COURSE INTRODUCTION

In the present era, bioinformatics has proven its pivotal role in the understanding and analyzing the information generated from biological experiments. This course covers provides a conceptual knowledge to the students for better understanding of the basics of bioinformatics, its aims, objectives and applications in the biological sciences. The entire course is organized into following three blocks:

Block 1 covers bioinformatics and biological databases

Block 2 deals the phylogeny and sequence databases

Block 3 describes protein database, simulation and drug designing

Rajarshi Tandon Open University, Prayagraj

Bioinformatics

Course Design Committee

Prof. K.N. Singh Uttar Pradesh Rajarshi Tandon Open University, Prayagraj	**Vice Chancellor**
Dr. (Prof.) Ashutosh Gupta, School of Science, UPRTOU, Prayagraj	**Chairman**
Prof. Abhay Kumar Pandey Department of Biochemistry University of Allahabad, Prayagraj	**Member**
Dr. Uma Rani Agarwal **Associate Professor** Department of Zoology CMP Degree College, Prayagraj	**Member**
Dr. Ravindra Pratap Singh Academic Consultant (Biochemistry) School of Science, UPRTOU, Prayagraj	**Member**
Dr. Dharmveer Singh Academic Consultant (Biochemistry) School of Science, UPRTOU, Prayagraj	**Member/Secretary**

Course Preparation Committee

Dr. Ravi Deval — **Author** — **Block-1& 2** (Unit: 1-4)
Asst. Professor
Department of biotechnology,
Invertis University, Bareilly.

Dr. Sachin Kumar — **Author** — **Block-3 (**Unit: 5&6**)**
Asst. Professor
Department of biotechnology,
Invertis University, Bareilly.

Dr. A.P Verma — **Editor** — **All Blocks and Units**
Associate Professor,
Department of Botany
Bappa Sri Narain Vocational
Post Graduate College,
Lucknow

Dr. Dharmveer Singh
(Course Coordinator)
School of Sciences, UPRTOU, Prayagraj

Block- I

Bioinformatics and Biological databases

UNIT -1

Introduction to bioinformatics

UNIT-2

Biological databases

Introduction

This is the first block on bioinformatics and biological databases consists of following two units.

Unit 1: This unit reveals the concepts on the usage of database management system, fundamentals of internet, browsers and their types and search engines etc. This will help the students to use and manage the internet as a tool for information access.

Unit 2: describes the concepts of databases, biological databases and their types. After completing this unit students will gain a fundamental knowledge about role and uses of biological sequences, sequence alignment tools, alignment viewers, and alignment matrices.

Unit 1: Bioinformatics and Biological databases

Structure

1.1. Introduction

Objectives

1.2. About bioinformatics

1.3. History

1.4. Aims

1.5. Approaches

1.6. Scopes of bioinformatics

1.7. Sequence analysis

1.7.1. Genome annotation

1.7.2. Computational evolutionary biology

1.8. Open-source bioinformatics software

1.9. Web services in bioinformatics

1.10. The Online Bioinformatics Resources Collection (OBRC)

1.11. Internet

1.11.1. World Wide Web (WWW)

1.11.2. URL - Uniform Resource Locator

1.11.3. Hyper text markup language (HTML)

1.11.4. Netscape

1.11.5. Internet Explorer (IE)

1.11.6. Google

1.11.6.1. Mozilla

1.11.6.2. Google Chrome

1.11.6.3. PubMed

1.11.7. HTTP Hyper text transfer protocol (HTTP)

1.12. Database Management System

1.13. Summary

1.14. Terminal questions

1.15. Suggested readings

1.1. Introduction

This unit describes the introduction, history, aims, approaches and scopes of bioinformatics. Open source softwares are discussed and their usages are illustrated. Introduction of internet is discussed briefly. The chapter summarizes the concepts on search engines such as google and PubMed, method of using a search engines and their usage etc. This chapter illustrates the topics such as World Wide Web (WWW), HTML and URL. The chapter outlined the browsers such as Internet explorer and Netscape. Database management system and their types are well illustrated along with examples.

Objectives

- To learn the basics of bioinformatics
- To gain a knowledge about aims, scope and objectives of bioinformatics
- To Understand the usage of open source bioinformatics softwares
- To learn about internet and its working
- To achieve a knowledge on search engines and their usage
- To understand the concept of database management system and its types

1.2. About bioinformatics

Bioinformatics is an interdisciplinary field that develops and improves on methods for storing, retrieving, organizing and analyzing biological data. A major activity in bioinformatics is to develop databases and software tools to generate useful biological knowledge through data analysis.

Bioinformatics uses many areas of computer science, mathematics and engineering to process biological data. Complex machines are used to read biological data at a faster rate. Databases and information systems are used to store and organize biological data. Analyzing biological data may involve algorithms in artificial intelligence, software computing, data mining, image

processing, and simulation. The algorithms in turn depend on theoretical foundations such as discrete mathematics, control theory, system theory, information theory, and statistics. Commonly used programming languages and software tools in the bioinformatics field include Java, C#, XML, Perl, C, C++, Python, R, SQL, CUDA, MATLAB.

1.3. History

Building on the recognition of the importance of information transmission, accumulation and processing in biological systems, in 1970 **Paulien Hogeweg** coined the term "Bioinformatics" to refer to the study of information processes in biotic systems. This definition placed bioinformatics as a field parallel to biophysics (the study of physical processes in biological systems) or biochemistry (the study of chemical processes in biological systems). Examples of relevant biological information processes studied in the early days of bioinformatics are the formation of complex social interaction structures by simple behavioral rules, and the information accumulation and maintenance in models of prebiotic evolution.

One early contributor to bioinformatics was **Elvin A. Kabat**, who pioneered biological sequence analysis in 1970 with his comprehensive volumes of antibody sequences released with Tai Te Wu between 1980 and 1991. Another significant pioneer in the field was **Margaret Oakley Dayhoff**, who has been hailed by **David Lipman**, director of the National Center for Biotechnology Information (NCBI), as the "mother and father of bioinformatics". This centre is part of United States national library of medicine (NLM), a branch of national institutes of health (NIH). The NCBI is located in Bethesda, Maryland and was refunded in 1998 through legislation sponsored by senator Claude Pepper.

At the beginning of the "genomic revolution", the term bioinformatics was re-discovered to refer to the creation and maintenance of a database to store biological information such as nucleotide sequences and amino acid sequences. Development of this type of database involved not only design issues but the development of complex interfaces whereby researchers could access existing data as well as submit new or revised data.

1.4. Aims

In order to study how normal cellular activities are altered in different disease states, the biological data must be combined to form a comprehensive picture of these activities. Therefore, the field of bioinformatics has evolved such that the most pressing task now involves the analysis

and interpretation of various types of data. This includes nucleotide and amino acid sequences, protein domains, and protein structures, etc. The actual process of analyzing and interpreting data is referred to as computational biology. Important sub-disciplines within bioinformatics and computational biology include:

- The development and implementation of tools that enable efficient access to, use and management of, various types of information.
- The development of new algorithms (mathematical formulas) and statistics with which to assess relationships among members of large data sets. For example, methods to locate a gene within a sequence, predict protein structure and/or function, and cluster protein sequences into families of related sequences.

The primary goal of bioinformatics is to increase the understanding of biological processes. What sets it apart from other approaches, however, is its focus on developing and applying computationally intensive techniques to achieve this goal. Examples include: pattern recognition, data mining, machine learning algorithms, and visualization. Major research efforts in the field include sequence alignment, gene finding, genome assembly (genome structure study of pathogen like COVID-19) drug design, drug discovery, protein structure alignment, protein structure prediction, prediction of gene expression ands protein–protein interactions, genome-wide association studies, and the modeling of evolution.

Bioinformatics now entails the creation and advancement of databases, algorithms, computational and statistical techniques, and theory to solve formal and practical problems arising from the management and analysis of biological data.

Over the past few decades rapid developments in genomic and other molecular research technologies and developments in information technologies have combined to produce a tremendous amount of information related to molecular biology. Bioinformatics is the name given to these mathematical and computing approaches used to glean understanding of biological processes.

1.5. Approaches

Common activities in bioinformatics include mapping and analyzing DNA finger printing and protein sequences, aligning different DNA and protein sequences to compare them, and creating and viewing 3-D models of protein structures.

There are two fundamental ways of modelling a biological system (e.g., living cell) namely "static and dynamic" both coming under bioinformatics approaches.

Static

Sequences – Proteins, nucleic acids and peptides

Interaction data among the above entities including microarray data and networks of proteins & metabolites.

Dynamic

Structures – Proteins, nucleic acids, ligands (including metabolites and drugs) and peptides (structures studied with bioinformatics tools are not considered static anymore and their dynamics is often the core of the structural studies)

1.6. Scopes of bioinformatics

Bioinformatics has become an important part of many areas of biology. In experimental molecular biology, bioinformatics techniques such as image and signal processing allow extraction of useful results from large amounts of raw data. In the field of genetics and genomics, it aids in sequencing and annotating genomes and their observed mutations. It plays a role in the textual mining of biological literature and the development of biological and gene ontologies to organize and query biological data. It plays a role in the analysis of gene and protein expression and regulation. Bioinformatics tools aid in the comparison of genetic and genomic data and more generally in the understanding of evolutionary aspects of molecular biology. At a more integrative level, it helps analyze and catalogue the biological pathways and networks that are an important part of systems biology. In structural biology, it aids in the simulation and modeling of DNA, RNA, and protein structures as well as molecular interactions. It helps in criminal investigation by use of DNA finger printing.

1.7. Sequence analysis

Since the phage Φ-X174 was sequenced in 1977, the DNA sequences of thousands of organisms began to be decoded and stored in databases. This sequence information is analyzed to determine genes that encode polypeptides (proteins), RNA genes, regulatory sequences, structural motifs, and repetitive sequences. A comparison of genes within a species or between different species can show similarities between protein functions, or relations between species (the use of molecular systematics to construct phylogenetic trees). With the growing amount of data, it has become impractical to analyze DNA sequences manually. Today, computer programs such as BLAST are used daily to search sequences from more than 260 000 organisms, containing over 190 billion nucleotides. These programs can compensate for mutations (exchanged, deleted or inserted bases) in the DNA sequence, to identify sequences that are related, but not identical. A variant of this sequence alignment is used in the sequencing process itself. The so-called shotgun sequencing technique (which was used, for example, by The Institute for Genomic Research (TIGR) to sequence the first bacterial genome, *Haemophilus influenzae*) does not produce entire chromosomes. TGIR is a non-profit research institute located in Rockvill, Maryland, united states, founded in 1992. Instead it generates the sequences of many thousands of small DNA fragments (ranging from 35 to 900 nucleotides long, depending on the sequencing technology). The ends of these fragments overlap and, when aligned properly by a genome assembly program, can be used to reconstruct the complete genome. Shotgun sequencing yields sequence data quickly, but the task of assembling the fragments can be quite complicated for larger genomes. For a genome as large as the human genome, it may take many days of CPU time on large-memory, multiprocessor computers to assemble the fragments, and the resulting assembly will usually contain numerous gaps that have to be filled in later. Shotgun sequencing is the method of choice for virtually all genomes sequenced today, and genome assembly algorithms are a critical area of bioinformatics research.

Another aspect of bioinformatics in sequence analysis is annotation. This involves computational gene finding to search for protein-coding genes, RNA genes, and other functional sequences within a genome. Not all of the nucleotides within a genome are part of genes. Within the genomes of higher organisms, large parts of the DNA do not serve any obvious purpose. This so-called junk DNA may, however, contain unrecognized functional elements. Bioinformatics helps to bridge the gap between genome and proteome projects — for example, in the use of DNA sequences for protein identification.

1.7.1. Genome annotation

In the context of genomics, annotation is the process of marking the genes and other biological features in a DNA sequence. The first genome annotation software system was designed in 1995 by **Owen White**, who was part of the team at TIGR that sequenced and analyzed the first genome of a free-living organism to be decoded, the bacterium *Haemophilus influenzae*. White built a software system to find the genes (fragments of genomic sequence that encode proteins), the transfer RNAs, and to make initial assignments of function to those genes. Most current genome annotation systems work similarly, but the programs available for analysis of genomic DNA, such as the GeneMark program trained and used to find protein-coding genes in *H.influenzae*, are constantly changing and improving.

1.7.2. Computational evolutionary biology

Evolutionary biology is the study of the origin and descent of species, as well as their change over time. Informatics has assisted evolutionary biologists in several key ways; it has enabled researchers to:

- Trace the evolution of a large number of organisms by measuring changes in their DNA, rather than through physical taxonomy or physiological observations alone,
- More recently, compare entire genomes, which permits the study of more complex evolutionary events, such as gene duplication, horizontal gene transfer, and the prediction of factors important in bacterial speciation,
- Build complex computational models of populations to predict the outcome of the system over time
- Track and share information on an increasingly large number of species and organisms.

1.8. Open-source bioinformatics software

Many free and open-source software tools have existed and continued to grow since the 1980s. The combination of a continued need for new algorithms for the analysis of emerging types of biological readouts, the potential for innovative *in silico* experiments, and freely available open code bases have helped to create opportunities for all research groups to contribute to both bioinformatics and the range of open-source software available, regardless of

their funding arrangements. The open source tools often act as incubators of ideas, or community-supported plug-ins in commercial applications. They may also provide de facto standards and shared object models for assisting with the challenge of bioinformation integration.

The range of open-source software packages includes titles such as Bioconductor, BioPerl, Biopython, BioJava, BioRuby, Bioclipse, EMBOSS, .NET Bio, Taverna workbench, and UGENE. In order to maintain this tradition and create further opportunities, the non-profit Open Bioinformatics Foundation have supported the annual Bioinformatics Open Source Conference (BOSC) since 2000.

1.9. Web services in bioinformatics

SOAP- and REST-based interfaces have been developed for a wide variety of bioinformatics applications allowing an application running on one computer in one part of the world to use algorithms, data and computing resources on servers in other parts of the world. The main advantages derive from the fact that end users do not have to deal with software and database maintenance overheads.

Basic bioinformatics services are classified by the EBI into three categories namely (SSS) Sequence Search Services, (MSA) Multiple Sequence Alignment, and (BSA) (Biological Sequence Analysis). The availability of these service-oriented bioinformatics resources demonstrate the applicability of web-based bioinformatics solutions, range from a collection of standalone tools with a common data format under a single, standalone or web-based interface, to integrative, distributed and extensible bioinformatics workflow management systems.

1.10. The Online Bioinformatics Resources Collection (OBRC) held at (University of Pittsburgh, USA) [http://www.hsls.pitt.edu/obrc] -contains annotations and links for 2826 bioinformatics databases and software tools e.g, DNA Sequence Databases and Analysis Tools (505), Enzymes and Pathways (282), Gene Mutations, Genetic Variations and Diseases (303), Genomics Databases and Analysis Tools (704), Immunological Databases and Tools (61), Microarray, SAGE, and other Gene Expression (215), Organelle Databases (29), Other Databases and Tools (Literature Mining, Lab Protocols, Medical Topics, and others) (178), Plant Databases (159), Protein Sequence Databases and Analysis Tools (492), Proteomics Resources

(74), RNA Databases and Analysis Tools (257), and Structure Databases and Analysis Tools (452). Some of the most commonly used resource websites of bioinformatics are as follows:

- **EBI -- European Bioinformatics Institute web services** (http://www.ebi.ac.uk/services/) - Find and use various bioinformatics tools from this huge collection at EBI.
- **INSDC -- International Nucleotide Sequence Database Collaboration** (http://www.insdc.org/) - Find nucleotide sequences.
- **NBRP: National BioResource Project** (http://www.nbrp.jp/) -Access database resources available through the NBRP.
- **NCBI -- the National Center for Biotechnology Information** (http://www.ncbi.nlm.nih.gov/) -Search a huge collection of molecular biology and bioinformatics databases.
- **The European Bioinformatics Institute's data resources - towards systems biology** (http://www.ebi.ac.uk/) - Search and use an extensive collection to molecular biology database and tools.
- **GenomNet**—Operated by Kyoto University Bioinformatics centre (http://www.genome.jp/en/) network of databases and computational services

1.11. Internet

Internet is one thing that we cannot imagine our lives without. It is used in every sphere of life. It has brought the world closer. Today, communicating with friends and relatives living in foreign lands is no longer a costly affair. You can connect with them at just the click of a button. Internet offers various means of communication including email, social media platforms, web calls and messengers. You can call or chat with your near and dear ones at any time of the day with the help of Internet.

Internet is also a great source of entertainment. In today's times when everyone is busy with their own lives internet can prove to be your best friend. From e-books to movies to music – everything you need for entertainment is available on the internet.

Internet has also proved to be a boon for the businessmen. It has become a platform to sell products and make a presence across the country as well as abroad sitting in your home town.

Everything today is being sold online. Even those who are not providing goods and services online are using this medium for promotion of their businesses.

Search Engine

Search engine is a service that allows internet users to search for content via the World Wide Web (WWW). A user enters keywords or key phrases into a search engine and receives a list of web content results in the form of websites, images, videos or other online data. The list of content returned via a search engine to a user is known as a search engine results page (SERP). Unlike web directories, which are maintained only by human editors, search engines also maintain real-time information by running an algorithm on a web crawler. Internet content that is not capable of being searched by a web search engine is generally described as the deep web.

The first search engine ever developed is considered Archie, which was used to search for FTP files and the first text-based search engine is considered Veronica. Today, the most popular and well-known search engine is Google. Other popular search engines include AOL, Ask.com, Baidu, Bing, and Yahoo.

Access to search engine

For users, a search engine is accessed through a browser on their computer, smartphone, tablet, or another device. Now-a-days, most new browsers use an omnibox, which is a text box at the top of the browser that shows the address where you can also search on the internet. You can also visit home page to perform a search.

Working of a search engine

Because large search engines contain millions and sometimes billions of pages, many search engines not only search the pages but also display the results depending on their importance. This importance is commonly determined by using various algorithms.

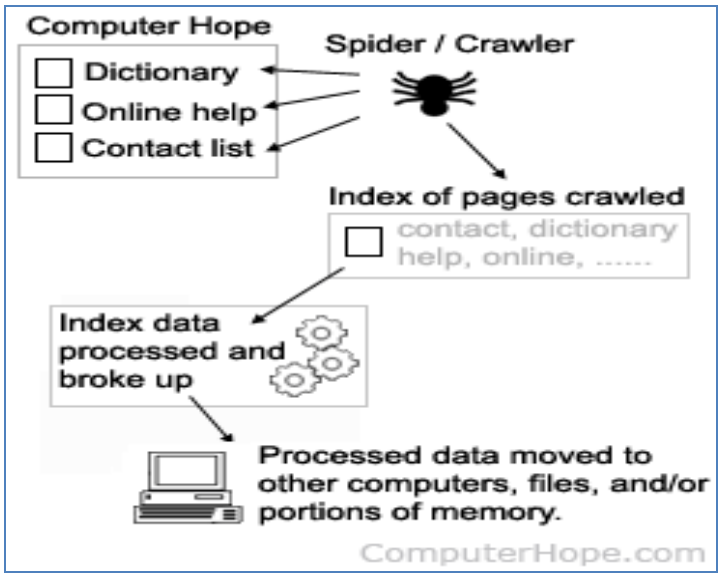


Figure 1.1: Sources of search engine

As illustrated in the Figure 1.1, the source of all search engine data is a spider or crawler, which automatically visits pages and indexes their contents.

- Once a page is crawled, the data contained in the page is processed and indexed. Often, this can involve the steps below.
- Strip out stop words.
- Record the remaining words in the page and the frequency they occur.
- Record links to other pages.
- Record information about any images, audio, and embedded media on the page.

The data collected above is used to rank the page and is the primary method a search engine uses to determine if a page should be shown and in what order.

Finally, once the data is processed, it is broken up into one or more files, moved to different computers, or loaded into memory where it can be accessed when a search is performed.

Search engines use proprietary algorithms to index and correlate data, so every search engine has its own approach to finding what you are trying to find. Its results may be based on where you are located, what else you have searched for, and what results were preferred by other users searching for the same thing. Each search engines will weigh these factors in a unique way, and offer you different results.

There is not one search engine that is better than all the others. Some people could argue that Google search engine is the best and it is probably the most popular and well-known. Often, if someone asks how to do something, or what something is, another person will suggest they "Google it." "Google," used as a verb, means to search for results using the Google search engine. Microsoft's Bing search engine is also popular and used by many people. Bing does an excellent job of finding information and answering questions. Yahoo's search engine, while not quite as popular as it used to be, still does an excellent job of searching for information.

1.11.1. World Wide Web (WWW)

The World Wide Web (WWW) is a network of online content that is formatted in HTML and accessed via HTTP. The term refers to all the interlinked HTML pages that can be accessed over the Internet. The www was originally designed in 1991 by **Tim Berners-Lee** while he was a contractor at CERN. The www is most often referred to simply as "the Web."

The www is what most people think of as the Internet. It is all the Web pages, pictures, videos and other online content that can be accessed via a Web browser. The Internet, in contrast, is the underlying network connection that allows us to send email and access the www. The early Web was a collection of text-based sites hosted by organizations that were technically gifted enough to set up a Web server and learn HTML. It has continued to evolve since the original design, and it now includes interactive (social) media and user-generated content that requires little to no technical skills.

We owe the free Web to Berners-Lee and CERN decision to give away one of the greatest inventions of the century.

1.11.2. URL - Uniform Resource Locator

URL is the abbreviation of **U**niform **R**esource **L**ocator and is defined as the global address of documents and other resources on the www. We all use URLs to visit webpages and other resources on the web. The URL is an address that sends users to a specific resource online, such as a webpage, video or other document or resource. When you search Google, for example, the search results will display the URL of the resources that match your search query. The title in search results is simply a hyperlink to the URL of the resource.

A URL is one type of ***U**niform **R**esource **I**dentifier (**URI**)*; the generic term for all types of names and addresses that refer to objects on the www.

The term "web address" is a synonym for a URL that uses the HTTP or HTTPS protocol. The URL was developed by Tim Berners-Lee in 1994 and the Internet Engineering Task Force (IETF) URI working group. Today, the format of the URL has not changed. The URL format is specified in RFC 1738 Uniform Resource Locators.

1.11.3. HTML

HTML is a abbreviated as ***H**yper **T**ext **M**arkup **L**anguage*, the authoring language used to create documents on the www. HTML is similar to SGML, although it is not a strict subset.

HTML defines the structure and layout of a web document by using a variety of tags and attributes. The correct structure for an HTML document starts with **<HTML><HEAD>** (enter here what document is about) **<BODY>** and ends with **</BODY></HTML>**. All the information you would like to include in your web page fits in between the **<BODY>** and **</BODY>** tags.

HTML Formatting Tags

There are hundreds of other tags used to format and layout the information in a web page. Tags are also used to specify hypertext links. These allow web developers to direct users to other web pages with only a click of the mouse on either an image or words. For a more complete list of HTML tags, check out the WC3 website.

HTML5

HTML5 is a W3C specification that defines the fifth major revision of the HTML. One of the major changes in HTML5 is in respect to how HTML addresses web applications.

1.11.4. Netscape

The Netscape web browser is the general name for a series of web browsers formerly produced by Netscape Communications Corporation, a former subsidiary of America Online (AOL). The original browser was once the dominant browser in terms of usage share, but as a result of the first browser war, it lost virtually its entire share to Internet Explorer.

Netscape was discontinued and support for all Netscape browsers and client products was terminated on March 1, 2008.

1.11.5. Internet Explorer (IE)

Internet Explorer (IE) is a www browser that comes bundled with the Microsoft Windows Operating System (OS). The browser was deprecated in Windows 10 in favour of Microsoft's new Edge Browser. It remains a part of the operating system even though it is no longer the default browser.

As of August 2016, Internet Explorer was the second-most widely used web browser on desktop operating systems, with 29.6% of the market, compared to 50.9% for the Google Chrome browser, according to Net Market Share. IE was the most popular browser from 1999, when it overtook Netscape Navigator, until 2012, when Chrome took the lead. Other competitors include Mozilla Firefox, an open source browser developed using the code from Netscape Navigator, and Apple's Safari.

Microsoft based the original version of Internet Explorer on technology licensed from Spyglass, developer of the pioneering Mosaic browser, and released it for Windows 95 in August 1995. Version 2, released that November, added support for secure sockets layer (SSL) encryption and cookies, and Version 3 followed in August 1996 with Java and cascading style sheets (CSS) -- all important technologies that are still in use today.

In 1998, the U.S. Department of Justice sued Microsoft for antitrust violations, accusing the company of, among other things, stifling web browser competition by bundling Internet Explorer with Windows. In 2001, the two sides reached a settlement that did not require Microsoft to unbundle IE from the operating system.

All told, Internet Explorer has gone through 11 versions and many patches in responses to targeted attacks on flaws in the programming of the application since its initial release. IE 11, released in 2013, is the last version of the web browser. Microsoft Edge replaced IE as the default browser in Windows with the release of Windows 10 in 2015. IE still ships with Windows 10, however, and it is also available as a download from Microsoft's website.

In the past, IE was also available for Unix and Apple's Mac OS X operating system. Microsoft has discontinued those versions as well.

1.11.6. Google

Google Search also referred to as Google Web Search or simply Google, is a web search engine developed by Google LLC. It is the most used search engine on the www across all platforms, with 92.62% market share as of June 2019.

The main purpose of Google Search is to hunt for text in publicly accessible documents offered by web servers, as opposed to other data, such as images or data contained in databases. It was originally developed in 1997 by **Larry Page, Sergey Brin**, and **Scott Hassan**. In June 2011, Google introduced "Google Voice Search" to search for spoken, rather than typed, words. In May 2012, Google introduced a Knowledge Graph semantic search feature in the U.S.

Analysis of the frequency of search terms may indicate economic, social and health trends. Data about the frequency of use of search terms on Google can be openly inquired via Google Trends and have been shown to correlate with flu outbreaks and unemployment levels, and provide the information faster than traditional reporting methods and surveys. As of mid-2016, Google's search engine has begun to rely on deep neural networks.

Competitors of Google include Baidu and Soso.com in China; Naver.com and Daum.net in South Korea; Yandex in Russia; Seznam.czin the Czech Republic; Yahoo in Japan, Taiwan and the US, as well as Bing and DuckDuckGo. Some smaller search engines offer facilities not available with Google, e.g. not storing any private or tracking information.

Within the U.S., as of July 2018, Microsoft Sites handled 24.2% of all search queries. During the same period of time, Oath (formerly known as Yahoo) had a search market share of 11.5%. Market leader Google generated 63.2% of all core search queries in the U.S.

1.11.6.1. Mozilla Browser: Firefox supports most basic Web standards including HTML, XML, XHTML, CSS, JavaScript, DOM, MathML, SVG, XSLT and XPath. Firefox's standards support and growing popularity have been credited as one reason Internet Explorer 7 was to be released with improved standards support. Firefox also implements a proprietary protocol from Google called "safebrowsing", which is not an open standard.

1.11.6.2. Google Chrome: It is a cross-platform web browser developed by Google. It was first released in 2008 for Microsoft Windows, and was later ported to Linux, macOS, iOS, and Android. The browser is also the main component of Chrome OS, where it serves as the platform

for web apps. Most of Chrome's source code comes from Google's open-source Chromium project, but Chrome is licensed as proprietary freeware. WebKit was the original rendering engine, but Google eventually forked it to create the Blink engine; all Chrome variants except iOS now use Blink.

As of July 2019, StatCounter estimates that Chrome has a 71% worldwide browser market share on traditional PCs and 63% across all platforms. Because of this success, Google has expanded the "Chrome" brand name to other products: Chrome OS, Chromecast, Chromebook, Chromebit, Chromebox, and Chromebase.

1.11.6.3. PubMed: PubMed comprises over 30 million citations for biomedical literature from MEDLINE, life science journals, and online books. PubMed citations and abstracts include the fields of biomedicine and health, covering portions of the life sciences, behavioral sciences, chemical sciences, and bioengineering. PubMed also provides access to additional relevant web sites and links to the other NCBI molecular biology resources.

PubMed is a free resource that is developed and maintained by the National Center for Biotechnology Information (NCBI), at the U.S. National Library of Medicine (NLM), located at the National Institutes of Health (NIH).

Publishers of journals can submit their citations to NCBI and then provide access to the full-text of articles at journal web sites using LinkOut.

PubMed also

- Links to full-text articles found in PubMed Central or at publisher web sites, and other related resources.
- Provides Advanced search, Clinical Queries search filters, and Special Queries pages.
- Links to related articles and provides discovery tools for other data that may be of interest.
- Includes automatic e-mailing of search updates, the ability to save records, and filters for search results using "My NCBI".
- Includes a spell checker feature.
- Links to NCBI molecular biology resources.
- Adds citations daily.

Searching contents on PubMed

1. Identify the key concepts for search.
2. Enter the terms (or key concepts) in the search box.
3. Suggestions will display after typing the search terms. Click Turn off to temporarily disable the autocomplete feature. The autocomplete feature is based on PubMed query log analysis described in "Finding Query Suggestions for PubMed."
4. After that "Click Search"

1.11.7. HTTP:

Hyper text transfer protocol is the underlying protocol used by the World Wide Web and this protocol is defines how message are formatted and transmitted, and what actions web servers and browsers should take in response to various commands.

1.12. Database Management System (DBMS)

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data. A DBMS relieves users of framing programs for data maintenance. Fourth-generation query languages, such as SQL, are used along with the DBMS package to interact with a database.

Some other DBMS examples include:

- MySQL
- SQL Server
- Oracle
- dBASE
- FoxPro

A database management system receives instruction from a database administrator (DBA) and accordingly instructs the system to make the necessary changes. These commands can be to load, retrieve or modify existing data from the system.

A DBMS always provides data independence. Any change in storage mechanism and formats are performed without modifying the entire application. There are four main types of database organization:

- **Relational Database:** Data is organized as logically independent tables. Relationships among tables are shown through shared data. The data in one table may reference similar data in other tables, which maintains the integrity of the links among them. This feature is referred to as referential integrity – an important concept in a relational database system. Operations such as "select" and "join" can be performed on these tables. This is the most widely used system of database organization.
- **Flat Database:** Data is organized in a single kind of record with a fixed number of fields. This database type encounters more errors due to the repetitive nature of data.
- **Object-Oriented Database:** Data is organized with similarity to object-oriented programming concepts. An object consists of data and methods, while classes group objects having similar data and methods.
- **Hierarchical Database:** Data is organized with hierarchical relationships. It becomes a complex network if the one-to-many relationship is violated.

1.13. Summary:

Bioinformatics is an interdisciplinary field that develops and improves on methods for storing, retrieving, organizing and analyzing biological data with aim to incease the understanding the biological world. Internet is one thing that we cannot imagine our lives without. Internet works on the basis of WWW, URL and HTML. All the information on the internet can be accessed by using a search engine like Google, Yahoo, msn, Netscape etc. The platform need to surf the internet is known as browser such as Google Chrome, Internet Explorer etc. A collection of data is known as database. A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure.

At the completion of this unit students should

- understand solid framework of bioinformatics
- understand the role and scope of bioinformatics

- gain knowledge on internet and its usage
- able to understand URL,WWW, HTML and HTTP
- able to select and carryout the usage of internet and browsers
- apply the theories of database management system

1.14. Terminal questions:

Q.1. What is Bioinformatics? Define its Scope?

Answer:----------

Q.2. What are the aims of bioinformatics?

Answer:----------

Q.3. What is Internet? Give some examples of usage of internet in modern life?

Answer:----------

Q.4. Define the term browsers? Name two browsers and explain them.

Answer:----------

Q.5. What is Database management system? Explain with examples.

Answer:----------

Q.6. What is PubMed? How to search in PubMed?

Answer:----------

1.15. Suggested reading

2. Bioinformatics: Principles and Applications by Zhumur Ghosh and Bibekanand Mallick, Oxford Press
3. Fundamental Concepts of Bioinformatics by Krane, Pearson Publications
4. Bioinformatics: Methods and Applications: Genomics, Proteomics and Drug Discovery by SC Rastogi, Prentice Hall of India
5. Fundamentals of Bioinformatics by by S. Harisha, L.K. International
6. Internet of Things by Jeeva Jose, Khanna Publishing
7. Bioinformatics: Sequence and Genome Analysis by Mount D., Cold Spring Harbor Laboratory Press, New York
8. Bioinformatics- a Practical Guide to the Analysis of Genes and Proteins by Baxevanis, A.D. and Francis Ouellellette, B.F., Wiley India Pvt Ltd.
9. Introduction to bioinformatics by Teresa K. Attwood, David J. Parry-Smith. Pearson Education

Unit 2-Biological Databases

Structure

2.1. Introduction
Objectives
2.2. Database
2.2.1. Biological databases
2.2.2. Sequence databases and file formats
2.2.3. Protein Data Bank
2.3. Sequence formats
2.3.1. Fasta format
2.3.2. Ontologies
2.4. Sequence conversion tools
2.5. A multiple alignment viewer
2.6. Sequence of File Formats in bioinformatics
2.7. PAM Matrix
2.8. BLOSUM- Blocks Substitution Matrix
2.9. Sequence alignment
2.10. Multiple sequence alignment
2.11. Alignment evaluation
2.12. Summary
2.13. Terminal questions
2.14. Suggested readings

2.1. Introduction

This unit explains the concepts of databases, classification of databases, types of databases, database management system and their importance. Biological databases and their applications are discussed. Important databases in bioinformatics such as NCBI and its subparts are summarized. The unit elaborates the topics such as EMBL, DDBJ and PDB and other important databases in the field of bioinformatics for storage and retrieval of biological sequences. Different sequence formats are compiled along with the methodology and tools required to perform sequence conversion. Also scoring matrices for alignment such as PAM and BLOSUM are illustrated. Sequence alignment and different type of alignment programs are elaborated. Multiple sequence alignment and its methodology is explained.

Objectives

- To understand the concept of database and database management system
- To gain a concept of biological databases and their usage

- To understand sequence conversion and sequence conversion tools
- To gain knowledge of PAM and BLOSUM matrices

2.2. Database

A database is an organized collection of data. Most databases contain multiple tables, which may each include several different fields. For example, a company database may include tables for products, employees, and financial records. Each of these tables would have different fields that are relevant to the information stored in the table.

Vocabulary

- *Entities*: The kind of things that we want to store in a database. E.g. Genes, DNA sequences, bibliographical references.
- *Records*: The particular things stored in the database. E.g. he gene BRCA1
- *Identifiers* or *key*: The unique name that identifies a record
- *Fields*: The properties that an entity has. E.g. he name, sequence and mutations of the gene

So if we think on the database as a table, the table would store information about one entity, the fields would be the column headers and the records would be the table rows.

It is quite common to store different entities in a database. For instance we could store movies, actors and directors or genes, sequences and mutations. In that case, the different entities could be stored in different tables and the records on those tables would be related by their unique identifiers. That structure would comprise a relational database.

The databases usually provide mechanisms to store, search, retrieve and modify the data.

2.2.1. Biological Databases

The data repositories more relevant to the biological sciences include:

- nucleotide and protein sequences
- protein structures
- genomes

- genetic expression
- bibliography

Main sequence databases:

- NCBI
- EMBL

Main protein databases:

- Uniprot
- PDB
- MMDB

Some genome databases:

- ENSEMBL (Human, mouse and others)
- SGD (Yeast)
- TAIR (Arabidopsis)

Bibliography:

- PubMed
- Web of Science

Human diseases:

- OMIM

Metabolic pathways:

- KEGG

2.2.2. Sequence databases and file formats:

A sequence database is a collection of DNA or protein sequences with some extra relevant information. The main nucleic acid sequence databases are GenBank and EMBL. Originally they were just sequence collections, but they have grown to store different biological databases heavily interconnected and they provide powerful interfaces to search and browse the stored information.

The sequences submitted to any of these databases are shared between them, so any sequence could be retrieved in the European or the American database. But they differ in the tools to search and browse the data and in some databases that provide extra information to the raw sequences like: mutations, coded proteins, bibliographical references, etc.

These databases are growing at an ever increasing fast pace. In June of 2007 there were 73 million sequences in GenBank and in August of 2015 there were 187 million (Figure 2.1).

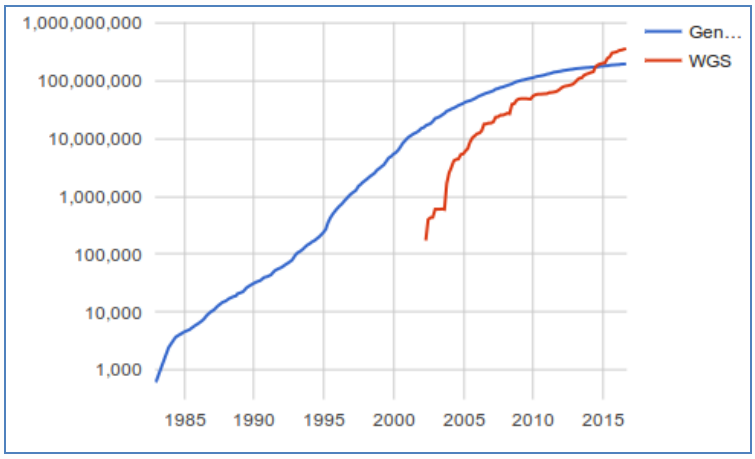


Figure 2.1: The number of bases and the number of sequence records in each release of GenBank, beginning with Release 3 in 1982.

The sequences are split in these databases in different sections to ease the search. Among others, there are sections for mRNAs, publised nucleotide sequences, genomes, and genes.

GenBank

GenBank is a public collection of annotated nucleic acid sequences hosted by the NCBI. Among other kinds of sequences GenBank includes messenger RNAs, genomic DNAs and ribosomic RNA.

Some characteristics:

- It is a public repository, anyone can submit sequences to it.
- There are sequences of different qualities, anything submitted is stored.
- There could be multiple sequences for the same gene or for the same mRNA.
- A sequence can have several versions that represent the modifications done by the authors.

Due to the huge amount of sequences stored to ease the search the databases are split in different divisions. These divisions follow two criteria: the species and type of sequence. Among the taxonomical divisions you can find: primate, rodent, other mammalian, invertebrate and others. The other divisions are related to the kind of sequences like: Expressed Sequence Tags (EST), Whole Genome Sequence (WGS), HT Genomic Sequence (HTGS), and many others. If you are looking for reads coming from the Next Generation Sequencing Technologies they are stored in a special division called Sequence Read Archive (SRA).

RefSeq

RefSeq is a reference database curated by NCBI. In Ref Seq, there are only well annotated and good quality sequences. It stores genomic, transcript and protein sequences and links the sequences that belong to a gene. It just has one representative sequence for each mRNA in a particular organism and, thus, it will have as many sequences as different transcripts and proteins coded for a particular gene in a particular organism.

It is not the aim of ReqSeq to have any sequence, but just to have a collection of well curated sequences. It is a secondary database. Since RefSeq requires extra curation work it, is not available for all organisms, but only for those with good quality sequences. As of July of 2016, it has 65M proteins and 15M transcripts for 60K organisms.

UniProt

UniProt is a protein database that includes information divided in two sections: Swiss-Prot and TrEMBL. UniProt aims to store sequence and functional information for the proteins.

TrEMBL is automatically annotated while Swiss-Prot is reviewed manually by humans that add information by reviewing the literature. Due to this effort Swiss-Prot has information of a higher quality, but it has less sequences than TrEMBL.

UniProt also hosts Uniref. This database aims to store one representative sequence for each protein without taking into account the species of origin. It clusters all the similar proteins and picks one for every cluster as a representative. There are clusters created at 100%, 90% and 50% identities.

PubMed

PubMed is a bibliographical database that comprises biomedical literature (MEDLINE), life science journals and on-line books. It is a good collection of publications related to biochemistry, cellular biology and medicine. As of 2016, PubMed stores 26 million citations.

For each record it stores:

- title
- authors
- abstract

There is a related database named PubMed Cental (PMC) that only includes citations of Free Access Journals. These citations include the complete text for the papers stored.

2.2.3. Protein Data Bank

PDB stores 3-D structures for proteins and nucleic acids. Every database provides one or more methods to search and query the data. It is quite common to provide a web interface in which to do text searches with some keyword, author, ID or any other text. GenBank has a powerful query web interface (**Figure 2.2)**.

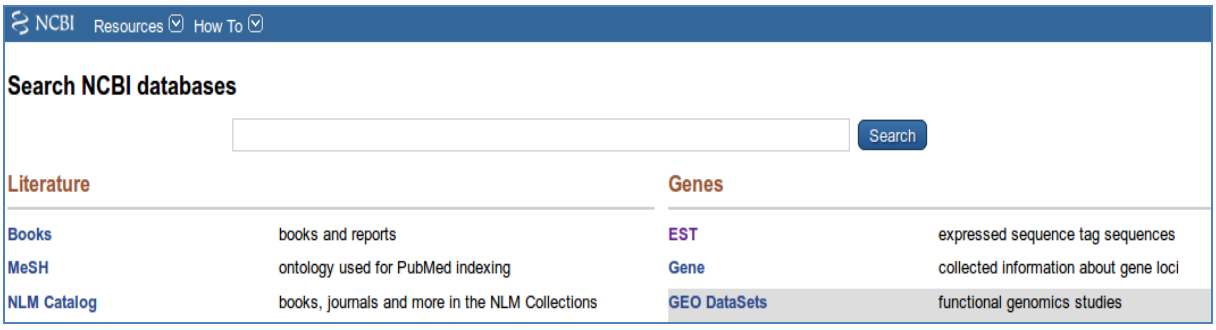


Figure 2.2: NCBI Web Interface

Each database shows the results in one or several formats. For instance, the GenBank sequences can be obtained in several formats.

GenBank format:

LOCUS EC750390 558 bp mRNA linear EST 03-JUL-2006

DEFINITION POE00005652 PL(light) Polytomella parva cDNA similar to frataxin protein

-related, mRNA sequence.

ACCESSION EC750390

VERSION EC750390.1 GI:110064507

KEYWORDS EST.

SOURCE Polytomella parva

ORGANISM Polytomella parva

Eukaryota; Viridiplantae; Chlorophyta; Chlorophyceae;Chlamydomonadales;

Chlamydomonadaceae; Polytomella.

REFERENCE 1 (bases 1 to 558)

AUTHORS Lee,R.W. and Borza,T.

TITLE The colorless plastid of the green alga Polytomella parva: a repertoire of its functions

JOURNAL Unpublished (2006)

COMMENT Contact: TBestDB

FEATURES Location/Qualifiers

source 1..558

/organism="Polytomella parva"

/mol_type="mRNA"

/db_xref="taxon:51329"

/clone_lib="PL(light)"

ORIGIN

1 gcggccgctt ttttttttt ttttttttt ttttcgtccg ttatttcttt tttaagaatg

61 cagtcatctg tacatcgtca agtattcgga gtgttatctc gttttgtggg aaacaaagcg

121 ggtattttta caaagcataa tcatggtgtc tcaaggttgt cttcatgcac ttcgtcatgc

181 gtaaagatgt atactagcaa caaggccccc gaggatcttc aaacgttcca ccggcaagca

241 gacgaaactc tagagcaagt cactgaagcc cttgaaaact atgtagatga gcatgaagtg

301 gaaggcagcg acattgagca tacgcaagga gtgcttacta ttaagcttgg aactcttgga

361 agttatgtaa ttaataaaca gactcctaat aagcagatat ggttatcctc tcccgtcagt

421 ggacccttcc gatatgatct taaagaaggt gcctgggttt atgaacgggc tggcgaggct

481 cggcgcgagc ttatttctca attagaaaca gaaatttcgg atttagttgg tgtcgaatta

541 aagataagta actgaacg

EMBL format:

ID EC750390; SV 1; linear; mRNA; EST; PLN; 558 BP.

XX

AC EC750390;

XX

DT 04-JUL-2006 (Rel. 88, Created)

DT 04-JUL-2006 (Rel. 88, Last updated, Version 1)

XX

DE POE00005652 PL(light) Polytomella parva cDNA similar to frataxin

DE protein-related, mRNA sequence.

XX

KW EST.

XX

OS Polytomella parva

OC Eukaryota; Viridiplantae; Chlorophyta; Chlorophyceae; Chlamydomonadales;

OC Chlamydomonadaceae; Polytomella.

XX

RN [1]

RP 1-558

RA Lee R.W., Borza T.;

RT "The colorless plastid of the green alga Polytomella parva: a repertoire of

RT its functions";

RL Unpublished.

XX

DR UNILIB; 42732; 19932.

XX

CC Contact: TBestDB

CC Departement de Biochimie, Universite de Montrealhttp://es.wikipedia.org/wiki/Base_de_datos

CC Montreal, Canada

CC Email: tbestdb-curator@bch.umontreal.ca

CC Plate: 4065.

XX

FH Key Location/Qualifiers

FH

FT source 1..558

FT /organism="Polytomella parva"

FT /mol_type="mRNA"

FT /clone_lib="PL(light)"

```
FT                   /db_xref="taxon:51329"
FT                   /db_xref="UNILIB:42732"
XX
SQ   Sequence 558 BP; 153 A; 105 C; 127 G; 173 T; 0 other;
     gcggccgctt ttttttttt ttttttttt ttttcgtccg ttatttcttt tttaagaatg      60
     cagtcatctg tacatcgtca agtattcgga gtgttatctc gttttgtggg aaacaaagcg     120
     ggtattttta caaagcataa tcatggtgtc tcaaggttgt cttcatgcac ttcgtcatgc     180
     gtaaagatgt atactagcaa caaggccccc gaggatcttc aaacgttcca ccggcaagca     240
     gacgaaactc tagagcaagt cactgaagcc cttgaaaact atgtagatga gcatgaagtg     300
     gaaggcagcg acattgagca tacgcaagga gtgcttacta ttaagcttgg aactcttgga     360
     agttatgtaa ttaataaaca gactcctaat aagcagatat ggttatcctc tcccgtcagt     420
     ggacccttcc gatatgatct taaagaaggt gcctgggttt atgaacgggc tggcgaggct     480
     cggcgcgagc ttatttctca attagaaaca gaaatttcgg atttagttgg tgtcgaatta     540
     aagataagta actgaacg                                                  558
```

Table 2.1: Main fields in the GenBank format

Field	Description	Search in Entrez
Locus name	Unique sequence name	[ACCN]
Sequence length	Sequence Length	[SLEN]
Molecule Type	DNA, genomic, mRNA, etc.	[PROP]
Genbank Division	Division for the sequence	[PROP]
Modification	Date for the last edit	[MDAT]

Field	Description	Search in Entrez
Date		
Definition	Brief description	[TITL]
Accession	Unique accession ID. It does not changes with modifications	[ACCN]
Version	Version number of the sequence	All fields
Keywords	keywords that describe the sequence	[KYWD]
Source	Common name for the source species	[ORGN]
Organism	Oficial name for the source species	[ORGN]
Reference	Related publications	[TITL][AUTH] [JOUR]
Features	Regions of interest	[FKEY]
CDS	Coding Sequence	[FKEY]

The Accession is the unique identifier for a sequence record. An accession number applies to the complete record and is usually a combination of a letter(s) and numbers, such as a single letter followed by five digits (e.g., U12345) or two letters followed by six digits (e.g., AF123456).

The records in GenBank can be updated by an author request, accession numbers do not change, even if information in the record is changed. So, a sequence can have several versions in GenBank. Version is an unique identifier that represents a single, specific sequence in the GenBank database. If there is any change to the sequence data (even a single base), the version number will be increased, e.g., U12345.1 → U12345.2, but the accession portion will remain stable.

Features holds information about genes and gene products, as well as regions of biological significance reported in the sequence. These can include regions of the sequence that code for proteins and RNA molecules.

Features example:

```
FEATURES             Location/Qualifiers
source          1..12401
                /organism="Homo sapiens"
                /mol_type="genomic DNA"
                /db_xref="taxon:9606"
                /chromosome="17"
                /map="17q24.3"
gene            complement(<1..4774)
                /gene="SOX9-AS1"
                /note="SOX9 antisense RNA 1"
                /db_xref="GeneID:400618"
                /db_xref="HGNC:HGNC:49321"
ncRNA           complement(<4744..4774)
                /ncRNA_class="lncRNA"
                /gene="SOX9-AS1"
                /product="SOX9 antisense RNA 1, transcript variant 2"
                /inference="similar to RNA sequence (same
                species):RefSeq:NR_103737.1"
                /exception="annotated by transcript or proteomic data"
                /transcript_id="NR_103737.1"
```

```
                /db_xref="GeneID:400618"
                /db_xref="HGNC:HGNC:49321"
     gene       5001..10401
                /gene="SOX9"
                /gene_synonym="CMD1; CMPD1; SRA1; SRXX2; SRXY10"
                /note="SRY-box 9"
                /db_xref="GeneID:6662"
                /db_xref="HGNC:HGNC:11204"
                /db_xref="MIM:608160"
     mRNA       join(5001..5803,6700..6953,7524..10401)
                /gene="SOX9"
                /gene_synonym="CMD1; CMPD1; SRA1; SRXX2; SRXY10"
                /product="SRY-box 9"
                /transcript_id="NM_000346.3"
                /db_xref="GeneID:6662"
                /db_xref="HGNC:HGNC:11204"
                /db_xref="MIM:608160"
     exon       5001..5803
                /gene="SOX9"
                /gene_synonym="CMD1; CMPD1; SRA1; SRXX2; SRXY10"
                /inference="alignment:Splign:2.1.0"
                /number=1
     CDS        join(5373..5803,6700..6953,7524..8368)
                /gene="SOX9"
```

/gene_synonym="CMD1; CMPD1; SRA1; SRXX2; SRXY10"

/note="SRY (sex-determining region Y)-box 9 protein;

SRY-related HMG-box, gene 9; SRY (sex determining region

Y)-box9"

/codon_start=1

/product="transcription factor SOX-9"

/protein_id="NP_000337.1"

/db_xref="CCDS:CCDS11689.1"

/db_xref="GeneID:6662"

/db_xref="HGNC:HGNC:11204"

/db_xref="MIM:608160"

/translation="MNLLDPFMKMTDEQEKGLSGAPSPTMSEDSAGSPCPSGSGSDTE

NTRPQENTFPKGEPDLKKESEEDKFPVCIREAVSQVLKGYDWTLVPMPVRVNGSSKNK

PHVKRPMNAFMVWAQAARRKLADQYPHLHNAELSKTLGKLWRLLNESEKRPFVEEAE
R

LRVQHKKDHPDYKYQPRRRKSVKNGQAEAEEATEQTHISPNAIFKALQADSPHSSSGM

SEVHSPGEHSGQSQGPPTPPTTPKTDVQPGKADLKREGRPLPEGGRQPPIDFRDVDIG

ELSSDVISNIETFDVNEFDQYLPPNGHPGVPATHGQVTYTGSYGISSTAATPASAGHV

WMSKQQAPPPPPQQPPQAPPAPQAPPQPQAAPPQQPAAPPQQPQAHTLTTLSSEPGQS

QRTHIKTEQLSPSHYSEQQQHSPQQIAYSPFNLPHYSPSYPPITRSQYDYTDHQNSSS

YYSHAAGQGTGLYSTFTYMNPAQRPMYTPIADTSGVPSIPQTHSPQHWEQPVYTQLTR
P"

STS 5374..5670

/gene="SOX9"

/gene_synonym="CMD1; CMPD1; SRA1; SRXX2; SRXY10"

/standard_name="PMC34415P1"

/db_xref="UniSTS:273201"

exon 6700..6953

/gene="SOX9"

/gene_synonym="CMD1; CMPD1; SRA1; SRXX2; SRXY10"

/inference="alignment:Splign:2.1.0"

/number=2

STS 6765..7647

/gene="SOX9"

/gene_synonym="CMD1; CMPD1; SRA1; SRXX2; SRXY10"

/standard_name="PMC351321P1"

/db_xref="UniSTS:273242"

STS 6771..7605

/gene="SOX9"

/gene_synonym="CMD1; CMPD1; SRA1; SRXX2; SRXY10"

/standard_name="MARC_71570-71571:1249593890:3"

```
                /db_xref="UniSTS:528379"
     STS        6876..8104
                /gene="SOX9"
                /gene_synonym="CMD1; CMPD1; SRA1; SRXX2; SRXY10"
                /standard_name="Sox9"
                /db_xref="UniSTS:502689"
     exon       7524..10401
                /gene="SOX9"
                /gene_synonym="CMD1; CMPD1; SRA1; SRXX2; SRXY10"
                /inference="alignment:Splign:2.1.0"
                /number=3
```

2.3. Sequence formats

Text files

There are different formats to store sequences in a text file. Text files should only include Plain text. Graphics or any other binary information are not allowed in text files. Microsoft Word files are not text file they are binary files that happen to represent documents. These documents can include text among many other things like images, charts or formats. Sequences are in plain files. We could store the sequence in a text file by just writing the sequence. These files would have to include only IUPAC characters.

```
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGC
CACCGCTGCCCTGCCCCTGGAGGGTAC

GGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGAA
AAGCAGCCTCCTGACTTTCCTCGCTTGGT

AGTGGACCTCCCAGGCCAGTGCCGGGCCCCTCATAGGAGAGGAAGCTCGGGAGG
TGGCCAGGCGGCAGGAAGGCGCACCCCC
```

ATCCGCGCGCCGGGACAGAATGCCCTGCAGGAACTTCTTCTGGAAGACCTTCTCC
TCCTGCAAATAAAA

This kind of file is seldom used because it lacks any metadata to identify the sequence.

2.3.1. Fasta format

The Fasta file includes a name for the sequence and, optionally, some description. The sequence should be preceded by a line that starts with the symbol >. The name will be written after that symbol. Spaces are not allowed in the sequence name. If there is a description it will be found after a space in the same line. Several sequences can be included in the same file. It is one of the most common formats.

>sequence1_name description

ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGC
CACCGCTGCCCTGCC

CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGG
AATAAGGAAAAGCAGC

CTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCC
CCTCATAGGAGAGG

AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGC
GCCGGGACAGAATGCC

CTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCAT
GAATGCTCACGC

>sequence2_name description

ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGC
CACCGCTGCCCTGCC

CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGG
A

If we want to include more information we could use the GenBank or EMBL formats. It is also very common in the sequences that come directly from a sequencing machine to include the quality information, for that purpose the most common format is FASTQ.

2.3.2. Ontologies

"An ontology is a formal naming and definition of the types, properties, and interrelationships of the entities that really or fundamentally exist for a particular domain of discourse".

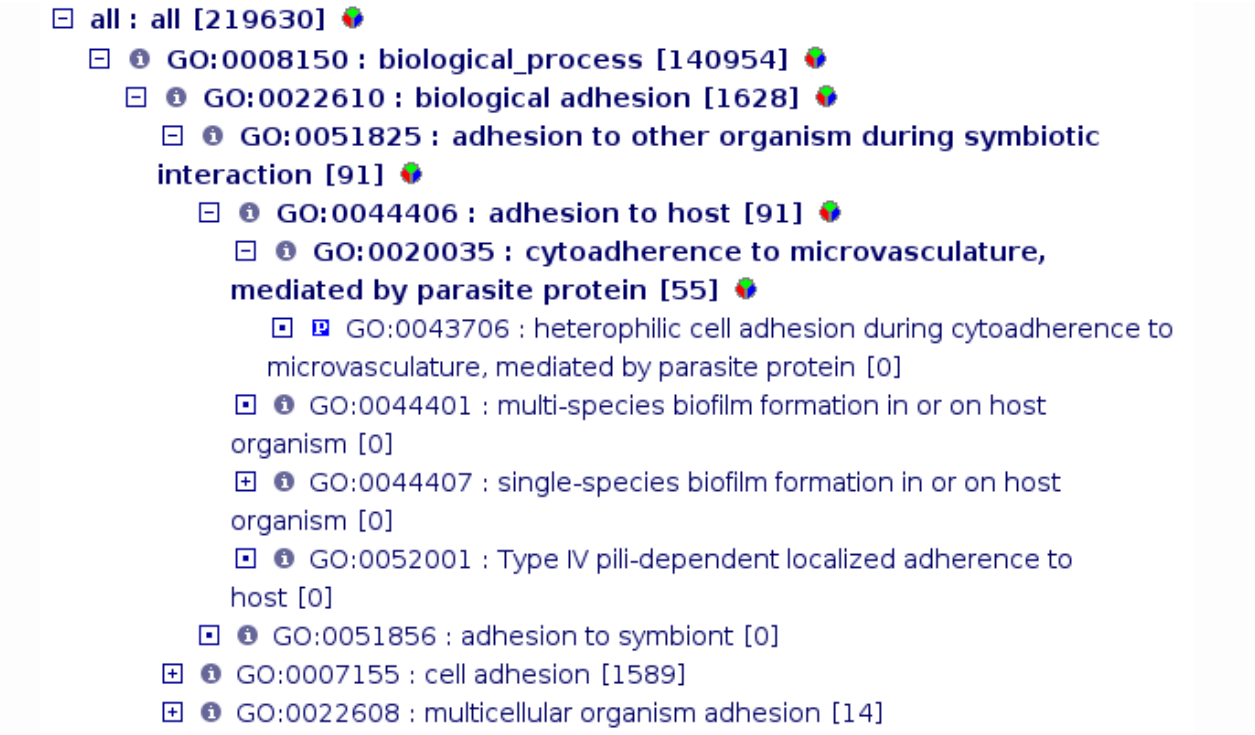
An ontology is a way of structure the knowledge by dividing it in the entities relevant to a particular field. For instance, we have been talking about sequences, so a term in our ontology could be *sequence*.

It is also quite common to create hierarchical ontologies. For instance, *nucleotide sequence* and *protein sequence* could be sub-terms of sequence. In an ontology the terms are precisely defined and, usually, there are no synonyms.

Defining the terms relevant to a field is very useful, especially if those terms are discussed and adopted by the whole community. Standard ontologies became powerful tools that enable automatic analyses and searches. Imagine that we want to look for all enzymes related to lipid metabolism in a database.

There are different biological ontologies, but the main ones are maintained by the Gene Ontology Consortium. The GO terms are used to define gene functions. Each GO term has an unique ID and a definition. There are three aspects covered by three hierarchical ontologies:

- *molecular function*: molecular activities of gene products
- *cellular component*: where gene products are active
- *biological process*: pathways and larger processes made up of the activities of multiple gene products.



GO hierarchies are browsed at a GO browser

A gene can have several GO terms associated with each ontology. A gene can be annotated with terms from different levels of the hierarchy.

The GO ontologies ease the search for information and allow complex automated analyses.

2.4. Sequence conversion tools

These tools read different biological sequence formats and can convert them to other formats.

Seqret (EMBOSS)

EMBOSS Seqret reads and reformats bio-sequences.

Launch Seqret

MView

Transform a Sequence Similarity Search result into a Multiple Sequence Alignment or reformat a Multiple Sequence Alignment using the MView program.

Launch MView

EMBOSS Seqret

EMBOSS Seqret reads and writes (returns) sequences. It is useful for a variety of tasks, including extracting sequences from databases, displaying sequences, reformatting sequences, producing the reverse complement of a sequence, extracting fragments of a sequence, sequence case conversion or any combination of the above functions.

2.5. A multiple alignment viewer

MView reformats the results of a sequence database search (BLAST, FASTA, etc) or a multiple alignment (MSF, PIR, CLUSTAL, etc) adding optional HTML markup to control colouring and web page layout. MView is not a multiple alignment program, nor is it a general purpose alignment editor.

Several sites are available for conversion of sequence from one format to another. These include:

- Galaxy is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research. This web server makes analysis tools, genomic data, tutorial demonstrations, persistent workspaces, and publication services available to any scientist. Extensive user documentation applicable to any public or local Galaxy instance is available. Offers a huge variety of tools for analysis and file interconversion.
- Sequence conversion (*Bioinf @ Bugaco*) - a huge suite of conversion tools. Also try Conversion.
- Readseq developed by D.G. Gilbert (Indiana University) reads and converts biosequences between a selection of common biological sequence formats, including EMBL, GenBank and fasta sequence formats is available at this platform.
- EMBOSS Seqret reads and writes (returns) sequences. It is useful for a variety of tasks, including extracting sequences from databases, displaying sequences, reformatting sequences, producing the reverse complement of a sequence, extracting fragments of a sequence, sequence case conversion or any combination of the above functions.
- Sequence editor - Convert DNA and RNA sequences. Generate antiparallel, complement and inverse sequences.

- Format Converter - This program takes as input a sequence or sequences (e.g., an alignment) in an unspecified format and converts the sequence(s) to a different user-specified format. Also converts *.gbk to *.gff3.
- ApolloRNA Convert data - Transformation of TransTermHP, CRISPRfinder, MOSAIC, PatScan, DARN! (GFF), GenBank output data in GFF and GAME XML format data that can be read by Apollo.
- GenBank Trans Extractor, it accepts a GenBank file as input and returns each of the protein translations described in the file in FASTA format. GenBank Trans Extractor should be used when you are more interested in the predicted protein translations of a DNA sequence than the DNA sequence itself.
- Sequence Manipulation Suite. It is a collection of java script programmes for generating , formatting and analysing short DNA and protein sequences. It is commonly used by molecular biologist, for teaching, and for program and algorithm testing.
- FeatureExtract 1.2L (light) Server - extracts sequence and feature annotation, such as intron/exon structure, from GenBank entries and other GenBank format files. (Reference: R. Wernersson (2005) Nucleic Acids Res. 33 (Web Server issue): W567–W569).
- Sequence editor - converts DNA and RNA sequences. Generate antiparallel, complement and inverse sequences.
- Format conversion - (single sequence, set of sequences, alignment, tree, matrix) and format are automatically recognized. Output: FASTA, NEXUS, PHYLIP, Clustal, EMBL, Newick, New Hampshire).
- Fasta dataset splitter - Part of FaBox
- GenBank 2 Sequin (*P. Lehwark & S. Greiner, Max-Planck Institute for Molecular Plant Physiology, Germany*) - this extremely usesful program is designed to convert revised GeSeq output into the Sequin format, required for NCBI submission. None the less, any custom GenBank file can be prepared for NCBI submission using GenBank 2 Sequin.

- JaMBW *(European Molecular Biology Laboratory of Heidelberg, Germany)*. Java based Molecular Biologist's Workbench.Select Chapter 1 for sequence format conversion (upper ⟷ lower case; T ⟷ U; reverse or complement sequence).
- Nucleic Acid Sequence Massager *(Allotron Biosensor Corporation)* which in addition to removing spurious material (numbers, breaks, HTML, spaces) changes the format (upper to low case, complement, reverse, RNA to DNA, and triplets).
- extractUpStreamDNA *(A. Villegas, Public Health Ontario)* - takes a GenBank flat file (*.gbk) as input and parses through and for every CDS that it finds, it extracts a pre-determined length of DNA upstream (length will be an argument; and will include 3 nt for the initiation codon). Output will be an FFN file of these upstream DNA sequences. N.B. this only WORKS for prokaryotic sequences because it does not handle Splits or Joins found in eukaryotic. This data then can be analyzed with programs such as MEME. This program is temporarily unavailable online, though one can download it from here.
- Convert GenBank to Fasta *(G. Rocap, School of Oceanography, University of Washington, U.S.A.)* - Select a GenBank formatted file containing a feature table. Select whether to extract translated peptide sequences, DNA sequence for each feature, or the entire DNA sequenceof the whole record. If you chose "Peptide Sequence", your feature table must have "translation"sub-features.
- FaBox *(Palle Villesen Fredsted, Aarhus University, Denmark)* - an online fasta sequence toolbox, including Fasta header editor, Fasta header replacer, Fasta sequence extractor, Fasta sequence subtractor, Fasta sequence joiner, Fasta dataset splitter/divider
- Feature Extract - this very useful service extracts sequence and feature annotation, such as intron/exon structure, from GenBank entries and other GenBank format files.
- Sequence editor - carries out numerous functions:s
- Antiparallel - Create the antiparallel DNA or RNA strand. For example the sequence ATGC will be converted into GCAT. It is a combination of the both functions Complement and Inverse.
- Complement - Create the complement DNA or RNA strand. For example the sequence ATGC will be converted into TACG.

- Inverse - Create the inverse DNA or RNA strand. For example the sequence ATGC will be converted into CGTA.
- T to U - Replace all thymidine by uracil. For example the sequence ATUGC will be converted into AUUGC.
- U to T - Replace all uracil by thymidine. For example the sequence ATUGC will be converted into ATTGC.
- UCase - Convert the sequence into upper case.
- LCase - Convert the sequence into lower case.
- Shuffle DNA and Sequence Randomizer permit one to randomize a sequence to compare with one's own.

2.6. Sequence of File Formats in bioinformatics

In the field of bioinformatics there exists many different file formats that store DNA and protein sequence information. There is no one sequence format that is ideal, many are used in different contexts, and can often be converted from one to another for easier access or sharing. Below is a list of file formats and a link to their respective file format specs and descriptions for anyone wishing to get to know the file formats a little better. While there are many different formats out there used by commercial software, this list focuses mainly on open, non-proprietary file formats.

- **GenBank** - quite possibly the standard in sequence file formats, the Genbank format is widely used by public databases such as NCBI. The Genbank file format is quite flexible and allows annotations, comments, and references to be included within the file. The file is plain text and thus can be read with a text editor. Genbank files often have the file extension '.gb' or '.genbank'.

- **EMBL** - similar in form to the GenBank file, the EMBL format is used by public databases such as European Molecular Biology Laboratory. The GenBank file format is quite flexible and allows annotations, comments, and references to be included within the file. The file is plain text and thus can be read with a text editor. GenBank files often have the file extension '.gb' or '.genbank'.

- **ABI** - ABI is a binary file format containing Sanger sequencing sequence and trace data. The format is used by sequencing facilities and requires special readers capable of reading the file format to view the trace data and extract the sequence. The file format is difficult to parse given its binary nature and the complexity of the spec.

- **PDB** - the PDB file format is used to store both sequence information, but more importantly stores 3-dimensional structure information. This information can be used to visualize the crystal structure of a given molecule (typically a protein). PDB files are simply text files, thus can be viewed with a text editor, and often have the file extension '.pdb'.

- **MDL** - While not technically containing sequence data, the MDL file format is worth including in this list. The MDL mol file contains information regarding small molecules, the spec being quite similar to that of the PDB file format. The MDL mol file contains information regarding 2d (and possibly 3d) molecule structure, such as atom type and atom connectivity.

- **BAM/SAM** - The BAM/SAM format contains next-generation sequencing data. The BAM is a binary file format while the SAM file format contains the same information but is text based. These files can be analyzed and viewed by several free software tools, such as the command line open source tool SAMTools and the user interface tool IGV. Both the BAM/SAM format contain not only the sequence data for next-generation sequencing reads, but also have the capability of storing alignment data of those reads to a reference sequence.

- **SFF** - The SFF file format specifies a binary file which contains next-generation sequence information. The name stands for *standard flowgram format*, and contains the actual flow information used on several next-generation DNA sequencers, including Ion-Torrent and Roche's '454'.

Scoring Matrix:

Scoring matrices are used to determine the relative score made by matching two characters in a sequence alignment. These are usually log-odds of the likelihood of two characters being derived from a common ancestral character. There are many flavors of scoring matrices for amino acid sequences, nucleotide sequences, and codon sequences, and each is derived from the alignment of "known" homologous sequences. These alignments are then used to determine the likelihood of one character being at the same position in the sequence as another character. (Figure 2.3)

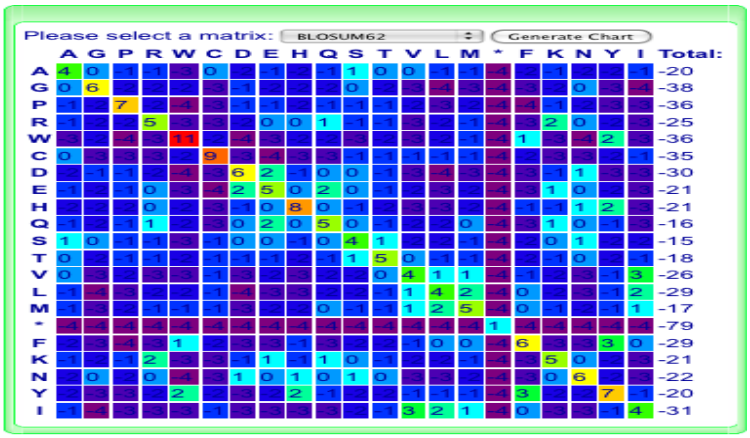


Figure 2.3: Matrix View: Sequence alignment matrix view

2.7. PAM matrices

PAM matrices are amino acid substitution matrices that encode the expected evolutionary change at the amino acid level. Each PAM matrix is designed to compare two sequences which are a specific number of PAM units apart. For example - the PAM120 score matrix is designed to compare between sequences that are 120 PAM units apart: The score it gives a pair of sequences is the (log of the) probabilities of such sequences evolving during 120 PAM units of evolution. For any specific pair (A_i, A_j) of amino acids the (i,j) entry in the PAM n matrix reflects the frequency at which A_i is expected to replace with A_j in two sequences that are n PAM units diverged. These frequencies should be estimated by gathering statistics on replaced amino acids.

Collecting statistics about amino acids substitution in order to compute the PAM matrices is relatively difficult for sequences that are distantly diverged, as mentioned in the previous section. But for sequences that are highly similar, i.e., the PAM divergence distance between them is small, finding the position correspondence is relatively easy since only few insertions and deletions took place. Therefore, in the first stage statistics were collected from aligned sequences that were believed to be approximately one PAM unit diverged and the PAM1 matrix could be computed based on this data, as follows: Let M_{ij} denote the observed frequency (= estimated probability) of amino acid A_i mutating into amino acid A_j during one PAM unit of evolutionary

change. *M* is a *20X* real matrix, with the values in each matrix column adding up to 1. There is a significant variance between the values in each column. **(**Table No. 2.2)

Table 2.2: The top left corner *5X5* of the PAM1 matrix. We write $10^4 M_{ij}$ for convenience.

	A	*R*	*N*	*D*	*C*
A	9867	2	9	10	3
R	1	9913	1	0	1
N	4	1	9822	36	0
D	6	0	42	9859	0
C	1	1	0	0	9973

Once *M* is known, the matrix M^n gives the probabilities of any amino acid mutating to any other during *n* PAM units. The (*i*,*j*) entry in the PAM *n* matrix is therefore:

$$\log \frac{f(j)M^n(i,j)}{f(i)f(j)} = \log \frac{M^n(i,j)}{f(i)}$$

where *f*(*i*) and *f*(*j*) are the observed frequencies of amino acids A_i and A_j respectively. This approach assumes that the frequencies of the amino acids remain constant over time, and that the mutational processes causing substitutions during an interval of one PAM unit operate in the same manner for longer periods. We take the log value of the probability in order to allow computing the total score of all substitutions using summation rather than multiplication. The PAM matrix is usually organized by dividing the amino acids to groups of relatively similar amino acids and all group members are located in consecutive columns in the matrix.

2.8. BLOSUM - Blocks Substitution Matrix:

The BLOSUM matrix is another amino acid substitution matrix used for sequence alignment of protein, first calculated by Steven *Henikoff* and Jorja *Henikoff* . For its calculation only blocks

of amino acid sequences with small change between them are considered. These blocks are called *conserved blocks* (Table 2.3). One reason for this is that one needs to find a multiple alignment between all these sequences and it is easier to construct such an alignment with more similar sequences. Another reason is that the purpose of the matrix is to measure the probability of one amino acid to change into another, and the change between distant sequences may include also insertions and deletions of amino acids.

Table 2.3: Alignment of several sequences. The conserved blocks are marked.

AABCDA...BBCDA
DABCDA.A.BBCBB
BBBCDABA.BCCAA
AAACDAC.DCBCDB
CCBADAB.DBBDCC
AAACAA...BBCCC

The first stage of building the BLOSUM matrix is eliminating sequences, which are identical in more than *x*% of their amino acid sequence. This is done to avoid bias of the result in favor of a certain protein. The elimination is done either by removing sequences from the block, or by finding a cluster of similar sequences and replacing it by a new sequence that represents the cluster. The matrix built from blocks with no more the *x*% of similarity is called BLOSUM-*x* (e.g. the matrix built using sequences with no more than 50% similarity is called BLOSUM-50.)

The second stage is counting the pairs of amino acids in each column of the multiple sequence alignment. For example in a column with the acids AABACA (as in the first column in the block in figure), there are 6 AA pairs, 4 AB pairs, 4 AC, and one BC. The probability $q_{i,j}$ for a unique pair of amino acids at a site (A_i and A_j) is compounded as well as the probability p_i of the unique amino acid to be A_i.

In the third stage the *log odds ratio* (s,i,j) is calculated as $s_{i,j} = \log_2 \frac{q_i}{p_i}$

As final result we consider the rounded $s_{i,j}$, this value is stored in the (i,j) entry of the BLOSUM-*x* matrix.

A verities of lock substitution matrix are available, whose utility depends on whether the user is comparing more highly divergent or les divergent sequences.

In contrast to the PAM matrices, more sequences are examined in the process of computing the BLOSUM matrix. Moreover, the sequences are of specific nature of resemblance, and therefore the two sets of matrices differ.

Comparing the efficiency of two matrices is done by calculating the ratio between the number of pairs of similar sequences discovered by a certain matrix but not discovered by another one and the number of pairs missed by the first but found by the other. According to this comparison BLOSUM-62 is found to be better than other BLOSUM-*x* matrices as well as than PAM-*x* matrices.

2.9. Sequence Alignment

When two symbolic representations of DNA or protein sequences are arranged next to one another so that their most similar elements are juxtaposed they are said to be **aligned**. Many bioinformatics tasks depend upon successful alignments. Alignments are conventionally shown as traces.

In a symbolic sequence each base or residue monomer in each sequence is represented by a letter. The convention is to print the single-letter codes for the constituent monomers in order in a fixed font (from the N-most to C-most end of the protein sequence in question or from 5' to 3' of a nucleic acid molecule). This is based on the assumption that the combined monomers evenly spaced along the single dimension of the molecule's primary structure. From now on we will refer to an alignment of two protein sequences.

Every element in a trace is either a **match** or a **gap**. Where a residue in one of two aligned sequences is identical to its counterpart in the other the corresponding amino-acid letter codes in the two sequences are vertically aligned in the trace, a match. When a residue in one sequence seems to have been deleted since the assumed divergence of the sequence from its counterpart, its "absence" is labelled by a dash in the derived sequence. When a residue appears to have been inserted to produce a longer sequence a dash appears opposite in the unaugmented sequence.

Since these dashes represent "gaps" in one or other sequence, the action of inserting such spacers is known as **gapping**.

A deletion in one sequence is symmetric with an insertion in the other. When one sequence is gapped relative to another a deletion in sequence **a** can be seen as an insertion in sequence **b**. Indeed, the two types of mutation are referred to together as **indels**. If we imagine that at some point one of the sequences was identical to its primitive homologue, then a trace can represent the three ways divergence could occur (at that point).

The first step to compare two sequences is, usually, to align them.

No alignment

```
CGATGCTAGCGTATCGTAGTCTATCGTAC
        |    ||
ACGATGCTAGCGTTTCGTATCATCGTA
```

Alignned

```
-CGATGCTAGCGTATCGTAGTCTATCGTAC
 |||||||||||| ||||||||||||||
ACGATGCTAGCGTTTCGTA-TC-ATCGTA-
```

Alignments could be used to:

- Quantify the phylogenetic distance between two sequences
- Look for functional domains
- Compare a mRNA with its genomic region
- Identify polymorphisms and mutations between sequences

Software for alignment

- Chimera - excellent molecular graphics package with support for a wide range of operations
- Clustal-W - the famous Clustal-W multiple alignment program
- Clustal-X - provides a window-based user interface to the Clustal-W multiple alignment program
- JAligner - a Java implementation of biological sequence alignment algorithms
- ModView - a program to visualize and analyze multiple biomolecule structures and/or sequence alignments
- Musca - alignment of amino acid or nucleotide sequences; uses pattern discovery
- MUSCLE - more accurate than T-Coffee, faster than Clustal-W
- PhyloDraw - a drawing tool for creating phylogenetic trees
- SAM - a collection of flexible software tools for creating, refining, and using linear Hidden Markov Models for biological sequence analysis
- SeaView - a graphical multiple sequence alignment editor
- ShadyBox - the first GUI based WYSIWYG multiple sequence alignment drawing program for Major Unix platforms
- UGENE - a graphical interface for Muscle3, Muscle4, KAlign and Phylip packages. Integrates both multiple alignment and phylogenetic tree editors

2.10. Multiple Sequence Alignment

A Multiple Sequence Alignment is an alignment of more than two sequences. We could align several DNA or protein sequences.

Some of the most usual uses of the multiple alignments are:

- phylogenetic analysis
- conserved domains
- protein structure comparison and prediction
- conserved regions in promoters

The multiple sequence alignment assumes that the sequences are homologous, they descend from a common ancestor. The algorithms will try to align homologous positions or regions with the same structure or function.

2.11. Multiple alignment algorithm

Multiple alignments are computationally much more difficult than pair-wise alignments. It would be ideal to use an analog of the Smith & Waterman algorithm capable of looking for optimal alignments in the diagonals of a multidimensional matrix given a scoring schema. This algorithm would have to create a multidimensional matrix with one dimension for each sequence. The memory and time required for solving the problem would increase geometrically with the length of every sequence. Given the number of sequences usually involved no algorithm

is capable of doing that. Every algorithm available reverts to a heuristic capable of solving the problem in a much faster time. The drawback is that the result might not be optimal.

Usually the multiple sequence algorithms assume that the sequences are similar in all its length and they behave like global alignment algorithms. They also assume that there are not many long insertions and deletions. Thus the algorithms will work for some sequences, but not for others.

These algorithms can deal with sequences that are quite different, but, as in the pair-wise case, when the sequences are very different they might have problems creating good algorithm. A good algorithm should align the homologous positions or the positions with the same structure or function.

If we are trying to align two homologous proteins from two species that are phylogenetically very distant we might align quite easily the more conserved regions, like the conserved domains, but we will have problems aligning the more different regions. This was also the case in the pair-wise case, but remember that the multiple alignment algorithms are not guaranteed to give back the best possible alignment.

These algorithms are not design to align sequences that do not cover the whole region, like the reads from a sequencing project. There are other algorithms to assemble sequencing projects.

Progressive construction algorithms

In Multiple Sequence Alignment it is quite common that the algorithms use a progressive alignment strategy. These methods are fast and allow to align thousands of sequences.

Before starting the alignment, as in the pair-wise case, we have to decide which is the scoring scheme that we are going to use for the matches, gaps and gap extensions. The aim of the alignment would be to get the multiple sequence alignment with the highest score possible. In the multiple alignment case, we do not have any practical algorithm that guarantees that it going to get the optimal solution, but we hope that the solution will be close enough, if the sequences comply with the restrictions assumed by the algorithm.

The idea behind the progressive construction algorithm is to build the pair-wise alignments of the more closely related sequences that should be easier to build, and to align progressively these alignments once we have them. To do it, we need first to determine which the closest sequence

pairs are. One rough and fast way of determining which the closest sequence pairs are is to align all the possible pairs and look at the scores of those alignments. The pair-wise alignments with the highest scores should be the ones between the more similar sequences. So the first step in the algorithm is to create all the pair-wise alignments and to create a matrix with the scores between the pairs. This matrix will include the similarity relations between all sequences.

Once we have this matrix we can determine the hierarchical relation between the sequences, which are the closest pairs and how these pairs are related and so on, by creating a hierarchical clustering, a tree. We can create these threes by using different fast algorithms like UPGMA or Neighbour joining. These trees are usually known as guide trees.

Example:

Secuences:

IMPRESIONANTE

INCUESTIONABLE

IMPRESO

Scores:

IMPRESIONANTE X IMPRESO 7/13

IMPRESIONANTE X INCUESTIONABLE 10/14

INCUESTIONABLE X IMPRESO 4/14

Scoring pair-wise matrix:

	IMPRESIONANTE	INCUESTIONABLE	IMPRESO
IMPRESIONANTE	1	10/14	7/13
INCUESTIONABLE	10/14	1	4/14
IMPRESO	7/13	4/14	1

Guide Tree:

```
    |--- IMPRESIONANTE
  |---|--- INCUESTIONABLE
  |
  |----- IMPRESO
```

The first alignment would be: IMPRESIONANTE x INCUESTIONABLE

```
    IMPRES-IONABLE
    INCUESTIANABLE
```

Now we align IMPRESO to the previous alignment.

```
    IMPRES-IONANTE
    INCUESTIONABLE
    IMPRES--O-----
```

We have no guarantee that the final is the one with the highest score.

The main problem of these progressive alignment algorithms is that the errors introduced at any point in the process are not revised in the following phases to speed up the process. For instance, if we introduce one gap in the first pair-wise alignment, this gap will be propagated to all the following alignments. If the gap was correct that is fine, but if it was not optimal it won't be fixed. These methods are especially prone to fail when the sequences are very different or phylogenetically distant.

Sequences to align already in the order given by a guide tree:

Seq A GARFIELD THE LAST FAT CAT

Seq B GARFIELD THE FAST CAT

Seq C GARFIELD THE VERY FAST CAT

Seq D THE FAT CAT

Step 1

Seq A GARFIELD THE LAST FAT CAT

Seq B GARFIELD THE FAST CAT

Step 2

Seq A GARFIELD THE LAST FA-T CAT

Seq B GARFIELD THE FAST CA-T

Seq C GARFIELD THE VERY FAST CAT

Step 3

Seq A GARFIELD THE LAST FA-T CAT

Seq B GARFIELD THE FAST CA-T

Seq C GARFIELD THE VERY FAST CAT

Seq D -------- THE ---- FA-T CAT

Historically the most used of the progressive multiple alignment algorithms was CLUSTALW. Nowadays CLUSTALW is not one of the recommended algorithms anymore because there are other algorithms that create better alignments like Clustal Omega or MAFFT. MAFFT was one of the best contenders in a multiple alignment software comparison.

T-Coffee is another progressive algorithm. T-Coffee tries to solve the errors introduced by the progressive methods by taking into account the pair-wise alignments. First it creates a library of all the possible pair-wise alignments plus a multiple alignment using an algorithm similar to the CLUSTALW one. To this library we can add more alignments based on extra information like the protein structure or the protein domain composition. Then it creates a progressive alignment, but it takes into accounts all the alignments in the library that relate to the sequences aligned at that step to avoid errors. The T-Coffe algorithm follows the steps:

1. Create the pair-wise alignments
2. Calculate the similarity matrix
3. Create the guide tree
4. Build the multiple progressive alignment following the tree, but taking into account the information from the pair-wise alignments.

T-Coffee is usually better than CLUSTALW and performs well even with very different sequences, especially if we feed it more information, like: domains, structures or secondary structure. T-Coffee is slower than CLUSTALW and that is one of its main limitations, it can not work with more than few hundred sequences.

Iterative algorithms

These methods are similar to the progressive ones, but in each step the previous alignments are revaluated. Some of the most popular iterative methods are Muscle and MAFFT (multiple sequence comparison by log expectation is claimed to achieve both better average accuracy and better speed than clustral W2 or T-Coffee) two popular examples of these algorithms.

Hidden Markov models

The most advanced algorithms to date are based on Hidden Markov Models and they have improvements in the guide tree construction, like the sequence embedding that reduce the computation time. Clustal Omega is one of these algorithms and can create alignments as accurate of the T-Coffee, but with many thousands of sequences.

2.11. Alignment evaluation

Once we have created our Multiple Sequence Alignment, we should check that the result is OK. We could open the multiple alignments in a viewer to assess the quality of the different regions of the alignment or we could automate this assessment. Usually not all the regions have an alignment of the same quality. The more conserved regions will be more easily aligned than the more variable ones. It is quite usual to remove the regions that are not well aligned before doing any further analysis, like a phylogenetic reconstruction. We can remove those regions manually or we can use an specialized algorithm like trimAl.

2.12. Summary

A database is an organized collection of data. The databases which usually provide mechanisms to store, search, retrieve and modify the biological data such as nucleotide or protein sequences are known as biological databases eg. NCBI, EMBL, PDB etc. Biological sequences are stored in a database in a specific format such as FASTA. The formats of the sequences can be changed by different tools/softwares. Multiple sequence alignment is a methodology to align biological sequences for searching their identity and evolutionary significance. CLUSTAL W_2 is(in a general purpose multiple sequence alignment program for DNA or proteins) the major tool used for these purpose.

Outcomes:

At the completion of this unit students should

- understand the concept of database and database management system
- develop a concept of biological databases and their usage
- able to explain the concepts of NCBI, its parts and uses
- sequence conversion and sequence conversion tools
- gain knowledge of PAM and BLOSUM matrices

- understand sequence alignment and tools
- perform Multiple sequence alignment

2.13. Terminal questions

Q.1. What is Database? Explain the terms Entities, Records, Keys and Fields in terms of Database.

Answer: ----------

Q.2. What are biological databases? Give examples.

Answer: ----------

Q.3. . What is Genbank? Explain

Answer: ----------

Q.4. What is FASTA format? Explain.

Answer: ----------

Q.5. Define some sequence conversion tools.

Answer: ----------

Q.6. What is a scoring matrices? Explain with examples?

Answer: ----------

2.14. Suggested reading:

1. Bioinformatics: Sequence and Genome Analysis by Mount D., Cold Spring Harbor Laboratory Press, New York
2. Bioinformatics- a Practical Guide to the Analysis of Genes and Proteins by Baxevanis, A.D. and Francis Ouellellette, B.F., Wiley India Pvt Ltd.
3. Introduction to bioinformatics by Teresa K. Attwood, David J. Parry-Smith. Pearson Education.
4. Fundamental Concepts of Bioinformatics by Krane, Pearson Publications
5. Bioinformatics: Methods and Applications: Genomics, Proteomics and Drug Discovery by SC Rastogi, Prentice Hall of India
6. Fundamentals of Bioinformatics by by S. Harisha, L.K., International
7. Internet of Things by Jeeva Jose, Khanna Publishing

Block- II

Phylogeny and sequence databases

UNIT -3

Molecular Phylogeny

UNIT-4

Biological databases

Introduction

This is the first block on phylogeny and a sequence database consists of following two units.

Unit-3: This unit covers the introduction of phenotypic, phylogeny and molecular phylogeny. The basic discussion on molecular clocks, methods of phylogeny, statistical evaluation of the obtained phylogenetic is also discussed briefly.

Unit-5: This unit covers the Biological sequence databases. In this unit the classification of biological data base has briefly discussed. The different biological databases agencies viz NCBI and their functions are revealed in unite. The tools, databases and sequence submission to NCBI also discussed here.

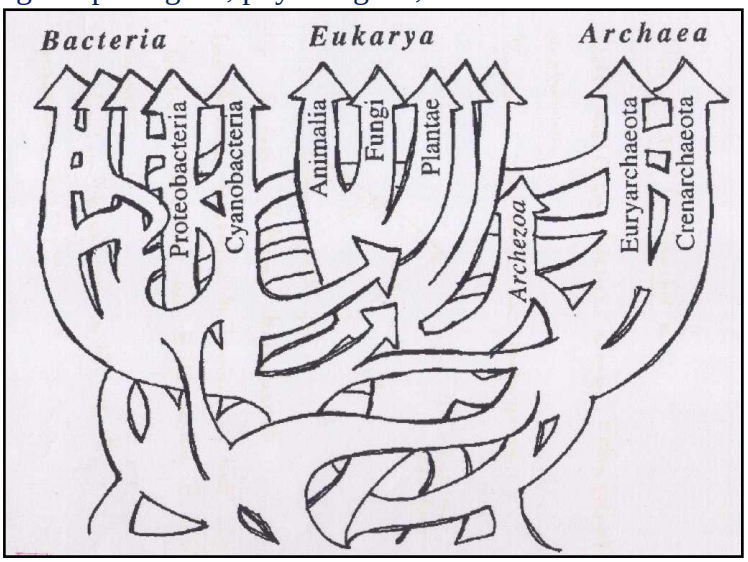
Unit-3: Molecular Phylogeny

Structure

3.1. Introduction
Objectives
3.2. Introduction to Molecular Phylogenetics
3.3. Molecular Evolution: Beyond Darwin
3.4. Phylogenetic Trees
3.5. Trees and Homology
3.6. Steps in Phylogenetic Analysis
3.6.1. Assemble and Align Datasets
3.6.2. Converting Alignment Data into a Phylogenetic Tree
3.6.3. Assessing the Accuracy of a Reconstructed Tree
3.7. Molecular clocks
3.8. Universal molecular clocks
3.9. Summary
3.10. Terminal questions
3.11. Suggested readings

3.1. Introduction to Phylogeny

According to modern evolutionary theory, all organisms on earth have descended from a common ancestor, which means that any set of species, extant or extinct, is related. This relationship is called a *phylogeny*. Term "Phylogeny" is derived from a combination of Greek words. Phylon stand for "tribe" or "clan" or "race" and genesis means "origin" or "source". The term can also be applied to the genealogy of genes derived from a common ancestral gene. Phylogeny is represented by phylogenetic trees, which graphically represent the evolutionary history related to the species of interest. Phylogenetics infers trees from observations about existing organisms using morphological, physiological, and molecular characteristics.



(Figure 3.1: Taken from Doolittle, Science 284, 1999)

3.2. Introduction to Molecular Phylogenetics

The similarity of biological functions and molecular mechanisms in living organisms strongly suggests that species descended from a common ancestor. Molecular phylogenetics uses the structure and function of molecules and how they change over long time to infer these evolutionary relationships. This branch of study emerged in the early 20th century but did not begin in earnest until the 1960s, with the advent of protein sequencing, PCR, electrophoresis, and other molecular biology techniques. Over the past 30 years, as computers have become more powerful and more generally accessible, and computer algorithms more sophisticated, researchers have been able to tackle the immensely complicated stochastic and probabilistic problems that define evolution at the molecular level more effectively. Within past decade, this field has been further reenergized and redefined as whole genome sequencing for complex organisms has become faster and less expensive. As mounds of genomic data becomes publically available, molecular phylogenetics is continuing to grow and find new applications.

The primary objective of molecular phylogenetic studies is to recover the order of evolutionary events and represent them in evolutionary trees that graphically depict relationships among species or genes over time. This is an extremely complex process, further complicated by the fact that there is no one right way to approach all phylogenetic problems. Phylogenetic data sets can consist of hundreds (even thousants) of different species, each of which may have varying mutation rates and patterns that influence evolutionary change.

Consequently, there are numerous different evolutionary models and stochastic methods available. The optimal methods for a phylogenetic analysis depend on the nature of the study and data used.

3.3. Molecular Evolution: Beyond Darwin

Evolution is a process by which the traits of a population change from one generation to another. In “the Origin of Species by Means of Natural Selection”, Darwin proposed that, given overwhelming evidence from his extensive comparative analysis of living specimens and fossils, all living organisms descended from a common ancestor. Darwin’s theory of evolution is based on three underlying principles: variation in traits exists among individuals within a population, these variations can be passed from one generation to the next via inheritance, and that some forms of inherited traits provide individuals a higher chance of survival and reproduction than others.

Although, Darwin developed his theory of evolution without any knowledge of the molecular basis of life, it has since been determined that evolution is actually a molecular process based on genetic information, encoded in DNA, RNA, and it proteins. At a molecular level, evolution is driven by the same types of mechanisms, Darwin observed it at the species level. One molecule undergoes diversification into many variations. One or more of those variants can be selected to be reproduced or amplified throughout in a population over many generations. Such variations at the molecular level can be caused by mutations, such as deletions, insertions, inversions, or substitutions at the nucleotide level, which inturn affect protein structure and biological function.

3.4. Phylogenetic Trees

Before exploring statistical and bioinformatics methods for estimating phylogenetic trees from molecular data, it is important to have a basic familiarity of the terms and elements common to these types of trees.

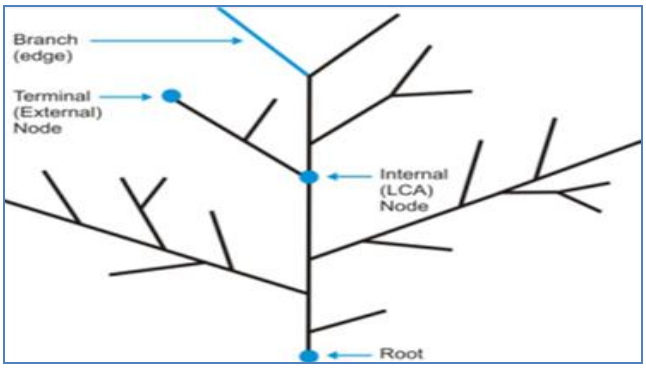


Figure 3.2: Basic Elements of a Phylogenetic Tree

Phylogenetic trees are composed of branches, also known as edges that connect and terminate at nodes. Branches and nodes can be internal or external (terminal). The terminal nodes at the tips of trees represent operational taxonomic units (OTUs). OTUs correspond to the molecular sequences or taxa (species) from which the tree was inferred. Internal nodes represent the last common ancestor (LCA) to all nodes that arise from that point. Trees can be made of a single gene from many taxa (a species tree) or multi-gene families (gene trees).

A tree is considered to be "rooted" if there is a particular node or outgroup (an external point of reference) from which all OTUs in the tree arises. The root is the oldest point in the tree and the common ancestor of all taxa in the analysis. In the absence of a known outgroup, the root can be placed in the middle of the tree or a rootless tree may be generated. Branches of a tree can be grouped together in different ways.

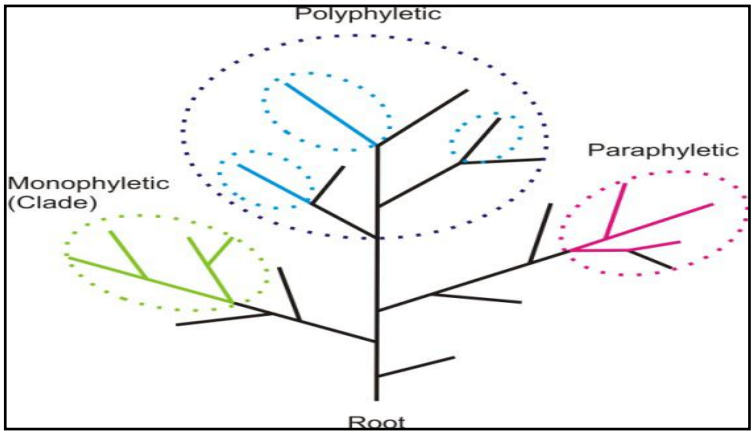


Figure 3.3: Groups and Associations of Taxonomical Units in Trees

A ***monophyletic* group** consists of an internal LCA node and all OTUs arising from it. All members within the group are derived from a common ancestor and have inherited a set of unique common traits. A ***paraphyletic* group** excludes some of its descendents (for examples all mammals, except the marsupialia taxa). Whereas a ***polyphyletic group*** is a collection of distantly related OTUs that are associated by a similar characteristic or phenotype, but are not directly descended from a common ancestor.

3.5. Trees and Homology

Evolution is shaped by homology, which refers to any similarity due to common ancestry. Similarly, phylogenetic trees are defined by homologous relationships. *Homologs* can be either paralogs or orthologs. *Paralogs* are homologous sequences separated by a gene duplication event. *Orthologs* are homologous sequences separated by a speciation event (when one species diverges into two).

Molecular phylogenetic trees are drawn so that branch length corresponds to amount of evolution (the percent difference in molecular sequences) between nodes.

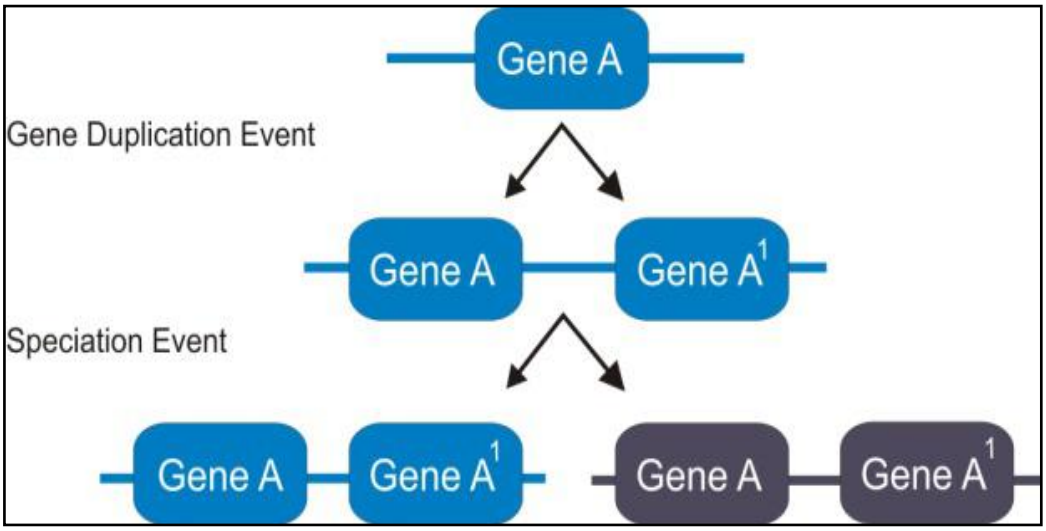


Figure 3.4: Understanding Paralogs and Orthologs

Paralogs are created by gene duplication events. Once a gene has been duplicated, all subsequent species in the phylogeny will inherit both copies of the gene, creating *orthologs*. Interestingly, evolutionary divergence of different species may result in many variations of a protein, all with similar structures and functions, but with very different amino acid sequences. Phylogenetic studies can trace the origin of such proteins to an ancestral protein family or gene.

3.6. Steps in Phylogenetic Analysis

Although the nature and scope of phylogenetic studies may vary significantly and require different datasets and computational methods, the basic steps in any phylogenetic analysis remain the same: assemble and align a dataset, build (estimate) phylogenetic trees from sequences using computational methods and stochastic models, and statistically test and assess the estimated trees.

3.6.1. Assemble and Align Datasets

The data used in reconstruction of a DNA-based phylogenetic tree are obtained by comparing nucleotide (basic unit of DNA)sequences. These comparisons are made by aligning the sequences so that nucleotide differences can be scored. This is the critical part of the entire enterprise because if the alignment is incorrect then the resulting tree will definitely not be the true tree. The first issue to consider is whether the sequences being aligned are homologous. If they are homologous then they must, by definition, be derived from a common ancestral sequence and so there is a sound basis for the phylogenetic study. If they are not homologous then they do not share a common ancestor. The phylogenetic analysis will find common ancestor because the methods used for tree reconstruction always produce a tree of some description, even if the data are completely erroneous, but the resulting tree will have no biological relevance. With some DNA sequences - for example, the β-globin genes of different vertebrates, there is no

difficulty in being sure that the sequences being compared are homologous, but this is not always the case, and one of the commonest errors that arises during phylogenetic analysis is the inadvertent inclusion of a non-homologous sequence.

Once it has been established that two DNA sequences are indeed homologous, the next step is to align the sequences so that homologous nucleotides can be compared. With some pairs of sequences this is a trivial exercise, but it is not so easy, if the sequences are relatively dissimilar and/or have diverged by the accumulation of insertions and deletions as well as point mutations. Insertions and deletions cannot be distinguished when pairs of sequences are compared so we refer to them as indels. Placing indels at their correct positions is often the most difficult part of sequence alignment.

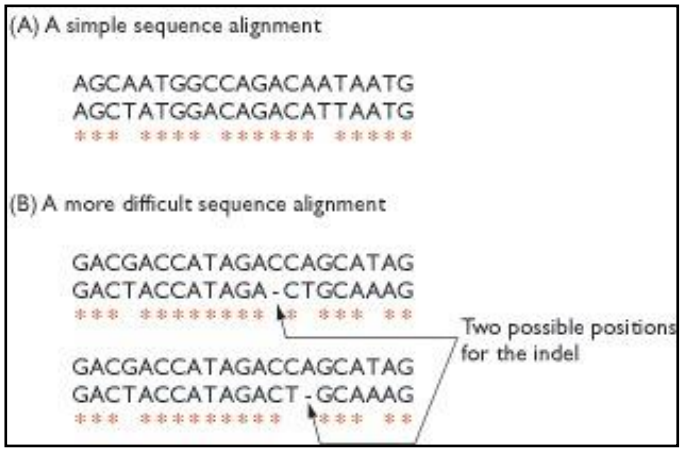


Figure 3.5: Sequence alignment: **(A)** Two sequences that have not diverged to any great extent, can be aligned easily by eye. **(B)** A more complicated alignment in which it is not possible to determine the correct position for an indel. If errors in indel placement are made in a multiple alignment then the tree reconstructed by phylogenetic analysis is unlikely to be correct. In this diagram, the red asterisks indicate nucleotides that are the same in both sequences indicated in Fig.3.5.

Some pairs of sequences can be aligned reliably by eye. For more complex pairs, alignment might be possible by the dot matrix method. The two sequences are written out on the x- and y-axes of a graph, and dots placed in the squares of the graph paper at positions corresponding to identical nucleotides in the two sequences. The alignment is indicated by a diagonal series of dots, broken by empty squares where the sequences have nucleotide differences, and shifting from one column to another at places where indels occur.

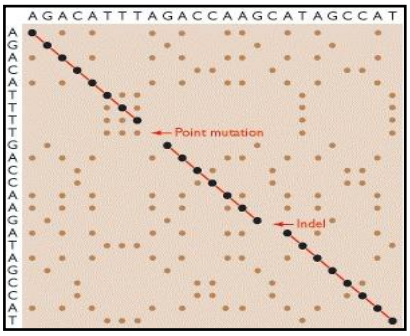


Figure 3.6 : The dot matrix technique for sequence alignment

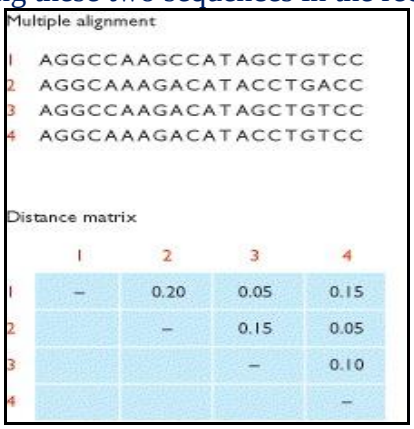
The correct alignment stands out because it forms a diagonal of continuous dots, broken at point mutations and shifting to different diagonal at indels. More rigorous mathematical approaches to sequence alignment have also been devised. The first of these is the **similarity approach** (Needleman and Wunsch, 1970), which aims to maximize the number of matched nucleotides - those that are identical in the two sequences. The complementary approach is the **distance method** (Waterman et al., 1976), in which the objective is to minimize the number of mismatches. Often the two procedures will identify the same alignment as being the best one. Usually the comparison involves more than just two sequences, meaning that a multiple alignment is required. This can rarely be done effectively with pen and paper so, as in all steps in a phylogenetic analysis, a computer program is used. For multiple alignments, Clustal is often the most popular choice (Jeanmougin et al., 1998). There are a bunch of tools available to visualize and annotate phylogenetic trees. Some of the most widely used software/tools are discussed below:

1. **MEGA:** MEGA is useful software in constructing phylogenies and visualizing them, and also for data conversion. It can easily convert alignment files to other formats such as nexus, paup, phylip, and fasta, and so on. The MEGA tree explorer is helpful in editing trees very easily, subtrees can also be selected and edited separately. Some tree image export options are also available. The input formats are newick, phylip, mega, and nexus. The phylogenetic tree can also be converted in newick format but it falls short on converting it into other formats such as phylip which is required in other analyses such as selection analysis.
2. **Dendroscope:** It is helpful in visualizing large trees and provides several options to export their graphics with a command line. Several different views are also available, trees can be easily re-rooted and node labels and branches can be easily formatted. It can export trees in newick and nexus format. Although, users will have to register themselves first to use this feature.
3. **FigTree:** It is actually designed to visualize trees that are produced by BEAST [4] program. Tip labels and node labels can be easily edited. It can easily export trees in nexus, newick, and JSON format with some graphics export options such as emf, pdf, sg, png, etc.
4. **Phylotree.js:** It is a javascript based library to visualize and annotate trees and offer some other customizations. It has a wide application in Datamonkey [6] comparative analyses. A user can upload trees using Phylotree.js where a user can easily select test and reference branches, and any changes can be mapped to their position on the corresponding structure. It is also good for comparison of trees with links between leaves known as a tanglegram, where crossings can represent evolutionary events. It also offers several export options and other *built-in* features.

3.6.2. Converting Alignment Data into a Phylogenetic Tree

Once the sequences have been aligned accurately, an attempt can be made to reconstruct the phylogenetic tree. Comparative tests have been run with artificial data, for which the true tree is known, but these have failed to identify any particular method as being better than any of the others (Felsenstein, 1988). The main distinction between the different tree-building methods is the way in which the multiple sequence alignment is converted into numerical data that can be analyzed mathematically in order to reconstruct the tree. The simplest approach is to convert the sequence information into a distance matrix, which is simply a table showing the evolutionary distances between all pairs of sequences in the dataset. The evolutionary distance is calculated

from the number of nucleotide differences between a pair of sequences and is used to establish the lengths of the branches connecting these two sequences in the reconstructed tree.


Multiple alignment

1 AGGCCAAGCCATAGCTGTCC
2 AGGCAAAGACATACCTGACC
3 AGGCCAAGACATAGCTGTCC
4 AGGCAAAGACATACCTGTCC

Distance matrix

	1	2	3	4
1	–	0.20	0.05	0.15
2		–	0.15	0.05
3			–	0.10
4				–

Figure 3.7: A simple distance matrix

The above matrix shows the evolutionary distance between each pair of sequences in the alignment. In this example the evolutionary distance is expressed as the number of nucleotide differences per nucleotide site for each sequence pair. For example, sequences 1 and 2 are 20 nucleotides in length and have four differences, corresponding to an evolutionary difference of 4/20 = 0.2. Note that this analysis assumes that there are no multiple substitutions (also called multiple hits). Multiple substitutions occur when a single site undergoes two or more changes (e.g. the ancestral sequence … ATGT … gives rise to two modern sequences: … AGGT … and … ACGT …). There is only one nucleotide difference between the two modern sequences, but there have been two nucleotide substitutions. If this multiple hit is not recognized then the evolutionary distance between the two modern sequences will be significantly underestimated. To avoid this problem, distance matrices for phylogenetic analysis are usually constructed using mathematical methods that include statistical devices for estimating the amount of multiple substitutions that has occurred.

The neighbour-joining method (Saitou and Nei, 1987) is a popular tree-building procedure that uses the distance matrix approach. To begin the reconstruction, it is initially assumed that there is just one internal node from which branches leading to all the DNA sequences radiate in a star-like pattern. This is virtually impossible in evolutionary terms but the pattern is just a starting point. Next, a pair of sequences is chosen at random, removed from the star, and attached to a second internal node, connected by a branch to the centre of the star. The distance matrix is then used to calculate the total branch length in this new 'tree'. The sequences are then returned to their original positions and another pair attached to the second internal node, and again the total branch length is calculated. This operation is repeated until all the possible pairs have been examined, enabling the combination that gives the tree with the shortest total branch length to be identified. This pair of sequences will be neighbours in the final tree; in the interim, they are combined into a single unit, creating a new star with one branch fewer than the original one. The whole process of pair selection and tree-length calculation is now repeated so that a second pair of neighbouring sequences is identified, and then repeated again so that a third pair is located, and so on. The result is a complete reconstructed tree.

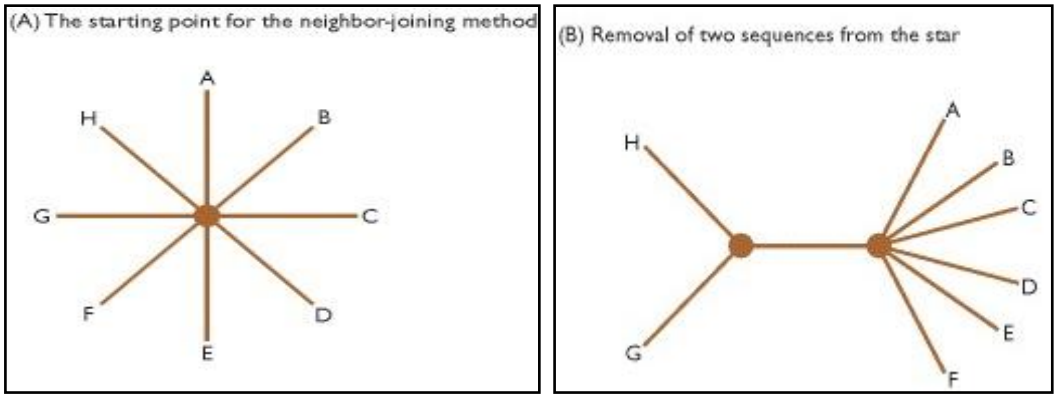


Figure 3.8: Manipulations carried out when using the neighbour-joining method for tree reconstruction

The advantage of the **neighbour-joining method** is that the data handling is relatively easy to carry out, largely because the information content of the multiple alignment has been reduced to its simplest form. The disadvantage is that some of the information is lost, in particular that pertaining to the identities of the ancestral and derived nucleotides (equivalent to ancestral and derived character states, defined in (figure 3.8) at each position in the multiple alignment. The **maximum parsimony method** (Fitch, 1977) takes account of this information, utilizing it to recreate the series of nucleotide changes that resulted in the pattern of variation revealed by the multiple alignments. The assumption, possibly erroneous, is that evolution follows the shortest possible route and that the correct phylogenetic tree is therefore the one that requires the minimum number of nucleotide changes to produce the observed differences between the sequences. Trees are therefore constructed at random and the number of nucleotide changes that they involve calculated until all possible topologies have been examined and the one requiring the smallest number of steps identified. This is presented as the most likely inferred tree.

The maximum parsimony method is more rigorous in its approach compared with the neighbour-joining method, but this increase in rigor inevitably extends the amount of data handling that is involved. This is a significant problem because the number of possible trees that must be scrutinized increases rapidly as more sequences are added to the dataset. With just five sequences there are only 15 possible uprooted trees, but for ten sequences there are 20,27,025 unrooted trees and for 50 sequences the number exceeds the number of atoms in the universe (Eernisse, 1998). Even with a high-speed computer it is not possible to check every one of these trees in a reasonable time, if at all, so often the maximum parsimony method is unable to carry out a comprehensive analysis. The same is true with many of the other more sophisticated methods for tree reconstruction.

3.6.3. Assessing the Accuracy of a Reconstructed Tree

The limitations to the methods used in phylogenetic reconstruction lead inevitably to questions about the veracity of the resulting trees. Statistical tests of the accuracy of a reconstructed tree have been devised (Hillis, 1997; Whelan et al., 2001) but these are necessarily complex because a tree is geometric rather than numeric and the accuracy of one part of the topology may be greater or lesser than the accuracy of the other parts.

The routine method for assigning confidence limits to different branch points within a tree is to carry out a bootstrap analysis. To do this we need a second multiple alignment that is

different from, but equivalent to, the real alignment. This new alignment is built up by taking columns, at random, from the real alignment. The new alignment therefore comprises sequences that are different from the original, but it has a similar pattern of variability. This means that when we use the new alignment in tree reconstruction we do not simply reproduce the original analysis, but we should obtain the same tree.

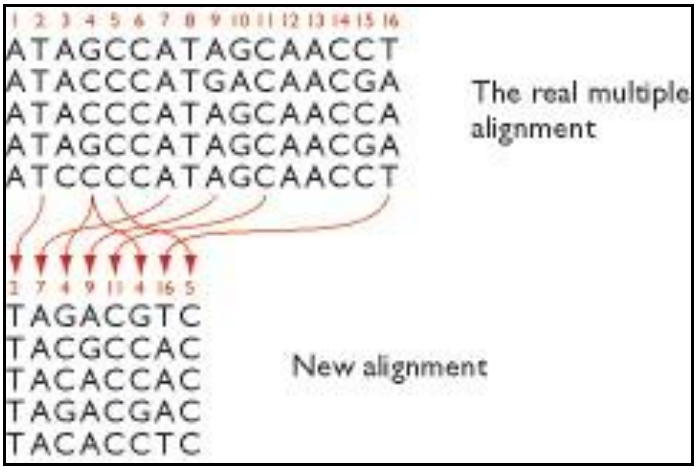


Figure 3.9: Constructing a new multiple alignment in order to bootstrap a phylogenetic tree: The new alignment is built up by taking columns at random from the real alignment

In practice, 1000 new alignments are created so 1000 replicate trees are reconstructed. A bootstrap value can then be assigned to each internal node in the original tree, this value being the number of times that the branch pattern seen at that node was reproduced in the replicate trees. If the bootstrap value is greater than 700/1000, then we can assign a reasonable degree of confidence to the topology at that particular internal node. Two common methods of Resampling are –

1. **Cross Validation**
2. **Bootstrapping**

1. **Cross Validation –**

Cross-Validation is used to estimate the test error associated with a model to evaluate its performance.

Validation set approach: This is the most basic approach. It simply involves randomly dividing the dataset into two parts: first training set and seconds a validation set or hold-out set. The model is fit on the training set and the fitted model is used to make predictions on the validation set.

- **Leave-one-out-cross-validation:** LOOCV is a better option than the validation set approach. Instead of splitting the entire dataset into two halves only one observation is used for validation and the rest is used to fit the model.
- **k-fold cross-validation –**This approach involves randomly dividing the set of observations into k folds of nearly equal size. The first fold is treated as a validation set and the model is fit on the remaining folds. The procedure is then repeated k times, where a different group each time is treated as the validation set.

2. **Bootstrapping –**Bootstrap is a powerful statistical tool used to quantify the uncertainty of a given model. However, the real power of bootstrap is that it could get applied to a wide range of models where the variability is hard to obtain or not output automatically.

3.7 Molecular clocks

Molecular clocks aimed to reconstruct evolutionary trees that show the relationships among species of interest. Internal nodes in the tree represent evolutionary divergence events. The timing of these events can be estimated using molecular clocks. A number of statistical methods are available for testing the molecular-clock hypothesis for a given set of DNA or protein sequences. When the molecular clock is rejected for a data set, one can use a statistical model to account for rate variation when estimating evolutionary timescales (Welch and Bromham, 2005). When estimating evolutionary timescales in a phylogenetic analysis, the molecular clock needs to be "calibrated." This can be done by assigning an absolute age to one or more nodes in the phylogeny, which can then act as a reference point for estimating the ages of the remaining nodes. Calibrations are often based on the fossil record, which can provide date estimates for the divergence events in the phylogeny. If a fossil taxon can be reliably assigned to one of the lineages in the tree, then the divergence of that lineage from its sister lineage must be older than the age of the fossil. Calibrations can also be based on geological events, including the separation of continents or the emergence of islands, if they are linked to evolutionary divergences. An example of this might involve two sister genera, one of which is endemic to the mainland and the other endemic to an offshore island that formed as a result of volcanic activity. The timing of the divergence between the two genera can be estimated by the age of the island which, in turn, can be estimated using radiometric dating. In some cases, previous genetic estimates of dates are employed as calibrations. These are known as "secondary" calibrations and are generally used when other calibrating information is unavailable.

Once the phylogenetic tree is calibrated by fixing or constraining the ages of one or more nodes in the phylogeny, the ages of the remaining nodes can be estimated using a molecular-clock analysis of the genetic data. This process is referred to as "molecular dating" or "divergence-time estimation." There is a large range of methods and models that have been developed for this purpose, implemented in various statistical frameworks. These methods are available in a range of computer programs, including most standard phylogenetic software. BEAST is a cross-platform program for Bayesian analysis of molecular sequences using MCMC. It is entirely orientated towards rooted, time-measured phylogenies inferred using strict or relaxed molecular clock models.

Strict clock

A strict clock model assumes that every branch in a phylogenetic tree evolves according to the same evolutionary rate. This is hence a 1-parameter model, the parameter of which represents the conversion rate between branch lengths and evolutionary time.

Fixed local clock

One of the first relaxations of a strict clock assumption consisted of assuming that one or more specific clades in the tree does not evolve according to this global rate. Instead, those clades/lineages are allowed to evolve according different evolutionary rates while rate constancy is assumed across the remainder of the tree. In BEAST, the 'Fixed local clock' option assumes that the user has created one or more taxon sets. BEAST then uses all of the taxon sets defined and assumes a change of evolutionary rate at their (time to) most recent common ancestor (MRCA). It's arguably to indicate in BEAUti that those taxon sets should be monophyletic.

Uncorrelated relaxed clock

Uncorrelated relaxed clocks allow each branch of a phylogenetic tree to have its own evolutionary rate (Drummond et al., 2006). These clocks are called 'uncorrelated' clocks because the evolutionary rate at one branch does not depend upon the rate at any of the neighboring

branches. This means that the evolutionary rate across branches can change abruptly, i.e. going from fast to slow or slow to fast suddenly, rather than needing to steadily increase or decrease over multiple adjoining branches.

Random local clock

Random local clocks permit an amount of variation in evolutionary rate across a tree which is more than the strict clock (which has no variation) but less than the relaxed clock (which has a different rate for each branch) (Drummond and Suchard, 2010). They work by proposing a series of local molecular clocks, each extending over a subregion of the full phylogeny. Each branch in a phylogeny (subtending a clade) is a possible location for a change of rate from one local clock to a new one. The number of changes may hence be zero (in which case the resulting clock is a strict clock), or it may be equal to the number of branches (in which case the resulting clock is a relaxed clock), or it may be somewhere in between. This is entirely determined by the data, as the MCMC chain samples over both the number of changes and their locations on the tree.

3.8 Universal molecular clocks

There is abundant evidence of evolutionary rate variation among species, dispelling any hope of a universal molecular clock across the tree of life. However, some researchers entertain the idea of homogeneous rates of mitochondrial evolution within certain groups of organisms, such as birds, mammals, and arthropods. This is a convenient assumption because it allows evolutionary rates to be applied in molecular-clock analyses when fossil or geological calibrations are otherwise unavailable.

A universal mitochondrial clock of 1 % per million years is often assumed for mammals. This rate was initially based on a mitochondrial study of primates and rodents (Brown et al., 1979), but it soon found support from estimates derived from other organisms (Wilson et al., 1985). Comprehensive analyses of mammalian DNA have demonstrated that there is substantial variation in rates among species, partly driven by differences in longevity (Nabholz et al., 2008; Welch et al., 2008).

The employment of "standard" mitochondrial clocks has often been criticized. This is because, the mitochondrial genome, is subject to differing degrees of natural selection among species, which can lead to heterogeneity in evolutionary rates. Similar reasoning applies to evolutionary rates in the nuclear genome. Given that rates can show substantial variation even among closely related species, the use of standard clocks has the potential to yield date estimates that are highly misleading.

3.9. Summary

- Phylogenetic, is the study of evolutionary relationships among nucleotide or protein sequences.
- There are many applications of phylogenetics, including forensics, pathogen surveillance, conservation and bioinformatics.
- There are several aspects of phylogenies that you need to understand in order to interpret your trees: topology, branch lengths, nodes and confidence.
- Careful interpretation is critical to understanding the biological meaning of phylogenies.
- The same phylogenetic tree can be visualised in many different ways.
- Evolutionary relationships can be unraveled by identifying the most recent common ancestor (MRCA) shared by species.
- EMBL-EBI has several resources and tools that are relevant to the field of phylogenetics including: Ensembl, ClustalW2 Phylogeny, Clustal Omega and Prank.

3.10. Terminal Questions

Q.1. Define Phylogeny and steps in Phylogenetic Analysis

Answer:----------

Q.2. Define dot matrix and distance matrix approaches

Answer:----------

Q.3. Explain Molecular Clocks.

Answer:----------

Q.4. Universal molecular clocks.

Answer:----------

3.11. Suggested readings

1. Bioinformatics: Principles and Applications by Zhumur Ghosh and Bibekanand Mallick, Oxford Press
2. Fundamental Concepts of Bioinformatics by Krane, Pearson Publications
3. Bioinformatics: Methods and Applications: Genomics, Proteomics and Drug Discovery by SC Rastogi, Prentice Hall of India
4. Fundamentals of Bioinformatics by by S. Harisha, L.K. International
5. Internet of Things by Jeeva Jose, Khanna Publishing
6. Bioinformatics: Sequence and Genome Analysis by Mount D., Cold Spring Harbor Laboratory Press, New York
7. Bioinformatics- a Practical Guide to the Analysis of Genes and Proteins by Baxevanis, A.D. and Francis Ouellellette, B.F., Wiley India Pvt Ltd.
8. Introduction to bioinformatics by Teresa K. Attwood, David J. Parry-Smith. Pearson Education

Unit-4: Biological Databases

Structure

4.1. Introduction
Objectives
4.1.1. Database
4.1.2. Biological Databases
4.2. National Canter for Biotechnology Information (NCBI)
4.3. NCBI's data-analytic software tools
4.3.1. Sequence submission to NCBI
4.4. Classification of Databases
4.4.1. Type 1
4.4.2. Type 2
4.4.2.1.Primary database
4.4.2.2.Secondary database
4.4.2.3.Composite Database
4.5. Primary Nucleotide Sequence Repository
4.5.1. GenBank
4.5.2. European Molecular Biology Laboratory(EMBL)
4.5.3. DNA Database of Japan (DDBJ)
4.5.3.1.DDBJ Flat File Format
4.6. Protein Sequence Databases
4.6.1. PIR
4.6.2. Swin Port
4.6.3. TrEML
4.6.4. UniParc
4.6.5. UniMES
4.7. Derived or Secondary databases of nucleotide sequences
4.7.1.ACeDB
4.7.2.Omniome
4.7.3.Gobae
4.8. Derived or Secondary databases of amino acid sequences
4.8.1.MHCPeP
4.8.2.CLuSTr
4.8.3.COGS
4.9. Derived or Secondary databases of amino acid sequences – Patterns and Signature
4.9.1. Prosite
4.9.2.Prints
4.9.3.ProDom
4.9.4.Pfam
4.10. Structure database
4.10.1.The primary structure database
4.10.2.Derived or Secondary databases of bimolecular structures
4.11. Basic Local Alignment Search Tool (BLAST)
4.12. Summary

4.13. Terminal questions
4.14. Suggested readings

4.1. Introduction

Biological databases are storehouses of biological information. It is a computerized archive used to store, organize and ease retrieval of sequence data. They can be defined as libraries containing data collected from scientific experiments, published literature and computational analysis. It provides users an interface to facilitate easy and efficient recording, storing, analyzing and retrieval of biological data through application of computer software.

O**bjectives:**

- T o determine the protein sequence database
- To define the biological database and their significance
- To briefly discuss nucleotide and amino acid sequence

4.1.1. Database

It is a collection of structured, searchable, periodically updated and cross-referenced data. It also includes associated tools and software necessary for database access/query, database updating, database information insertion and database information deletion. The data stored in these databases is persistent and organized. Database Management System (DBMS) is a software application that deals with the user, other applications, and the database itself in order to perform analysis and capture data in a systematic manner.

4.1.2. Biological Databases

Bioinformatics databases or biological databases are storehouses of biological information. It is a computerized archive used to store, organize and ease retrieval of sequence data. They can be defined as libraries containing data collected from scientific experiments, published literature and computational analysis. It provides users an interface to facilitate easy and efficient recording, storing, analyzing and retrieval of biological data through application of computer software. Biological data comes in several different formats like text, sequence data, structure, links, etc. and these needs to be taken into account while creating the databases. There are

various criteria on the basis of which the databases can be classified. On the basis of structure, databases can be classified as text files, flat file, object oriented and relational databases. On the basis of information, they can be classified as general and specialized databases.

It is well acknowledged that scientific information is being generated at an exponentially increasing rate. One recent molecular biology endeavour is of particular public interest: The Human Genome Project (HGP) sequenced and mapped the complete human genome.

4.2. National Centre for Biotechnology Information (NCBI)

The National Centre for Biotechnology Information (NCBI) is a multi-disciplinary research group that serves as a resource for molecular biology information. It was formed in 1988 as a complement to the activities of the National Institutes of Health (NIH) and the National Library of Medicine (NLM) https://www.ncbi.nlm.nih.gov/. Its facilities are located in Bethesda, Maryland, USA. Initially, NCBI's creation was intended to aid in understanding the molecular mechanisms that affect human health and disease with the following goals: to create and maintain public databases, develop software to analyze genomic data, and to conduct research in computational biology. In time, and through widespread use of the Internet, NCBI became increasingly aware of the role of pure biological research.

NCBI began offering services as well:

- Developing new methods to deal with the volume and complexity of data researching into methods that can analyze the structure and function of macromolecules.
- Creating computerized systems for storing and analysing data about molecular biology.
- Providing access to analysis and computing tools (which facilitate the use of databases and software) to researchers and the public.

In the process of database development, NCBI formed database standards such as database nomenclature that are also used by other non-NCBI databases. One NCBI database is GenBank, the nucleic acid sequence database that contains sequence information from over 200 000 different organisms. GenBank is probably the most popular database in use, and actually predates NCBI.

Database retrieval systems offered by NCBI include Locus Link, the Taxonomy Browser, and Gene. Locus Link offers descriptive information about genes and is based on curated data. The Taxonomy Browser offers information on lineage of organisms that have corresponding sequences in GenBank. Taxonomic and phylogenetic trees can also be viewed through the Taxonomy Browser. Gene is poised to become the successor of Locus link, with greater scope, and integration into NCBI's Entrez system. Most commonly, they are classified on the basis of the type of data stored in primary, secondary and composite databases.

4.3. NCBI's data-analytic software tools

Analytic software tools allow for the conducting of scientific experiments, the rejection of hypotheses, and the drawing of conclusions concerning molecular biology. Many data-analytic tools exist at NCBI, UBiC and other places on the web. Due to the overwhelming number of techniques available for analysing data, and to the relatively new analytic software, conditions for the use of any of these tools may be confusing. Mistakes due to unfamiliarity with the tools remain quite common. Other tools have gained widespread use simply by being easy to use. One such tool is the Basic Local Alignment Search Tool (BLAST), which is most commonly used to analyze nucleic acid sequences from GenBank.

4.3.1. Sequence submission to NCBI

- **BankIt** , a www-based submission tool with wizards to guide the submission process
- **tbl2asn**, a command-line program, automates the creation of sequence records for submission to GenBank using many of the same functions as Sequin. It is used primarily for submission of complete genomes and large batches of sequences and is available by FTP for use on MAC, PC and Unix platforms.
- **Submission Portal**, a unified system for multiple submission types. Currently only ribosomal RNA (rRNA), rRNA-ITS, Influenza or Norovirus sequences can be submitted with the GenBank component of this tool. Genome and Transcriptome Assemblies can be submitted through the Genomes and TSA portals, respectively.
- **Sequin**, NCBI's stand-alone submission tool to propagate features from one record to another is available by FTP for use on for MAC, PC, and UNIX platforms. NCBI is phasing out support of the Sequin submission tool.

Records with simple annotation may be submitted by **BankIt** or **Sequin**, while records with complicated annotation may be more easily submitted via **Sequin**.

Group of nucleotide sequences for the *same* gene or locus: Includes

- population studies (sequences for a single organism)
- phylogenetic studies (sequences for multiple organisms)
- environmental samples (such as cultured or uncultured bacteria or metagenomic samples)

4.4. Classification of databases

There are various criteria on the basis of which the databases can be classified. On the basis of structure, databases can be classified as text files, flat file, object oriented and relational databases. On the basis of information, they can be classified as general and specialised databases. Commonly, they are classified on the basis of the type of data stored in primary, secondary and composite databases.

4.4.1. Type 1

Databases can be classified on the basis of structure as Abstract Syntax Notation (ASN.1), Flat files, Object oriented databases, Relational databases, and XML.

1. **ASN.1:** This format comprises of a syntax and description of how a particular data type can be represented physically in a data stream or sequential file. This format has been adopted by NCBI for the representation of sequential data. It is one of the major file formats in GenBank.
2. **Flat Files:** This implementation is based on only one table, which incorporates the complete data i.e. all the attributes for each variable. Each row of the table specifies a different record. Specified delimiters are used to differentiate among records. Maintenance of data stored is a major drawback of this type of databases. Integration of two or more databases is difficult due to redundancy in data and variation in the format used.
3. **Object Oriented Databases:** Object oriented databases can handle complex data types and can be easily integrated with Object Oriented Programming Languages (OOPL). They can be defined as a collection of objects. Objects represent an instance of an entity and comprise attributes as well as methods.

4. **Relational Databases:** Relational database systems can be defined as a collection of relations or tables. In a relational database, the data is organized in the form of a table where each row contains a record and each column specific an attribute of the record. The ordering of tuples attributes or values within a tuple do not make any impact on the relation. The data is subjected to various constraints for validation.
5. **XML:** XML can be defined as an advanced flat file format. It provides greater support for representation of complex nested data structures. It contains data definitions and supports new definitions and tags upon requirement. The major advantages of this type are fast accessibility, reliability, and scalability.

4.4.2. Type 2

The databases can be classified into three categories on the basis of the information stored. They are primary, secondary and composite databases.

4.4.2.1. Primary Databases

Primary databases contain data that is derived experimentally. They usually store information related to the sequences or structures of biological components. They can be further divided into protein or nucleotide databases which can be further divided as sequence or structure databases. The most commonly used primary databases are: DNA Data Bank of Japan (DDBJ), European Molecular Biology Laboratory (EMBL) Nucleotide Sequence Database, GenBank, and Protein Data Bank (PDB).

4.4.2.2. Secondary Databases

A secondary database contains derived information from the primary database. It contains information like the conserved sequence, signature sequence and active site residues of the protein families arrived by multiple sequence alignment of a set of related proteins. A secondary structure database contains entries of the PDB in an organized way. These contain entries that are classified according to their structure like all alpha proteins, all beta proteins, etc. These also contain information on conserved secondary structure motifs of a particular protein. Some of the secondary database created and hosted by various researchers at their individual laboratories includes SCOP, developed at Cambridge University; CATH developed at University College of London, PROSITE of Swiss Institute of Bioinformatics, eMOTIF at Stanford.

4.4.2.3. Composite Database

Composite database amalgamates a variety of different primary database sources, which obviates the need to search multiple resources. Different composite database use different primary database and different criteria in their search algorithm. Various options for search have also been incorporated in the composite database. NCBI hosts these nucleotide and protein databases in their large high available redundant array of computer servers provides free access to the various persons involved in research. This also has link to OMIM (Online Mendelian Inheritance in Man) which contains information about the proteins involved in genetic diseases.

4.5. Primary Nucleotide Sequence Repository

4.5.1. GenBank

It is located in the USA. NCBI since 1992 has provided access to GenBank DNA sequence database through NCBI gateway server and hence is accessible freely. The three nucleotide sequence databases GenBank, EMBL and DDBJ coordinate among themselves so that all three of them are updated with the latest findings. The three letter code for GenBank divisions is depicted in Table 4.1.

Table 4.1: Three letter code for the divisions in GenBank

Division	**Sequence Subset**
PRI	Primate
ROD	Rodent
MAM	Mammalian
VRT	Vertebrate
INV	Invertebrate
PLN	Plant
BCT	Bacteria
RNA	Structure RNA
VRL	Viral

BHG	Bacteriophage
SYN	Synthetic
UNA	Unnotated
EST	Expressed Sequence Tag
PAT	Patent
STS	Sequence Tagged Site
GSS	Genome Survey Sequence
HTG	High Throughput Genome Sequence

A detailed structure of a nucleotide sequence file format in this database includes the following:

1. **Locus:** This can be defined as a title given by GenBank itself to name the sequence entry. It includes the following:
    a) *Locus Name:* Similar to accession number for the sequence.
    b) *Sequence Length:* Tells the number of bases existing in the sequence.
    c) *Molecule-Type:* Identifies the type of nucleic acid sequence. The various types are mRNA (which is present as cDNA), rRNA, snRNA, and DNA.
    d) *GB Division:* Postulates class of the data according to classification criteria of GenBank.
    e) *Modification Date:* The date on which the record was modified.
2. **Definition:** This denotes the name of the nucleotide sequence.
3. **Accession:** This covers accession number, accession version, and GI number. Accession number can be defined as the unique identifier associated with eachnucleotide sequence present in the database. If more than one record is created for a particular sequence then it will have the same accession number but all records will have different versions associated with that accession number.
4. **Keyword:** Defined words that were used to index the entries.
5. **The Source:** This describes organism from which sequences have been obtained.

The accepted common name is mentioned first and then the scientific name is mentioned. In the end, the taxonomic lineage according to GenBank is specified.

6. **The Citation:** Includes the journal from which with the sequence was derived as initially the sequences were obtained only from published literature.

7. **Features:** These consist of the information derived from the sequence such as biological source, coding region, exon, intron, promoters, alternate splice patterns, mutations, etc.

8. **Sequence:** Contains the following:

   a) Count of presence of each nucleotide in the sequence,

   b) Whole nucleotide sequence,

   c) Beginning of sequence is determined by keyword "ORIGIN", and

   d) End is marked as "\\".

4.5.2. European Molecular Biology Laboratory (EMBL)

The nucleotide sequence database from European Bioinformatics Institute (EBI) includes sequences both form direct author submission and genome sequence group and from scientific literature and patent applications (https://www.embl.org/). The database is developed in collaboration with GenBank and DDBJ. In January, 1998, EMBL contains a more than million entries representing around 15,500 species. But with model system predominantly *Homo sapiens*, *sea elegans*, *cerevisiae*, *musculus*, *A. thaliana*, together contain more than 50% of the resource. The information can be retrieved from EMBL using SRS (Sequence Retrieval System).

The various aims of the EBI organization are as follows:

- To provide freely available data and bioinformatics services to all facets of the scientific community.
- To contribute to the advancement of biology through basic investigator-driven research.
- To provide advanced bioinformatics training to scientists at all levels.
- To help disseminate cutting-edge technologies to industry.
- To coordinate biological data provision throughout Europe (www.ebi.ac.uk/).

4.5.3. DNA Database of Japan (DDBJ)

The (DDBJ) belongs to National Institute of Genetics (NIG) in Japan https://www.ddbj.nig.ac.jp/index-e.html. DDBJ is the only nucleotide sequence data bank currently present in Asia. Although DDBJ essentially has Japanese researchers as contributors but it also accepts the data from researchers of other countries. It is an associate of the International Nucleotide Sequence Database Collaboration (INSDC). The major driving force

behind DDBJ operations is the advancement of the quality of INSD as the nucleotide sequence accounts organism development more directly than other biological constituents. Key tasks of DDBJ Center are as follows:

1. Construction and operation of INSDC which offers nucleotide and amino acid sequence data along with the patent request.
2. Provides searching and analysis of biological data.
3. Training course and journal.

4.5.3.1. DDBJ Flat File Format

The data submitted in DDBJ is managed and retrieved according to the DDBJ format (flat file). The flat file includes the sequence and the information of who submitted the data, references, source organisms, and information about the feature, etc.

4.6. Protein Sequence Databases

PIR-PSD or protein information resource – protein sequence database, at the NBRF (National Biomedical Research Foundation, USA), and SWISS-PROT at the SBI (Swiss Biotechnology Institute), Switzerland are protein sequence databases.

4.6.1. Protein Information Resource (PIR)

The PIR-PSD is a collaborative endeavour between the PIR, the MIPS (Munich Information Centre for Protein Sequences, Germany) and the JIPID (Japan International Protein Information Database, Japan) https://proteininformationresource.org/pirwww/index.shtml. The PIR-PSD is now a comprehensive, non-redundant, expertly annotated, object relational DBMS. A unique characteristic of the PIR-PSD is its classification of protein sequences based on the super family concept. Sequence in PIR-PSD is also classified based on homology domain and sequence motifs. Homology domains may correspond to evolutionary building blocks, while sequence motifs represent functional sites or conserved regions. The classification approach allows a more complete understanding of sequence function structure relationship.

4.6.2. SWISS-PROT

It is a well-known and extensively used protein database (http://www.expasy.ch/sprot). Like the PIR-PSD, this curated proteins sequence database also provides a high level of annotation. The data in each entry can considered separately as core data and annotation (Table 4.2). The core data consists of the sequences entered in common single letter amino acid code,

and the related references and bibliography. The taxonomy of the organism from which the sequence was obtained also forms part of this core information. The annotation contains information on the function or functions of the protein, post-translational modification such as phosphorylation, acetylation, etc., functional and structural domains and sites, such as calcium binding regions, ATP-binding sites, zinc fingers, etc., known secondary structural features as for examples alpha helix, beta sheet, etc., the quaternary structure of the protein, similarities to other protein if any, and diseases that may rise due to different authors publishing different sequences for the same protein, or due to mutations in different strains of an described as part of the annotation.

Table 4.2: Code in SWISS-PROT database

Code	Expansion	Remarks
ID	Identification	Occurs at the beginning of the entry. Contains a unique name for the entry, plus information on the status of the entry. If it has been checked and conforms to SWISS-PROT standards, it is called STANDARD.
AC	Accession numbers	This is a stable way of identifying the entry. The name may change but not the AC. If the line has more than one number, it means that the entry was constituted by merging other entries.
DT	Date	There are three dates corresponding to the creation date of the entry and modification dates of the sequence and the annotation respectively
DE	Description	Lines that start with the identifier contain general description about the sequence.
GN	Gene name	The name of the gene (or genes) that codes for the protein
OS, OG,OC	Organism name, Organelle,	The name and taxonomy of the organism, and information regarding the organelle containing the gene

	Organism classification	e.g. mitochondria or chloroplast, etc.
RN, RP,RX,RA RT,RL	Reference number, Position, comments, cross-reference, authors, title and location.	Bibliographic reference to the sequence. This includes information (following the code RP) on the extent of work carried out b the authors.
CC	Comments	These are free text comments that provide any relevant information pertaining to the entry.
DR	Database cross-reference	This line gives cross-references to other databases where information regarding this entry is also found. As for example to structural information for the protein in the PDB.
KW	Keywords	This line gives a list of keywords that can be used in indexes. Search programs very often simply go through such indices to identify required information
FT	Features Table	These lines describe regions or sites of interest in the sequence, e.g. post-transitional modifications, binding sites, enzyme active sites and local secondary structures
SQ	Sequence Header	This line indicates the beginning of the sequence data and gives a brief summary of its contents.

Both PIR-PSD and SWISS-PROT have software that enables the user to easily search through the database to obtain only the required information. SWISS-PROT has the SRS or the sequence retrieval system that searches also through the other relevant databases on the site, such as TrEMBL.

4.6.3. TrEMBL (Translated EMBL)

It a computer-annotated protein sequence database that is released as a supplement to SWISS-PROT https://www.uniprot.org/statistics/TrEMBL. It contains the translation of all coding sequences present in the EMBL Nucleotide database, which have not been fully annotated. Since the sequence data were being generated at a very high rate with respect to the ability of the SwissProt in performing the annotation hence TrEMBL Nucleotide Sequence Data Library was created so as to facilitate computer-annotated data for those proteins which could not be entered in Swiss-Prot. The records present in this database are automatically annotated and have been analyzed computationally with high quality. Automatic processing and insertion of the translation of annotated coding sequences present in the three major nucleotide sequence database are done through this database. It also takes into account the sequences from PDB, Ensembl, RefSeq and CCDS.

4.6.4. UniParc

UniProt Archive (UniParc) stores proteins sequences from publicly available protein sequence database in a non-redundant manner and it is updated on a regular basis https://www.uniprot.org/help/uniparc. Since proteins may exist in several databases and there are high chances that a single sequence is present multiple times in the same database. Hence to avoid redundancy of data, each unique sequence is presented only once in this database. The identical sequences are merged even if they belong to different species. A unique identifier called UPI is given to each sequence which enables the identification of a same protein from various source databases. The protein sequences present in this database are without any annotation. Database cross-references are provided in order to facilitate the retrieval of more detailed information from the source databases.

4.6.5. UniMES:

UniProt Metagenomic and Environmental Sequences (UniMES) database has been created for environmental and metagenomic data (Consortium, 2010) https://www.uniprot.org/help/unimes. In order to improve the original data through more analysis, proteins that have already been predicted are merged using InterPro. UniMES is the source containing data from Global Ocean Sampling Expedition (GOS). UniProt Reference Clusters or UniProt Knowledgebase does not contain data of environmental sample of this database.

4.7. Derived or Secondary databases of nucleotide sequences

Many of the secondary databases are simply sub-collection of sequences culled from one or the other of the primary databases such as GenBank or EMBL. There is also usually a great deal of value addition in terms of annotation, software, presentation of the information and the cross-references. There are other secondary databases that do not present sequences at all, but only information gathered from sequences databases.

4.7.1. ACeDB:

More than a database, this is a database management system that was originally developed for the *C. elegans* (a nematode worm) genome project https://www.sanger.ac.uk/science/tools/acedb. It is a repository of not only the sequence, but also the genetic map as well as phenotypic information about the C. elegans nematode worm.

4.7.2. Omniome

The comprehensive Microbial Resource maintained by TIGR (The Institute for Genomic Research) at http://www.tigr.org allows access to a database called Omniome. This contains all the focus on one organism. Omniome has not only the sequence and annotation of each of the completed genomes, but also has associated information about the organisms (such as taxon and gram stain pattern), the structure and composition of their DNA molecules, and many other attributes of the protein sequences predicted from the DNA sequences. The presence of all microbial genomes in a single database facilitated meaningful multi-genome searches and analysis, for instance, alignment of entire genomes, and comparison of the physical proper of proteins and genes from different genomes etc.

4.7.3. GOBASE

A database of the genomes of mitochondria and other such organelles is available at the Organelle Genome Database at the University of Montreal, Canada, and is called GOBASE (http://megasun.bch.umontreal.ca/gobase).

4.8. Derived or Secondary databases of amino acid sequences

4.8.1. MHCPep

It is a database comprising over 13000 peptide sequences known to bind the Major Histocompatibilty Complex of the immune system (http://wehih.wehi.edu.au/mhcpep/). Each entry in the database contains not only the peptide sequence, which may be 8 to 10 amino acid

long, but in addition has information on the specific MHC molecules to which it binds, the experimental method used to assay the peptide, the degree of activity and the binding affinity observed , the source protein that, when broken down gave rise to this peptide along with other, the positions along the peptide where it anchors on the MHC molecules and references and cross links to other information.

4.8.2. CluSTr

The **CluSTr** (Cluster of SWISS-PROT and TrEMBL proteins at http://ebi.ac.uk.clustr) database offers an automatic classification of the entries in the SWISS-PROT and TrEMBL databases into groups of related proteins. The clustering is based on the analysis of all pair wise comparisons between protein sequences.

4.8.3. COGS

Similar to CluSTRr is the COGS or Cluster of Orthologous Groups of database that is accessible at htp://ncbi.nlm.nih.gov/COG. An orthologous group of proteins is one in which the members are related to each other by evolutionary descent. Such orthology may not be just from one protein to another, and then to another and so on down the line. It may involve one-to-many and many-to-many evolutionary relationships, and hence termed as ‘groups’. COGS is thus a database of phylogenetic relationships.

4.9. Derived or Secondary databases of amino acid sequences – Patterns and Signature

A set of databases collects together patterns found in protein sequences rather than the complete sequences. The patterns are identified with particular functional and/or structural domains in the protein, for example, ATP binding site or the recognition site of a particular substrate. The patterns are usually obtained by first aligning a multitude of sequences through multiple sequence alignment techniques. This is followed by further processing by different methods, depending on the particular database.

4.9.1. PROSITE

PROSITE is one such pattern database, which is accessible at http://www.expasy.ch/prosite. The protein motif and pattern are encoded as “regular expressions”. The information corresponding to each entry in PROSITE is of the two forms – the patterns and the related descriptive text. The regular expression is placed in a format reminiscent of the SWISS-PROT

entries, with a two letter identifier at beginning of the each line specifying the type of information the line contains. The expression itself is placed on line identified by “PA”. The entry also contains references and links to all the proteins sequences that contains that pattern. The related descriptive text is placed in a documentation file with the accession number making the connection to the expression data.

4.9.2. PRINTS

In the PRINTS database (http://www.bioinfo.man.ac.uk/dbbrowser/PRINTS), the protein sequence patterns are stored as ‘fingerprints’. A finger print is a set of motifs or patterns rather than a single one. The information contained in the PRINT entry may be divided into three sections. In addition to entry name, accession number and number of motifs, the first section contains cross links to other databases that have more information about the characterized family. The second section provides a table showing how many of the motifs that make up the finger print occurs in the how many of the sequences in that family. The last section of the entry contains the actual finger prints that are stored as multiply aligned set of sequences, the alignment being made without gaps. There is therefore one set of aligned sequences for each motif.

4.9.3. ProDom

The ProDom protein domain database (http://www.toulouse.inrs.fr/prodom.html) is a compilation of homologous domains that have been automatically identified sequence comparison and clustering methods using the program PSI-BLAST. No identification of patterns is made. The focus is here to look for complete and self-contained structural domains and the search methods includes signals for such features. A graphical user interface allows easy interactive analysis of structural and therefore functional homology relationships among protein sequences.

4.9.4. Pfam

A database called Pfam contains the profiles used using Hidden markov models. HMMs build the model of the pattern as a series of match, substitute, insert or delete states, with scores assigned for alignment to go from one state to another http://www.sanger.ac.uk/Software/Pfam. Each family or pattern defined in the Pfam consists of the four elements. The first is the annotation, which has the information on the source to make the entry, the method used and

some numbers that serve as figures of merit. The second is the seed alignment that is used to bootstrap the rest of the sequences into the multiple sequence alignments and then the family. The third is the HMM profile. The fourth element is complete alignment of all the sequences identified in that family.

4.10. Structure Databases

Structure databases like sequence databases comes in two varieties, primary and secondary. Strictly speaking there is only one database that stores primary structural data of biological molecules, namely the PDB. In the context of this database, term macromolecule stretches to cover three orders of magnitude of molecular weight from 1000 Daltons to 1000 kilo Daltons Small biological and organic molecules have their structures stored in another primary structure database the CSD, which is also widely used in biological studies. This contains the three dimensional structure of drugs, inhibitors and fragments or monomers of the macromolecule.

4.10.1. The primary structure database

a) Protein Databank (PDB)

Inspite of the name, PDB archive the three-dimensional structures of not only proteins but also all biologically important molecules, such as nucleic acid fragments, RNA molecules, large peptides such as antibiotic gramicidin and complexes of protein and nucleic acids. The database holds data derived from mainly three sources. Structure determined by X-ray crystallography form the large majority of the entries. This is followed by structures arrived at by NMR experiments. The data in the PDB is organized as flat files, one to a structure, which usually means that each file contain one molecule, or one molecular complex.

b) The Cambridge Structural Database (CSD)

It was originally a project of the University of Cambridge, which is set up to collect together the published three-dimensional structure of small organic molecules. This excludes proteins and medium sized nucleic acid fragments, but small peptides such as neuropeptides, and monomer and dimmers of nucleic acid finds a place in the CSD. Currently CSD holds crystal structures information for about 2.5 lakhs organic and metal organic compounds. All these crystal structures have been obtained using X-ray or neuron diffraction technique. For each entry in the CSD there are three distinct types of information stored. These are categorized as bibliographic

information, chemical connectivity information and the three- dimensional coordinates. The annotation data field incorporates all of the bibliographic material for the particular entry and summarized the structural and experimental information for the crystal structure.

4.10.2. Derived or Secondary databases of bimolecular structures

a) Nucleic acid databases (NDB):

It is a relational database of three-dimensional structures containing nucleic acid. This encompasses DNA and RNA fragments, including those with unusual chemistry such as NDB, and collections of patterns and motifs such as SCOP, PALI etc. The structures are the same as those found in the PDB and therefore the NDB qualifies to be called a specialized sub collection. However a substantial amount, and, unlike the PDB, the NDB is much more than just a collection of files. The structure of DNA has been classified into A, B and Z polymorphic forms, based on the information specified by authors. Other classes include RNA structures, unusual structures and protein-nucleic acid complexes. These classes of structures are arranged in the form of an ATLAS of Nucleic Acid Containing Structures, which can be browse and searched to obtain the structure or structures required. Each entry in the atlas has information on the sequence, crystallisation condition, references and details of the parameters and the figures of the merit used in structure solution. The entry has links not only to the coordinated but also to automatically generated graphical views of the molecule. NDB also has also have archives of structural geometries calculated for all the structures or for a subset of them. And finally, the database stores average geometrical parameters for nucleic acids, obtained by statistical analysis of the structures. These parameters are widely used in computer simulations of nucleic acids and their interactions. The NDB may be accessed at http://ndbserve.rutgers.edu/NDB/.

b) The SCOP database (Structural Classification of Proteins:

It is a manual classification of protein structures in a hierarchical scheme with many levels (http://scop.mrc-lmb.cam.ac.uk/scop/). The principal classes are the family, the super family and the fold. SCOP is a searchable and browsable database. In other words, one may either enter SCOP at the top of the hierarchy or examine different folds and families as one pleases, or one may supply a keyword or a phrase to be search the database and retrieve corresponding entries. Once a structure, or a set of structures, has been selected, they may be obtained or viewed wither

as graphical images. Each entry also has other annotation regarding function, etc., and links to other databases, including other structural classification such as CATH.

c) CATH:

CATH stands for Class, Architecture, Topology and Homologous super family. The name reflects the classification hierarchy used in the database. The structures chosen for classification are a subset of PDB, consisting of those that have been determined to a high degree of accuracy.

4.11. Basic Local Alignment Search Tool (BLAST)

The Basic Local Alignment Search Tool (BLAST) is a powerful way to carry out sequence similarity searching https://blast.ncbi.nlm.nih.gov/Blast.cgi. First, as a bioinformatician you have an obligation to correctly use the terms: **homology** and **similarity**. Many scientists use these terms interchangeably when they actually mean quite different things. Similarity is a measure of how related two sequences are, whereas homology is a conclusion about the evolutionary relatedness of two sequences based on an assessment of their similarity. Two sequences can be said to be 68% similar but these same two sequences are either homologous or not. There is no degree to homology, two sequences are either related or not.

BLAST is, of course, the tool that many scientists use to infer relationships of homology based on the degree of sequence similarity between two sequences. BLAST is a local alignment tool which means that it searches for regions of similarity instead of trying to align the entire length of the sequences. Although **global alignment methods** result in the most mathematically optimized alignments of full length sequences, they may miss local regions of sequence similarity. Often, **local alignment methods** can detect small regions of similarity resulting in a more biologically significant alignment. In addition, global alignment methods are computationally intensive and slow. On the other hand, local alignment tools, like BLAST, are based on computationally efficient search algorithms which break the large problem of finding similar sequences down into smaller pieces, making the method faster.

4.11.1. Understanding the BLAST Algorithm and BLAST Statistics

As with any sequence similarity searching method, it is important to understand how the method works and what measures or statistics are presented to aid in your evaluation of the results. By

understanding the BLAST algorithm and a few key BLAST statistics, you will be better able to interpret BLAST results.

Scoring Matrices:

The measure of similarity between two sequences is captured by a scoring scheme in BLAST which is based on scoring matrices. Scoring matrices are empirical weighting schemes that are used in comparing sequences and capture information about residue conservation, residue frequency, and evolutionary models. In BLAST, substitution matrices are used for amino acid alignments whereas nucleotide matrices are compared using identity matrices. These matrices are "look-up" tables in which each possible residue is given a score reflecting the probability that it is related to the corresponding residue in the query. The two most commonly used substitution matrices are the BLOSUM and PAM scoring matrices.

The **PAM (point accepted mutation)** scoring matrices are based on global alignments of closely related proteins. The PAM1 matrix is calculated by looking at the amino acid substitutions that occur in proteins with no more than 1% divergence (1 change per 100 amino acids). In an effort to model evolutionary changes, the other PAM matrices are extrapolated from PAM1 by matrix multiplication.

The **BLOSUM (blocks substitution matrices)** are based on local alignments where the BLOSUM62 matrix is calculated from comparisons of sequences with a certain threshold. This process of scanning a database with small sequence fragments is far faster than scanning a database with a large sequence.

4.11.2. A Brief Description of the BLAST Algorithm

The BLAST algorithm finds regions of local alignments by breaking the query sequence down into smaller chunks of sequence called **words.** These words are then indexed by the computer along with information about where each is found in the intact sequence.

The BLAST algorithm then starts by seeding the search with this small subset of letters from the query sequence. These words from your query sequence are then used to scan the database for matches above a certain threshold. This process of scanning a database with small sequence fragments is far faster than scanning a database with a large sequence.

Once the initial word hit has been found, the BLAST algorithm attempts to extend the match in the immediate sequence neighbourhood. Extension proceeds and the cumulative score is calculated using the scoring matrices discussed above. As long as positive matches and conservative substitutions outweigh the negative scores for gaps and mismatches, the cumulative score will increase. When the cumulative score starts to drops off, the BLAST algorithm measures the rate of decay and, once past a certain point, stops trying to extend the alignment. The result is an extended sequence alignment that was initially seeded by a word hit.

This is called an **HSP**, or **high-scoring segment pair**. All HSPs that have a cumulative score above a certain threshold are reported in BLAST reports. Because the BLAST algorithm carries out these searches using all possible query words, it is possible that more than one HSP may be found for any given pair of sequences.

4.11.3. BLAST Statistics

After an HSP is identified, it is important to determine whether this match is significant or not. Two BLAST statistics, the **score (S)** and the **E-value (E)** are particularly helpful in making this interpretation. First, the **score (S)** is a good measure of the quality of an alignment because it is calculated as the sum of substitution and gap scores for each aligned residue. Recall that substitution scores are given by look-up tables (PAM, BLOSUM) whereas gap scores are assigned empirically. Second, the **E-value (E),** or expectation value is a good measure of the significance of the alignment. The E-value is the number of different alignments, with scores equivalent to or better than S, that are expected to occur in a database search by chance. The lower the E-value, the more significant the alignment result.

BLAST is actually a family of programs (all included in the blastall executable). These include:

1. **Nucleotide-nucleotide BLAST (blastn):** This program, given a DNA query, returns the most similar DNA sequences from the DNA database that the user specifies.
2. **Protein-protein BLAST (blastp):** This program, given a protein query, returns the most similar protein sequences from the protein database that the user specifies.
3. **Position-Specific Iterative BLAST (PSI-BLAST):** This program is used to find distant relatives of a protein. First, a list of all closely related proteins is created. These proteins are combined into a general "profile" sequence, which summarises significant features present

in these sequences. A query against the protein database is then run using this profile, and a larger group of proteins is found. This larger group is used to construct another profile, and the process is repeated.

By including related proteins in the search, PSI-BLAST is much more sensitive in picking up distant evolutionary relationships than a standard protein-protein BLAST.

4. **Nucleotide 6-frame translation-protein (blastx):** This program compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database.
5. **Nucleotide 6-frame translation-nucleotide 6-frame translation (tblastx):** This program is the slowest of the BLAST family. It translates the query nucleotide sequence in all six possible frames and compares it against the six-frame translations of a nucleotide sequence database. The purpose of tblastx is to find very distant relationships between nucleotide sequences.
6. **Protein-nucleotide 6-frame translation (tblastn):** This program compares a protein query against the all six reading frames of a nucleotide sequence database.
7. **Large numbers of query sequences (megablast):** When comparing large numbers of input sequences via the command-line BLAST, "megablast" is much faster than running BLAST multiple times. It concatenates many input sequences together to form a large sequence before searching the BLAST database, then post-analyze the search results to glean individual alignments and statistical values.

4.11.4. Uses of BLAST:

BLAST can be used for several purposes. These include identifying species, locating domains, establishing phylogeny, DNA mapping, and comparison.

- **Identifying Species:** With the use of BLAST, you can possibly correctly identify a species and/or find homologous species. This can be useful, for example, when you are working with a DNA sequence from an unknown species.
- **Locating Domains**: When working with a protein sequence you can input it into BLAST, to locate known domains within the sequence of interest.
- **Establishing Phylogeny:** Using the results received through BLAST you can create a phylogenetic tree. It should be noted that phylogenies based on BLAST alone are less

reliable than other purpose-built computational phylogenetic methods, so should only be relied upon for "first pass" phylogenetic analyses.

- **DNA Mapping**: When working with a known species, and looking to sequence a gene at an unknown location, BLAST can compare the chromosomal position of the sequence of interest, to relevant sequences in the database(s).

Comparison: When working with genes, BLAST can locate common genes in two related species, and can be used to map annotations from one organism to another.

4.12. Summary: As biology has increasingly turned into a data rich science, the need for storing and communicating large datasets has grown tremendously. The obvious examples are the nucleotide sequences, the protein sequences, and the 3D structural data produced by X-ray crystallography and NMR. Biological databases are an important tool in assisting scientists to understand and explain a host of biological phenomena from the structure of biomolecules and their interaction, to the whole metabolism of organisms and to understanding the evolution of species. This knowledge helps facilitate the fight against diseases, assists in the development of medications and in discovering basic relationships amongst species in the history of life.

4.13. Terminal Questions:

Q.1. What are primary databases?

Answer:----------

Q.2. Explain BLAST. Types of BLAST

Answer:----------

Q.3. Write expansion

Answer:----------
---

a) EMBL

b) DDBJ

Q.4. Discus importance of biological databases in bioinformatics.

Answer:----------

Q.5. What is meant by secondary database? What are the major secondary databases?

Answer:----------

Q.6. Define PAM and BLOSSUM.

Answer:----------

4.14. Suggested Readings

1. Bioinformatics: Principles and Applications by Zhumur Ghosh and Bibekanand Mallick, Oxford Press
2. Fundamental Concepts of Bioinformatics by Krane, Pearson Publications
3. Bioinformatics: Methods and Applications: Genomics, Proteomics and Drug Discovery by SC Rastogi, Prentice Hall of India
4. Fundamentals of Bioinformatics by by S. Harisha, L.K. International
5. Internet of Things by Jeeva Jose, Khanna Publishing
6. Bioinformatics: Sequence and Genome Analysis by Mount D., Cold Spring Harbor Laboratory Press, New York
7. Bioinformatics- a Practical Guide to the Analysis of Genes and Proteins by Baxevanis, A.D. and Francis Ouellellette, B.F., Wiley India Pvt Ltd.
8. Introduction to bioinformatics by Teresa K. Attwood, David J. Parry-Smith. Pearson Education.

Block- III

Protein database, Simulation and drug designing

UNIT -5

Protein database

UNIT-6

Molecular Simulation

Introduction

This is the first block on protein database, simulation and drug designing consists of following two units.

Unit-3: This unit covers the Blast, PSO- blast. The different database viz. nucleotide and protein and gene expression are briefly discussed in this unit. Molecular Simulation and drug designing is also discussed briefly.

Unit-5: This unit covers the Protein 3D structure and classification database. The protein database bank, harnessing data from PDB, data deposition tools, PDB Data are briefly discussed. The RCSB, PDB structural genomics information portal, retrieval of structural database from MMDB, converted domain database (CDD) reveal the basic information of database.

Unit 5: Protein Database

5.1. Introduction
Objectives
5.2. Introduction of Proteins and Some Important Facts about Structures
Objectives
5.2.1. Protein Classification
5.3. Protein Three-Dimensional (3d) Molecular Structure
5.3.1. Coordinates, Sequences and Chemical Graphs
5.3.2. Atoms, Bonds and Completeness
5.4. Protein Databases
5.4.1. Protein Structure Classification Databases
5.5. PDB Files
5.5.1. Sequences from Structure Records
5.5.2. Validating PDB Sequences
5.6. Harnessing Data from PDB
5.6.1. Guessing the 3-D Structure of your Protein
5.7. Protein Sequence Databases
5.7.1. RefSeq
5.7.2. UniProt
5.7.3. D Gel Databases
5.7.4. Chemistry Databases
5.7.5. Enzyme and Pathway Databases
5.7.5.1. MetaCyc and BioCyc
5.7.5.2. BRENDA
5.7.6. Reactome
5.7.7. Family and Domain Databases
5.7.7.1. InterPro
5.7.7.2. Pfam
5.7.7.3. PIRSF
5.7.7.4. PROSITE
5.8. Structural Database
5.8.1. Searching a structural Database
5.9. Sequence Databases: Primary, Secondary and Other Databases
5.9.1. Primary Databases
5.9.2. Secondary Databases Nucleic Acids and Proteins

5.10 Nucleic Acid Secondary Databases

5.10.1 Gene Expression Databases: Expression Atlas

5.10.2 Genome Annotation Databases

5.10.2.1 Ensembl

5.10.2.2 Entrez Gene

5.10.2.3 UCSC

5.11 Organism Specific Databases

5.11.1 FlyBase

5.11.2 MGD

5.12 Polymorphism and Mutation Databases: dbSNP

5.13 Summary

5.14 Terminal question

5.15 Further readings

5.1 Introduction

Use of high-throughput technologies to study molecular biology systems in the past decades has revolutionized biological and biomedical research, allowing researchers to systematically study the genomes of organisms (Genomics), the set of RNA molecules (Transcriptomics), and the set of proteins including their structures and functions (Proteomics). Since proteins occupy a middle ground molecularly between gene and transcript and many higher levels of molecular and cellular structure and organization, and most physiological and pathological processes are manifested at the protein level, biological and biomedical scientists are increasingly interested in applying high-throughput proteomics techniques to achieve a better understanding of basic molecular biology and disease processes. The richness of proteomics data allows researchers to ask complex biological questions and gain new scientific insights. To support data-driven hypothesis generation and biological knowledge discovery, many protein-related bioinformatics databases, query facilities, and data analysis software tools have been developed to organize and provide biological annotations for proteins to support sequence, structural, functional and evolutionary analyses in the context of pathway, network and systems biology. With the recent extraordinary advances in genome sciences and Next-Generation Sequencing (NGS) technologies that have uncovered rich genomic information in a huge number of organisms, new protein bioinformatics databases are also being introduced and many existing databases have been enhanced. As more and more genomes are sequenced, the protein sequences archived in databases have increased dramatically in recent years (see Figure 1 for an example). This poses new challenges for computational

biologists in building new infrastructure to support protein science research in the age of Big Data.

Objectives

- Describe how protein structures are determined.
- Describe some of the relevant protein databases including PDB and associated file formats and file viewers.
- List the resources available for protein structure prediction
- Define the criteria for assessing and refining a predicted protein structure

5.2 Introduction of Proteins and Some Important Facts about Structures

The polypeptide chain is first assembled on the ribosome using the codon sequence on mRNA as a template, as illustrated in Figure 5.1. The resulting linear chain forms secondary structures through the formation of hydrogen bonds between amino acids in the chain. Through further interactions among amino acid side groups, these secondary structures then fold into a three-dimensional structure. Chaperone proteins and membranes may assist with this process. For the protein to have biological activity, processing of the protein by cleavage or chemical modification may also be necessary. Therefore, protein structure is largely specified by amino acid sequence, but how one set of interactions of the many possible occurs is not yet fully understood.

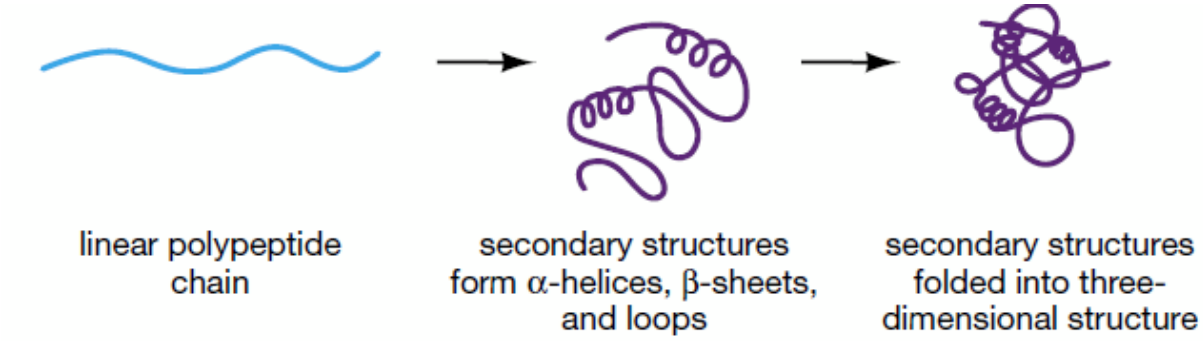


Fig. 5.1: Is describing how a linear protein molecule or chains of amino acids forms a three dimensional protein structure.

Some protein sequences have distinct amino acid motifs that always form a characteristic structure. Prediction of these structures from sequence is quite achievable

using presently available methods and information. For most proteins, however, the accuracy of secondary structure prediction is approximately 70–75%. Methods for matching sequence to three-dimensional structure have been formulated, but they are not yet very reliable.

However, great forward strides have been made, and there is a very active community of structural biochemists and bioinformaticists working on improvements. The need for such an effort is revealed by the rapid increases in the number of protein sequences and structures.

As of June 2000, more than 12,500 protein structures had been deposited in the Brookhaven Protein Data Bank (PDB), and 86,500 protein sequence entries were in the SwissProt protein sequence database, a ratio of approximately 1 structure to 7 sequences. The number of protein sequences can be expected to increase dramatically as more sequences are produced by research laboratories and the genome sequencing projects.

It has first been estimated that there are approximately 1,000 protein families composed of members that share detectable sequence similarity Thus, as new protein sequences are obtained, they will be found to be similar to other sequences already in the databases and can be expected to share structural features with these proteins. Whether this low number represents physical restraints in folding the polypeptide chain into a three-dimensional structure or merely the selection of certain classes of three-dimensional structure by evolution has yet to be discovered. Understanding these relationships is fundamentally important because this information can greatly assist with structural predictions. As discussed below, information from amino acid substitutions at a particular sequence position as obtained from a multiple sequence alignment has been found to increase significantly the prediction of secondary structures from protein sequences. A second major advance in protein structure analysis has been the revelation that proteins adopt a limited number of three-dimensional configurations.

The imagery of protein and nucleic acid structures has become a common feature of biochemistry textbooks and research articles. This imagery can be beautiful and intriguing enough to blind us to the experimental details an image represents—the underlying

biophysical methods and the effort of hard-working X-ray crystallographers and nuclear magnetic resonance (NMR) spectroscopists. The data stored in structure database records represents a practical summary of the experimental data. It is, however, not the data gathered directly by instruments, nor is it a simple mathematical transformation of that data. Each structure database record carries assumptions and biases that change as the state of the art in structure determination advances. Nevertheless, each biomolecular structure is a hard-won piece of crucial information and provides potentially critical information regarding the function of any given protein sequence.

5.2.1 Protein classification

Proteins may be classified according to both structural and sequence similarity. For structural classification, the sizes and spatial arrangements of secondary structures described in the above section are compared in known three-dimensional structures.

Terms Used for Classifying Protein Structures and Sequences

- **Active site** is a localized combination of amino acid side groups within the tertiary (three-dimensional) or quaternary (protein subunit) structure that can interact with a chemically specific substrate and that provides the protein with biological activity. Proteins of very different amino acid sequences may fold into a structure that produces the same active site.
- **Architecture** describes the relative orientations of secondary structures in a three dimensional structure without regard to whether or not they share a similar loop structure. In contrast, a fold is a type of architecture that also has a conserved loop structure. Architecture is a classification term used by the CATH database (http://www.biochem.ucl.ac.uk/bsm/cath/).
- **Blocks** is a term used to describe a conserved amino acid sequence pattern in a family of proteins. The pattern includes a series of possible matches at each position in the represented sequences, but there are not any inserted or deleted positions in the pattern or in the sequences. By way of contrast, sequence profiles are a type of scoring matrix that represents a similar set of patterns that includes insertions and deletions.

❖ **Class** is a term used to classify protein domains according to their secondary structural content and organization. Four classes were originally recognized and several others have been added in the SCOP database described below.

❖ **Core** is the portion of a folded protein molecule that comprises the hydrophobic interior of α helices and β sheets. The compact structure brings together side groups of amino acids into close enough proximity so that they can interact. When comparing protein structures, as in the SCOP database, core refers to the region common to most of the structures that share a common fold or that are in the same super family.

❖ **Domain** refers to a segment of a polypeptide chain that can fold into a three-dimensional structure irrespective of the presence of other segments of the chain. The separate domains of a given protein may interact extensively or may be joined only by a length of polypeptide chain. A protein with several domains may use these domains for functional interactions with different molecules.

❖ **Family** is a group of proteins of similar biochemical function that are more than 50% identical when aligned. This same cutoff is still used by the Protein Information Resource (PIR). A protein family comprises proteins with the same function in different organisms (orthologous sequences) but may also include proteins in the same organism (paralogous sequences) derived from gene duplication and rearrangements.

❖ **Fold** is a term with similar meaning to structural motif, but in general refers to a somewhat larger combination of secondary structural units in the same configuration. Thus, proteins sharing the same fold have the same combination of secondary structures that are connected by similar loops.

❖ **Motif** refers to a conserved pattern of amino acids that is found in two or more proteins. In the Prosite catalog, a motif is an amino acid pattern that is found in a group of proteins that have a similar biochemical activity, and that often is near the active site of the protein.

5.3 Protein Three-Dimensional (3D) Molecular Structure

Let us begin with a mental exercise in recording the three-dimensional data of a biopolymer. Consider how we might record, on paper, all the details and dimensions of a three-dimensional ball-and-stick model of a protein like myoglobin (Figure 5.2). One way to begin is with the sequence, which can be obtained by tracing out the backbone of the three-dimensional model. Beginning from the NH2-terminus, we identify each amino acid side chain by comparing the atomic structure of each residue with the chemical structure of the 20 common amino acids, possibly guided by an illustration of amino acid structures from a textbook.

Once the sequence has been written down, we proceed with making a two dimensional sketch of the biopolymer with all its atoms, element symbols, and bonds, possibly taking up several pieces of paper. The same must be done for the heme ligand, which is an important functional part of the myoglobin molecule. After drawing its chemical structure on paper, we might record the three-dimensional data by measuring the distance of each atom in the model starting from some origin point, along some orthogonal axis system. This would provide the x-, y-, and z-axis distances to each atomic ‘‘ball’’ in the ball-and-stick structure.

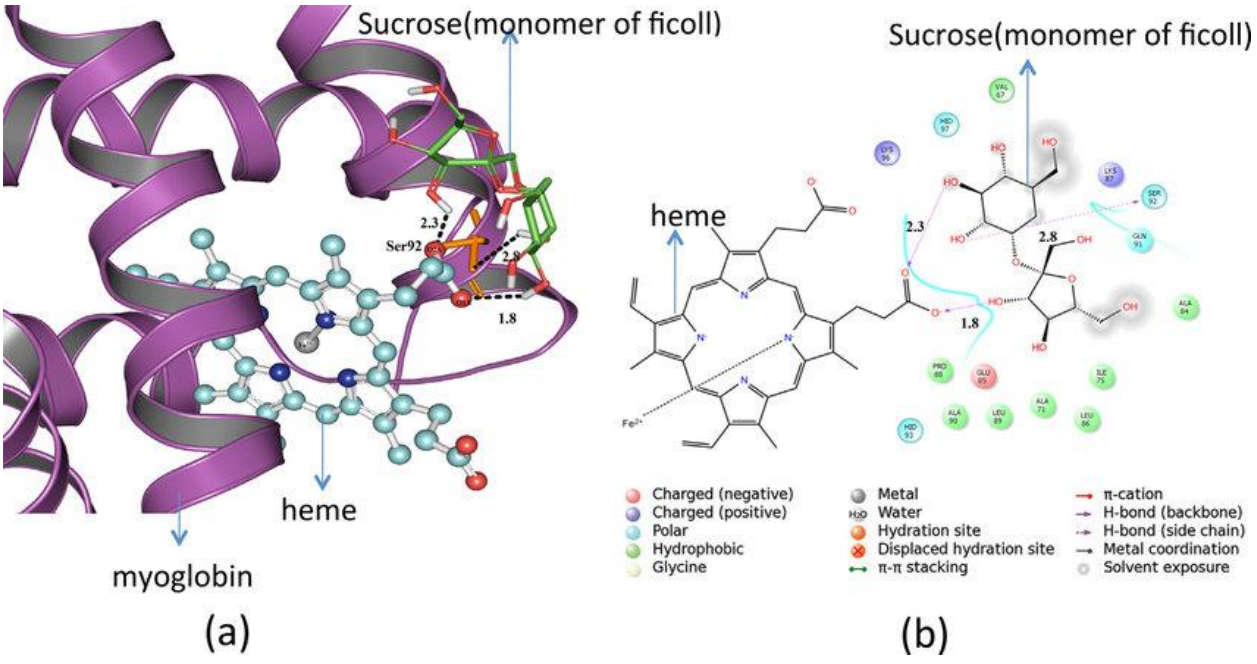


Figure 5.2: Showing the myogoblin a) with heme side chain b) with different metals association (https://www.researchgate.net/figure/Docking-study-of-myoglobin-with-ficoll-a-The-docked-complex-of-myoglobin-cartoon_fig5_319151931).

The next step is to come up with a bookkeeping scheme to keep all the (x, y, z) coordinate information connected to the identity of each atom. The easiest approach may be to write the (x, y, z) value as a coordinate triple on the same pieces of paper used for the two-dimensional sketch of the biopolymer, right next to each atom. This associates the (x, y, z) value with the atom it is attached to. This mental exercise helps to conceptualize what a three-dimensional structure database record ought to contain. There are two things that have been recorded here: the chemical structure and the locations of the individual atoms in space. This is an adequate ''human-readable'' record of the structure, but one probably would not expect a computer to digest it easily. The computer needs clear encoding of the associations of atoms, bonds, coordinates, residues, and molecules, so that one may construct software that can read the data in an unambiguous manner. Here is where the real exercise in structural bioinformatics begins.

5.3.1 Coordinates, Sequences, and Chemical Graphs

The most obvious data in a typical three-dimensional structure record, regardless of the file format in use, is the coordinate data, the locations in space of the atoms of a molecule. These data are represented by (x, y, z) triples, distances along each axis to some arbitrary origin in space (Figure 5.3). The coordinate data for each atom is attached to a list of labeling information in the structure record: which element, residue, and molecule each point in space belongs to. For the standard biopolymers (DNA, RNA, and proteins), this labeling information can be derived starting with the raw sequence. Implicit in each sequence is considerable chemical data. We can infer the complete chemical connectivity of the biopolymer molecule directly from a sequence, including all its atoms and bonds, and we could make a sketch, just like the one described earlier, from sequence information alone. We refer to this ''sketch'' of the molecule as the chemical graph component of a three-dimensional structure. Every time a sequence is presented in this book or elsewhere, remember that it can encode a fairly complete description of the chemistry of that molecule.

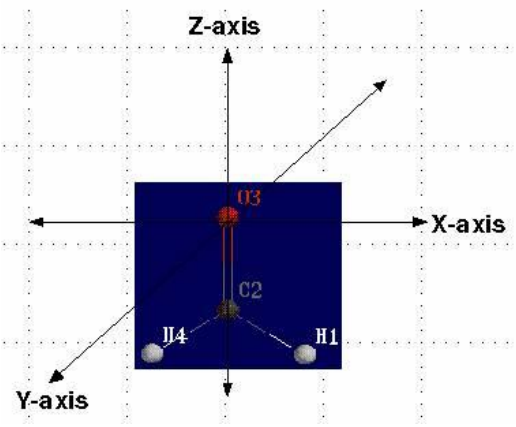


Figure 5.3: An explanation how both are arranged in different axis in methyl group (http://biosiva.50webs.org/genomics.htm).

When we sketch all the underlying atoms and bonds representing a sequence, we may defer to a textbook showing the chemical structures of each residue, lest we forget a methyl group or two. Likewise, computers could build up a sketch like a representation of the chemical graph of a structure in memory using a residue dictionary, which contains a table of the atom types and bond information for each of the common amino acid and nucleic acid building blocks. What sequence is unable to encode is information about posttranslational modifications. For example, in the structure databases, a phosphorylated tyrosine residue is indicated as ‘‘X’’ in the one letter code—essentially an unknown! Any residue that has had an alteration to its standard chemical graph will, unfortunately, be indicated as X in the one-letter encoding of sequence.

5.3.2 Atoms, Bonds, and Completeness

Molecular graphics visualization software performs an elaborate ‘‘connect-the-dots’’ process to make the wonderful pictures of protein structure we see in textbooks of biomolecular structure, like the structure for insulin (3INS). The connections used are, of course, the chemical bonds between all the atoms. In current use, three-dimensional molecular structure database records employ two different ‘‘minimalist’’ approaches regarding the storage of bond data.

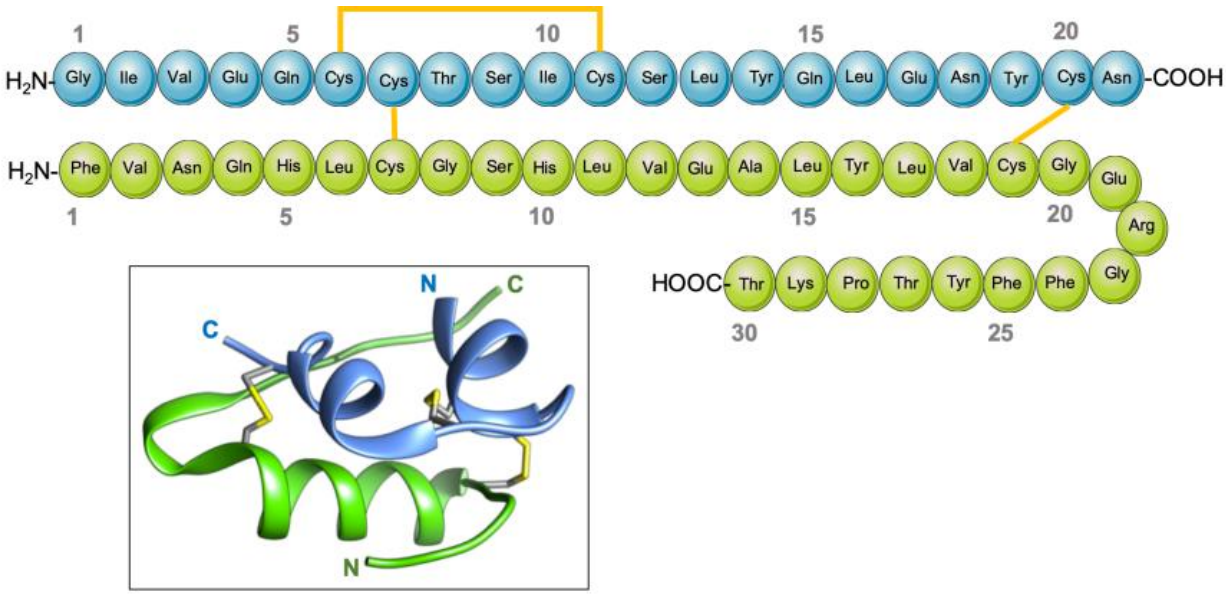


Figure 5.4: Linear change of amino acids and its three dimensional configuration (https://pdb101.rcsb.org/global-health/diabetes-mellitus/drugs/insulin/insulin).

The original approach to recording atoms and bonds is something we shall call the chemistry rules approach. The rules are the observable physical rules of chemistry, such as, ‘‘the average length of a stable C—C bond is about 1.5 angstroms.’’ Applying these rules to derive the bonds means that any two coordinate locations in space that are 1.5 A° apart and are tagged as carbon atoms always form a single bond.

5.4 Protein Databases

There are many publically available databases that are currently being used for different structural characterization of protein. All the data that have been deposits in these types of databases are being generated through high throughput screening methods which include methods or experimentation of molecular biology. Some the keen renowned data bases are given below:

Table 5.1: The major protein databases used for various structure analysis.

Name of resource	Resources available	Internet address
Protein data bank (PDB) at	atomic coordinates of structures as	http://www.rcsb.org/pdb

the State University of New Jersey	PDB files, models, viewers, links to many other Web sites for structural analysis and classification	
National Center for Biotechnology Information Structure Group	Molecular Modelling Database (MMDB), Vector Alignment Search Tool (VAST) for structural comparisons, viewers, threader software	http://www.ncbi.nlm.nih.gov/Structure/
Structural Classification of Proteins at Cambridge University	SCOP database of structural relationships among known protein structures classified by super family, family, and fold	http://scop.mrc-lmb.cam.ac.uk/scop
Swiss Institute of Bioinformatics, Geneva	basic types of protein analysis databases, the Swiss-Model resource for prediction of protein models, Swiss-PdbViewer	http://www.expasy.ch/

5.4.1 Protein Structure Classification Databases

The following databases are accessible on the Web and provide up-to-date structural comparisons for the proteins currently in the Brookhaven PDB and access to the sequences of these proteins. The methods used to classify the protein structures in these databases vary from manual examination of structures to fully automatic computer algorithms. Hence, although one can expect to find roughly the same groupings in each database, there will be some structural relationships that are only identified by one of these methods. Each database has useful information that may be lacking in the others.

1. **The SCOP database:**

The SCOP (Structural Classification of Proteins) database based on expert definition of structural similarities, is located at http://scop.mrc-lmb.cam.ac.uk/scop/. Following

classification by class, SCOP additionally classifies protein structures by a number of hierarchical levels to reflect both evolutionary and structural relationships; namely family, superfamily, and fold.

2. The CATH database:

The CATH (classification by Class, Architecture, Topology, and Homology) protein structure database resides at University College, London (http://www.biochem.ucl.ac.uk/bsm/cath/). Proteins are classified first intohierarchical levels by class, similar to the SCOP classification except that α/β and α + β proteins are considered to be in one class. Instead of a fourth class for α + β proteins, the fourth class of CATH comprises proteins with few secondary structures. Following class, proteins are classified by architecture, fold, superfamily, and family.

3. The FSSP database:

The FSSP (**F**old classification based on **S**tructure-**S**tructure **A**lignment of proteins) is based on a structural alignment of all pair-wise combinations of the proteins in the Brookhaven structural database by the structural alignment program DALI (http://www2.embl-ebi.ac.uk/dali/fssp/fssp.html). PDB has a number of redundant structures of proteins whose sequences and structures are 25% or more identical. A subset of representative structures in PDB without these redundant entries was first produced by aligning all of the PDB structures with DALI. Each protein in the subset was then subdivided into individual domains. These domains were then aligned structurally with DALI to identify the common folds. Redundant folds were again eliminated, and a set of representative folds was chosen. From 8320 PDB entries, 947 representative structures, 1484 domains, and 540 structurally distinct fold types were identified in 1997. These fold types represent a unique configuration of secondary structural elements in the domains. For example, one fold might be composed of helix-strand-helix-6 strands joined by loops in a particular configuration.

4. MMDB (Molecular Modelling Database):

Proteins of known structure in the Brookhaven PDB have been categorized into structurally related groups in MMDB by the VAST (Vector Alignment Search Tool)

structural alignment program. VAST aligns three-dimensional structures based on a search for similar arrangements of secondary structural elements. This method provides a method for rapidly identifying PDB structures that are statistically out of the ordinary. MMDB has been further incorporated into the ENTREZ sequence and reference database at http://www.ncbi.nlm.nih.gov/Entrez. Accordingly, it is possible to perform a simultaneous search for similar sequences and structures, designated neighbors, at the ENTREZ Web site. Structural neighbors within MMDB are based on detailed residue-by-residue alignments.

5. The SARF database:

The SARF (spatial arrangement of backbone fragments) database at http://www-lmmb.ncifcrf.gov/_nicka/sarf2.html/ (Alexandrov and Fischer 1996) also provides a protein database categorized on the basis of structural similarity. Like VAST, SARF can find structural similarity rapidly based on a search for secondary structural elements. These structural hierarchies found by this method are in good agreement with those found in the SCOP, CATH, and FSSP databases with several interesting differences. The method also found several new groupings of structural similarity. The SARF Web site provides a similarity-based tree of structures at http://www-lmmb.ncifcrf.gov/_nicka/ tree.html/ and some excellent representations of overlaid structures

5.5 PDB FILES

Every software package that reads in PDB data files must reconstruct the bonds based on these rules. However, the rules we are describing have never been explicitly codified for programmers. This means that interpreting the bonding in PDB files is left for the programmer to decide, and, as a result, software can be inconsistent in the way it draws bonds, especially when different algorithms and distance tolerances are used. The PDB file approach is minimalist in terms of the data stored in a record, and deciphering it often requires much more sophisticated logic than would be needed if the bonding information and chemical graph were explicitly specified in the record. Rarely is this logic properly implemented, and it may in fact be impossible to deal with all the exceptions in the PDB file format (Fig. 5.5). Each exception to the bonding rules needs to be captured by complicated logic statements programmed on a case-by-case basis.

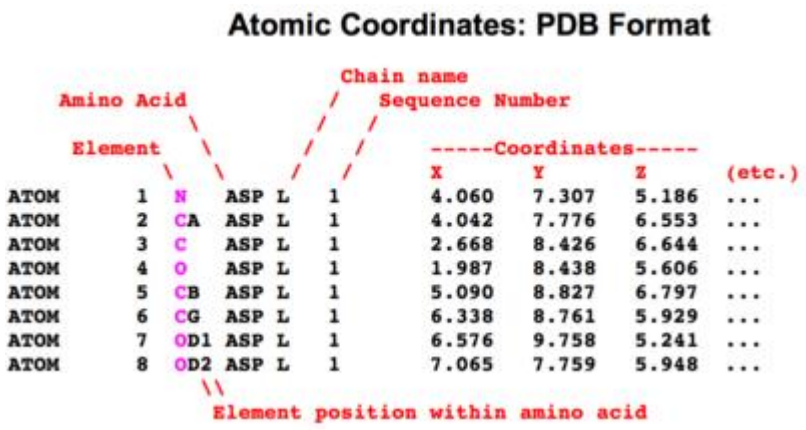


Fig. 5.5: PDB file format arranged with different amino acids and the information of their x, y and z co ordinates (https://proteopedia.org/wiki/index.php/Atomic_coordinate_file).

The second approach to describing a molecule is what we call the explicit bonding approach, the method that is used in the database records of the Molecular Modelling Database (MMDB), which is, in turn, derived from the data in PDB. In the MMDB system, the data file contains all of its own explicit bonding information. MMDB uses a standard residue dictionary, a record of all the atoms and bonds in the polymer forms of amino acid and nucleic acid residues, plus end-terminal variants. Such data dictionaries are common in the specialized software used by scientists to solve X-ray or NMR structures.

The software that reads in MMDB data can use the bonding information supplied in the dictionary to connect atoms together, without trying to enforce (or force) the rules of chemistry. As a result, the three-dimensional coordinate data are consistently interpreted by visualization software, regardless of type. This approach also lends itself to inherently simpler software, because exceptions to bonding rules are recorded within the database file itself and read in without the need for another layer of exception-handling codes.

5.5.1 Sequences from Structure Records

PDB file-encoded sequences are notoriously troublesome for programmers to work with. Because completeness of a structure is not always guaranteed, PDB records contain two copies of the sequence information: an explicit sequence and an implicit sequence. Both are required to reconstruct the chemical graph of a biopolymer. Explicit sequences in a PDB file are provided in lines starting with the keyword SEQRES. Unlike other sequence

databases, PDB records use the three-letter amino acid code, and nonstandard amino acids are found in many PDB record sequence entries with arbitrarily chosen three-letter names. Unfortunately, PDB records seem to lack sensible, consistent rules. In the past, some double-helical nucleic acid sequence entries in PDB were specified in a 3' to-5' order in an entry above the complementary strand, given in 5'-to-3' order. Although the sequences may be obvious to a user as a representation of a double helix, the 3'to-5' explicit sequences are nonsense to a computer. Fortunately, the NDB project has fixed many of these types of problems, but the PDB data format is still open to ambiguity disasters from the standpoint of computer readability.

As an aside, the most troubling glitch is the inability to encode element type separately from the atom name. Examples of where this becomes problematic include cases where atoms in structures having FAD or NAD cofactors are notorious for being interpreted as the wrong elements, such as neptunium (NP to Np), actinium (AC to Ac), and other nonsense elements. Because three-dimensional structures can have multiple biopolymer chains, to specify a discrete sequence, the user must provide the PDB chain identifier. SEQRES entries in PDB files have a chain identifier, a single uppercase letter or blank space, identifying each individual biopolymer chain in an entry.

The implicit sequences in PDB records are contained in the embedded stereochemistry of the (x, y, z) data and names of each ATOM record in the PDB file. The implicit sequences are useful in resolving explicit sequence ambiguities such as the backward encoding of nucleic acid sequences or in verifying nonstandard amino acids. In practice, many PDB file viewers (such as RasMol) reconstruct the chemical graph of a protein in a PDB record using only the implicit sequence, ignoring the explicit SEQRES information. If this software is asked to print the sequence of certain incomplete molecules, it will produce a non physiological and biologically irrelevant sequence. The implicit sequence, therefore, is not sufficient to reconstruct the complete chemical graph. Consider an example in which the sequence ELVISISALIVES is represented in the SEQRES entry of a hypothetical PDB file, but the coordinate information is missing all (x, y, z) locations for the subsequence ISA. Software that reads the implicit sequence will often report the PDB sequence incorrectly from the chemical graph as ELVISLIVES.

5.5.2 Validating PDB Sequences

To properly validate a sequence from a PDB record, one must first derive the implicit sequence in the ATOM records. This is a nontrivial processing step. If the structure has gaps because of lack of completeness, there may only be a set of implicit sequence fragments for a given chain. Each of these fragments must be aligned to the explicit sequence of the same chain provided within the SEQRES entry. This treatment will produce the complete chemical graph, including the parts of the biological sequence that may be missing coordinate data. This kind of validation is done on creation of records for the MMDB and mmCIF databases.

The best source of validated protein and nucleic acid sequences in single-letter code derived from PDB structure records is NCBI's MMDB service, which is part of the Entrez system.

5.6 Harnessing Data from PDB

Like other data repositories, the Protein Data Bank (PDB) offers a rather daunting interface that wasn't particularly designed with the non specialist in mind. Yet, in those rare cases where you know precisely what you're looking for and even know what you're doing, you may want to retrieve a protein 3-D structure dataset directly from one of the PDB sites. Before you query the PDB, be sure to collect some precise information about the structure you're looking for such as the exact protein name or (even better) its PDB identifier.

You can usually obtain this identifier from such user-friendly sources as the ExPASy/Swiss-Prot server or by using the various NCBI query tools. Here's how to obtain and display a PDB structure. For now, let's assume that we are looking for the structure of an *Escherichia coli (E. coli)* protein named TolB, with PDB ID code 1CRZ.

1. Point your browser to www.rcsb.org/pdb/. This takes you to the PDB home page.

2. Enter the PDB ID code 1CRZ in the search box in the middle at the very top of the page.

3. Click the adjacent SEARCH button. Figure 5.6 shows the resulting output. This one-page Structure Summary form presents the essential information on this protein structure

including a small graphic, a bibliographical reference, its function, its organism of origin and for the brave some more obscure crystallographic parameters. You can get additional details on various aspects of this protein by clicking the various tags at the top of the form (see Figure 5.6). For now, we only want to see what the protein molecule looks like in 3-D.

4. Click the All Images link in the Display Options below the structure image (refer to Figure 5.6). A new page appears, with an enlarged still view of the structure on the top, and an interactive view (a Java applet) at the bottom.

5. Right-click the Still Image (in color). A menu of options now allows you to either print the image, copy it, or save it to your own hard drive. It is now suitable for inclusion in reports or presentations. But wait, there's more! You can actually get a better feeling for the shape of this protein using the bottom gray-scale Interactive View as follows:

6. Right-click the Interactive View picture. You are offered a large menu of viewing options. Explore them to see which functions of this Java applet actually work on your specific system. For instance, selecting Spin On puts the molecule in an endless rotation. (To stop the merry-go-round, select Spin Off.) This is a great way to gain a better understanding of the 3D shape of the molecule.

7. Left-click and drag from any position in the structure, as shown in Figure 5.7. This sets the molecule in controlled motion. Use this mode to inspect specific structural regions at your own pace, from any angle that you choose. (You can zoom in as well.) In addition, double-clicking allows you to select a given position from which you can measure distances and angles. Try it out for yourself to see what you can do; it's fun, and you may impress your colleagues with your brand new expertise in structural biology.

Feel free to play with the other display options (KiNG, WebMol, . . ., etc.) and see which one works best for you. See the section "Exploring the sequence/ PDB structure relationship the interactive way," later in this chapter, for a tutorial on a more advanced tool that requires downloading and installing the CnD3 free software on your machine. The preceding steps list shows you how to jump directly from the PDB entry to a visual representation of the molecule. This may not be enough, and you may also want to keep the PDB entry for further study on your own computer. Read on to find out how.

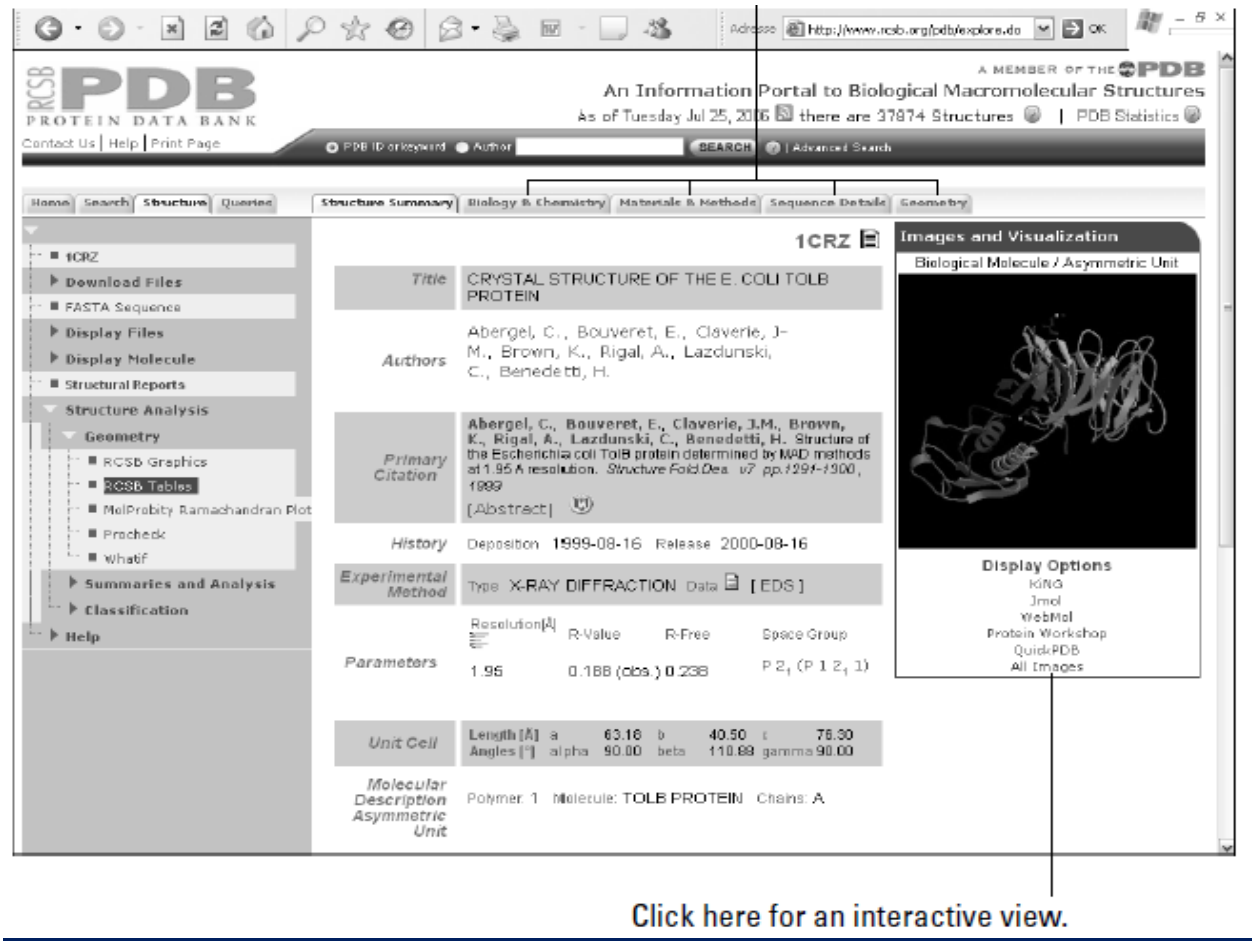


Fig. 5.6: The PDB structure summary output form for query 1CRZ.

PDB files aren't meant to be read by nonspecialists. They are rather long because they contain a vast amount of data. For instance, the 1CRZ contains the detailed x, y, z coordinates of 3,488 atoms as well as additional information about their connectivity.

5.6.1 Guessing the 3-D structure of your protein

The previous section shows how to retrieve and quickly display a known 3-D–structure recorded in the PDB. This is definitely useful but falls short of addressing the questions we listed at the beginning of this chapter. Basically, we still do not know what the interesting bits of our own sequence look like in their 3-D context.

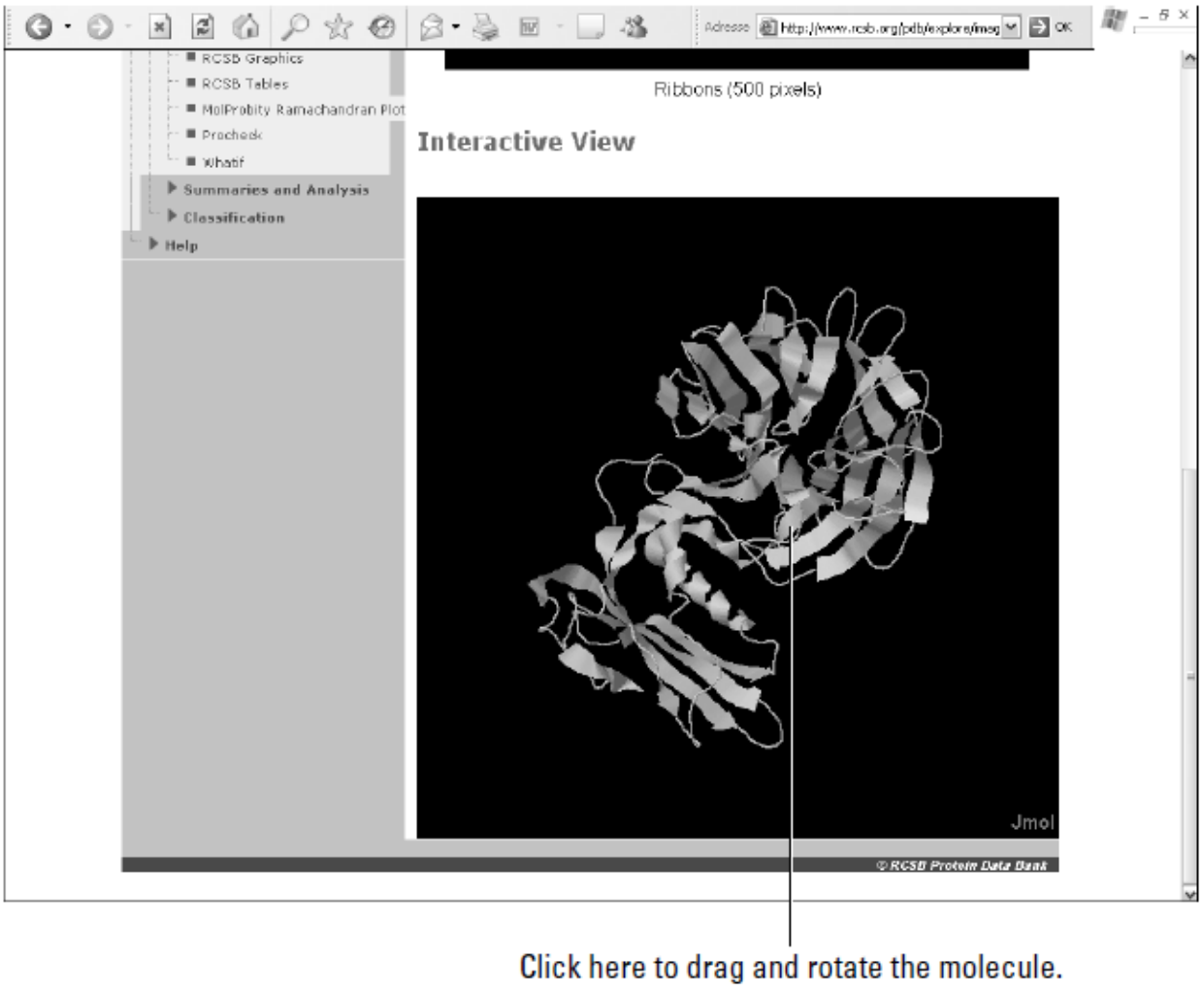


Fig. 5.7: Interactive view of the 1CRZ PDB entry.

In the remaining sections of this chapter, we follow a realistic scenario to show you how you can sometimes get a fairly good idea of the 3-D structure of your protein from its sequence alone by doing a handful of simple operations. We assume that you have determined the sequence and the secondary structure of your protein and that you are now asking this basic question: How does it fold? A simple way to answer this question is to look for a homologous protein with a known 3-D structure in the PDB. The next steps list shows you how. Imagine, if you will, that we've just determined the sequence of the TolB gene of the bacteria *Rickettsia conorii* and we are curious about its structure. To satisfy our curiosity, we do the following:

1. Fetch the protein sequence from NCBI:

a. Open a window in your browser and go to www.ncbi.nlm.nih.gov.
b. Choose Protein from the Search drop-down menu.
c. Type the identifier NP_360043 in the query window and then click Go.
d. When the answer comes back, change the Display format to FASTA. The *Rickettsia conorii* TolB protein sequence is now ready for use.

2. Open a new browser window and point it to the NCBI BLAST server at www.ncbi.nlm.nih.gov/BLAST/.
3. Click the Standard Protein-Protein BLAST [blastp] link. It is the first choice in the upper-right corner. The blastp input page appears.
4. Select pdb from the Choose database drop-down menu.
5. Copy the *Rickettsia conorii* TolB sequence from the other browser window and paste it into the blastp query window.
6. Deselect the Do CD-search box (to simplify your output).
7. Click the BLAST! button.
8. An intermediate page appears, telling you that your request has been successfully submitted and put into the Blast Queue.
9. Click the Format, button on the intermediate page and wait for the Blastp search to finish.

There are two strong matches with E-values much lower than 1. They are identified as PDB entries 1C5K and 1CRZ. This is good news. If you click these matches, their identifiers reveal that they both correspond to the TolB protein from *E. coli* the structure of which has been determined by two different laboratories (Figure 5.8).

This result is very interesting for two important reasons:

1. The homologous PDB sequences have 29 percent identical residues with our query.
2. The homology region covers the entire sequences.

By experience, we know that, given such a percentage of identical residues, the 3-D structure of our query protein (the Rickettsia TolB) is probably quasiidentical to the 3-D structure of its E. coli homologue. And so, we have a nearly perfect 3-D model to work with.

The next section shows you how to use this model to further interpret the TolB protein sequence.

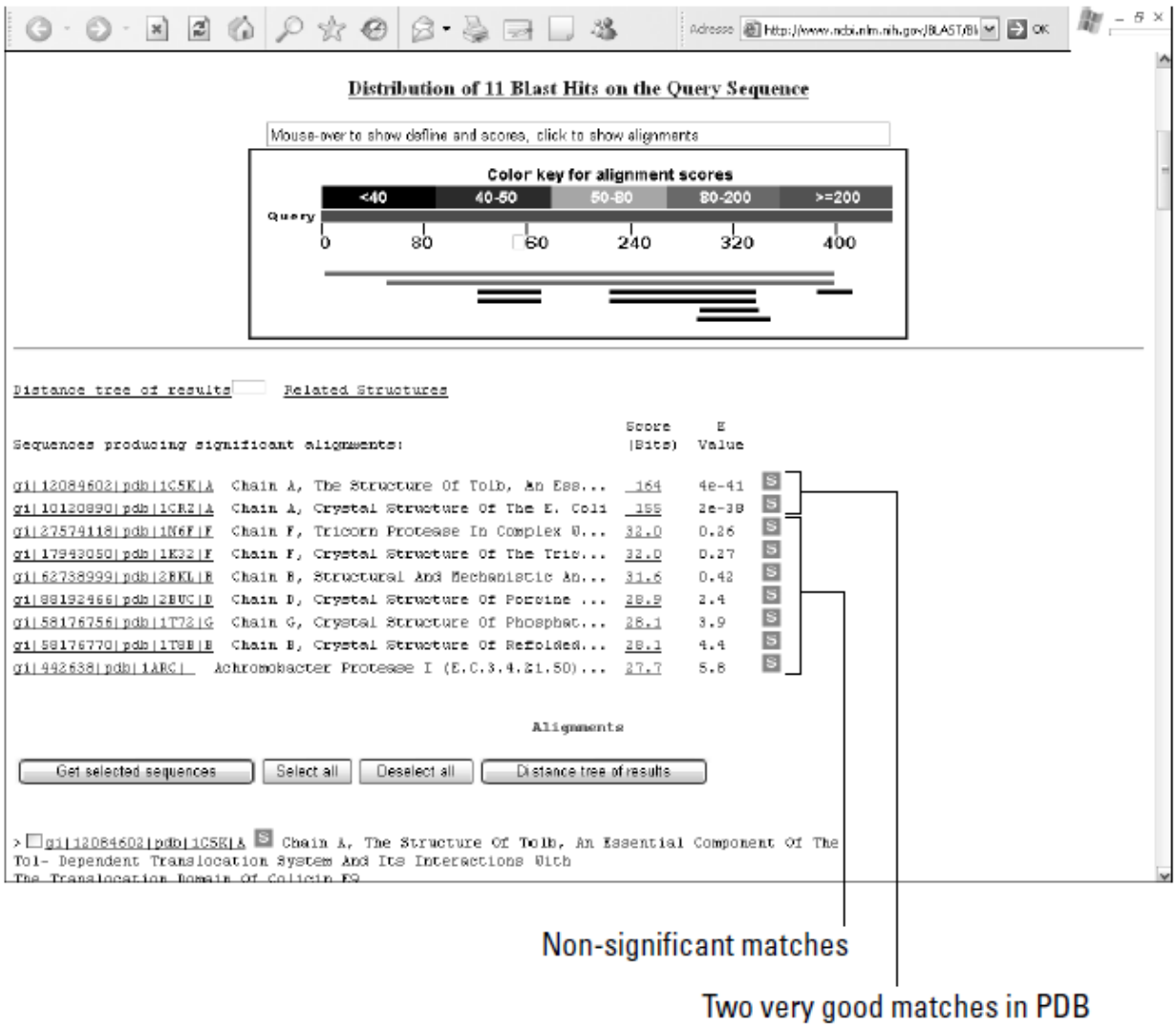


Figure 5.8: Blastp output of the *Rickettsia conorii* TolB sequence against PDB.

5.7. Protein Sequence Databases

5.7.1 RefSeq: The National Center for Biotechnology Information Reference Sequence (NCBI RefSeq) database provides curated non-redundant sequences of genomic regions, transcripts and proteins for taxonomically diverse organisms including Archaea, Bacteria, Eukaryotes, and Viruses. RefSeq database is derived from the sequence data available in

the redundant archival database GenBank. RefSeq sequences include coding regions, conserved domains, variations etc. and enhanced annotations such as publications, names, symbols, aliases, Gene IDs, and database cross-references. The sequences and annotations are generated using a combined approach of collaboration, automated prediction, and manual curation. The RefSeq release 73 on November 6, 2015 includes 54,766,170 proteins, 12,998,293 transcripts and 55,966 organisms. The RefSeq records can be directly accessed from NCBI web sites by search of the Nucleotide or Protein databases, BLAST searches against selected databases and FTP downloads. RefSeq records are also available through indirect links from other NCBI resources such as Gene, Genome, BioProject, dbSNP, ClinVar and Map Viewer etc. In addition, RefSeq supports programmatic access through Entrez Programming Utilities.

5.7.2 UniProt: The UniProt Consortium consists of research teams from the European Bioinformatics Institute (EBI), the Swiss Institute of Bioinformatics (SIB) and the Protein Information Resource (PIR). The UniProt Consortium provides a central resource for protein sequences and functional annotations with four core database components to support protein bioinformatics research. The UniProt Knowledgebase (UniProtKB) is the predominant data store for functional information on protein sequences with rich and accurate annotations (protein name or description, taxonomic information, classification, crossreference and literature citation). The UniProtKB consists of two parts: UniProtKB/Swiss-Prot, which contains manually annotated records with information extracted from literature and curator-evaluated computational analysis, and UniProtKB/TrEMBL, which contains computationally analyzed records with automatic annotation and classification. Comparative analysis and query for proteins are supported by UniProtKB extensive cross-references, functional and feature annotations, classification, and literature-based evidence attribution. The 2015_12 release on December 09, 2015 of UniProtKB/SwissProt contains 550,116 sequence entries, comprising 196,219,159 amino acids, and 55,270,679 UniProtKB/TrEMBL sequence entries comprising 18,388,518,872 amino acids.

The UniProt Archive (UniParc) is a comprehensive and non-redundant archival protein sequence database from all major publicly accessible resources. UniParc contains

protein sequences and cross-references to their source databases. UniParc stores each unique protein sequence with a stable and unique identifier and tracks sequence changes in its source databases.

The UniProt Reference Clusters (UniRef) are clustered sets of sequences from the UniProt Knowledgebase (including isoforms) and selected UniParc records. UniRef merges sequences and sub-fragments with 100% (UniRef100), ≥ 90% (UniRef90), or ≥ 50% (UniRef50) identity and 80% overlap with the longest sequences in the cluster (seed) into a single UniRef entry and select the highest ranked protein sequences as the cluster representatives. The UniProt Proteomes provides sets of proteins that are considered to be expressed by organisms whose genomes have been completely sequenced. A UniProt proteome consists of all UniProtKB/Swiss-Prot entries plus those UniProtKB/TrEMBL entries mapped to Ensembl Genomes for that proteome. Some well-studied model organisms and other organisms of interest to biomedical research and phylogeny have been manually and computationally selected as reference proteomes. The UniProt web site (http://www.uniprot.org) is the primary access point to its data and documentation. The site provides batch retrieval using UniProt identifiers; BLAST-based sequence similarity search; Clustal Omega based sequence alignment; and Database identifier mapping. The UniProt FTP download site provides batch download of protein sequence data in various formats, including flat file TEXT, XML, RDF and FASTA.

5.7.3 D Gel Databases: World-2DPAGE The World-2DPAGE Constellation is an effort of the Swiss Institute of Bioinformatics to promote and publish two-dimensional gel electrophoresis proteomics data online through the ExPASy proteomics server. The World-2DPAGE Constellation consists of three components:

World-2DPAGE List (http://world-2dpage.expasy.org/list/) contains references to known federated 2-D PAGE databases, as well as to 2-D PAGE related servers and services.

World-2DPAGE Portal (http://world-2dpage.expasy.org/portal/) is a dynamic portal that serves as a single interface to query simultaneously worldwide gelbased proteomics databases that are built using the Make2D-DB package.

World-2DPAGE Repository (http://world-2dpage.expasy.org/repository/) is a public repository for gel-based proteomics data with protein identifications published in the literature. Mass-spectrometry based proteomics data from related studies can also be submitted to the PRIDE database so that interested readers can explore the data in the views of 2D-gel and/or MS.

The World-2DPAGE Constellation also provides a set of tools:

Make2D-DB package (ver. 3.10.2) is open source packages that can be used to build a user's own 2-D PAGE web site, access and integrate federated 2D-PAGE databases, portals or data repositories.

5.7.4 Chemistry Databases: ChEMBL is a large-scale bioactivity database containing binding, functional, in vivo absorption, distribution, metabolism, excretion, and toxicity (ADMET) information about drug-like bioactive compounds. ChEMBL data are manually curated from the published literature together with data drawn from other databases. ChEMBL are standardized for using in many types of chemical biology and drug-discovery research problems. ChEMBL database can be accessed from a web-based interface where a variety of search and browsing functionality are provided. ChEMBL data is freely available from their FTP site in the formats of Oracle, MySQL, PostgreSQL, structure-data file (SDF), FASTA and RDF. Programmatic access is also supported by a set of RESTful web services. The ChEMBL release 20 (prepared on Jan 14, 2015) contains 1,715,135 compound records, 1,463,270 compounds (of which 1,456,020 have mol files), 13,520,737 activities, 1,148,942 assays, 10,774 targets, and 59,610 documents.

5.7.5 Enzyme and Pathway Databases

5.7.5.1 MetaCyc and BioCyc: MetaCyc is a reference database of non-redundant, experimentally elucidated metabolic pathways and enzymes curated from the scientific literature. MetaCyc stores pathways, compounds, proteins, protein complexes and genes associated with these pathways with extensive links to protein sequence databases, nucleic acid sequence databases, protein structure databases and literature. MetaCyc can also be used as a reference database to predict the metabolic network in sequenced genomes by Pathway Tools software using machine-learning methods. The 2015 release of MetaCyc

includes 2,411 metabolic pathways, 13,074 reactions, 10,789 enzymes, 10,928 genes, 12,792 chemical compounds, 2,740 organisms, and 47,838 citations. BioCyc is a collection of Pathway/Genome Databases (PGDBs). Each BioCyc PGDB contains the metabolic network of one organism predicted by the Pathway Tool software using MetaCyc as a reference database. The BioCyc databases are organized into three tiers: Tier 1 databases are those that have received at least one person-year of literature-based curation. Tier 2 and Tier 3 databases are computationally predicted metabolic pathways. Web-based query, browsing, visualization and comparative analysis tools are also provided from MetaCyc and BioCyc web sites. A collection of data files in different formats is provided for download. BioCyc also provides RESTful web services, MySQL server and Perl, Java and Lisp APIs access to its data. The 2015 release of BioCyc includes 7,667 Pathway/Genome Databases.

5.7.5.2 BRENDA: BRENDA (BRaunschweig ENzyme DAtabase) is an information system for functional and molecular properties of enzymes and enzyme-ligands obtained by manual extraction from literature, text and data mining, data integration and computational predictions. BRENDA stores enzyme data in textual, single numeric, numeric range, and graphic formats. The content of BRENDA is based on the IUBMB (International Union of Biochemistry and Molecular Biology) enzyme classification system. BRENDA includes the following databases generated by text mining approach.

• KENDA contains kinetic values and kinetic expressions mined from PubMed abstracts.

• DRENDA contains disease-related enzyme information (causal interaction, therapeutic application, diagnostic usage, and ongoing research) mined from PubMed abstracts using MeSH terms.

• FRENDA contains references found in PubMed abstracts that have the enzyme name and organism combination.

• AMENDA is a subset of FRENDA providing organism-specific information on the enzyme sources and the subcellular localization.

The user can access the data and information in BRENDA by searching (Quick Search, Advanced Search, Full text Search, Substructure Search, and Sequence Search) and browsing (TaxTree Explorer, EC Explorer, Ontology Explorer, and Genome Explorer). The

search results can be downloaded as CSV file. The BRENDA release 2015.2 in July 2015 contains 6,759 enzymes.

5.7.6 Reactome: Reactome is an open source, expert-curated and peer-reviewed database of biological reactions and pathways with cross-references to major molecular databases. Reactome provides the visual representation of classical intermediary metabolism, signaling, innate and acquired immune function, transcriptional regulation, apoptosis and disease process etc. Reactome website supports the navigation of pathway knowledge and pathway-based analysis and visualization of experimental or computational data. Interaction, reaction and pathway data are downloadable as flat file, MySQL, BioPAX,

SBML and PSI-MITAB files. They are also accessible through RESTful web services. Software tools such as Pathway Browser, Analyze Data, Species Comparison, Reactome FI Network are provided to support data mining and analysis of large-scale data sets. The Reactome release 54 in September 2015 contains 101,670 proteins, 74,357 complexes, 68,659 reactions, and 20,261 pathways.

5.7.7 Family and Domain Databases

5.7.7.1 InterPro: InterPro is an integrated resource of predictive models or 'signatures' representing protein domains, families, regions, repeats and sites from major protein signature databases including CATH-Gene3D, HAMAP, PANTHER, Pfam, PIRSF, PRINTS, ProDom, PROSITE, SMART, SUPERFAMILY and TIGRFAMs. Each entry in the InterPro database is annotated with a descriptive abstract name and cross-references to the original data sources, as well as to specialized functional databases. The search by sequence or domain architecture is provided by InterPro web site. The InterPro signatures in XML format are available via anonymous FTP download. InterPro also provides a software package InterProScan that can be used locally to scan protein sequences against InterPro's signatures. Programmatic access to InterProScan is possible via RESTful and SOAP web service APIs. The InterPro BioMart allows users to retrieve InterPro data from a query-optimized data warehouse that is synchronized with the main InterPro database, and to build simple or complex queries and control the query results through a unified interface. The InterPro release 54.0 on October 15, 2015 includes 28,462 entries containing

signatures of 19,110 families, 8,191 domains, 284 repeats, 115 active sites, 74 binding sites, 672 conserved sites and 16 PTMs.

5.7.7.2 Pfam: Pfam is a database of protein families represented as multiple sequence alignments and Hidden Markov Models (HMMs). Pfam entries can be classified as Family (related protein regions), Domain (protein structural unit), Repeat (multiple short protein structural units), Motifs (short protein structural unit outside global domains). Related Pfam entries are grouped into clans based on sequence, structure or profile-HMM similarity. The Pfam database web site provides search interface for querying by sequence, keyword, domain architecture, taxonomy, and browse interfaces for analyzing protein sequences for Pfam matches and viewing Pfam annotations in domain architectures, sequence alignments, interactions, species and protein structures in PDB. The Pfam data can be downloaded from its FTP site or programmatically accessed through RESTful web service APIs. The Pfam release 28.0 in May 2015 contains 16,230 families.

5.7.7.3 PIRSF: The PIRSF classification system provides comprehensive and non-overlapping clustering of UniProtKB sequences into a hierarchical order to reflect their evolutionary relationships based on whole proteins rather than on the component domains. The PIRSF system classifies the protein sequences into families, whose members are both homologous (evolved from a common ancestor) and homeomorphic (sharing full-length sequence similarity and a common domain architecture). The PIRSF family classification results are expert-curated based on literature review and integrative sequence and functional analysis. The classification report shows the information on PIRSF members and general statistics, family and function/structure relationships, database cross-references and graphical display of domain and motif architecture of seed members or all members. The web-based PIRSF system has been demonstrated as a useful tool for studying the function and evolution of protein families. It provides batch retrieval of entries from the PIRSF database. The PIRSF scan allows searching a query sequence against the set of fully curated PIRSF families with benchmarked Hidden Markov models. The PIRSF membership hierarchy data is also available for FTP download. The current release of PIRSF contains 11,800 families, which cover 5,407,000 UniProtKB protein sequences.

5.7.7.4 PROSITE: PROSITE is a database of documentation entries describing protein domains, families and functional sites as well as associated patterns and profiles to identify them. The entries are derived from multiple alignments of homologous sequences and have the advantage of identifying distant relationships between sequences. PROSITE includes a collection of ProRules based on profiles and patterns of functionally and/or structurally critical amino acids that can be used to increase PROSITE's discriminatory power. The PROSITE web site provides keyword-based search and allows browsing by documentation entry, ProRule description, taxonomic scope and number of positive hits. The software tool ScanProsite supports three options for users to scan proteins for matches to PROSITE motifs or their own sequence patterns:

1) Scan protein sequence against the PROSITE motifs;

2) Scan motifs against a protein sequence database;

3) Submit protein sequences and motifs and scan them against each other.

The PROSITE documentation entries and related tools can be downloaded from its FTP site. The PROSITE release 20.120 on November 4, 2015 contains 1,742 documentation entries, 1,309 patterns, 1,139 profiles and 1,138 ProRules.

5.8 Structural Database

In biology, a protein structure database is a database that is modeled around the various experimentally determined protein structures. The aim of most protein structure databases is to organize and annotate the protein structures, providing the biological community access to the experimental data in a useful way.

Structural databases are essential tools for all crystallographic work and often needed to be consulted at several stages of the process of producing, solving, refining and publishing the structure of a new material. Examples of such uses are:

i. Before deciding to synthesise a new compound, the database could be used to check how many compounds with a particular chemical composition have been reported.

ii. After synthesising and indexing the unit cell of a material the database can be searched to see if a material with the same or a similar unit cell is already known.

iii. If a material is found in the database with a similar unit cell to the new material, then its structure may be close enough (i.e. same symmetry and similar unit cell contents) to be used as the starting model for the Rietveld refinement of the new material.

iv. To verify the results of a structure refinement, the database can be consulted to find structures that have comparable bond distances, bond angles or coordination environments to the new structure.

The common information found in a structural database for each entry is:

1. Bibliographic information - author(s) names, journal reference.

2. The chemical compound name, formula and oxidation states of the elements present.

3. The contents (number of formula units per unit cell), dimensions and symmetry (crystal system and space group) of the unit cell.

4. The symmetry of the structure, atomic coordinates occupancies and thermal parameters (isotropic or anisotropic).

5. Comments on any special features of the experiment to collect the diffraction data, for example, the temperature, and on any problems found in the structure itself.

Not all the above information is found in each database for all entries, but all databases contain the information listed in points (1) to (3). All the above data for each entry is explicit. The explicit data can be used to generate data that is implicit in this stored information but that must be calculated from the stored data. Implicit data includes such things as, interatomic distances, bond angles, torsion angles, coordination numbers and structural representations that are generated by software within the database suites.

The structures in the databases have been solved using X-ray, neutron and electron diffraction techniques on samples that are generally single crystals, but with the advances in structural solution using powder diffraction data, may be powders. There are some

entries whose structures are predicted from computational modelling and some determined using NMR spectroscopy, these entries generally occur for protein samples.

One important point to note is the difference between these structural databases and the database of powder diffraction files (ICDD-PDF). The latter contains the "fingerprint" powder diffraction pattern of crystalline materials whose structure may or may not be known. The structural databases contain structural information for each material (the unit cell at least) derived from analysis of diffraction data.

5.8.1 Searching a Structural Database

Searching a structural database is similar to searching any database in principle. There are three main areas to search which are:

i. Bibliographic information

ii. Chemical information

iii. Crystallographic/ Structural information

To search a database, one can search for exact matches, for example, if you use the search term M. P. Artfield in an author search, or you can search for ranges, for example, 500 to 520 $Å^3$ in a unit cell volume search. Often the results of searches produce many matching structures the number of which can be reduced by intersecting with the results from other searches. For instance, if you want to find the number of structures with space group $P2_1/c$ and that contain the element indium, then searching the data base of inorganic structures for structures with the space group $P2_1/c$ produces 2316 structures and structures that contain indium gives 867 structures.

5.9 Sequence Databases: Primary, Secondary and Other Databases

Anyone entering the heart of the biological internet encounters a bewildering number of accession numbers, identifiers and gene names. To get to grips with this flood of terminology, it is important to understand the difference between primary and secondary databases and their associated accession numbers. This is not proposed as a rigorous definition but it does have a utility for understanding the information flow between sequence databases.

5.9.1 Primary Databases

Primary accession numbers have a number of key attributes; they refer to nucleic acid sequences derived directly from a sequencing experiment, the results are submitted by authors in a standardized format to GenBank, EMBL or DDBJ, the accession numbers are both unique and stable (if they are updated or amended by the submitting authors the accession number will signify a version change as .1, .2 etc.), the data records from every accession number can be retrieved, a contactable submitter is included in every record, they are explicitly redundant in that all submissions are accepted regardless of partial or complete overlap with existing entries and lastly the growth rate remains close to exponential and now exceeds 16 million sequence records.

The concept of authors' needs stretches to encompass consortia that run high-throughput sequencing projects. One of the most valuable and perhaps overlooked principals of these unique public repositories is that there is always (with the exception of patent data, see below) an identified individual or laboratory representative listed with the sequence record who can be contacted for any queries regarding experimental details, data quality and availability of source material. There is a large amount of information associated with primary sequence records. These include primary accession numbers, version numbers, protein ID numbers, gene identifier (GI) numbers, header records and feature identifiers. These cannot be covered in detail here but full descriptions are given in database guides (http://www.ebi.ac.uk/embl/index.html) and release notes (ftp://ftp.ncbi.nih.gov/genbank/gbrel.txt).

Geneticists should be encouraged to contact submitting authors in cases where anything seems non-obvious about primary data records for an mRNA or a finished genomic clone. They may have an extra information that has a crucial bearing on the interpretation of genetic experiments. Authors may be difficult to track down if they have moved institutions but they are usually pleased to assist in the utilization of their data, because as with scientific publishing, this is the principle behind public sequence databases. Technical errors, anomalies, miss-annotation in submissions or artifacts are entirely the responsibility of submitting authors not the database administrators. Although we should be sanguine concerning anomalies in the high-throughput data divisions (EST,

GSS, STS, HTG, HTC and SNP), if problems are pointed out authors can certainly amend or update their entries or in some cases may withdraw them. The primary data is deposited in good faith so authors should certainly not be harshly judged if an error has occurred in the rough and tumble of cloning, sequencing and submitter annotation. The exception to author responsibility for GenBank records is the patent division (gbPAT) where inventors are not equivalent to academic authors. These sequence records are processed by the US, European and Japanese patent offices and forwarded on to the databases. Although author contact may not be practical database users should be aware that patent applications are public documents and for an increasing number of gbPAT records the documentation can be accessed via the patent number on-line and free of charge (http://ec.espacenet.com/espacenet/ and http://www.uspto.gov/patft/). It is also possible to get to these patent full-text links directly from sequence entries via SRS

5.9.2 Secondary Databases Nucleic Acids and Proteins

By definition secondary databases are derived from the primary data. The word secondary should not be taken to imply lower value; indeed they include sources of the highest utility for genetic research. However, they are defined, it is important to understand how they are linked back to the experimental data. The good news for geneticists is that there is now a comprehensive selection of high quality secondary databases that extract and collate subsets of mRNA, genomic or protein sequences from primary GenBank entries. The bad news is that the proliferation of features that make secondary databases so powerful also presents a bewildering range of options to the user. Testimony to both, the good and bad news, is given by the 2002 update of the Molecular Biology Database Collection (http://nar.oupjournals.org/cgi/content/full/30/1/1/DC1). This covers no less than 355 databases, up from 281 in 2001, of which the primary databases, GenBank, EMBL and DDJB, constitute only three entries. Although, this compendium includes many non-human data sources, almost all of these secondary databases contain information that could be pertinent to mammalian genetics. These review issues appear every January in Nucleic Acids Research and are definitely worth browsing. Are the genome portals secondary databases? This is where the definitions become blurred. Because NCBI generate their own genomic accessions numbers (NT

numbers) and Ensembl generate their own exon and gene identifiers, they could be considered secondary databases. In so far as the UCSC genome portal marks up only external sequence record identifiers (primary and secondary), they are not strictly a secondary database. However, because they usefully give every type of gene prediction in the display a retrievable identity number, they could be considered as a secondary database.

The value of secondary databases includes the following:

- Distilling down a massive number of overlapping and/or redundant primary GenBank entries to a manageable range of genomic sections, unique transcripts and translated protein sequences
- Maintaining a running total of gene products, they partition human gene products and other vertebrates with extensive genomic data such as mouse, rat and zebra fish
- The inclusion of informative graphic displays for sequence features • Providing access to a vast amount of pre-processed bioinformatic data
- Extensive interconnectivity through web hot-links
- Many of them are backed up by extensive institutional resources and expertise

However, users of these secondary databases also need to be aware of their shortcomings:

- They all suffer from the snapshot problem i.e. the time to re-build or update massive data sets means they are always out of date with respect to the new data cascading into the primary databases (given the complexity of the processes, this is entirely expected but they often do not display the dates when the primary records were extracted)
- They all have different look-and-feel interfaces thereby necessitating regular practice to get the best out of them
- The web-based interoperativity can leave a lot to be desired; e.g. broken links, linkouts to databases that are not maintained to the same standards and overkill by linking out to too many similar sources
- Their automated annotation schema can be confounded by sequence artefacts
- The overlap between utility and content between major databases is extensive but is never enough for any of them to be the mythical ‘one-stop-shop’

- Non-redundant transcript and protein collections may seem conceptually similar but because they diverge in schema details and update frequency they all give different statistics
- Some secondary databases such as SwissProt keep sequence identifiers both unique and stable but for technical reasons others, such as UniGene EST clusters or Ensembl genes, may change identifiers between builds
- Many specialized 'boutique' databases are never updated when their originators move on or run out of resources
- Last but not least, some secondary databases that initially had free access can become commercial and require a subscription fee

5.10 Nucleic Acid Secondary Databases

For the analysis of their results, the geneticist must become acquainted with these feature rich sources of gene product information. A key example, based around nucleic acid sequence but including protein of secondary databases is LocusLink/RefSeq (LLRS) for mRNAs. The LLRS system is built round a reference sequence (RefSeq) which is usually the longest available mRNA of those coding for the same protein. RefSeq includes splice variants and if only genomic sequence is available, such as for many of the 7TM receptors, the system defaults to the predicted coding sequence annotated as a 'CDS' in the database entry. For example there is no experimentally determined human rhodopsin mRNA in GenBank, only a model mRNA predicted from the genomic sequence U49742. This presents an immediate problem for the geneticist, as the untranslated region (UTR) of the rhodopsin locus, which defines the boundaries and functional regions of the gene may be extensive. Chapter 4 takes a detailed look at approaches to help define the true extent of gene loci.

The end-product of the RefSeq pipeline is a unique mRNA, coding sequence (CDS), or set of splice variants for those gene products where data or predictions are available. The LocusLink side of things, as suggested by the title, is directed towards mapping the RefSeq gene products onto the genomic sequence and checking the consistency between the two. LocusLink has linked sections of key importance to the geneticist. These are:

variation which assigns SNP data, OMIM which includes verified monogenic disease links, homologene which indicates close homologues in other species, UniGene which specifies ESTs clusters associated with the gene product, and PubMed that links to all publications that can be specifically linked to the primary GenBank accession numbers. There are also links to all three genome portals, NCBI, UCSC and Ensembl. There has been some confusion in the past, where the portals could not synchronize their builds and track displays with GP version updates, but this problem has been addressed and they should all be on version 28 (from December 2001) at the time of writing.

The RefSeq identifier is secondary in the sense that it is a supplementary identifier assigned to one particular mRNA chosen as the reference sequence. These accession numbers have the prefix NM for mRNA entries and NP for protein entries. The LocusLink/RefSeq system goes one step further in assigning a third identifier, XM for nucleic acid and XP for proteins, which are the genomic counterparts of the NM and NP numbers. A BLAST search against the NCBI protein database will show all three entries, the primary accession number, the NM and the XM entries. There is the added complication that the XP sequences have a variable evidence support level and include ab-initio genomic predictions both with and without EST support. Secondary accession numbers are also important for ESTs. ESTs can be considered as mRNA fragments that, with sufficient sampling (now just exceeding 4 million human entries in dbEST) can be clustered or assembled to form a contiguous extended transcription product and in some cases, the splice variants from the tissue types sampled for EST preparation. The main post-genomic utility of EST collections is as exon detectors. In addition to splice variants these can reveal possible gene transcription activity where no extended mRNA has been experimentally verified. The primary data source for ESTs is the dbEST division of GenBank.

The geneticist should be aware of two major secondary EST databases, UniGene (Wheeler et al., 2002) and the TIGR human gene index (Liang et al., 2000). The principles by which these different databases are constructed, are explained in the appropriate source references but in fact they both converge to a similar set of ‘virtual’ surrogate transcripts.

In the TIGR case, the virtual transcripts assembled from overlapping ESTs can be retrieved; in the Unigene case, the individual EST reads can be batch downloaded. As with most secondary databases, built from the same source data, the two databases have both overlap and complementarity. The TIGR assemblies are particularly useful for extending the 3_ UTR of known mRNAs but the assemblies are re-compiled at long time intervals.

UniGene is updated more frequently and is fully interlinked to the LocusLink/RefSeq system but the clusters are built on mRNAs from the preceding version of GeneBank.

STSs and SNPs: These are two of the most important data sources for the geneticist involved in disease mapping. The dbSTS database contains sequence and mapping data on short genomic landmark sequences. Although, they have a primary sequence record and GB accession number, they also have a number of alternative marker names. These have been cross-referenced into a secondary database called UniSTS that integrates all available marker and mapping data (http://www.ncbi.nlm.nih.gov/genome/sts/). The dbSNP database is an interesting exception in that it is not a division of GenBank, so it is not strictly a primary database. The submissions numbers (SS numbers) are equivalent to a primary record but overlapping sequences with the same polymorphism are collapsed into the Reference SNP Cluster Report with an RS number. This can be considered a secondary database where the RS numbers are non-redundant and stable. These RS numbers, currently at 2,640,509 for human, are integrated with other NCBI genomic data and primary GenBank records containing overlapping sequences deduced or stated to be from the same location. The HGVbase has a smaller set of 984,093 highly curated records (http://hgvbase.cgb.ki.se/). They have their own secondary accession/ID number and these can be queried and retrieved from the Ensembl genome annotation.

5.10.1 Gene Expression Databases: Expression Atlas

The Expression Atlas database provides gene, protein and splice variant expression patterns in different cell types, organism parts, biological and experimental conditions. The high quality Microarray and RNA-Seq data imported from ArrayExpress and Gene Expression Omnibus were manually curated, annotated and processed using standardized

analysis methods to detect the expression patterns under the original experimental conditions. Expression Atlas consists of two components: Baseline Atlas and Differential Atlas. The Baseline Atlas is about genes and their expression pattern under the "normal" conditions using only RNA-Seq data. The Differential Atlas is about genes that are up- or down- regulated in differential biological or experimental conditions using both Microarray and RNA-Seq data. Expression Atlas web interface supports query both the Baseline Atlas and Differential Atlas by gene, protein and splice variant. The search for sample attributes and experimental conditions are also supported. All Expression Atlas analysis results can be downloaded from their FTP site. The differential expression data and meta-data can be used in R Bioconductor (https://www.bioconductor.org/) package. The APIs to programmatically access Expression Atlas is under development. The October 29, 2015 release of Expression Atlas contains 2,373 datasets (93,057 assays).

5.10.2 Genome Annotation Databases

5.10.2.1 Ensembl: Ensembl is a genome annotation database that provides up-to-date annotations for chordates and model organism genomes. Additional metazoan genomes are available from EnsemblMetazoa, Plant and fungal genomes are available from EnsemblPlants and EnsemblFungi, Unicellular eukaryotic and prokaryotic genomes are available from EnsemblProtists and EnsemblBacteria. Ensembl supports variety of access routes to their data. Small data set can be exported from online search results. Large dataset or complex analyses can be accessed from MySQL server, Perl and RESTful APIs. Complex cross databases queries are supported by BioMart data mining tool. The whole database can be downloaded from FTP site in FASTA, EMBL, GenBank, GVF, VCF, VEP, GFF formats or through MySQL dumps. In addition, Ensembl also provides a set of data processing software tools. For example, Variant Effect Predictor, BLAST/BLAT, Assembly converter, ID History converter etc. The Ensembl release v83 in September 2015 contains 69 species with annotations for gene and transcript, gene sequence evolution, genome evolution, sequence and structural variants and regulatory elements.

5.10.2.2 Entrez Gene: Entrez Gene is a NCBI gene-specific database that provides GeneIDs (unique integer identifiers) for genomes that have been completely sequenced. The data in Entrez Gene database (nomenclature, map location, gene products and attributes, markers,

phenotypes, citations, sequences, variations, maps, expression, homologs, protein domains etc.) are results of manual curation and automated computational analysis of data from RefSeq and many other NCBI databases. The data in Entrez Gene database can be accessed in several ways:

1) Query Entrez from the NCBI home page and display the results in Gene,

2) Enter a query in any Entrez query bar and restrict the database search to Gene,

3) Cross links from other NCBI resources such as GenBank, BLAST, RefSeq, Map Viewer.

Entrez Gene data can be downloaded from NCBI FTP site and accessed by Entrez Programming Utilities [145]. The Entrez Gene release on December 4, 2015 includes 13,778 taxa and 12,841,400 genes.

5.10.2.3 UCSC: UCSC Genome Browser database contains large collection of genome assemblies and annotations for vertebrate and selected model organisms. The major sources of genome annotations include RefSeq, GENCODE, Ensembl, GenBank, ENCODE, RepeatMasker, dbSNP, the 1000 Genome project and other resources. In addition to Genome Browser, the UCSC bioinformatics group also provides web-based and command-line based tools to facilitate the use of genome annotations data. For example, BLAT can be used to quickly find sequences of 95% and greater similarity and 25 bases or more in length. The Table Browser can retrieve the data associated with a track in Genome Browser and calculate intersections between tracks. The Variant Annotation Integrator can associate UCSC Genome Browser annotations with the user-uploaded variants. The Gene Sorter can be used to show expression, homology and other information on groups of genes. User data can be viewed together with UCSC annotations via 'custom track', 'track data hubs', 'assembly hub' and 'Genome Browser in a Box (GBiB)'. Genome data and source codes are downloadable. UCSC Genome Bioinformatics group also provides public MySQL server access. Currently (December 11, 2015), there are 95 genomes in UCSC Genome Browser database.

5.11 Organism Specific Databases

5.11.1 FlyBase: FlyBase is a database of Drosophila melanogaster related genetic and genomic information. The sequence and annotation data for Drosophila melanogaster genome assembly can be downloaded from FlyBase FTP site in multiple formats (GFF3, FASTA, GTF, Chado XML, and Chado PostgreSQL dump). FlyBase uses generic genome browser 2 (GBrowse 2) to display the genome annotations and genome-aligned evidence on the reference genome assembly. FlyBase database can be searched for genes, alleles, aberrations and other genetic objects, phenotypes, sequences, stocks, images and movies, controlled terms. FlyBase provides a standalone BLAST server for 50 different arthropod genomes and supports query results analysis such as hit list refinement and batch download. The latest FlyBase is FB2015_05 released on November 20, 2015 that consists of 212,991 references, 141,104 stocks and 1,258 images.

5.11.2 MGD: The Mouse Genome Database (MGD) is a database of integrated genomic, genetic and biological data on the laboratory mouse that is a model for translational research. MGD integrates mouse genome annotations from NCBI, Ensembl and Havana into a single non-redundant resource. MGD is the authoritative source for the unified catalog of mouse genome features, Gene Ontology (GO) annotations (functional associations) of mouse protein-coding genes, and mouse phenotype annotations. The Human-Mouse: Disease Connection (http://www.diseasemodel.org) is a translational research tool that provides simultaneous access to human-mouse genomic, phenotypic and genetic disease information. MGD uses a powerful new genome browser called JBrowse to integrate mouse gene and protein annotations with large-scale sequence data. In addition to online search tools for genes, genome features and maps, phenotypes, alleles and disease models, gene expression, GO functional annotations, strains, SNPs and polymorphisms, sequences, references, vocabularies, MGD also provides bulk data download as FTP reports and batch query tool and programmatic access by Web services and BioMart. MGD is updated on a weekly basis.

5.12 Polymorphism and Mutation Databases: dbSNP

The NCBI dbSNP database is a database for short genetic variations from variety of organisms. dbSNP catalogs single nucleotide variations, short nucleotide insertions and

deletions, short tandem repeats and microsatellites. dbSNP homepage provides search interface for querying variations by simple term or complex queries. The details of matched variation record is displayed as the Reference SNP Cluster Report that contains summary of the allele, mapping information in Human Genome Variation Society (HGVS) nomenclature, gene-centric view, map table with chromosomal coordinates, variation view, and link to the 1000 Genomes Browser. dbSNP integrates disease-related variations collected by OMIM [86]. dbSNP variation data are accessible through links from other NCBI databases. dbSNP data can also be downloaded from a FTP site and accessed by EUtils API (https://www.ncbi.nlm.nih.gov/books/NBK25500/). dbSNP build 146 on November 24, 2015 for Homo sapiens contains 150,482,731 RefSNP Clusters, among them 100,135,281 are validated.

5.13 Summary

Protein databases have become a crucial part of modern biology. Huge amounts of data for protein structures, functions, and particularly sequences are being generated. Searching databases is often the first step in the study of a new protein. Comparison between proteins or between protein families provides information about the relationship between proteins within a genome or across different species, and hence offers much more information than can be obtained by studying only an isolated protein. In addition, secondary databases derived from experimental databases are also widely available. These databases reorganize and annotate the data or provide predictions. The use of multiple databases often helps researchers understand the structure and function of a protein. Although some protein databases are widely known, they are far from being fully utilized in the protein science community. This unit provides a starting point for readers to explore the potential of protein databases on the Internet.

5.14 Terminal questions

Q.1: Which of the following statements about SCOP is incorrect regarding its features?

a) Proteins with the same shapes but having little sequence or functional similarity are placed in different super families, and are assumed to have only a very distant common ancestor

b) Proteins having the same shape and some similarity of sequence and/or function are placed in 'families', and are assumed to have a closer common ancestor

c) SCOP was created in 1994 in the Centre of Protein Engineering and the University College London

d) It aims to determine the evolutionary relationship between proteins

Answer: (d)

Q.1: What is the source of protein structures in SCOP and CATH?

a) Uniprot

b) Protein Data Bank

c) Ensemble

d) InterPro?

Answer: (b)

Q.2: A Protein Data Bank (PDB) data file for a protein structure contains only x, and z coordinates of atoms.

a) True

b) False

Answer: (b)

Q.3: Ribbon diagrams use cylinders or spiral ribbons to represent α-helices and broad, flat arrows to represent β-strands.

a) True

b) False

Answer: (a)

Q.4: Which of the following is wrong about Swiss-PDB Viewer?

a) It is a structure viewer for multiple platforms

b) It is a structure viewer for single platforms

c) It is essentially a Swiss-Army knife for structure visualization and modeling

d) It is capable of structure visualization, analysis, and homology modeling

Answer: (b)

Q.5: Which of the following is incorrect about the intramolecular approach?

a) The method works by generating a distance matrix between residues of the same protein

b) It generates a string between residues of the same protein

c) In comparing two protein structures, the distance matrices from the two structures are moved relative to each other to achieve maximum overlaps

d) By overlaying two distance matrices, similar intramolecular distance patterns representing similar structure folding regions can be identified

Answer: (b)

Q.6: Which of the following is incorrect about SSAP?

a) It is a web server that uses an intramolecular distance–based method

b) Matrices are built based on the Cβ distances of all residue pairs

c) Dynamic programming approach is not used here

d) Dynamic programming approach is used?

Answer: (c)

Q.7: The justification behind Rosetta stone method is that when two domains are fused in a single protein, they have to be in _____ proximity to perform a common function.

a) distant

b) close

c) extremely distant

d) extremely close

Answer: (d)

5.11. Further readings

- Barnes MR, Gray IC. Bioinformatics for geneticists. Chichester, West Sussex, England ; Hoboken, N.J.: Wiley; 2003. xiv, 408 p., 8 p. of plates p.
- Baxevanis AD, Ouellette BFF. Bioinformatics : a practical guide to the analysis of genes and proteins. 3rd ed. Hoboken, N.J.: Wiley; 2005. xviii, 540 p. p.
- Bergeron BP. Bioinformatics computing. Upper Saddle River, NJ: Prentice Hall/Professional Technical Reference; 2003. xxii, 439 p. p.
- Claverie J-M, Notredame C. Bioinformatics for dummies. 2nd ed. Hoboken, N.J.: Wiley Pub.; 2007. xviii, 436 p. p.
- Lesk AM. Introduction to bioinformatics. 3rd ed. Oxford ; New York: Oxford University Press; 2008. xxii, 474 p. p.
- Mount DW. Bioinformatics: sequence and genome analysis. 2nd ed. Cold Spring Harbor, N.Y.: Cold Spring Harbor Laboratory Press; 2004. xii, 692 p. p.
- Xu D, Xu Y. Protein databases on the internet. Curr Protoc Mol Biol. 2004; Chapter 19:Unit–19.4. doi:10.1002/0471142727.mb1904s68.

Unit- 6: Molecular Simulation

6.1. Introduction

Objectives

6.2. PDB: Protein Data Bank at The Research Collaboratory for Structural Bioinformatics (RCSB)
- 6.2.1. PDB Query and Reporting
- 6.2.2. Submitting Structures
- 6.2.3. PDB-ID Codes
- 6.2.4. Database Searching, PDB File Retrieval, mm CIF File Retrieval and Links

6.3. MMDB: Molecular Modelling Database At NCBI
- 6.3.1. Free Text Query Of Structure Records
- 6.3.2. MMDB Structure Summary
- 6.3.3. Structure File Formats
- 6.3.4. Retrieval Of Structural Information From Mmdb
- 6.3.5. Visualizing Structural Information

6.4. Database Structure Viewers
- 6.4.1. Visualization Tools
- 6.4.2. Rasmol And Rasmol-Based Viewers
- 6.4.3. Mmdb Viewer: Cn3d
- 6.4.4. Other 3d Viewers: Mage, Cad, And Vrml
- 6.4.5. Making Presentation Graphics

6.5. Modelling And Drug Designing
- 6.5.1. Fundamentals of modelling
- 6.5.2. Components of modelling
- 6.5.3. Process involved in modelling

6.6. Simulation and its types
- 6.6.1. Continuous Simulation
- 6.6.2. Discrete Simulation
- 6.6.3. Hybrid Simulation
- 6.6.4. Numeric Considerations
- 6.6.5. Errors
- 6.6.6. Perspectives

6.7. Algorithms
- 6.7.1. Monte Carlo Method
- 6.7.2. Metropolis Algorithm
- 6.7.3. Hardware

6.8. Drug and Protein Structure
- 6.8.1. Ab Initio Methods
- 6.8.2. Heuristic Methods
- 6.8.3. Template Selection

6.9. Conserved Domain Database (CDD)

6.10. Nucleotide and Protein Sequence Databases

6.10.1. Entrez
6.10.2. Sequence Retrieval Server (SRS)
6.11. Drug Designing
6.12. Summary
6.13. Terminal question
6.14. Further readings

6.1. Introduction

Computer modelling is used to provide insight and understanding of how complex systems behave beyond what theory and experiment could deliver separately. It bridges theory and experiment by solving state equations numerically. Molecular dynamics (MD) is a computer simulation method for analyzing the physical movements of atoms and molecules. The atoms and molecules are allowed to interact for a fixed period of time, giving a view of the dynamic "evolution" of the system. In the most common version, the trajectories of atoms and molecules are determined by numerically solving Newton's equations of motion for a system of interacting particles, where forces between the particles and their potential energies are often calculated using interatomic potentials or molecular mechanics force fields. The method is applied mostly in chemical physics, materials science, and the biophysics. Because molecular systems typically consist of a vast number of particles, it is impossible to determine the properties of such complex systems analytically; MD simulation circumvents this problem by using numerical methods. However, long MD simulations are mathematically ill-conditioned, generating cumulative errors in numerical integration that can be minimized with proper selection of algorithms and parameters, but not eliminated entirely.

Objectives

- Gain knowledge of biomolecular structures, molecular visualization programs and biological data bases.
- Understanding the theory at the basis of molecular dynamics simulations and Monte Carlo methods.

- Demonstrate skills in molecular simulations and ability to set up and to interpret classical molecular dynamics simulations in the light of the knowledge acquired on principles of the statistical mechanics.

6.2 PDB: Protein Data Bank at the Research Collaboratory For Structural Bioinformatics (RCSB)

The use of computers in biology has its origins in biophysical methods, such as X ray crystallography and NMR spectroscopy. Thus, it is not surprising that the first ‘‘bioinformatics’’ database as built to store complex three-dimensional data. The Protein Data Bank, originally developed and housed at the Brookhaven National Laboratories, is now managed and maintained by the Research Collaboratory for Structural Bioinformatics (RCSB). RCSB is a collaborative effort involving scientists at the San Diego Supercomputing Center, Rutgers University, and the National Institute of Standards and Technology. The collection contains all publicly available three-dimensional structures of proteins, nucleic acids, carbohydrates, and a variety of other complexes experimentally determined by X-ray crystallographers and NMR spectroscopists. This section focuses briefly on the database and bioinformatics services offered through RCSB.

The World Wide Web site of the Protein Data Bank at the RCSB offers a number of services for submitting and retrieving three-dimensional structure data. The home page of the RCSB site provides links to services for depositing three-dimensional structures, information on how to obtain the status of structures undergoing processing for submission, ways to download the PDB database, and links to other relevant sites and software.

6.2.1 PDB Query and Reporting

Starting at the RCSB home page, one can retrieve three-dimensional structures using two different query engines. The SearchLite system is the one most often used, providing text searching across the database. The SearchFields interface provides the additional ability to search specific fields within the database. Both of these systems report structure matches to the query in the form of Structure Summary pages, an example of which is shown in Figure 6.1. The RCSB Structure Summary page links are to other Web

pages that themselves provide a large number of links, and it may be confusing to a newcomer to not only sift through all this information but to decide which information sources are the most relevant ones for biological discovery.

6.2.2 Submitting Structures

For those who wish to submit three-dimensional structure information to PDB, the RCSB offers its ADIT service over the Web. This service provides a data format check and can create automatic validation reports that provide diagnostics as to the quality of the structure, including bond distances and angles, torsion angles, nucleic acid comparisons, and crystal packing. Nucleic acid structures are accepted for deposition at NDB, the Nucleic Acids Database. It has been the apparent working policy of PDB to reject three-dimensional structures that result from computational three-dimensional modelling procedures rather than from an actual physical experiment; submitting data to the PDB from a non experimental computational modelling exercise is strongly discouraged.

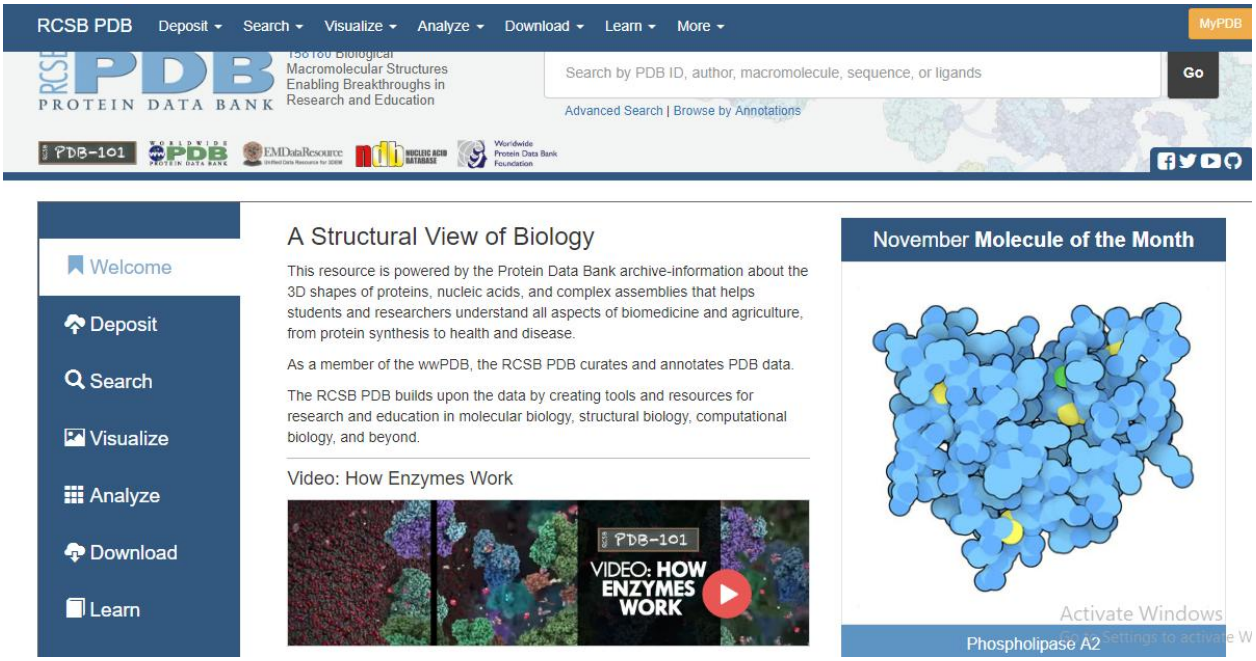


Figure 6.1: How webpage of PDB looks like and other information to explore for exact structure of protein.

6.2.3 PDB-ID Codes

The structure record accessioning scheme of the Protein Data Bank is a unique four-character alphanumeric code, called a PDB-ID or PDB code. This scheme uses the digits 0 to 9 and the uppercase letters A to Z. This allows for over 1.3 million possible combinations and entries. Many older records have mnemonic names that make the structures easier to remember, such as 3INS, the record for insulin shown earlier. A different method is now being used to assign PDB-IDs, with the use of mnemonics apparently being abandoned.

6.2.4 Database Searching, PDB File Retrieval, mmCIF File Retrieval, and Links

PDB's search engine, the Structure Explorer, can be used to retrieve PDB records, as shown in Figure 5.5. The Structure Explorer is also the primary database of links to third-party annotation of PDB structure data. There are a number of links maintained in the Structure Explorer to Internet-based three-dimensional structure services on other Web sites. The Structure Explorer also provides links to special project databases maintained by researchers interested in related topics, such

Links to visualization tool-ready versions of the structure are provided, as well as authored two-dimensional images that can be very helpful to see how to orient a three-dimensional structure for best viewing of certain features such as binding sites.

6.3 MMDB: Molecular Modelling Database at NCBI

NCBI's Molecular Modelling Database (MMDB) is an integral part of NCBI's Entrez information retrieval system. It is a compilation of all the Brookhaven Protein Data Bank three-dimensional structures of biomolecules from crystallographic and NMR studies. MMDB records are in ASN.1 format rather than in PDB format. Despite this, PDB formatted files can also be obtained from MMDB. By representing the data in ASN.1 format, MMDB records have value-added information compared with the original PDB entries. Additional information includes explicit chemical graph information resulting from an extensive suite of validation procedures, the addition of uniformly derived secondary structure definitions, structure domain information, citation matching to MEDLINE, and the molecule-based assignment of taxonomy to each biologically derived protein or nucleic acid chain.

6.3.1 Free Text Query of Structure Records

The MMDB database can be searched from the NCBI home page using Entrez. (MMDB is also referred to as the NCBI Structure division.) Search fields in MMDB include PDB and MMDB ID codes, free text from the original PDB REMARK records, author name, and other bibliographic fields. For more specific, fielded queries, the RCSB site is recommended.

6.3.2 MMDB Structure Summary

MMDB's Web interface provides a Structure Summary page for each MMDB structure record, as shown in Figure 5.9. MMDB Structure Summary pages provide the FASTA-formatted sequences for each chain in the structure, links to MEDLINE references, links to the 3DBAtlas record and the Brookhaven PDB site, links to protein or nucleic acid sequence neighbours for each chain in the structure, and links to VAST structure-structure comparisons for each domain on each chain in the structure.

6.3.3 Structure File Formats

PDB

The PDB file format is column oriented, like that of the punched cards used by early FORTRAN programmers. The exact file format specification is available through the PDB Web site. Most software developed by structural scientists is written in FORTRAN, whereas the rest of the bioinformatics world has adopted other languages, such as those based on C. To the uninitiated, the most obvious problem is that the information about biopolymer bonds is missing, obliging one to program in the rules of chemistry, clues to the identity of each atom given by the naming conventions of PDB, and robust exception handling. PDB parsing software often needs lists of synonyms and tables of exceptions to correctly interpret the information.

Two newer chemical-based formats have emerged: mmCIF (Macro Molecular Chemical Interchange Format) and MMDB (Molecular Modelling Database Format). Both of these file formats are attempts to modernize PDB information. Both start by using data description languages, which are consistently machine parsable. The data description languages use ‘‘tag value’’ pairs, which are like variable names and values used in a

programming language. In both cases, the format specification is composed in a machine-readable form, and there is software that uses this format specification document to validate incoming streams of data. Both file formats are populated from PDB file data using the strategy of alignment-based reconstruction of the implicit ATOM and HETATM chemical graphs with the explicit SEQRES chemical graphs, together with extensive validation, which is recorded in the file. As a result, both of these file formats are superior for integrating with biomolecular sequence databases over PDB format data files, and their use in future software is encouraged.

mmCIF

The mmCIF file format (Figure 6.2) was originally intended to be a biopolymer extension of the CIF (Chemical Interchange Format) familiar small-molecule crystallographers and is based on a subset of the STAR syntax. CIF software for parsing and validating format specifications is not forward-compatible with mmCIF, since these have different implementations for the STAR syntax. The underlying data organization in an mmCIF record is a set of relational tables. The mmCIF project refers to their format specification as the mmCIF dictionary, kept on the Web at the Nucleic Acids Database site. The mmCIF dictionary is a large document containing specifications for holding the information stored in PDB files as well as many other data items derivable from the primary coordinate data, such as bond angles. The mmCIF data specification gives this data a consistent interface, which has been used to implement the NDB Protein Finder,

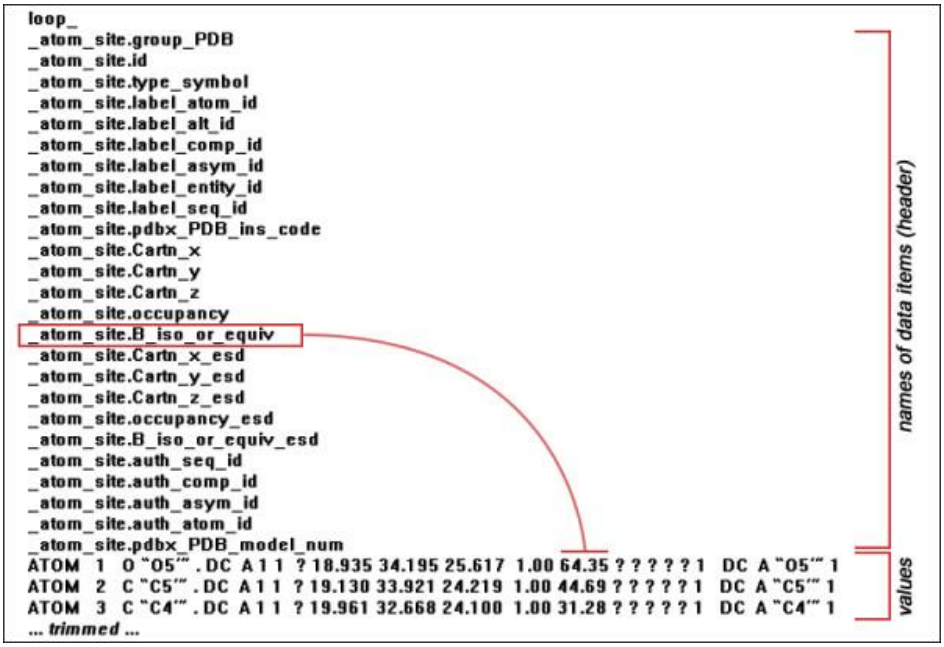


Figure 6.2: A representation how protein structure are arranged in flat file format. (https://www.researchgate.net/figure/The-basic-principles-of-the-mmCIF-yntax_fig1_51045128)

Web-based query format in a relational database style (Figure 5.8), and is also used as the basis for the new RCSB software systems. Validating an incoming stream of data against the large mmCIF dictionary entails significant computational time; hence, mmCIF is probably destined to be an archival.

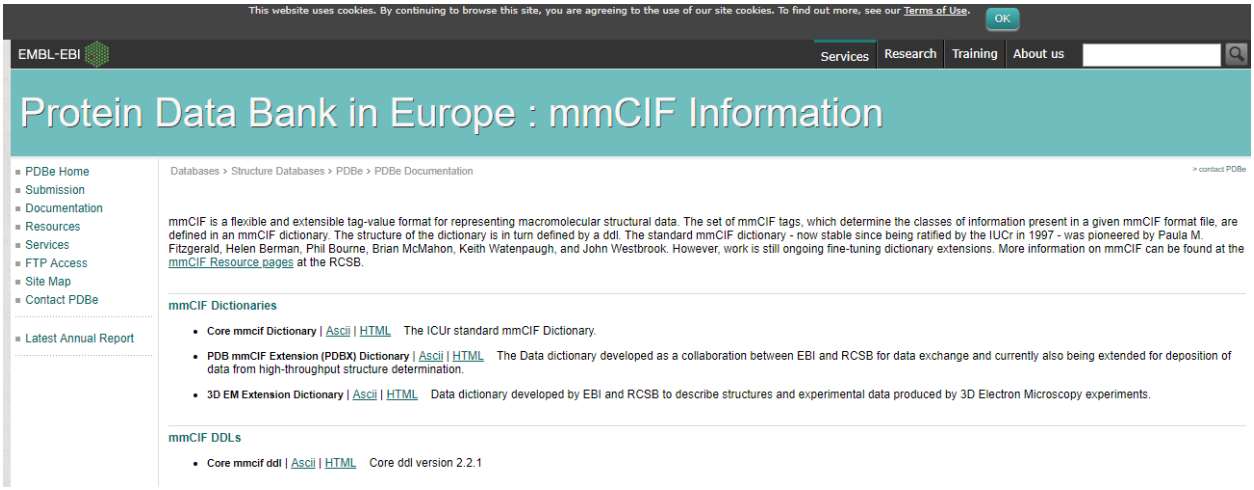


Figure 5.8: Showing a web page of mmCIF, which can be used for retrieval of protein structure.

MMDB

The MMDB file format is specified by means of the ASN.1 data description language, which is used in a variety of other settings, surprisingly enough including applications in telecommunications and automotive manufacturing. Because the US National Library of Medicine also uses ASN.1 data specifications for sequence and bibliographic information, the MMDB format borrows certain elements from other data specifications, such as the parts used in describing bibliographic references cited in the data record. ASN.1 files can appear as human-readable text files or as a variety of binary and packed binary files that can be decoded by any hardware platform. The MMDB standard residue dictionary is a lookup table of information about the chemical graphs of standard biopolymer residue types. The MMDB format (Figure 6.3) specification is kept inside the NCBI toolkit distribution, but a browser is available over the Web for a quick look. The MMDB ASN.1 specification is much more compact and has fewer data items than the mmCIF dictionary, avoiding derivable data altogether.

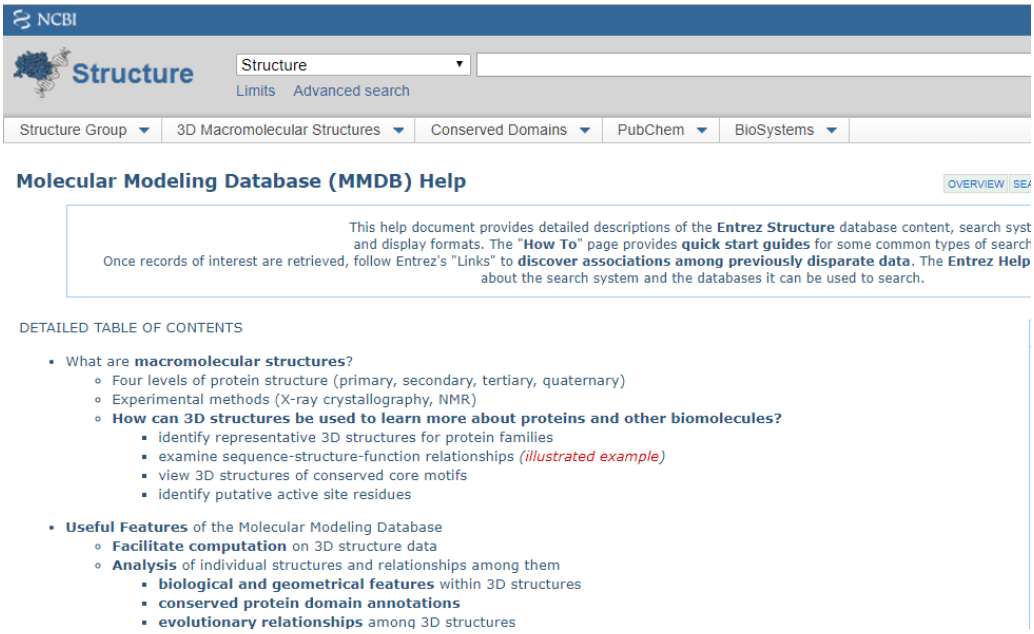


Figure 6.3: Web page of MMDB on NCBI (https://www.ncbi.nlm.nih.gov/Structure/MMDB/docs/mmdb_help.html)

In contrast to the relational table design of mmCIF, the MMDB data records are structured as hierarchical records. In terms of performance, ASN.1-formatted MMDB files

provide for much faster input and output than do mmCIF or PDB records. Their nested hierarchy requires fewer validation steps at load time than the relational scheme in mmCIF or in the PDB file format; hence, ASN.1 files are ideal for three dimensional structure database browsing.

A complete application programming interface is available for MMDB as part of the NCBI toolkit, containing a wide variety of C code libraries and applications. Both an ASN.1 input/output programming interface layer and a molecular computing layer (MMDB-API) are present in the NCBI toolkit. The NCBI toolkit supports x86 and alpha-based Windows' platforms, Macintosh 68K and PowerPC CPUs, and a wide variety of UNIX platforms. The three-dimensional structure database viewer (Cn3D) is an MMDB-API-based application with source code included in the NCBI toolkit.

6.3.4 Retrieval of Structural Information From MMDB

On its own, a protein sequence is an important piece of information, but it becomes more revealing and valuable when you compare it with others from a variety of different species. The role of multiple alignments to identify the most significant regions of a sequence: Conserved residues (or, alternatively, highly variable ones) are often key to predicting or understanding a protein function. Going further, by precisely locating these conserved residues in space, it's possible to come up with additional clues as to their biological roles: Such residues, for example, can delineate a cavity at the active site of an enzyme or, if they are found at the surface, one can assume they are good candidates for interactivity with other molecules, and so on.

We can use the example of the TolB protein family (of relatively unknown function) to illustrate the interplay between multiple alignments and structural analysis. Here's how it's done:

1. First fetch several TolB homologue sequences from various bacterial species from NCBI:

   a. Open a window in your browser and go to www.ncbi.nlm.nih.gov.
   b. Choose Protein from the drop-down Search menu.

c. In the query window, type in the following identifiers: NP_360043 *(R. conorii)*, NP_415268 *(E. coli)*, NP_404737 *(Y. pestis)*, NP_249663 *(P. aeruginosa)*, and NP_438543 *(H. influenzae)*; then click Go.
d. When the answer is returned, change the Display format to FASTA.
e. Finally, use Send to Text to get rid of any parasitic characters.

Five TolB protein sequences are now ready for use. Now we can build a multiple alignment out of these sequences.

2. Open a new browser window and go to www.igs.cnrs-mrs.fr/Tcoffee/.

3. Click on Regular in the very top TCOFFEE option line.

4. Copy the five TolB sequences from the other browser window and paste them into the Tcoffee input window. Be sure to include their FASTA headers.

5. Click Submit (at the bottom of the form). The Tcoffee Results form automatically appears after an intermediary waiting page.

6. Choose score_html from the Multiple Alignment Output menu. This displays the multiple alignment in color with the most reliable regions in red. This part of the alignment is shown in Figure 6.4. It clearly reveals a segment of strictly conserved (invariant) residues shared by all these TolB proteins. Our challenge in the remaining pages of this chapter is to figure out where in the TolB structure these residues are located.

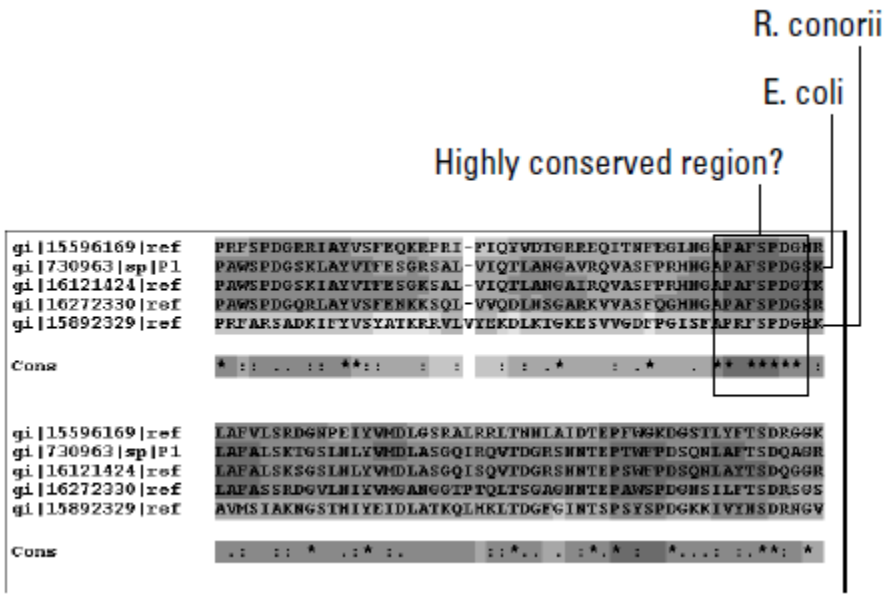


Fig. 6.5: An intriguing segment of strictly conserved residues in the TolB proteins.

Exploring the sequence/PDB structure relationship the interactive way To interpret the significance of an invariant segment in our TolB proteins, we must be able to display its 3-D structure and turn this structure around interactively so we can look at it from any angle. Simultaneously, we must be able to refer to the protein sequence, and to highlight the residues we find interesting.

In short, we need a protein model that we can rotate around and analyze in parallel with its sequence. Not long ago, this kind of business was still the privilege of protein crystallographers working on expensive computers. Not anymore! If you can bear with us until the end of this chapter, you'll be doing it on your own PC in just a few minutes. Just follow our instructions:

1. Point your browser to www.ncbi.nlm.nih.gov/Structure/. The structure server of the NCBI appears.

2. In the Search Entrez Structure/MMDB window at the top of the page, enter the PDB code of your relevant model. (MMDB stands for "Molecular Modeling DataBase.") For our example, we'll use 1CRZ.

If you have the choice of several structures, a simple rule is to choose the one with the best resolution; for our example, the best resolution is 1CRZ (that is, 1.95 Angstroms). You'll find the Angstrom value in the ID card of each PDB entry (refer to Figure 6.5).

3. Click the Go button. Wait for a new page to appear.

4. Click the 1CRZ link. At this point, you should be looking at the MMDB structure card, as shown in Figure 6.5. This MMDB form exhibits a prominent View 3-D Structure button. What will happen next depends on the software already installed on your computer.

5. Click the View 3-D Structure button. Doing so downloads a file with the atomic coordinates of the 1CRZ protein structure. This file is formatted for the 3-D structure viewer Cn3D program. If nobody ever used your computer to display a protein structure in 3-D, a dialog box will pop up, asking you what to do with this file.

It's telling you that your system doesn't contain the Cn3D program required to display the structure. In order to continue, you have to install the program.

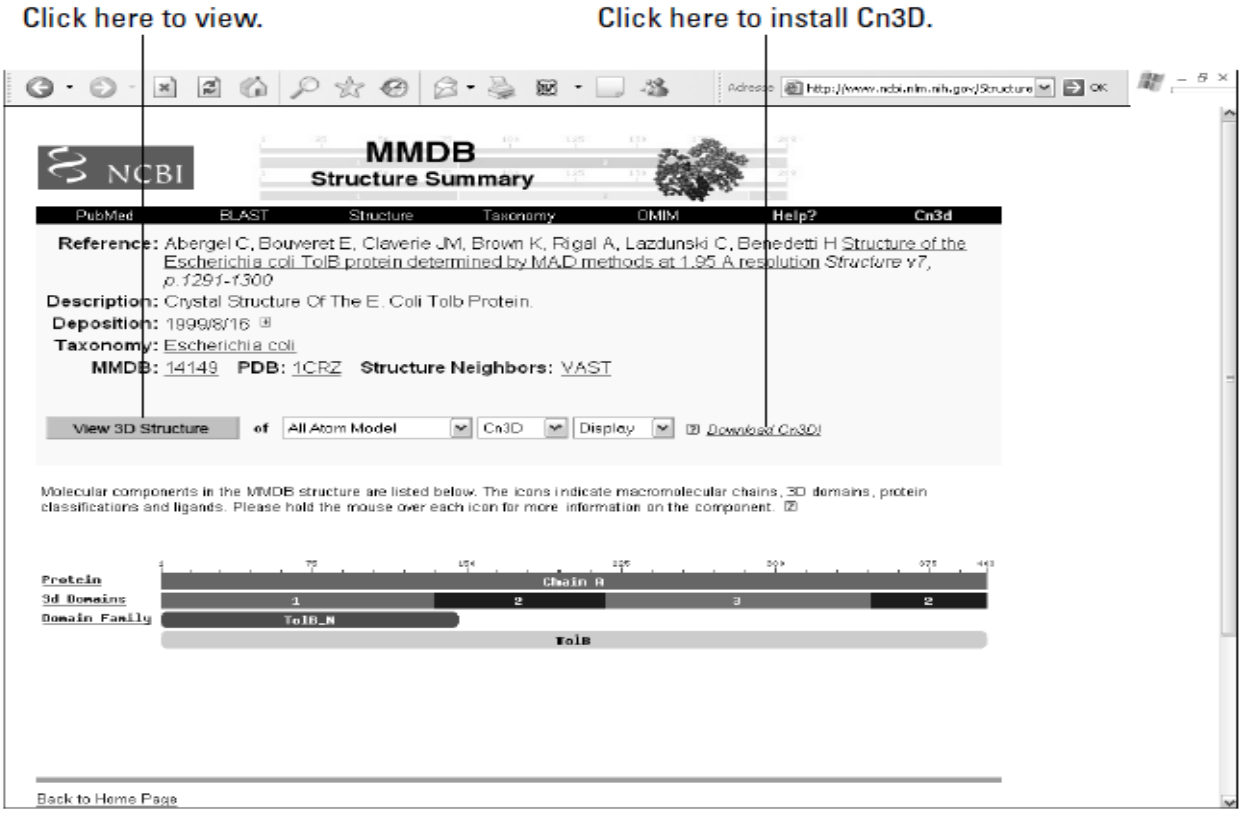


Figure 6.5: MMDB structure card for PDB entry 1CRZ.

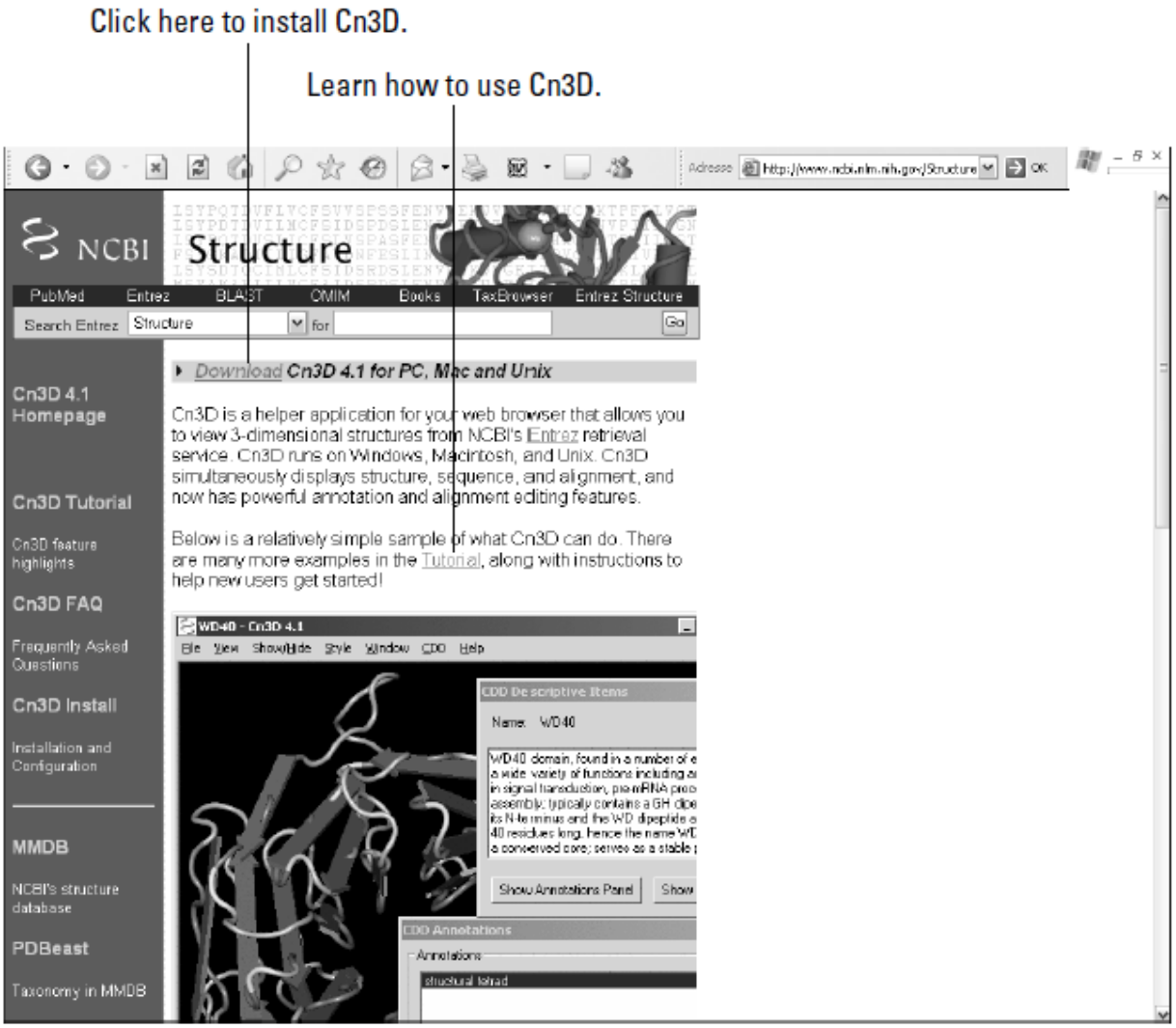


Figure 6.6: The Cn3D home page.

6.3.5 Visualizing Structural Information

Multiple Representation Styles

We often use multiple styles of graphical representation to see different aspects of molecular structure. Typical images of a protein structure are shown in Figure 6.7 (see also color plate). Here, the enzyme barnase 1BN1 appears both in wire-frame and space-filling model formats, as produced by RasMol. Because the protein structure record 1BN1 has three barnase molecules in the crystallographic unit, the PDB file has been hand-edited using a text editor to delete the superfluous chains. Editing data files is an accepted and

widespread practice in three-dimensional molecular structure software, forcing the three-dimensional structure viewer to show what the user wants. In this case, the crystallographic data recorded in the three-dimensional structure does not represent the functional biological unit. In our example, the molecule barnase is a monomer; however, we have three molecules in the crystallographic unit.

The wire-frame image in Figure 6.7a clearly shows the chemistry of the barnase structure, and we can easily trace of the sequence of barnase on the image of its biopolymer in an interactive computer display. The space-filling model in Figure 6.7b gives a good indication of the size and surface of the biopolymer, yet it is difficult to follow the details of chemistry and bonding in this representation. The composite illustration in Figure 6.7c shows an carbon backbone in a typical pseudo-structure representation. The lines drawn are not actual chemical bonds, but they guide us along the path made by the carbons of the protein backbone. These are also called virtual bonds.'' The purple tryptophan side chains have been selected and drawn together with a dot surface. This composite illustration highlights the volume taken up by the three tryptophan side chains in three hydrophobic core regions of barnase, while effectively hiding most of the structure's details.

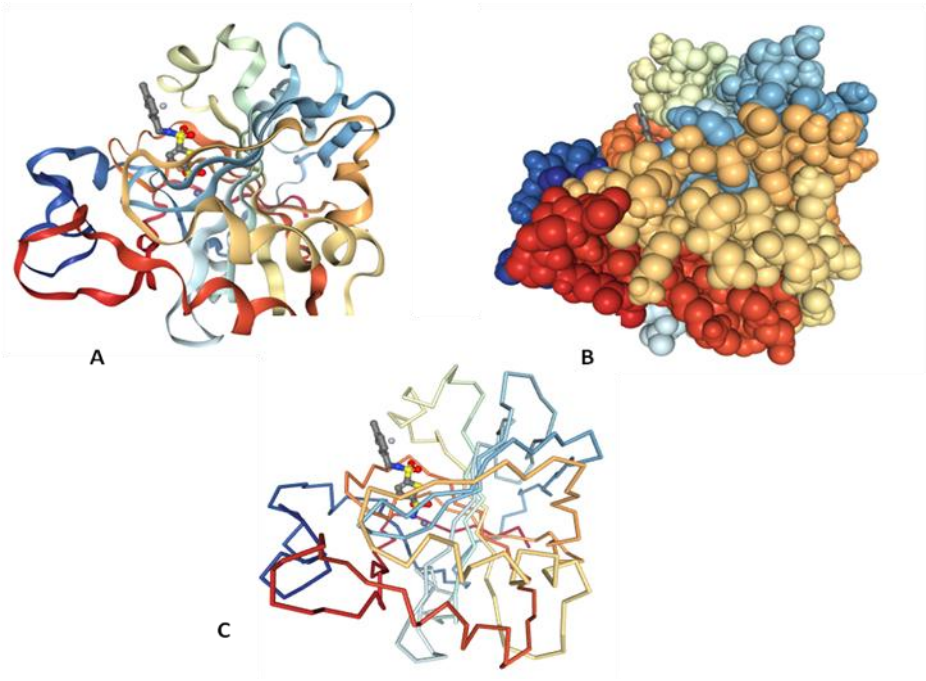


Figure 6.7: Showing different structure types of views of 1BN1 where in A) is ribbon type, B) is Space fill and C) is Lines viewing style.

The ribbon model in Figure 6.7a shows the organization of the structural path of the secondary structure elements of the protein chain (helix and sheet regions). This representation is very often used, with the arrowheads indicating the N-to-C-terminal direction of the secondary structure elements, and is most effective for identifying secondary structures within complex topologies. The variety of information conveyed by the different views in Figure 5.10 illustrates the need to visualize three-dimensional biopolymer structure data in unique ways are not common to other three-dimensional graphics applications. This requirement often precludes the effective use of software from the ‘‘macroscopic world,’’ such as computer-aided design (CAD) or virtual reality modelling language (VRML) packages.

Picture the Data: Populations, Degeneracy, and Dynamics

Both X-ray and NMR techniques infer three-dimensional structure from a synchronized population of molecules—synchronized in space as an ordered crystal lattice or synchronized in behavior as nuclear spin states are organized by an external magnetic field. In both cases, information is gathered from the population as a whole. The coordinate (x, y, z) locations of atoms in a structure are derived using numerical methods. These fit the expected chemical graph of the sample into the three-dimensional data derived from the experimental data. The expected chemical graph can include a mixture of biopolymer sequence-derived information as well as the chemical graph of any other known small molecules present in the sample, such as substrates, prosthetic groups, and ions.

One somewhat unexpected result of the use of molecular populations is the assignment of degenerate coordinates in a database record, i.e., more than one coordinate location for a single atom in the chemical graph. This is recorded when the population of molecules has observable conformational heterogeneity.

6.4 Database Structure Viewers

In the past several years, the software used to examine and display structure information has been greatly improved in terms of the quality of visualization and, more importantly, in terms of being able to relate sequence information to structure information.

6.4.1 Visualization Tools

Although the RCSB Web site provides a Java-based three-dimensional applet for visualizing PDB data, the applet does not currently support the display of non protein structures. For this and other reasons, the use of RasMol v2.7 is instead recommended for viewing structural data downloaded from RCSB; more information on RasMol appears in the following section. If a Java-based viewer is preferred. With the advent of many homemade visualization programs that can easily be downloaded from the Internet, the reader is strongly cautioned to only use mature, well-established visualization tools that have been thoroughly tested and have undergone critical peer review.

6.4.2 RasMol and RasMol-Based Viewers

As mentioned above, several viewers for examining PDB files are available. The most popular one is RasMol. RasMol represents a breakthrough in software-driven three-dimensional graphics, and its source code is a recommended study material for anyone interested in high-performance three-dimensional graphics. RasMol treats PDB data with extreme caution and often recomputes information, making up for inconsistencies in the underlying database. It does not try to validate the chemical graph of sequences or structures encoded in PDB files. RasMol does not perform internally either dictionary-based standard residue validations or alignment of explicit and implicit sequences. RasMol 2.7.1 contains significant improvements that allow one to display information in correlated disorder ensembles and select different NMR models. It also is capable of reading mmCIF-formatted three-dimensional structure files and is thus the viewer of choice for such data. Other data elements encoded in PDB files, such as disulfide bonds, are recomputed based on rules of chemistry, rather than validated.

RasMol contains many excellent output formats and can be used with the Molscript program to make wonderful PostScript_ ribbon diagrams for publication. To make optimal

use of RasMol, however, one must master its command- line language, a familiar feature of many legacy three-dimensional structure programs. Several new programs are becoming available and are free for academic users. Based on RasMol's software-driven three-dimensional-rendering algorithms and sparse PDB parser, these programs include Chime_, a Netscape_ plug-in. Another program, WebMol, is a Java-based three-dimensional structure viewer apparently based on RasMol-style rendering.

6.4.3 MMDB Viewer: Cn3D

Cn3D (for ''see in 3-D'') is a three-dimensional structure viewer used for viewing MMDB data records. Because the chemical graph ambiguities in data in PDB entries have been removed to make MMDB data records and because all the bonding information is explicit, Cn3D has the luxury of being able to display three-dimensional database structures consistently, without the parsing, validation, and exception-handling overhead required of programs that read PDB files. Cn3D's default image of a structure is more intelligently displayed because it works without fear of misrepresenting the data. However, Cn3D is dependent on the complete chemical graph information in the ASN.1 records of MMDB, and, as such, it does not read in PDB files.

Cn3D 3.0 has a much richer feature set than its predecessors, and it now allows selection of subsets of molecular structure and independent settings of rendering and coloring aspects of that feature. It has state-saving capabilities, making it possible to color and render a structure, and then save the information right into the ASN.1 structure record, a departure from the hand-editing of PDB files or writing scripts. This information can be shared with other Cn3D users on different platforms. This provides graphics for publication-quality images that are much better than previous versions, but the original Viewer3D version of Cn3D 3.0 is available for computers that are not capable of displaying OpenGL or that are too slow.

Also unique to Cn3D is a capacity to animate three-dimensional structures. Cn3D's animation controls resemble tape recorder controls and are used for displaying quickly the members of a multiple structure ensemble one after the other, making an animated three-dimensional movie. The GO button makes the images animated, and the user can rotate or

zoom the structure while it is playing the animation. This is particularly useful for looking at NMR ensembles or a series of time steps of structures undergoing motions or protein folding. The animation feature also allows Cn3D to provide superior multiple structure alignment.

6.4.4 Other 3D Viewers: Mage, CAD, and VRML

A variety of file formats have been used to present three-dimensional biomolecular structure data lacking in chemistry-specific data representations. These are viewed in generic three-dimensional data viewers such as those used for ‘‘macroscopic’’ data, like engineering software or virtual-reality browsers. File formats such as VRML contain three-dimensional graphical display information but little or no information about the underlying chemical graph of a molecule. Furthermore, it is difficult to encode the variety of rendering styles in such a file; one needs a separate VRML file for a space-filling model of a molecule, a wire-frame model, a ball-and-stick model, and so on, because each explicit list of graphics objects (cylinders, lines, spheres) must be contained in the file.

Biomolecular three-dimensional structure database records are currently not compatible with ‘‘macroscopic’’ software tools such as those based on CAD software. Computer-aided design software represents a mature, robust technology, generally superior to the available molecular structure software. However, CAD software and file formats in general are ill-suited to examine the molecular world, owing to the lack of certain ‘‘specialty’’ views and analytical functions built in for the examination of details of protein structures.

6.4.5 Making Presentation Graphics

To get the best possible publication-quality picture out of any molecular graphics software, first consider whether a bitmap or a vector-based graphic image is needed. Bitmaps are made by programs like RasMol and Cn3D—they reproduce exactly what you see on the screen, and are usually the source of trouble in terms of pixellation, a bitmap of 380–400 pixels. High quality print resolution is usually at 300–600 dots per inch, but monitors have far less information in pixels per inch (normally 72 dpi), so a big image on a

screen is quite tiny when printed at the same resolution on a printer. Expanding the image to fit a page causes exaggeration of pixel steps on diagonal lines.

The best advice for bitmaps is to use as big a monitor/desktop as possible, maximizing the number of pixels included in the image. This may mean borrowing a colleagues' 21-in monitor or using a graphics card that offers a ''virtual desktop'' that is larger than the monitor being used in pixel count. In any case, always fill the entire screen with the viewer window before saving a bitmap image for publication.

The Vector Alignment Search Tool (VAST) provides a similarity measure of three-dimensional structure. It uses vectors derived from secondary structure elements, with no sequence information being used in the search. VAST is capable of finding structural similarities when no sequence similarity is detected. VAST, like BLAST, is run on all entries in the database in an N- N manner, and the results are stored for fast retrieval using the Entrez interface. More than 20,000 domain substructures within the current three-dimensional structure database have been compared with one another using the VAST algorithm, the structure-structure superpositions recorded, and alignments of sequence derived from the superposition. The VAST algorithm focuses on similarities that are surprising in the statistical sense. One does not waste time examining many similarities of small substructures that occur by chance in protein structure comparison. For example, very many small segments of sheets have obvious, but not surprising, similarities. The similarities detected by VAST are often examples of remote homology, undetectable by sequence comparison. As such, they may provide a broader view of the structure, function, and evolution of a protein family.

The VAST system stands out amongst these comparative tools because:

a) It has a clearly defined similarity metric leading to surprising relationships
b) It has an adjustable interface that shows nonredundant hits for a quick first look at the most interesting relationships, without seeing the same relationships lots of times
c) It provides a domain-based structure comparison rather than a whole protein comparison

d) It has the capability to integrate with Cn3D as a visualization tool for inspecting surprising structure relationships in detail.

In addition to a listing of similar structures, VAST-derived structure neighbours contain detailed residue-by-residue alignments and three-dimensional transformation matrices for structural superposition. In practice, refined alignments from VAST appear conservative, choosing a highly similar ‘‘core’’ substructure compared with DALI super positions. With the VAST superposition, one easily identifies regions in which protein evolution has modified the structure, whereas DALI super positions may be more useful for comparisons involved in making structural models. Both VAST and DALI super positions are excellent tools for investigating relationships in protein structure, especially when used together with the SCOP database of protein families.

6.5 Modelling And Drug Designing

Experimental molecular biology research is often a painstakingly slow process that typically involves a long sequence of carefully performed experiments, using a variety of equipment and laboratory specialists. For example, positively identifying a protein by structure may take years of work. The protein must be isolated, purified, crystallized, and then imaged. Because each step may involve dozens of failed attempts, many scientists not primarily interested in the experimental methods, but simply needing the structure data, look to other non-experimental methods. In determining protein structure, the primary alternative to experimental or wet-lab techniques is bioinformatics.

A wet lab technique is one when drugs, chemicals and other types of biological matter can be analyzed and tested by using various liquids. On the other hand, a dry lab environment focuses more of applied or computational mathematical analysis via the creation of computer model or simulation.

Although computational methods may be able to deliver a solution to a molecular biology problem such as structure determination in days or weeks instead of months or years, the solution is only as good as the formulation of the problem. In the case of protein structure determination or prediction, formulating the problem entails creating a model of the molecule and the major environmental factors that may influence its structure. With a

valid model definition, arriving at a solution—that is, using the model to drive a simulation of the molecule's behavior and structure—is simply a matter of executing a program and then evaluating the results. In order to appreciate the significance of modelling and simulation in bioinformatics, consider that the first "killer app" on the desktop micro computer, the one application that raised the status of the technology from a hobbyist's plaything to a "must have" in business and in the laboratory—was the now-defunct electronic spreadsheet, VisiCalc. This spreadsheet enabled accountants, engineers, and physicists to interactively run a variety of what-if scenarios or implicit attempts at problem formulation to predict the outcomes of virtually any activity that they could express mathematically. Equipped with a spreadsheet and a few equations, a molecular biologist might define a model of a neural network that can learn to recognize amino acid sequences and assign protein structures to certain sequences, as in Figure 6.8. The model of a single neuron in the artificial neural network defines the output of the neuron as the weighted sum of inputs to the neuron, including feedback from the output: The model of the entire neural network additionally specifies the interconnection of the individual neuron models. Mathematically, the model of an individual neuron that can accept four outputs (o) with their associated fixed weights (w) can be expressed as:

$$Output = \sum_{i=0}^{4} o_i w_i$$

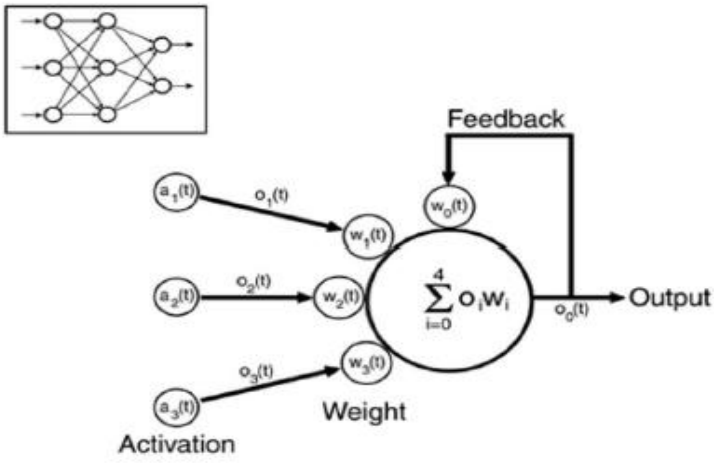


Fig. 6.1: Model of a Single Neuron.

This model is used in the simulation of a neural network (inset) that can be used to classify patterns, such as protein structures associated with specific amino acid sequences. The model and associated simulation can be created in a general-purpose spreadsheet or

in a computational environment specifically designed for the simulation of neural networks.

Note that this model for an individual neuron is a greatly simplified representation of the function of an actual neuron in the human nervous system. For example, neurons in the brain are regularly bathed in substances—from naturally produced endorphins to drugs such as seratonin release inhibitors—that dynamically alter the strength of connections, represented by the fixed weights (w) in the neuron model. The advantage of ignoring the intricacies of the actual nervous system is computational efficiency and lower overhead associated with developing a model. A simpler model is also easier to develop and maintain compared to developing and maintaining a more complex model.

The challenge is defining a model that is simple enough computationally and yet is rich enough to accurately define the behavior of the system. Although spreadsheets are still used for modelling and simulation applications in business, science, and engineering, all but the simplest modelling is performed with software optimized for particular domains. For example, nuclear physicists use custom modelling and simulation programs running on supercomputers to simulate the power of nuclear explosions. Similarly, life scientists use a variety of microcomputer-based simulations to explore everything from population dynamics to the docking of proteins.

The downside of using a general-purpose spreadsheet as a platform for modelling and simulation is related to performance, flexibility, visualization capabilities, standards, and startup time. A general purpose spreadsheet, like a general-purpose language such as eXtensible Markup Language (XML) or C++, is designed to solve a variety of problems. As such, it represents a compromise between flexibility and performance. Although a spreadsheet can be used to prototype virtually any type of simulation, the simulation will likely run several orders of magnitudes slower than a simulation developed in an environment designed for modelling and simulation.

Similarly, coding a simulation in C++ may result in a system with a higher performance than can be obtained with a dedicated simulation system. However, the startup time associated with a domain specific simulation will likely be several orders of

magnitude lower that that associated with the general-purpose language. For example, classification systems based on a neural network simulation are typically outperformed by classification systems developed in C++ or some other compiled language. However, creating a classification system with a neural network system may take only minutes. Neural network systems typically provide a library of predefined models that the user can incorporate in a neural network by connecting icons graphically instead of making extensive use of mathematical equations. Like using a high-level programming language, there is no need to develop or even fully understand low-level neuron model operation in these systems to create functional classifiers.

Even if a general-purpose language is used to develop a simulation, there are numerous reasons for going through the time and hassle of developing a model of a real-world system. Simulations allow conditions in the real world to be evaluated in compressed or expanded time and under a variety of conditions that would be too dangerous, too time-consuming, occur too infrequently, or that would otherwise be impractical in the real world. Instead of taking days or weeks to set up and run a series of biological experiments on the population dynamics of yeast under a variety of environmental conditions, the effect of, for example, an increase in temperature, can be explored in a few minutes through a simulation.

Common uses of modelling and simulation include predicting the course and results of certain actions, and exploring the changes in outcome that result when actions are modified. Several bioinformatics R&D groups are focused on developing simulation-based systems to determine, for example, if a candidate molecule for a new drug will exhibit toxicity in patients before money is invested in actually synthesizing the drug. In this regard, simulation is a means of identifying problem areas and verifying that all variables are known before construction of the drug development facility is begun. As an analysis tool, simulations help to explain why certain events occur, where there are inefficiencies, and whether specific modifications in the system will compensate for or remove these inefficiencies. As listed in Table 6.1, the range of possible applications of modelling and simulation in bioinformatics is extensive. These applications range from understanding basic metabolic pathways to exploring genetic drift. One of the most promising application

areas of modelling and simulation in bioinformatics and the most heavily funded is as a facilitator of drug discovery, which in turn depends on modelling and simulating protein structure and function. Given the exponentially increasing rate, at which models of proteins are being added to the Protein Data Bank (PDB), modelling and simulation of proteins and their interaction with other molecules are the most promising means in our lifetimes of linking protein sequence, structure, function, and expression with the clinical relevance of the proteome.

Table 6.1: Now a day's sample of the applications of Modelling and simulation are found in the different fields which are given below:

- Bioinformatics.
- Clinical What-If Analysis
- Drug Discovery and Development
- Experimental Toxicology
- Exploring Genetic Drift
- Exploring Molecular Mechanisms of Action
- Personal Health Prediction
- Drug Efficacy Prediction
- Drug Side-Effects Prediction
- Gene Expression Prediction
- Protein Folding Prediction
- Protein Function Prediction
- Protein Structure Prediction
- Vaccine Discovery

Pharma, the primary backer of bioinformatics R&D worldwide, is keenly interested in automating and speeding the drug discovery and development process. The typical drug discovery and development process, shown in Figure 6.9, involves an often arduous series of events that starts with perhaps 5,000 candidate drug molecules and ends with a single product that can be brought to market.

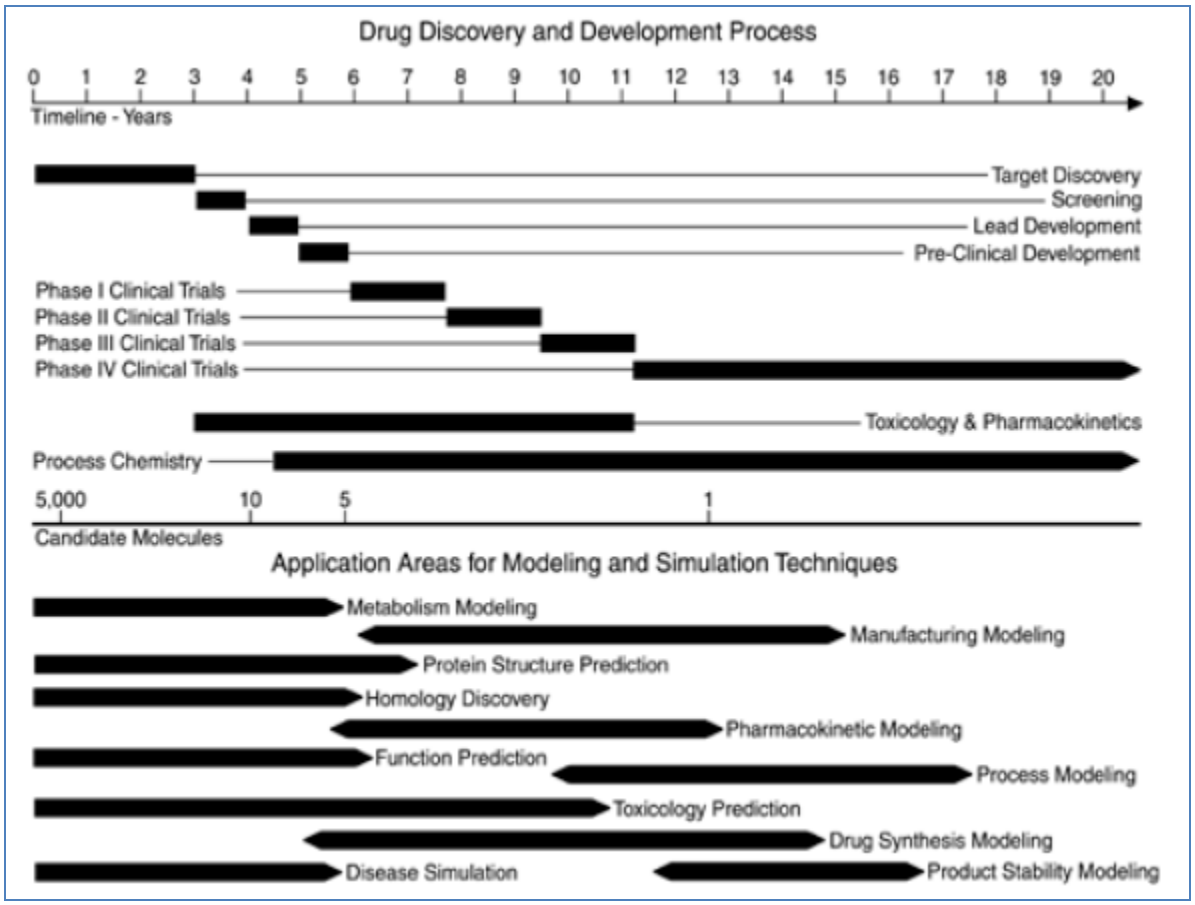


Fig. 6.9: The Drug Discovery and Development Process (top) and Application Areas of Modelling and Simulation (bottom), save the industry billions of dollars, there is considerable R&D involved in replacing or supplementing the drug discovery process with modelling and simulation.

A better understanding of the underlying metabolism of a particular disease or condition can suggest which molecules will be most effective for treatment, and which ones may cause toxic reactions in a patient. Similarly, assuming that protein molecules with similar structure also have similar function, modelling protein structure and comparing it with known drugs can potentially serve as a more effective screener for candidate drugs, compared to wet-lab techniques.

Later in the drug discovery process, modelling and simulation of pharmacokinetics and of drug absorption can potentially be used to shorten clinical trials. Currently, each

phase of the clinical trials takes a year or more. Phase I, involving about 100 subjects, deals with safety. Phase II, which involves about 200 subjects, deals with evidence for efficacy at various dosages. Phase III, involving up to about 5,000 subjects, deals with assessing the clinical value of a molecule. Phase IV, which begins with the release of the drug, involves monitoring patients for adverse reactions. The FDA approves only about 1 molecule in 5 that makes it to Phase I clinical trials.

As illustrated in the bottom half of Figure 6.2, modelling and simulation techniques can also be applied to various aspects of the drug development process. For example, process modelling can be used, starting around years of the drug discovery and development process, to develop the most efficient and cost-effective development processes. Similarly, the manufacturing process can be modelled to determine the best use of materials, product stability, and best method of product synthesis—all without modifying the actual process. Before delving into one of the key modelling and simulation areas, protein structure determination and prediction, consider the following review of the fundamentals of modelling and simulation.

6.5.1 Fundamentals of Modelling

The numerous potential applications of modelling and simulation in the drug discovery process illustrate that whether the intent is to predict the toxicity of a candidate drug or to streamline the screening process, the fundamental components and processes are identical. However, as described here, the drug discovery process also illustrates how there are also domain and implementation specific issues, including numeric considerations, selecting the most optimum algorithms for a given problem, determining which simulation perspective best fits the problem, and hardware requirements.

6.5.2 Components for Modelling

Every modelling and simulation system is composed of a model, a database, a simulation engine, and a visualization engine. The user and some form of feedback device, such as a computer monitor, are normally considered key elements as well. These components aren't necessarily separate entities as illustrated in Figure 6.10, but may be combined and integrated in various ways. For example, the model and data may be

combined within the simulation engine, or the simulation engine and visualization engine may be combined. Regardless of how they are represented in a system, each component is necessary for operation of the simulation.

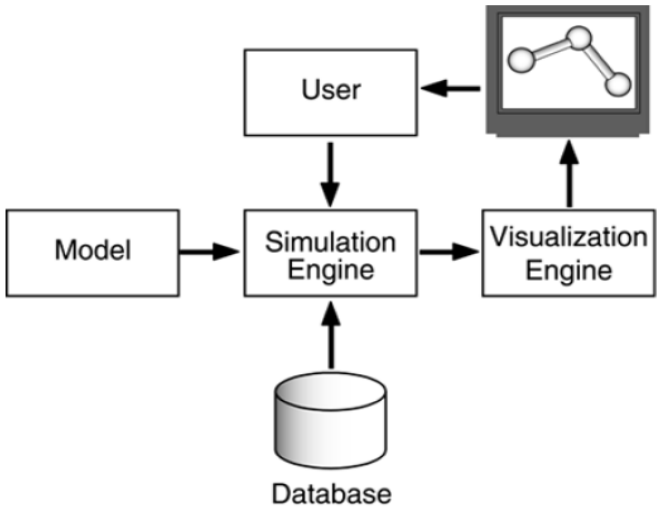


Figure 6.10: Components of a modelling and simulation system include a model, database, simulation engine, and visualization engine.

The components of a simulation system typically vary in form, complexity, and completeness, as a function of what is being modeled and the required fidelity of the simulation. For example, the model, which can be a mathematical equation, a logical description encoded as rules, or a group of algorithms that describes objects and their interrelationships in the real world, defines the underlying nature of the simulation. The database may take the form of a few lines of data imbedded as statements within the model code, or consist of a separate text file that describes variables and constants that can be used with the underlying model. However, in most bioinformatics applications, the database consists of a large, complex system that contains libraries of data that can be applied to the underlying model. The contents of the database typically range from physical constants, such as the bond lengths of covalently bound atoms, to user-defined input, such as heuristics regarding situations in which the underlying model can be applied.

The simulation engine consists of functions that are evaluated over time, and triggered by time, events, or the value of intermediate simulation results. The simulation engine takes the model, data from the database, and direction from the user to create an output that corresponds to a condition in the real world, such as a description of the folding of a protein molecule in an aqueous solution. Finally, the visualization engine takes the output of the simulation engine and formats it into a more user-friendly form. For example, a string of digits can be formatted into a 3D rendition of a protein structure. The visualization engine may be little more than a text-formatting utility or it can take the form of a high-performance, real-time, high-resolution 3D rendering engine.

6.5.3 Process involved in Modelling

The basic modelling and simulation process outlined in Figure 6.11 is applicable to most problems in bioinformatics. The first step is to define the problem space, such as predicting protein structure from amino acid sequence data—one application of modelling among the many depicted by Figure 6.11. Defining the problem space involves specifying the objectives and requirements of the simulation, including the required accuracy of results. This phase of the process also involves establishing how an observer in some experimental frame observes or interacts with some part of reality. The experimental frame defines the set of conditions under which a system will be observed, including initial states, terminal conditions, specifications for data collations, and observable variables and their magnitudes. The system represents a collection of objects, their relationships, and the behaviors that characterize them as some part of reality. The underlying assumption in defining the problem space is that the phenomenon or problem to be modeled can be positively identified and measured.

Once the problem space has been defined, the next phase in the modelling and simulation process is conceptual modelling, which involves mapping the systems objects, relationships, processes, and behaviours to some sort of organized structure. For example, in predicting protein structure from sequence data, the conceptual modelling might entail using ab initio methods—that is, working from first principles, such as bond lengths and angles—to construct the protein's secondary and tertiary structures.

Activities at this phase of the process also include documenting assumptions about the system so that the appropriate simulation methods can be selected. For example, if ab initio methods are going to be used to predict protein structure from sequence data, then an underlying assumption is that the data on amino acid sequence, bond length, bond angles, and related atomic-level data are not only available, but the data are accurate to some verifiable level.

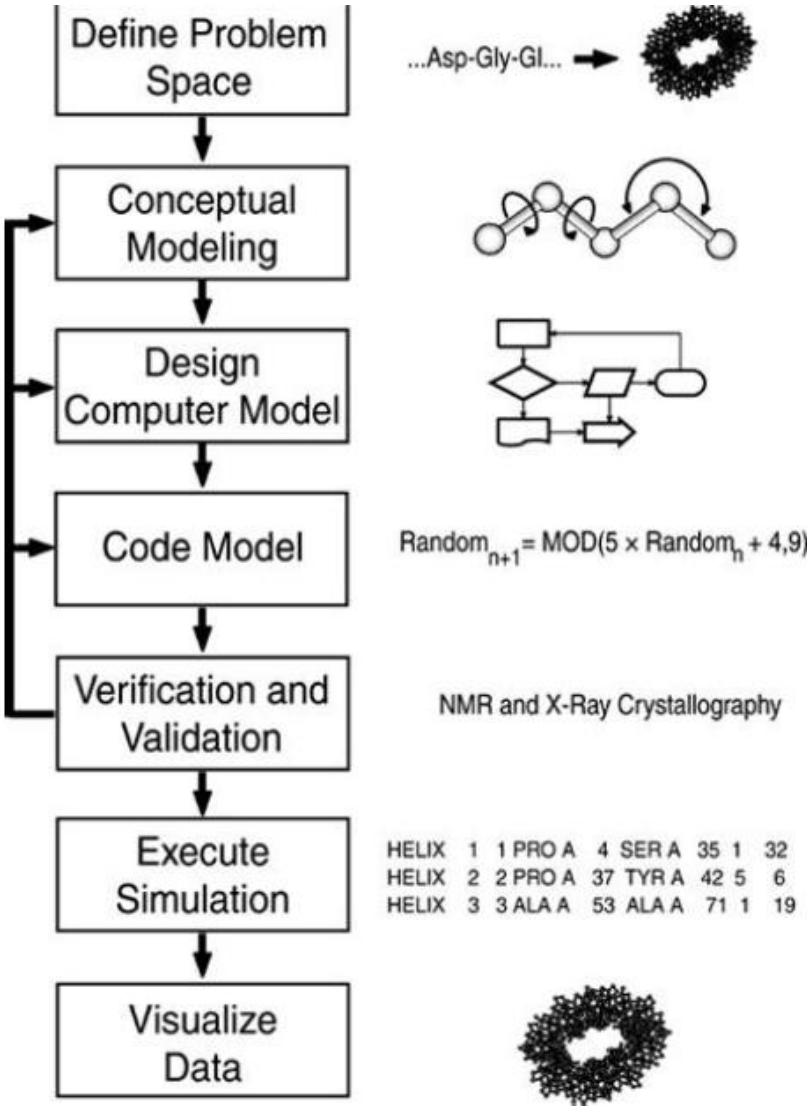


Fig. 6.11: The Modelling and Simulation Process.

Given the underlying data and a conceptual model, the next phase of the modelling and simulation process is translating the conceptual model into data structures and high-

level descriptions of computational procedures. Designing the computer model involves extracting from the conceptual model only those characteristics of the original system that are deemed essential, as determined by the model's ultimate purpose. For example, the purpose of predicting protein structure from sequence data may be to allow the end-user to visualize the protein structure, so that a high degree of accuracy isn't that essential. In this example, the purpose of the model is to simplify and idealize, and the characteristics selected from the conceptual model should reflect this purpose.

Designing the computer model, like defining the problem space and conceptual modelling, is largely an art. Designing a simple model that adequately mimics the behaviour of the system or process under study is a creative process that incorporates certain assumptions. The art of making good assumptions may well be the most challenging component of modelling, considering success depends as much as on the domain experience of the modeller as it does on the nature of the system to be modelled. Biological systems are seldom presented in a quantitative manner, often requiring that the model designer derive or invent the needed mathematical formalisms or heuristics.

Coding of the computer model involves transferring the symbolic representations of the system into executable computer code. Model coding marks the transition of the modelling process from an artistic endeavor to a predominantly scientific one, defined by software engineering principles. Model coding may involve working with a low-level computer language, such as C++, or a high-level shell designed specifically for modelling and simulation. Once a model is in the form of executable code, it should be subject to verification and validation. Verification is the process of determining that the model coded in software accurately reflects the conceptual model by testing the internal logic of a model to confirm that it is functioning as intended, for example. The simulation system and its underlying model are validated by assessing whether the operation of the software model is consistent with the real world, usually through comparison with data from the system being simulated. For example, in a system designed to predict protein structure, the validation process would include comparing model data with protein structure data from NMR spectroscopy and X-ray crystallography. Validating X-ray crystallography data might involve comparing it with the pattern resulting from bombarding the crystal lattice

of a purified protein with X-rays. In contrast, validating NMR data might involve comparing it with actual data produced by scanning a pure protein in solution.

Validation also involves certifying that the output of the system as a whole is adequate for the

Intended purpose and is consistent with the presumptions of expert opinion. As such, validation is at least in part a subjective call. The validity of a model is a function of the objectives of the model designer and the context of its application. For example, the usefulness of a model of protein structure for a decision-making application is a function of the accuracy of prediction. There are no concepts such as "best" or "correct" in model validity assessment, considering that the degree to which a model needs to reflect or mimic a real-world system varies with each case. In addition, because verification is a check for internal consistency, it's possible for a model to be verifiable and yet fail validation because of errors in the conceptual model.

Executing the simulation ideally generates the output data that can illustrate or answer the problem initially identified in the problem space. Depending on the methods used, the amount of process and time required to generate the needed data may be extensive. For example, predicting protein structure using ab initio methods can involve thousands of iterations and take days of supercomputer time in order to arrive at statistically reliable results. Visualizing the output data opens the simulator output to human inspection, especially if the output is in the form of 3D graphics that can be assessed qualitatively instead of in tables of textual data. For example, even though the structure of a protein may be described completely in a text file that follows the PDB format, the data take on more meaning when they can be visualized as a 3D structure that can be rotated in 3D space using a visualization program such as RasMol, Chimera, or SWISS-PDBViewer. Data are typically subject to numerical analysis as well as visualization, in order to provide a quantitative measure of accuracy and to determine whether the underlying model needs to be improved upon.

6.6 Simulation and its types

6.6.1 Continuous Simulation

Continuous simulation refers to a computer model of a physical system that continuously tracks system response according to a set of equation typically involving different equation.

The continuous simulation methods are most appropriate when what is of primary interest is the time varying nature of objects or processes in some real-world system. The variables in a continuous model are assumed to vary continuously with advancing time. Because there is no instant of time when the system is not in flux, continuous simulations are said to be time-driven. Behaviour patterns modelled as a mixture of differential and algebraic equations provide the basis for this simulation perspective.

A differential equation defines a relationship between a continuous variable and its own rate of change. To take an example from pharmacokinetics, consider the time-varying nature of the plasma level of a drug ingested. Given the initial concentration of the drug in the body, the time since the drug was ingested, and the rate at which the drug is absorbed in the gut, we can model the current concentration of drug in the body with the following relationship:

$$[Drug]_{plasma} = \frac{Dose}{Volume_{plasma}} \times e^{-KT}$$

In this equation, the fraction of the drug lost from the plasma per-unit time is represented by KT, where K is a constant and T is time. The elimination of constant K is a function of the type of drug administered, administration route, method of elimination or conversion, health of the patient, and renal function. Drugs with a large number for K will be eliminated faster from the body than those with a smaller number for K.

When this model of drug elimination is coded, the formula for drug plasma concentration becomes a DO LOOP in which the value for T is incremented by an appropriate value, dt, with each loop. Depending on what is being studied, dt might be 1 millisecond or 10 seconds. In pseudocode form, the solution to the preceding equation during the first 100 seconds after drug administration, assuming an initial dose of 1000 milligrams, a plasma volume of 6000 ml, an elimination constant of 0.4, and a dt of 0.1 seconds, appears as:

```
DOSE = 1000

PLASMA VOLUME = 6000

T = 0

K = 0.4

DT = 0.1

FOR INDEX = 1 TO 1000

DO

CONCENTRATION = (DOSE/plasma volume) x EXP(–K x T)

T = T + DT

LOOP
```

This differential equation is solved by advancing time in relatively small increments dt and recomputing the continuous variable concentration at each step. Larger steps may be taken to decrease computation time at the expense of greater approximation error. Termination of the program occurs after 1000 iterations of the DO LOOP. However, termination could also be linked to a maximum runtime, or a maximum or minimum concentration, or some combination of the two. The drug concentration, as described in the preceding differential equation, isn't limited to integer values, but is instead most accurately expressed in real values, such as 3.457 mg per ml. When run, the output of the simulation results in a plasma drug concentration that initially decreases rapidly and then more slowly as the concentration approaches zero.

6.6.2 Discrete Simulation

A discrete event simulation perspective lends itself to modelling systems in which an object or process arrives at a stage, waits in a queue until it receives attention, and then moves on to the next stage. Discrete event simulation is characterized by relatively large quantities of time during which the underlying system doesn't change. Advancing the simulation from one event to the next simulates time. Another characteristic of discrete

methods is that the progresses of objects or processes moving through the system are typically measured as integers.

6.6.3 Hybrid Simulation

Hybrid simulation is a testing method for examining the seismic response of structure using a hybrid model comprised of both physical and numerical substation.

Hybrid simulation methods are useful when the system to be modelled displays a variety of behaviours, some of which lend themselves to discrete event methods, and some of which are more easily solved through continuous simulation techniques. Consider the challenges faced by a modeller attempting to simulate a complex neuromuscular system involving individual packets of neurotransmitter substances, receptor sites, and resulting muscular contraction. Describing the release, transport, and subsequent absorption of neurotransmitter packets might be most easily mapped in a discrete event model. The resulting time-varying contraction, however, is likely to be most easily described in terms of differential equations within a continuous simulation model.

In general, any system can be simulated with models adhering to continuous, discrete, or hybrid perspectives. However, the perspective that most closely maps to the actual system characteristics will minimize development effort. The optimal modelling perspective is also a function of the characteristic of the system to be modeled. For example, a system can be modelled with discrete and continuous methods, with each method answering a different question. In addition, in extremely complex simulations, computation consideration may dictate the most appropriate perspective. For example, it's often more economical, in terms of computational time and hardware requirements, to approximate an event-driven system with a continuous simulation.

6.6.4 Numeric Considerations

The algorithms underlying a model necessarily reflect the scope and nature of the simulation. Depending on the simulation requirements, the algorithms used may vary from simple and approximate to very complex, computationally expensive, and as accurate as possible.

6.6.5 Errors

There is a limit to the degree of accuracy available in every simulation, as dictated by the software and hardware available. For example, all complex digital computations, especially those employing multiple operations on floating-point numbers, are prone to errors. Because of the way in which the two components of a floating-point number are handled, computations involving numbers in this format are not exact. Given enough iterations, the cumulative errors of multiple operations will become significant.

Floating-point relationships such as 2/3 (0.666666…) are represented in a digital computer system to only so many decimal places. Errors of this type, sometimes referred to as roundoff errors, can be minimized at the expense of computational speed by working the highest precision possible. For example, double-precision variables can be used for operators in computations. Rearranging the sequence of computational events so that significant figures aren't lost can also minimize round-off errors. In comparison, computations involving strictly integer numbers are exact as long as the results are within the range of the data type used. The primary benefit of using an integer over a floating-point number is speed.

Round-off errors are due to computer hardware limitations. They can be minimized by the judicious use of appropriate data types and algorithms. The other major type of error, truncation error, is independent of computer hardware, and is attributable instead to the algorithms used in the simulation. These errors occur when the algorithms use approximations to arrive at an answer. For example, instead of computing the sum of an infinite series, a practical algorithm might stop after a sufficient number of elements have been added. Truncation error can best be thought of as the difference between the actual answer and the answer obtained by way of a practical calculation. Unlike round-off errors, which are a function of the computer hardware, operating system, and programming language, truncation errors are a function of the algorithms used to solve a given problem.

6.6.6 Perspectives

During the design of a computer model, one of the major decisions is what perspective to use. The three basic simulation perspectives are continuous, discrete, and hybrid discrete/continuous. These perspectives, which differ in how the system states

change with time and events, define the tools, methods, and algorithms that should be used in the model coding phase of the modelling and simulation process.

6.7. Algorithms used for modelling and simulations

Modelling in bioinformatics is a multidisciplinary activity that borrows algorithms from statistics, mathematics, Artificial Intelligence (AI), and even robotics. For example, robotics algorithms are being used to explore the manipulation of proteins by chaperone molecules. Instead of defining a rotation or unfolding of a protein in 3D space, the space is split into n-dimensions. As a result, the movements of molecules can be described with simple linear functions that are much less computationally intensive than vector algebra. In addition to the esoteric algorithms that are useful in niche areas of bioinformatics, there are several algorithms that have general applicability in modelling and simulation, notably the Monte Carlo methods.

6.7.1 Monte Carlo Method

An approach developed through the collaboration of a computer scientist, physicist, and mathematician, the Monte Carlo method, forms the basis for much modelling and simulation activity in bioinformatics. The Monte Carlo method, named after the famous Monaco casino, involves running multiple repetitions of a model, gathering statistical data, and deriving behaviors of the real-world system based on these models. Each run of a model represents chance behaviours that cannot be modelled exactly, but only characterized statistically. Monte Carlo methods are particularly useful in modelling systems that have a large number of degrees of freedom and quantities of interest. The first uses of the method were in nuclear physics and various military applications. Today, Monte Carlo methods are used in bioinformatics for applications ranging from optimizing the drug discovery process to protein structure prediction.

6.7.2 Metropolis Algorithm

The most important variant of the basic Monte Carlo method used in bioinformatics work is the Metropolis Algorithm. The Metropolis Algorithm is useful in the minimization problems that are common in performing likelihood fits and optimization problems. For example, consider the function graphed in Figure 6.12. Within the boundaries defined by

x1 and x2, A and B are local minima and C is the global minimum. The general problem is to find x so that it minimizes f(x) with as few function calls as possible. The caveat is that the formula for solving f(x) is non-trivial and may be computationally intractable using ordinary means. One of the pitfalls of solving for f(x) through ordinary means is that the solution may be stuck at a local minima, such as A or B in the figure. That is, the algorithm determines that f(x) increases to either side of a local minima, and therefore settles down in the local minima, ignoring the global minimum.

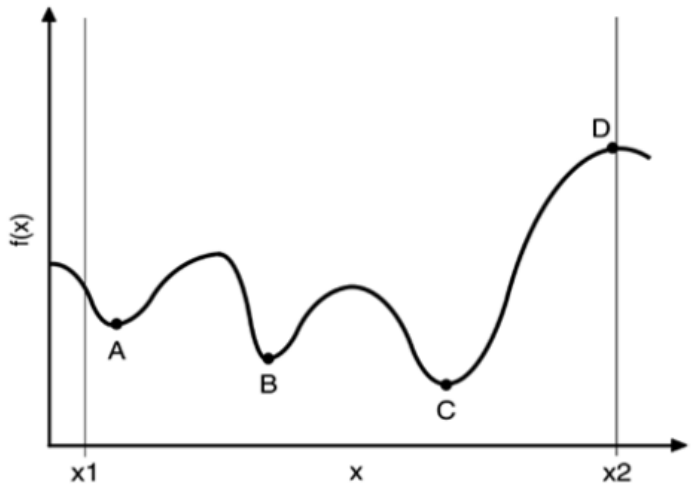


Figure 6.12. A function with local minima (A and B), global minimum (C), and global maximum (D) within the boundaries defined by x1 and x2.

The value of using the Metropolis Algorithm is that it offers a means of maximizing the odds of jumping out of a local minima and into the global minimum—and staying there. In nature, molten materials, such as quartz, when allowed to cool slowly, find the global local minimum state—they crystallize. However, when the material is cooled quickly, the material ends up in local minima—an amorphous state. Algorithmically, the probability that a system at temperature T is in a state of energy E (not at the global minimum) appears as:

$$p(E) \sim e^{-E/kT}$$

6.7.3 Hardware

Simulations, especially those involving tens of thousands of data points and relationships, such as those dealing with protein structure prediction, are extremely hardware-intensive. Many simulations are beyond the capabilities of all but the most powerful general-purpose desktop workstations operating at over 1 GHz with dual CPUs and several GB of RAM—and even these systems may take days of processing time per simulation. The most affordable general-purpose alternatives to mainframe hardware are to create a Linux cluster of affordable, modest-power workstations. A cluster of 20 or more workstations can provide the computational power approaching that of a mainframe at a fraction of the cost.

Depending on the nature of the simulation, specialized hardware may be available to make some modelling and simulation tenable on desktop systems. For example, there are graphics accelerator cards to enhance the rendering of molecules and other 3D structures. Similarly, for neural network–based simulations, there are cards designed to represent the individual nodes in hardware, speeding the lengthy learning process by several orders of magnitude.

6.8 Drug and Protein Structure

Knowledge of protein structure is generally considered a prerequisite to understanding protein function and, by extension, a cornerstone of proteomics research. Because months and sometimes years are involved in verifying protein structure through experimental methods, computational methods of modelling and predicting protein structure are currently viewed as the only viable means of quickly determining the structure of a newly discovered protein. This section explores the role of modelling and simulation methods in determining protein structure.

There are two main computational alternatives to experimental methods of determining or predicting secondary and tertiary protein structures from sequence data. The first approach is based on ab initio methods, which involve reasoning from first principles. The second approach, often termed heuristic methods, is based on some form of pattern matching, using knowledge of existing protein structures. Ab initio methods rely on molecular physics, and ignore any relationship of the molecule with other proteins.

Heuristic methods, in contrast, use information contained in known protein structures. Figure 6.12 shows a flowchart of the methods available for determining or predicting protein structure from protein sequence data.

The difference between the two approaches can be appreciated with parallel approaches in archaeology. When a fossilized skeleton of a small animal is discovered, one approach to reconstructing the physical structure of the animal and its lifestyle is to reason from first principles, using the size, arrangement, thickness of the various bones, the size of the brain case, and other physical indicators, such as the bowing of the long bones (which indicate the amount of musculature present). Wear patterns on the teeth might suggest a diet rich in grains, and the presence of canines may suggest the animal was omnivorous.

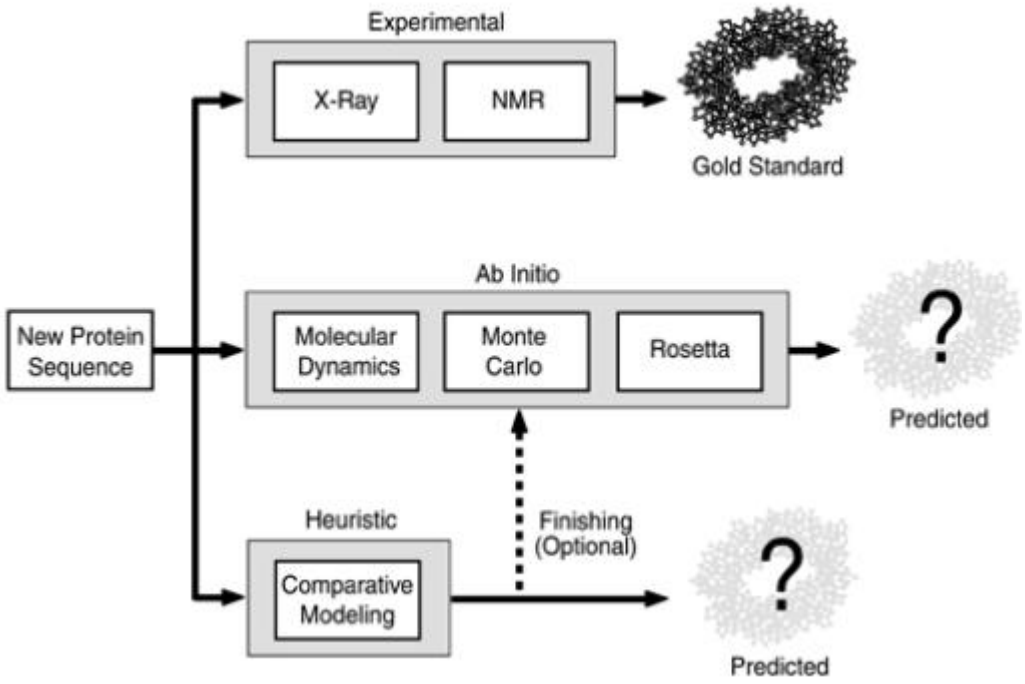


Fig. 6.12. Computational Methods of Protein Structure Prediction versus Experimental Protein Structure Determination Methods. Ab initio and heuristic methods promise to provide less accurate but more timely results, compared to experimental methods that can require a year or more of research per molecule.

The second approach to assessing the fossil of the extinct animal is to compare the skeleton with those of known animals. The leg and arm bones may approximate those of small modern monkey, for example. The teeth may approximate those of a modern primate, with large, flat molars and prominent canines. The relative size of the brain case, when compared to present-day monkeys, might give an indication of the relative intelligence and social lifestyle of the extinct animal, based on current primates.

Comparing fossilized skeletons of animals with those of modern animals is frequently practiced because it's easy, rapid, and to the best of our knowledge, fairly accurate. Reasoning from first principles is usually reserved for those cases where there is nothing resembling the newly discovered fossil in the current fossil record. In many cases, the methods overlap and complement each other. For example, first principles may be used to reconstruct the general body shape and stature of the extinct animal, but give no indication of the skin or hair coloring. However, by extrapolating current behavior and habitat knowledge of current species, a good guess can be made as to the composition and color of the skin, fur, or feathers. Similarly, in bioinformatics, ab initio and heuristic methods of determining protein structure are commonly used in parallel or sequentially because of the accuracy limitations of either approach when used alone. For example, hand editing is commonly applied to ab initio data to improve the accuracy of the results. The primary methods used in the two basic approaches are reviewed here, with an emphasis on the underlying modelling and simulation techniques involved in each method.

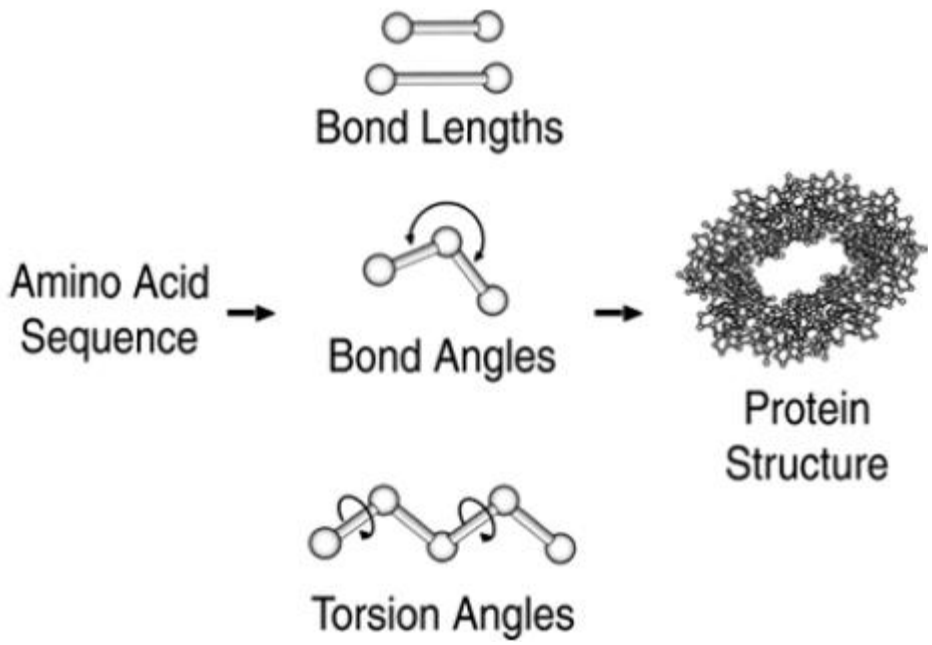
6.8.1 Ab Initio Methods

Pure ab initio methods of determining protein structure are based on sequence data and the physics of molecular dynamics. Newtonian physics, atomic-level forces, and solving equations for the most stable (minimum free energy) conformation or structure form the basis for these methods. Reasoning from first principles assumes that the shape of a protein can be defined as a function of the amino acid sequence, the temperature, pressure, pH, and other local conditions without knowledge of the biology associated with the molecule. For example, the fact that a protein unfolds or becomes denatured at elevated temperatures and reverts to its normal, active, folded state can be modelled irrespective of the structure or function of the protein. However, unlike our knowledge of

physics or other hard sciences, our understanding of the first principles of molecular biology is largely incomplete. As a result, attempts thus far at using first principles as the basis for determining protein structure have been successful primarily as a means of defining limited areas (finishing) of the global protein architecture. For example, with the overall protein structure approximately known, reasoning from first principles can be used to define a particular bend in the structure.

Because of the computational demands associated with ab initio methods, assumptions and simplifications are required for all but the smallest proteins. For example, just as the models of individual neurons discussed earlier are composed of simple equations, instead of considering the dozens of variables affecting each atom and bond in a real neuron, a common simplifying assumption is that protein structure can be computed from bond lengths, bond angles, and torsion (dihedral) angles (see Figure 6.13).

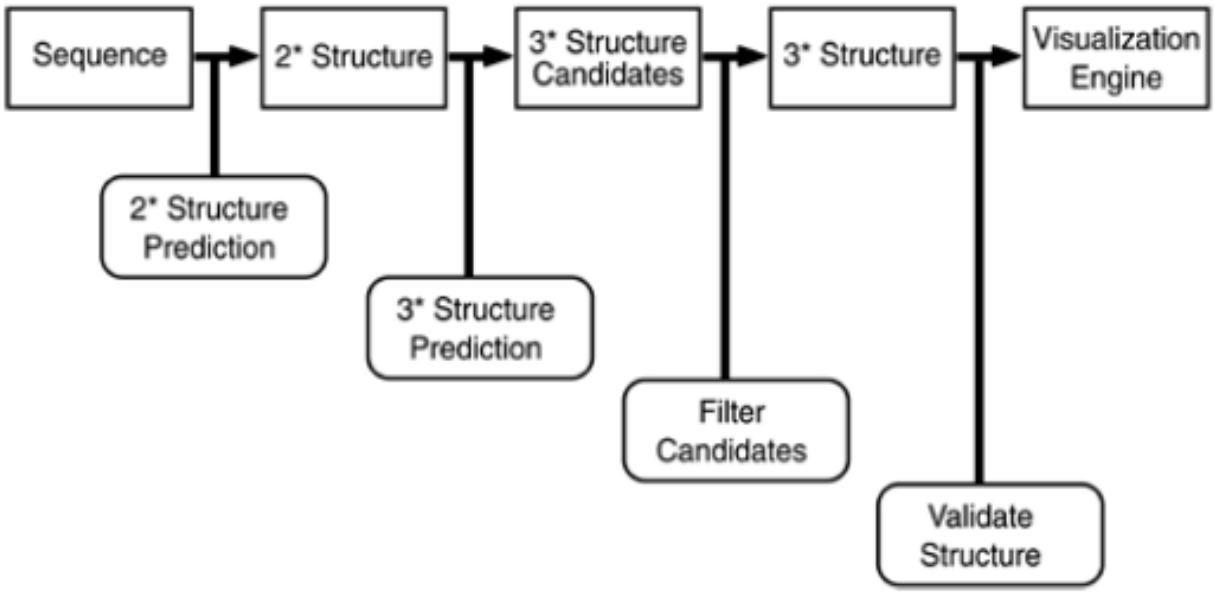
Fig. 6.13. Ab Initio Protein Structure Determination. Based on the protein's amino acid sequence (primary structure), secondary and tertiary structures are computed. Tertiary structures typically takes the form of xyz coordinates for each atom in the protein molecule. Many ab initio methods assume that protein secondary and tertiary structures are a function of bond lengths, bond angles, and torsion angles.



The assumption that a protein's secondary structure can be completely defined as a function of bond lengths, bond angles, and torsion angles, while not always valid, greatly simplifies the computations involved. However, in some instances, even limiting consideration of protein structure to bond lengths, bond angles, and torsion angles is too computationally intensive. For example, modelling protein-protein interactions, with each protein molecule composed of perhaps several thousand atoms, in an aqueous environment with several hundred-thousand water molecules, is currently practically impossible on desktop hardware and may require days of supercomputer time. As a means of simplifying the computations, protein molecules are commonly simplified by representing certain chemical groups as points or ellipses that are either attracted to or repelled by surrounding water molecules.

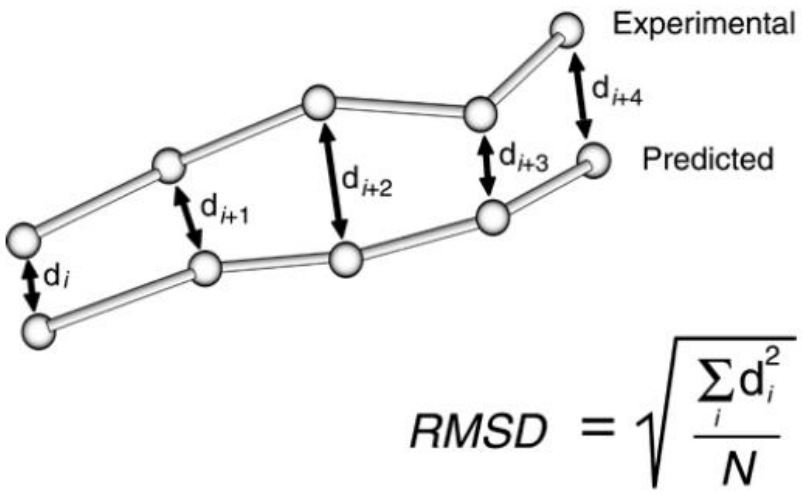
The overall process of determining or predicting tertiary protein structure from a known primary structure or sequence is illustrated in Figure 6.14. Given a sequence of amino acids, the first step is to generate a reasonable secondary structure by using bond lengths, angles, and torsion angles. The next phase of the process, generating the tertiary structure, involves methods such as molecular dynamics and Monte Carlo methods to create a library of tertiary protein structure candidates.

Figure 6.14: General Ab Initio Protein Structure Prediction Process.



Once the top protein structure candidate is identified, it is validated and visualized. Validation typically refers to comparing the predicted protein structure with a structure derived from NMR and X ray crystallography experiments. That is, ab initio methods are still being perfected. Eventually, ab initio methods may provide enough accuracy and handle molecules large enough to supplant experimental methods. However, for now, validation involves assigning a figure of merit to the predicted structure, based on comparison to the gold standard. The most often-used figure of merit in protein structure comparison is the root mean squared deviation (RMSD). The calculation for RMSD, expressed in Angstroms, is shown in Figure 6.15.

Figure 6.15: Root Mean Squared Deviation (RMSD) Calculation. N = number of atoms. D = the distance in Angstroms between corresponding atoms in the experimental and predicted protein structures.



Perfectly identical structures would have an RMSD of 0; matching short to moderate-length protein structures typically have RMSDs in the 1–3 Angstrom range. A problem with RMSD is that it doesn't take the size of the protein into account, and therefore the significance of the RMSD score can't be taken as an absolute measure across all proteins. An RMSD of 5 or 6 Angstroms may be intolerable in a molecule with only 50 residues, but perfectly acceptable in large protein molecules for applications such as searching structure databases for known protein structures. However, even as a relative

measure, RMSD is valuable when working within a single family of proteins because the size of structures will be about the same.

In addition to the RMSD measure, a variety of visualization techniques are available to provide qualitative measures of similarity. Visualizing the protein structure is typically performed through the use of any number of freely available protein rendering engines on the Web, such as RasMol

6.8.2 Heuristic Methods

While, ab initio methods of protein structure prediction can be used to identify novel structures from sequence data alone, they're too computationally intensive to work with all but the smallest proteins. For most proteins of unknown structure, short of X-ray crystallography and nuclear magnetic resonance (NMR) studies, heuristic methods offer the fastest, most accurate means of deriving structure from amino acid sequence data. Heuristic methods use a database of protein structures to make predictions about the structure of newly sequenced proteins. A basic premise of heuristic methods is that most newly sequenced proteins share structural similarities with proteins whose structures and sequences are known, and that these structures can serve as templates for new sequences. It's also assumed that because relatively substantial changes in amino acid sequence may not significantly alter the protein structure, similarity in sequences implies similarity in structure.

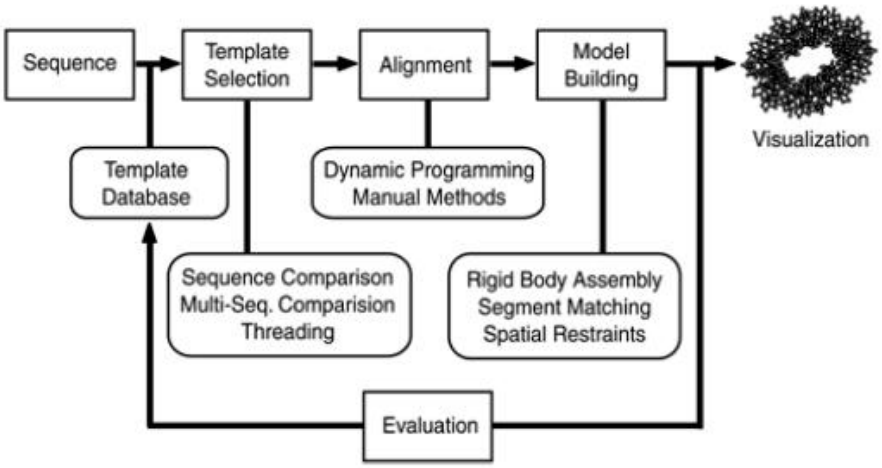
The primary limitation of a heuristic approach to protein structure prediction is that it can't model a novel structure. There must be a suitable template—meaning that the sequences of the template and the new protein can be aligned—available to work with as a starting point. For this reason, heuristic approaches often have difficulty with novel mutations that induce structural changes in the new (target) protein molecule. Within the constraints of these assumptions and limitations, the advantages of heuristic methods over ab initio methods are significant, and include improved accuracy and an ability to work with large protein molecules as opposed to protein fragments. In addition, the potential time savings of heuristic over experimental methods is a driving force for investment in heuristic methods from the pharmaceutical and private investment communities.

The main heuristic method of predicting protein structure from amino acid sequence data is comparative modelling—that is, finding similarities in amino acid sequence, independent of the molecule's lineage. Comparative modelling is sometimes confused with homology modelling. However, homology implies ancestral relationships, and assumes that proteins from the same families share folding motifs even if they don't share the same sequences. In contrast, comparative modelling assumes that proteins with similar amino acid sequences share the same basic 3D structure. The basis for comparative modelling is typically the PDB, which contains descriptions of 3D structures of proteins and other molecules as determined by NMR and X-ray crystallography experiments.

Another source of modelling data is the Molecular Modelling Database (MMDB), which combines PDB data with cross references to sequence, chemical, and structural data. It's important to note that the protein structures defined within PDM, MMDB, and virtually every other protein structure database are based on assumptions that may not be completely valid. For example, the common assumption that similar amino acid sequences result in similar protein structures is known to have exceptions.

Comparative modelling is an iterative, multi-phase process. As outlined in Figure 6.16, given protein sequence data, the main phases of the process are template selection, alignment, model building, and evaluation. 3D visualization is often performed as part of the evaluation phase. The key activities in each phase of the comparative modelling process are outlined here.

Fig. 6.16. Comparative Modelling Process. Not illustrated is the optional use of ab initio methods at the end of the model building phase to reduce errors in the computed structure.



6.8.3 Template Selection

The first phase of the comparative modelling process, template selection, involves searching a template database for the closest match or matches to the new (target) molecule, based on the target's amino acid sequence. The goal of template selection is to discover a link between the target protein and a known protein structure. This usually involves the use of a protein structure template databases, such as the PDB. Selecting an appropriate group of database entries from the database to serve as structure templates is typically based on some form of sequence comparison or threading.

Pairwise sequence comparison involves searching sections of the template candidate for amino acid sequences that are similar to sequences in the target protein. A key decision in sequence comparison is how similar is similar enough. Multiple sequence comparison relies on an iterative algorithm that expands the template search to include all reasonable candidate templates from the template databases. As a result, multiple sequence comparison is more sensitive and more likely to find suitable templates in the template database.

Threading involves aligning the sequence of the target protein with the 3D structure of a template to determine whether the amino acid sequence is spatially and chemically similar to the template. Threading can be thought of as searching through a bin of factory-second gloves, looking for a glove that fits, where the hand is the target protein and the gloves represent templates. Some gloves may be able to accommodate only four

fingers (no thumb), whereas others might have a channel for a sixth finger. These gloves represent templates that don't match the target protein. Gloves that, on visual inspection (the gloves aren't actually tried on—yet) can accommodate five fingers on the proper hand—assuming the hand is "normal"—are retained as potential templates. Similarly, templates that best fit the target protein are identified for use later in the comparative modelling process.

There are various forms of threading. For example, in contact potential threading, which is based on the analysis of the number and closeness of contacts between amino acids in the protein core, the idea is to position amino acids and compute empirical energies from the observed associations of amino acids. The most energetically stable conformation is the most likely protein structure. A more complex form of threading involves modelling energies from first principles. This method is based on dynamic programming techniques and is a recursive method of solving a problem that involves saving intermediate results in a matrix or table so that they can be used for future calculations.

Developing modelling and simulation systems de novo requires knowledge of advanced computing techniques, from Markov Modelling to network computing and numerical calculus. Fortunately, a wide variety of modelling simulation tools is available on the Web and from commercial vendors. As listed in Table 6.2, there are tools specifically designed to aid modelling and simulation in bioinformatics as well as tools for general-purpose modelling. For example, a tool such as Prospect (Protein Structure Prediction and Evaluation Computer Toolkit), a threading-based protein structure prediction program, can be used as part of a comparative modelling process. A commercial system, such as Extend, can be used to determine the most cost-effective means of staffing the research lab, based on a model of individual researcher output and the overall protein structure modelling process.

Table 6.2: Modelling and Simulation Tools.

Tool	Examples
Databases	CATH, GenBank, GeneCensus, ModBase, PDB, SWISS-PROT+TrEMBL

Template Search	123D, BLAST, DALI, FastA, Matchmaker, PHD, PROFIT, Threader
Sequence Alignment	BCM Server, BLAST, Block Maker, CLUSTAL, FASTA3, Multalin
Modelling	Congen, CPH Models, Dragon, ICM, InsightII, Modeller, Look etc.
Verification	Anolea, Aqua, Biotech, Errat, Procheck, ProCeryon, Prosall etc.
Visualization	CHIMERA, SWISS-PDBViewer, RasMol, Pymol

6.9. Conserved Domain Database (CDD)

The Conserved Domain Database (CDD) is a database of well-annotated multiple sequence alignment models and derived database search models, for ancient domains and full-length proteins.

Domains can be thought of as distinct functional and/or structural units of a protein. These two classifications coincide rather often, as a matter of fact, and what is found as an independently folding unit of a polypeptide chain also carries specific function. Domains are often identified as recurring (sequence or structure) units, which may exist in various contexts. In molecular evolution such domains may have been utilized as building blocks, and may have been recombined in different arrangements to modulate protein function. CDD defines conserved domains as recurring units in molecular evolution, the extents of which can be determined by sequence and structure analysis.

The goal of the NCBI conserved domain curation project is to provide database users with insights into how patterns of residue conservation and divergence in a family relate to functional properties, and to provide useful links to more detailed information that may help to understand those sequence/structure/function relationships. To do this, CDD Curators include the following types of information in order to supplement and enrich the traditional multiple sequence alignments that form the foundation of domain

models: 3-dimensional structures and conserved core motifs, conserved features/sites, phylogenetic organization, links to electronic literature resources.

CDD content includes NCBI manually curated domain models and domain models imported from a number of external source databases (Pfam, SMART, COG, PRK, TIGRFAMs). What is unique about NCBI-curated domains is that they use 3D-structure information to explicitly define domain boundaries, align blocks, amend alignment details, and provide insights into sequence/structure/function relationships. Manually curated models are organized hierarchically if they describe domain families that are clearly related by common descent. To provide a non-redundant view of the data, CDD clusters similar domain models from various sources into superfamilies.

Searching the database

The collection is also part of NCBI's Entrez query and retrieval system, crosslinked to numerous other resources. CDD provides annotation of domain footprints and conserved functional sites on protein sequences. Precalculated domain annotation can be retrieved for protein sequences tracked in NCBI's Entrez system, and CDD's collection of models can be queried with novel protein sequences via * "the CD-Search service"*. United States National Center for Biotechnology Information.*, or at* "the Batch CD-Search"*. United States National Center for Biotechnology Information.*, that allows the computation and download of annotation for large sets of protein queries.

6.10 Nucleotide and Protein Sequence Databases

There are three major organizations that collaborate to collect publicly available nucleotide and protein sequences. These organizations share data on a daily basis but they are distinguished by different international catchment areas for submissions, different formats and sometimes differences in the nature of their submitter annotations. Genbank is maintained by the NCBI in the United States (http://www.ncbi.nlm.nih.gov/Genbank/index.html). EMBL is maintained by the European Bioinformatics Institute in the United Kingdom (http://www.ebi.ac.uk/). The third member is the DNA Database of Japan (DDBJ) in Mishima, Japan (http://www.ddbj.nig.ac.jp/). All three organizations offer a wide range of tools for

sequence searching and analysis but two integrated database query tools have become pre-eminent. These are Entrez from the NCBI and SRS from the EBI.

6.10.1 Entrez

Entrez (http://www.ncbi.nlm.nih.gov/Entrez/) is the backbone of the NCBI database infrastructure. It is an integrated database retrieval system that allows the user to search and browse all the NCBI databases through a single gateway. Entrez provides access to DNA and protein sequences derived from many sources, including genome maps, population sets and, as already described, the biomedical literature via PubMed and Online Mendelian Inheritance in Man (OMIM). New search features are being added to Entrez on a regular basis. Most recently facilities have been added to allow searches for DNA by 'ProbeSet' data from gene-expression experiments and for proteins by molecular weight range, by protein domain or by structure in the Molecular Modelling Database of 3D structures (MMDB).

6.10.2 Sequence Retrieval Server (SRS)

The sequence retrieval server (SRS) serves a similar role to Entrez, for the major European sequence databases. SRS is a flexible sequence query tool which allows the user to search a defined set of sequence databases and knowledge-bases by accession number, keyword or sequence similarity. SRS encompasses a very wide range of data, including all the major EMBL sequence divisions (Table 2.5). SRS goes one step further than Entrez by enabling the user to create analysis pipelines by selecting retrieved data for processing by a range of analysis tools, including ClustalW, BLAST and InterProScan.

6.15. Summary

Molecular dynamics (MD) and related methods are close to becoming routine computational tools for drug discovery. Their main advantage is in explicitly treating structural flexibility and entropic effects. This allows a more accurate estimate of the thermodynamics and kinetics associated with drug–target recognition and binding, as better algorithms and hardware architectures increases their use. Here, this chapter has the theoretical background of MD and enhanced sampling methods, focusing on free-

energy perturbation, metadynamics, steered MD, and other methods most consistently used to study drug–target binding. Here, we tried to give a brief about algorithms used in MD simulations that nowadays allow the observation of unsupervised ligand–target binding, assessing how these approaches help optimizing target affinity and drug residence time toward improved drug efficacy. Later on at the end we discussed some of the nucleotide databases which include the basic from where the story starts and how the things need to be learn.