Karp, P. 1995. A strategy for database interoperation. *J Comput Biol* 2:573–86.
3. Davidson, S. B., C. Overton, and P. Buneman. 1995. Challenges in integrating biological data sources. *J Comput Biol* 2:557–72.
4. The Gene Ontology Consortium. 2000. Gene ontology: Tool for the unification of biology. *Nat Genet* 25:25–9.
5. Discala, C., X. Benigni, E. Barillot, and G. Vaysseix. 2000. DBcat: A catalog of 500 biological databases. *Nucleic Acids Res* 28:8–9.
6. Uschold, M., and M. Gruninger. 1996. Ontologies: Principles, methods and applications. *Knowledge Engineer Rev* 1:93–155.
7. Ogden, C. K., and I. A. Richards. 1946. *The meaning of meaning*. New York: Harcourt, Brace and World.
8. Winston, M., R. Chaffin, and D. Herrmann. 1987. A taxonomy of part–whole relations. *Cogn Sci* 11:417–44.

9. Gruber, T. R. 1993. Towards principles for the design of ontologies used for knowledge sharing. *Proceedings of the International Workshop on Formal Ontology,* Padova, Italy.
10. Webb, E. C. 1984. *Enzyme nomenclature 1984: Recommendations of the nomenclature committee of the international union of biochemistry on the nomenclature and classification of enzyme-catalyzed reactions*. Orlando, FL: Academic Press.
11. Hadley, C., and D. T. Jones. 1999. A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. *Structure* 7:1099–12.
12. Stevens, R., C. A. Goble, and S. Bechhofer. 2000. Ontology-based knowledge representation for bioinformatics. *Brief Bioinformatics* 1:398–416.
13. Stevens, R., C. Wroe, P. W. Lord, and C. A. Goble. 2004. Ontologies in bioinformatics. In *Handbook on ontologies*, ed. S. Staab and R. Studer, 635–58. New York: Springer.
14. Altman, R., M. Bada, X. J. Chai, M. W. Carillo, R. O. Chen, and N. F. Abernethy. 1999. RiboWeb: An ontology-based system for collaborative molecular biology. *IEEE Intell Sys* 14 (5):68–76.
15. Karp, P. D., M. Riley, M. Saier, I. T. Paulsen, S. M. Paley, and A. Pellegrini-Toole. 2000. The EcoCyc and MetaCyc databases. *Nucleic Acids Res* 28:56–9.
16. Klein, T. E., J. T. Chang, M. K. Cho, K. L. Easton, R. Fergerson, M. Hewett, Z. Lin, et al. 2001. Integrating genotype and phenotype information: An overview of the PharmGKB project. *Pharmacogenomics J* 1:167–70.
17. Stoeckert, C. J., and H. Parkinson. 2003. The MGED ontology: A framework for describing functional genomics experiments. *Compar Funct Genomics* 4:127–32.
18. Ludäscher, B., A. Gupta, and M. E. Martone. 2001. Model-based mediation with domain maps. In *17th International Conference on Data Engineering (ICDE),* Heidelberg, Germany.
19. Goble, C. A., R. Stevens, G. Ng, S. Bechhofer, N. W. Paton, P. G. Baker, M. Peim, and A. Brass. 2001. Transparent access to multiple bioinformatics information sources. *IBM Systems Journal,* Special Issue 40:532–52.
20. Wroe, C., R. Stevens, C. Goble, A. Roberts, and M. Greenwood. 2003. A suite of DAML+OIL ontologies to describe bioinformatics Web services and data. *Int J Coop Informat Sys* 12:597–624.
21. Noy, N. F., and C. D. Hafner. 1998. Representing scientific experiments: Implications for ontology design and knowledge sharing. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, 615–22. Cambridge, MA: AAAI Press/MIT Press.
22. Bada, M., R. Stevens, C. Goble, Y. Gil, M. Ashburner, J. A. Blake, J. M. Cherry, M. Harris, and S. Lewis. 2004. A short study on the success of the Gene Ontology. *Web Semantics Science, Services and Agents on the World Wide Web* 1:235–40.
23. Antoniou, G., and F. V. Harmelen. 2004. *A semantic web primer.* Cambridge, MA: MIT Press.
24. Stevens, R., C. Goble, I, Horrocks, and S. Bechhofer. 2002. Building a bioinformatics ontology using OIL. *IEEE Trans Inf Technol Biomed* 6:135–41.
25. Stevens, R., C. Goble, I. Horrocks, and S. Bechhofer. 2002. OILing the way to machine understandable bioinformatics resources. *IEEE Trans Inf Technol Biomed* 6:129–34.
26. Rogers, J. E., W. D. Solomon, A. Rector, and P. Zanstra. 1997. From rubrics to dissections to GRAIL to classifications. In *Medical Informatics Europe (MIE97),* 241–5. Thesalonika, Greece: IOS Press.

27. Wroe, C. J., R. Stevens, C. A. Goble, and M. Ashburner. 2003. A methodology to migrate the Gene Ontology to a Description Logic environment using DAML+OIL. In *Proceedings of the 8th Pacific Symposium on Biocomputing,* 624–36.
28. Ogren, P.V., K. B. Cohen, G. Acquaah-Mensah, J. Eberlein, and L. Hunter. 2004. The compositional structure of Gene Ontology terms. In *Proceedings of the Pacific Symposium on Biocomputing 2004,* 214–25.
29. Berners-Lee, T. 1999. *Weaving the Web.* New York: HarperCollins.
30. Stevens, R., H. J. Tipney, C. J. Wroe, T. M. Oinn, M. Senger, P. W. Lord, C. A. Goble, A. Brass, and M. Tassabehji. 2004. Exploring Williams-Beuren Syndrome Using myGrid. *Bioinformatics* 20:303–10.
31. Borgida, A., R. J. Brachman, D. L. McGuinness, and L. A. Resnick. 1989. CLASSIC: A Structural data model for objects. In *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*, 59–67.
32. Sowa, J. F. 1992. Conceptual graphs summary. In *Conceptual structures: Current research and practice*, ed. P. Eklund, T. Nagle, J. Nagle, and L. Gerholz, 3–52. New York: Ellis Horwood.
33. McEntire, R., P. Karp, N. Abernethy, F. Olken, R. E. Kent, M. DeJongh, P. Tarczy-Hornoch, et al. 2000. An evaluation of ontology exchange languages for bioinformatics. In *Proceedings of the Eight International Conference on Intelligent Systems for Molecular Biology*, 239–50.
34. Horrocks, I. 1998. Using an expressive Description Logic: FaCT or fiction? In *Proceedings of KR–98*, ed. A. G. Cohn, L. Schubert, and S. C. Shapiro, 636–47. San Francisco: Morgan Kaufmann.
35. Haarslev, V., and R. Möller. 1999. RACE system description. In *Proceedings of the International Workshop on Description Logics 99*, ed. P. Lambrix, A. Borgida, M. Lenzerini, R. Moller, and P. Patel-Schneider, 130–2. Linkoping, Sweden: CEUR.