

SHRIMATI INDIRA GANDHI COLLEGE
(Nationally Accredited at “A” Grade (3rd cycle) By NAAC)
Tiruchirappalli -2

INSTRUCTION MATERIAL
BIOINFORMATICS

By
Ms.A.Shanmuga priya, Associate Professor,
Department of Biochemistry
2017 -2018



DEPARTMENT OF BIOCHEMISTRY
SHRIMATI INDIRA GANDHI COLLEGE

INSTRUCTION MATERIAL FOR ADVANCE LEARNERS
BIOINFORMATICS

| CONTENTS | TOPICS | PG NO. |
|-----------------|---|---------------|
| UNIT I | Introduction | 4 |
| UNIT II | Workstation | 10 |
| UNIT III | Databases | 18 |
| UNIT IV | Database searches and sequence alignment | 24 |
| UNIT V | Applications | 44 |

SYLLABUS

Unit 1 Introduction

Introduction to bioinformatics, scope of bioinformatics, role of computers in biology. The internet, the World Wide Web, useful search engines- Boolean searching, search engine algorithms. Finding scientific articles- Pub med. running computer software, computer operating systems. Software downloading and installation.

Unit 2 Workstation

The bioinformatics workstation, UNIX system, files and directories in UNIX, working on a UNIX system. Scripting languages- Perl and Python, markup languages- HTML, XML.

Unit 3 Databases

Database concepts- Database, database system, database management systems- Hierarchical, Rational and Network, Database security. Biological databases, Types- sequence and structure databases. Genome and organism specific databases. Miscellaneous databases. Data submission data retrieval with Entrez, DBGET / Link DB and SRS.

Unit 4 Database searches and sequence alignment

Searching sequence database sequence similarity searches, amino acid substitution matrices, Database searches: FASTA and BLAST, sequence filters, Iterative database searches and PSIBLAST. Multiple sequence alignment- gene and protein families. Phylogenetics- building phylogenetic trees, Evolution of macromolecular sequences, Sequence annotation.

Unit 5 Applications

Prediction and visualization of protein structure. Drug discovery and development, combinatorial chemistry and docking. Pharmacogenomics. Pharmacogenetics. Toxicogenomics. Functional genomics, metabolomics. E-cell. Metabolic pathways- Kegg and Wit, primer design, Micro fluidics.

UNIT I: INTRODUCTION

1. What is Bioinformatics?

Bioinformatics has been defined many different ways, since practitioners do not always agree upon the scope of its use within the biological and computer sciences, but it is always considered a combination of both sciences, along with other contributing disciplines.

Bioinformatics as a biological science

It is debatable whether bioinformatics and the discipline computational biology, literally "biology that involves computation," are the same or distinct. To some, both bioinformatics and computational biology are defined as *any* use of computers for processing *any* biologically-derived information, whether DNA sequences or breast X-rays. Therefore, there are other fields, e.g. medical imaging / image analysis that might be considered part of bioinformatics. This would be the broadest definition of the term. But, in practice, the definition used by most people is even narrower; bioinformatics to them is a synonym for computational molecular biology: any use of computers to characterize the molecular components of living things.

Bioinformatics as a computer science

To others, bioinformatics is a grammatical contraction of "biological informatics" and is therefore related to the computer science disciplines of *information science* and/or *information technology*. This definition would thus emphasize the *information* contained within the biological data, also implying that large amounts of data would be managed and/or analyzed.

2. What is the scope of Bioinformatics?

Bioinformatics and computational biology involve the use of techniques including applied mathematics, informatics, statistics, computer science, artificial intelligence, chemistry and biochemistry to solve biological problems usually on the molecular level. Research in computational biology often overlaps with systems biology. Major research efforts in the field include sequence alignment, gene finding, genome assembly, protein structure alignment, protein structure prediction, prediction of gene expression and protein-protein interactions, and the modeling of evolution.

3. Role of Computers in Biology

- Collecting and processing signals detected by laboratory equipment: DNA sequences, CCD devices, spectrophotometers, and just about any other device that can be connected to a computer via an analog to digital converter.
- Tracking samples and managing experiments in industrial-style laboratories (e.g., in gene sequencing centers). Smaller labs don't have the resources to invest in automated laboratory management, but using software to manually maintain lab-notebook-style electronic records is rapidly becoming more common.
- Storing data in public databases, and more importantly, public access to the database via sophisticated Web searches and deposition mechanisms. NCBI, home of Genbank, PubMed, and other public databases, is the premier example of the kind of information services that can be built onto a public biological database.
- Extracting patterns and rules from large data collections and using these observed patterns to characterize and predict features in new data. This is the core of bioinformatics: developing tools which can recognize pattern matches and feature signatures within an otherwise inscrutable data set.
- Annotation: using automatic computational methods to assign functional meaning to uncharacterized data and to create informative links between different data collections. For example, many annotation systems use automated sequence comparison searches to identify potential genes in new genome data.

- Simulation: using known information about a system, along with a mathematical or physicochemical model, to simulate properties of the system. This category is incredibly diverse, from simulating the motions of interacting protein molecules to modeling the flow of chemicals through biochemical pathways.

4. Internet

Internet is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/IP) to link several billion devices worldwide. It is a network of networks that consists of millions of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. The Internet carries an extensive range of information resources and services, such as the inter-linked hypertext documents and applications of the World Wide Web (WWW), the infrastructure to support email, and peer-to-peer networks for file sharing and telephony.

The origins of the Internet date back to research commissioned by the United States government in the 1960s to build robust, fault-tolerant communication via computer networks.^[1] This work, combined with efforts in the United Kingdom and France, led to the primary precursor network, the ARPANET, in the United States. The interconnection of regional academic networks in the 1980s marks the beginning of the transition to the modern Internet. From the early 1990s, the network experienced sustained exponential growth as generations of institutional, personal, and mobile computers were connected to it.

The Internet, referring to the specific global system of interconnected IP networks, is a proper noun and written with an initial capital letter. In the media and common use it is often not capitalized, viz. the internet. Some guides specify that the word should be capitalized when used as a noun, but not capitalized when used as a verb or an adjective. The Internet is also often referred to as the Net.

Historically the word *internet* was used, uncapitalized, as early as 1883 as a verb and adjective to refer to interconnected motions. The designers of early computer networks used *internet* both as a noun and as a verb in shorthand form of internetwork or internetworking, meaning interconnecting computer networks with gateways. The terms Internet and World Wide Web are

often used interchangeably in everyday speech; it is common to speak of "going on the Internet" when invoking a web browser to view web pages. However, the World Wide Web or the Web is only one of a large number of Internet services. The Web is a collection of interconnected documents (web pages) and other web resources, linked by hyperlinks and URLs. As another point of comparison, Hypertext Transfer Protocol, or HTTP, is the language used on the Web for information transfer, yet it is just one of many languages or protocols that can be used for communication on the Internet. In addition to the Web, a multitude of other services are implemented on the Internet.

5. World Wide Web

The **World Wide Web (WWW, W3)** is an information system of interlinked hypertext documents that are accessed via the Internet. It has also commonly become known simply as *the Web*. Individual document pages on the World Wide Web are called web pages and are accessed with a software application running on the user's computer, commonly called a web browser. Web pages may contain text, images, videos, and other multimedia components, as well as web navigation features consisting of hyperlinks.

Tim Berners-Lee, a British computer scientist and former CERN employee, is considered the inventor of the Web. On 12 March 1989, Berners-Lee wrote a proposal for what would eventually become the World Wide Web. The 1989 proposal was meant for a more effective CERN communication system but Berners-Lee eventually realised the concept could be implemented throughout the world. Berners-Lee and Belgian computer scientist Robert Cailliau proposed in 1990 to use hypertext "to link and access information of various kinds as a web of nodes in which the user can browse at will", and Berners-Lee finished the first website in December of that year. The first test was completed around 20 December 1990 and Berners-Lee reported about the project on the newsgroup on 7 August 1991.

he World Wide Web Consortium (W3C) was founded by Tim Berners-Lee after he left the European Organization for Nuclear Research (CERN) in October 1994. It was founded at the Massachusetts Institute of Technology Laboratory for Computer Science (MIT/LCS) with support from the Defense Advanced Research Projects Agency (DARPA), which had pioneered

the Internet; a year later, a second site was founded at INRIA (a French national computer research lab) with support from the European Commission DG InfSo; and in 1996, a third continental site was created in Japan at Keio University. By the end of 1994, the total number of websites was still relatively small, but many notable websites were already active that foreshadowed or inspired today's most popular services.

Connected by the existing Internet, other websites were created around the world, adding international standards for domain names and HTML. Since then, Berners-Lee has played an active role in guiding the development of web standards (such as the markup languages to compose web pages in), and has advocated his vision of a Semantic Web. The World Wide Web enabled the spread of information over the Internet through an easy-to-use and flexible format. It thus played an important role in popularizing use of the Internet.^[26] Although the two terms are sometimes conflated in popular use, *World Wide Web* is not synonymous with *Internet*.^[27] The Web is an information space containing hyperlinked documents and other resources, identified by their URIs.^[28] It is implemented as both client and server software using Internet protocols such as TCP/IP and HTTP.

Function

The World Wide Web functions as a layer on top of the Internet, helping to make it more functional. The advent of the Mosaic web browser helped to make the web much more usable.

The terms Internet and World Wide Web are often used without much distinction. However, the two things are not the same. The Internet is a global system of interconnected computer networks. In contrast, the World Wide Web is one of the services transferred over these networks. It is a collection of text documents and other resources, linked by hyperlinks and URLs, usually accessed by web browsers, from web servers. Viewing a web page on the World Wide Web normally begins either by typing the URL of the page into a web browser, or by following a hyperlink to that page or resource. The web browser then initiates a series of background communication messages to fetch and display the requested page. In the 1990s, using a browser to view web pages—and to move from one web page to another through hyperlinks—came to be known as 'browsing,' 'web surfing,' (after channel surfing), or 'navigating

the Web'. Early studies of this new behavior investigated user patterns in using web browsers. One study, for example, found five user patterns: exploratory surfing, window surfing, evolved surfing, bounded navigation and targeted navigation.

The following example demonstrates the functioning of web browser when accessing a page at the URL http://example.org/wiki/World_Wide_Web. The browser resolves the server name of the URL (example.org) into an Internet Protocol address using the globally distributed Domain Name System (DNS). This lookup returns an IP address such as 203.0.113.4. The browser then requests the resource by sending an HTTP request across the Internet to the computer at that address. It requests service from a specific TCP port number that is well known for the HTTP service, so that the receiving host can distinguish an HTTP request from other network protocols it may be servicing.

6. Boolean Searching

A type of search allowing users to combine keywords with operators such as AND, NOT and OR to further produce more relevant results. For example, a Boolean search could be "hotel" AND "New York". This would limit the search results to only those documents containing the two keywords.

7. Finding scientific articles in pub med

1. Search the PubMed with a search term, author name, or PubMed ID. Author name can be entered as follows: smith aj[au].
2. Click on the title of an entry of interest.
3. Look for icons in the upper-right-hand corner of the record:
 - Click on the PubMed Central link or a Publisher's link to access the full text of the article. Articles in PubMed Central are freely available. Articles on Publisher's websites are either freely available or can be accessed with a fee. Contact the specific publisher for questions about their site.

- For PubMed records with no icons in the upper-right-hand corner, Loan some Doc can be accessed to order the article following these directions: PubMed Help.

UNIT II:

WORK STATION

1. Workstation

A **workstation** is a special computer designed for technical or scientific applications. Intended primarily to be used by one person at a time, they are commonly connected to a local area network and run multi-user operating systems. The term *workstation* has also been used loosely to refer to everything from a mainframe computer terminal to a PC connected to a network, but the most common form refers to the group of hardware offered by several current and defunct companies such as Sun Microsystems, Silicon Graphics, Apollo Computer, DEC, HP and IBM which opened the door for the 3D graphics animation revolution of the late 1990s.

Workstations offered higher performance than mainstream personal computers, especially with respect to CPU and graphics, memory capacity, and multitasking capability. Workstations were optimized for the visualization and manipulation of different types of complex data such as 3D mechanical design, engineering simulation (e.g. computational fluid dynamics), animation and rendering of images, and mathematical plots. Typically, the form factor is that of a desktop computer, consist of a high resolution display, a keyboard and a mouse at a minimum, but also offer multiple displays, graphics tablets, 3D mice (devices for manipulating 3D objects and navigating scenes), etc. Workstations were the first segment of the computer market to present advanced accessories and collaboration tools.

The increasing capabilities of mainstream PCs in the late 1990s have blurred the lines somewhat with technical/scientific workstations. The workstation market previously employed proprietary hardware which made them distinct from PCs; for instance IBM used RISC-based CPUs for its workstations and Intel x86 CPUs for its business/consumer PCs during the 1990s and 2000s. However by the late 2000s this difference disappeared, as workstations now use highly commoditized hardware dominated by large PC vendors, such as Dell and HP& Fujitsu, selling

Microsoft Windows or GNU/Linux systems running on x86-64 architecture such as Intel Xeon or AMD

Current workstation market

Decline of RISC-based workstations

As of January 2009, all RISC-based workstation product lines have been discontinued:

- SGI ended general availability of its MIPS-based SGI Fuel and SGI Tezro workstations in December 2006.
- Hewlett-Packard withdrew its last HP 9000 PA-RISC based desktop products from the market in January 2008.
- Sun Microsystems announced end-of-life for its last Sun Ultra SPARC workstations in October 2008. IBM retired its Intelli station POWER product line on January 2, 2009. Change to x86-64 workstations

The current workstation market uses x86-64 microprocessors. Operating systems available for these platforms include MS Windows, FreeBSD, the different GNU/Linux distributions, Apple Mac, and Oracle Solaris. Some vendors also market commodity mono socket systems as workstations.

Three types of products are marketed under the workstation umbrella:

1. Workstation blade systems (IBM HC10 or Hewlett-Packard xw460c. Sun Visualization System is akin to these solutions)
2. Ultra high-end workstation (SGI virtu VS3xx)
3. Deskside systems containing server-class CPUs and chipsets on large server-class motherboards with high-end RAM (HP Z-series workstations & Fujitsu CELSIUS workstations)

2. Unix system

Unix (all-caps **UNIX** for the trademark) is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, developed in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.

Initially intended for use inside the Bell System, AT&T licensed Unix to outside parties from the late 1970s, leading to a variety of both academic and commercial variants of Unix from vendors such as the University of California, Berkeley (BSD), Microsoft (Xenix), IBM (AIX) and Sun Microsystems (Solaris). AT&T finally sold its rights in Unix to Novell in the early 1990s, which then sold its Unix business to the Santa Cruz Operation (SCO) in 1995, but the UNIX trademark passed to the industry standards consortium The Open Group, which allows the use of the mark for certified operating systems compliant with the Single UNIX Specification. Among these is Apple's OS X,^[5] which is the Unix version with the largest installed base as of 2014.

From the power user's or programmer's perspective, Unix systems are characterized by a modular design that is sometimes called the "Unix philosophy," meaning the OS provides a set of simple tools that each perform a limited, well-defined function, with a unified filesystem as the main means of communication^[3] and a shell scripting and command language to combine the tools to perform complex workflows. Aside from the modular design, Unix also distinguishes itself from its predecessors as the first portable operating system: virtually the entire OS is written in the C programming language which allowed it to outgrow the 16-bit PDP-11 minicomputer for which it was originally developed.

3. Unix files and directories

In UNIX, all data are stored in repositories called **files**. For example, the RESOLVE/C++ programs that you write in this course will be stored as UNIX files. You could also use files to store any reports that you write or to save the e-mail that you receive.

Like in the "real" world, it is not a good idea to have lots of files "lying around" in a disorganized manner. Unix allows you to organize your files into **directories**. A "directory" is a location where files are kept in a list. For instance, you could create a directory to store all your files for

the first lab and call it *Lab1*. You could create another directory called *Lab2* to store your files for the second lab. If you are already familiar with either Apple Macintosh computers or Windows File Manager, just think of Unix directories as being the same as folders.

When you login to a Unix system, the system puts you in your **home directory**. Your home directory is the directory that is assigned to you to store all your files. Unix has commands that you can use to create and delete files and directories within your home directory. Unix also gives you commands to change from your home directory to other directories. The directory that you're in at any given point of time is called your **current working directory**.

The ls command

The Unix command *ls*, lists the contents of the current working directory. The contents of a directory are the files and sub-directories inside that directory.



- In the xterm, enter the command: *ls*

Since your working directory at this point is your home directory, the output of the *ls* command tells you the contents of your home directory.

- In the xterm, enter the command: *Ls*

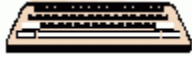
You'll get an error message of the form:

Ls: Command not found.

Important Note: Unix is case-sensitive. That means it treats *ls* and *Ls* as two distinct commands. In fact, there is no command *Ls* in Unix. Similarly, if you have a file named *foo*, you cannot access it using the name *FOO*.

The mkdir command

You can use the *mkdir* command to create new directories. In the home directory, let us create a directory called *Testdir1*.

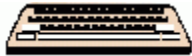


Enter the command: `mkdir Testdir1`

If you repeat the *ls* command, you'll see that *Testdir1* appears among the contents listed. Let us use the *cd* command to change our working directory to *Testdir1*.

The cd command

You can move around from one directory to another with the *cd* command.



Enter the command: `cd Testdir1`

At this point, you're not in your home directory anymore but in the sub-directory *Testdir1*. What command do you think you should choose if you wanted to create a sub-directory *Testdir2* within this directory?

Go ahead and enter the command to create a new sub-directory *Testdir2* in your xterm. Use the *ls* command to verify that the directory was created.

Now *Testdir2* is a sub-directory of *Testdir1*, which is a sub-directory of your home directory.

4. Perl Scripting Language

Perl is a family of high-level, general-purpose, interpreted, dynamic programming languages. The languages in this family include Perl 5 and Perl 6. Though Perl is not officially an

acronym, here are various backronyms in use, the most well-known being "Practical Extraction and Reporting Language". Perl was originally developed by Larry Wall in 1987 as a general-purpose Unix scripting language to make report processing easier. Since then, it has undergone many changes and revisions. Perl 6, which began as a redesign of Perl 5 in 2000, eventually evolved into a separate language. Both languages continue to be developed independently by different development teams and liberally borrow ideas from one another.

The Perl languages borrow features from other programming languages including C, shell scripting (sh), AWK, and sed. They provide powerful text processing facilities without the arbitrary data-length limits of many contemporary Unix commandline tools, facilitating easy manipulation of text files. Perl 5 gained widespread popularity in the late 1990s as a CGI scripting language, in part due to its unsurpassed regular expression and stringparsing abilities. In addition to CGI, Perl 5 is used for graphics programming, system administration, network programming, finance, bioinformatics, and other applications. It has been humorously nicknamed "the Swiss Army chainsaw of scripting languages" because of its flexibility and power, and possibly also because of its "ugliness". In 1998, it was also referred to as the "duct tape that holds the Internet together", in reference to both its ubiquitous use as a glue language and its perceived inelegance.

5. Python

Python is a widely used general-purpose, high-level programming language.^{[17][18][19]} Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java.^{[20][21]} The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems. Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into stand-alone executable programs for some of the

most popular operating systems, allowing for the distribution of Python-based software for use on those environments without requiring the installation of a Python interpreter.

CPython, the reference implementation of Python, is free and open-source software and has a community-based development model, as do nearly all of its alternative implementations. CPython is managed by the non-profit Python Software Foundation.

6. HTML

HyperText Markup Language commonly referred to as **HTML** is the standard markup language used to create web pages. It is written in the form of HTML elements consisting of *tags* enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example . The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*).

Web browsers can read HTML files and compose them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages.

Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML.⁷ XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. It is

defined by the W3C's XML 1.0 Specification and by several other related specifications all of which are free open standards.

The design goals of XML emphasize simplicity, generality and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures^[6] such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while many application programming interfaces (APIs) have been developed to aid the processing of XML data.

Applications of XML

As of 2009, hundreds of document formats using XML syntax have been developed,^[7] including RSS, Atom, SOAP, and XHTML. XML-based formats have become the default for many office-productivity tools, including Microsoft Office (Office Open XML), OpenOffice.org and LibreOffice (OpenDocument), and Apple's iWork. XML has also been employed as the base language for communication protocols, such as XMPP. Applications for the Microsoft.NET Framework use XML files for configuration. Apple has an implementation of a registry based on XML.

XML has come into common use for the interchange of data over the Internet. IETF RFC 7303 gives rules for the construction of Internet Media Types for use when sending XML. It also defines the media type's *application/xml* and *text/xml*, which say only that the data are in XML, and nothing about its semantics. The use of *text/xml* has been criticized as a potential source of encoding problems and it has been suggested that it should be deprecated.^[9]

RFC 7303 also recommends that XML-based languages be given media types ending in *+xml*; for example *image/svg+xml* for SVG.

Further guidelines for the use of XML in a networked context may be found in RFC 3470, also known as IETF BCP 70 a document which covers many aspects of designing and deploying an XML-based language.

UNIT III:

DATABASES

1. Database

A **database** is an organized collection of data. The data is typically organized to model aspects of reality in a way that supports processes requiring information. For example, modelling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

Database management systems are computer software applications that interact with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs include MySQL, PostgreSQL, Microsoft SQL Server, Oracle, SAP and IBM DB2. A database is not generally portable across different DBMSs, but different DBMS can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Database management systems are often classified according to the database model that they support; the most popular database systems since the 1980s have all supported the relational model as represented by the SQL language. Sometimes a DBMS is loosely referred to as a 'database'

2. Biological database

Biological databases are libraries of life sciences information, collected from scientific experiments, published literature, high-throughput experiment technology, and computational analyses. They contain information from research areas including genomics, proteomics, metabolomics, microarray gene expression, and phylogenetics. Information contained in biological databases includes gene function, structure, localization (both cellular and

chromosomal), clinical effects of mutations as well as similarities of biological sequences and structures.

Biological databases can be broadly classified into sequence and structure databases. Nucleic acid and protein sequences are stored in sequence databases and structure database only store proteins. These databases are important tools in assisting scientists to analyze 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, predicting certain genetic diseases and in discovering basic relationships among species in the history of life.

Biological knowledge is distributed among many different general and specialized databases. This sometimes makes it difficult to ensure the consistency of information. Integrative bioinformatics is one field attempting to tackle this problem by providing unified access. One solution is how biological databases cross-reference to other databases with accession numbers to link their related knowledge together.

Relational database concepts of computer science and Information retrieval concepts of digital libraries are important for understanding biological databases. Biological database design, development, and long-term management is a core area of the discipline of bioinformatics. Data contents include gene sequences, textual descriptions, attributes and ontology classifications, citations, and tabular data. These are often described as semi-structured data, and can be represented as tables, key delimited records, and XML structures.

3. Nucleic Acids Research Database Issue

An important resource for finding biological databases is a special yearly issue of the journal *Nucleic Acids Research* (NAR). The Database Issue of NAR is freely available, and categorizes many of the publicly available on line databases related to biology and bioinformatics. A companion database to the issue called the Online Molecular Biology Database Collection lists 1,380 online databases. Other collections of databases exist such as MetaBase and the Bioinformatics Links Collection.

Access

Most biological databases are available through web sites that organise data such that users can browse through the data online. In addition the underlying data is usually available for download in a variety of formats. Biological data comes in many formats. These formats include text, sequence data, protein structure and links. Each of these can be found from certain sources, for example:

- Text formats are provided by PubMed and OMIM.
- Sequence data is provided by GenBank, in terms of DNA, and UniProt, in terms of protein.
- Protein structures are provided by PDB, SCOP, and CATH.

4. Species-specific databases

Species-specific databases are available for some species, mainly those that are often used in research. For example, Colibase ([1]) is an *E. coli* database. Other popular species specific databases include Mouse Genome Informatics for the laboratory mouse, *Mus musculus*, the Rat Genome Database for *Rattus*, ZFIN for *Danio Rerio* (zebrafish), FlyBase for *Drosophila*, WormBase for the nematodes *Caenorhabditis elegans* and *Caenorhabditis briggsae*, and Xenbase for *Xenopus tropicalis*

5. Sequence database

In the field of bioinformatics, a **sequence database** is a type of biological database that is composed of a large collection of computerized ("digital") nucleic acid sequences, protein sequences, or other polymer sequences stored on a computer. The UniProt database is an example of a protein sequence database. As of 2013 it contained over 40 million sequences and is growing at an exponential rate. Historically, sequences were published in paper form, but as the number of sequences grew this storage method became unsustainable.

Current issues

Records in sequence databases are deposited from a wide range of sources, from individual researchers to large genome sequencing centers. As a result, the sequences themselves, and especially the biological annotations attached to these sequences, may vary in quality. There is much redundancy, as multiple labs may submit numerous sequences that are identical, or nearly identical, to others in the databases.¹

6. Protein structure 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. Data included in protein structure databases often includes three-dimensional coordinates as well as experimental information, such as unit cell dimensions and angles for x-ray crystallography determined structures. Though most instances, in this case either proteins or a specific structure determinations of a protein, also contain sequence information and some databases even provide means for performing sequence based queries, the primary attribute of a structure database is structural information, whereas sequence databases focus on sequence information, and contain no structural information for the majority of entries.

The Protein Data Bank

The Protein Data Bank (PDB) was established in 1971 as the central archive of all experimentally determined protein structure data. Today the PDB is maintained by international consortia collectively known as the Worldwide Protein Data Bank (wwPDB). The mission of the wwPDB is to maintain a single archive of macromolecular structural data that is freely and publicly available to the global community.

List of other protein structure databases

Database of Macromolecular Movements

Describes the motions that occur in proteins and other macromolecules, particularly using movies

Dynameomics

a data warehouse of molecular dynamics simulations and analyses of proteins representing all known protein fold families

JenaLib

the Jena Library of Biological Macromolecules is aimed at a better dissemination of information on three-dimensional biopolymer structures with an emphasis on visualization and analysis.

SCOP

the Structural Classification of Proteins [2] a detailed and comprehensive description of the structural and evolutionary relationships between all proteins whose structure is known.

SWISS-MODEL Repository

a database of annotated protein models calculated by homology modeling

TOPSAN

the Open Protein Structure Annotation Network — a wiki designed to collect, share and distribute information about protein three-dimensional structures.

7. Data submission

GEO accepts many categories of high-throughput functional genomic data, including all array-based applications and some high-throughput sequencing data. This page summarizes deposit options and formats.

We aim to make data deposit procedures as straightforward as possible and will provide as much assistance as you require to get your data submitted to GEO. If you have problems or questions about the submission procedures, just e-mail us at geo@ncbi.nlm.nih.gov with a brief description of the type of data you are trying to submit, and one of our curators will quickly get back to you.

- **Data types**
 - Array submissions
 - General
 - Affymetrix
 - Agilent
 - Nimblegen
 - Illumina
 - RT-PCR submissions
 - High-throughput sequence submissions
 - Traditional SAGE submissions
- Submission format options
- Basic requirements for submissions
- Fast facts about submitting data

Submission format options [Back to top](#)

Deciding which method to use depends on the amount of data you have to submit, the format in which your data currently exist and what applications you are familiar with. Regardless of the deposit method you choose, your final GEO records will look the same and contain equivalent information.

All deposit options described above can be used for any data type. However, the majority of GEO submitters uses common commercial arrays (Affymetrix, Agilent, Illumina or Nimblegen) each of which has unique properties and file types. It is recommended that submitters who use the 4 common commercial arrays see these recommendations:

8. Data Retrieval

In this section, you will find information on how to retrieve data directly, a series of user manuals to help with data retrieval and downloading, including processing of data after download, and documents explaining the structure of the Registry database.

The EBMT realises how important it can be for centres to have access to the data they submit and has ensured that centres can always do this. The best way is to access the data directly using the same system used for data entry. This ensures centres can retrieve their data how and when they want. If centres are unable or unwilling to do so, they can request that a copy of their data be forwarded to them by the Registry.

Retrieving data directly

Users can run columnar reports on their own data filtering the output by data items such as year of the HSCT, type of donor, diagnosis, etc. They can also run reports on aggregated data in the form of frequency tables or cross-tabulations. Centres that are members of the EBMT can also run reports on aggregated data from the whole database.

UNIT IV:

1. Searching sequence database

The most obvious first stage in the analysis of any new sequence is to perform comparisons with sequence databases to find homologues. These searches can now be performed just about anywhere and on just about any computer. In addition, there are numerous web servers for doing searches, where one can post or paste a sequence into the server and receive the results interactively:

There are many methods for sequence searching. By far the most well-known are the BLAST suite of programs. One can easily obtain versions to run locally (either at NCBI or Washington University), and there are many web pages that permit one to compare a protein or DNA sequence against a multitude of gene and protein sequence databases. To name just a few:

- National Center for Biotechnology Information (USA) Searches
- European Bioinformatics Institute (UK) Searches
- BLAST search through SBASE (domain database; ICGEB, Trieste)
- and others too numerous to mention.

One of the most important advances in sequence comparison recently has been the development of both gapped BLAST and PSI-BLAST (position specific iterated BLAST). Both of these have made BLAST much more sensitive, and the latter is able to detect very remote homologues by taking the results of one search, constructing a *profile* and then using this to search the database again to find other homologues (the process can be repeated until no new sequences are found). It is essential that one compares any new protein sequence to the database with PSI-BLAST to see if known structures can be found prior to doing any of the other methods discussed in the next sections.

Other methods for comparing a single sequence to a database include:

- The FASTA suite (William Pearson, University of Virginia, USA)
- SCANPS (Geoff Barton, European Bioinformatics Institute, UK)
- BLITZ (Compugen's fast Smith Waterman search)
- and others.

It is also possible to use multiple sequence information to perform more sensitive searches. Essentially this involves building a *profile* from some kind of multiple sequence alignment. A profile essentially gives a score for each type of amino acid at each position in the sequence, and generally makes searches more sensitive. Tools for doing this include:

- PSI-BLAST (NCBI, Washington)
- ProfileScan Server (ISREC, Geneva)
- HMMER Hidden Markov Model searching (Sean Eddy, Washington University)
- Wise package (Ewan Birney, Sanger Centre; this is for protein versus DNA comparisons)
- and several others.

A different approach for incorporating multiple sequence information into a database search is to use a MOTIF. Instead of giving every amino acid some kind of score at every position in an alignment, a motif ignores all but the most invariant positions in an alignment, and just describes the key residues that are conserved and define the family. Sometimes this is called a "signature". For example, "H-[FW]-x-[LIVM]-x-G-x(5)-[LV]-H-x(3)-[DE]" describes a family of DNA binding proteins. It can be translated as "histidine, followed by either a phenylalanine or

In bioinformatics, the **BLOSUM (BLOcks SUbstitution Matrix)** matrix is a substitution matrix used for sequence alignment of proteins. BLOSUM matrices are used to score alignments between evolutionarily divergent protein sequences. They are based on local alignments. BLOSUM matrices were first introduced in a paper by Henikoff and Henikoff.^[1] They scanned the BLOCKS database for very conserved regions of protein families (that do not have gaps in the sequence alignment) and then counted the relative frequencies of amino acids and their substitution probabilities. Then, they calculated a log-odds score for each of the 210 possible substitution pairs of the 20 standard amino acids. All BLOSUM matrices are based on observed alignments; they are not extrapolated from comparisons of closely related proteins like the PAM Matrices.

Biological background

The genetic instructions of every replicating cell in a living organism are contained within its DNA.^[2] Throughout the cell's lifetime, this information is transcribed and replicated by cellular mechanisms to produce proteins or to provide instructions for daughter cells during cell division, and the possibility exists that the DNA may be altered during these processes.^{[2][3]} This is known as a mutation. At the molecular level, there are regulatory systems that correct most — but not all — of these changes to the DNA before it is replicated. The functionality of a protein is highly dependent on its structure. Changing a single amino acid in a protein may reduce its ability to carry out this function, or the mutation may even change the function that the protein carries out. Changes like these may severely impact a crucial function in a cell, potentially causing the cell — and in extreme cases, the organism — to die. Conversely, the change may allow the cell to continue functioning albeit differently, and the mutation can be passed on to the organism's offspring. If this change does not result in any significant physical disadvantage to the offspring, the possibility exists that this mutation will persist within the population. The possibility also exists that the change in function becomes advantageous.

The 20 amino acids translated by the genetic code vary greatly by the physical and chemical properties of their side chains. However, these amino acids can be categorised into groups with similar physicochemical properties. Substituting an amino acid with another from the same

category is more likely to have a smaller impact on the structure and function of a protein than replacement with an amino acid from a different category.

Sequence alignment is a fundamental research method for modern biology. The most common sequence alignment for protein is to look for the similarity between different sequences in order to understand the evolutionarily divergent protein sequences on the molecular level, so that researchers could predict the functions initiated by those mutated genes. Matrices are applied as algorithms to calculate the similarity of different sequences of proteins; however, the utility of Dayhoff Matrix which is a widely used method before is limited due to the requirement of sequences with a similarity more than 85%. In order to fill in this gap, Henikoff and Henikoff introduced BLOSUM (BLOCKS Substitution Matrix) matrix which led to marked improvements in alignments and in searches using queries from each of the groups of related proteins.

Terminology

BLOSUM: Blocks Substitution Matrix, a substitution matrix used for sequence alignment of proteins.

Scoring metrics (statistical versus biological): When evaluating a sequence alignment, one would like to know how meaningful it is. This requires a scoring matrix, or a table of values that describes the probability of a biologically meaningful amino-acid or nucleotide residue-pair occurring in an alignment. Scores for each position are obtained frequencies of substitutions in blocks of local alignments of protein sequences. Several sets of BLOSUM matrices exist using different alignment databases, named with numbers. BLOSUM matrices with high numbers are designed for comparing closely related sequences, while those with low numbers are designed for comparing distant related sequences. For example, BLOSUM80 is used for less divergent alignments, and BLOSUM45 is used for more divergent alignments. The matrices were created by merging (clustering) all sequences that were more similar than a given percentage into one single sequence and then comparing those sequences (that were all more divergent than the given percentage value) only; thus reducing the contribution of closely related sequences. The percentage used was appended to the name, giving BLOSUM80 for example where sequences that were more than 80% identical were clustered.

BLOSUM r: the matrix built from blocks with no more than r% of similarity – E.g., BLOSUM62 is the matrix built using sequences with no more than 62% similarity. – Note: BLOSUM 62 is the default matrix for protein BLAST. Experimentation has shown that the BLOSUM-62 matrix is among the best for detecting most weak protein similarity

Construction of BLOSUM matrices

BLOSUM matrices are obtained by using blocks of similar amino acid sequences as data, then applying statistical methods to the data to obtain the similarity scores. Statistical Methods Steps :

Eliminating Sequences

Eliminating the sequences that are more than r% identical. There are two ways to eliminate the sequences. It can be done either by removing sequences from the block or just by finding similar sequences and replace them by new sequences which could represent the cluster. Eliminating is done to avoid bias of the result in favor of a certain protein.

Calculating Frequency & Probability

A database storing the sequence alignments of the most conserved regions of protein families. These alignments are used to derive the BLOSUM matrices. Only the sequences with a percentage of identity higher are used. By using the block, counting the pairs of amino acids in each column of the multiple alignment. odd ratio .

3. BLAST

In bioinformatics, **BLAST** for **B**asic **L**ocal **A**lignment **S**earch **T**ool is an algorithm for comparing primary biological sequence information, such as the amino-acid sequences of different proteins or the nucleotides of DNA sequences. A BLAST search enables a researcher to compare a query sequence with a library or database of sequences, and identify library sequences that resemble the query sequence above a certain threshold.

Different types of BLASTs are available according to the query sequences. For example, following the discovery of a previously unknown gene in the mouse, a scientist will typically

perform a BLAST search of the human genome to see if humans carry a similar gene; BLAST will identify sequences in the human genome that resemble the mouse gene based on similarity of sequence. The BLAST algorithm and program were designed by Stephen Altschul, Warren Gish, Webb Miller, Eugene Myers, and David J. Lipman at the NIH and was published in the Journal of Molecular Biology in 1990.

Background

BLAST is one of the most widely used bioinformatics programs for sequence searching.^[2] It addresses a fundamental problem in bioinformatics research. The heuristic algorithm it uses is much faster than other approaches, such as calculating an optimal alignment. This emphasis on speed is vital to making the algorithm practical on the huge genome databases currently

Before fast algorithms such as BLAST and FASTA were developed, doing database searches for protein or nucleic sequences was very time consuming because a full alignment procedure (e.g., the Smith–Waterman algorithm) was used.

While BLAST is faster than Smith-Waterman, it cannot "guarantee the optimal alignments of the query and database sequences" as Smith-Waterman does. The optimality of Smith-Waterman "ensured the best performance on accuracy and the most precise results" at the expense of time and computer power.

BLAST is more time-efficient than FASTA by searching only for the more significant patterns in the sequences, yet with comparative sensitivity. This could be further realized by understanding the algorithm of BLAST introduced below.

The BLAST algorithm and the computer program that implements it were developed by Stephen Altschul, Warren Gish, and David Lipman at the U.S. National Center for Biotechnology Information (NCBI), Webb Miller at the Pennsylvania State University, and Gene Myers at the University of Arizona. It is available on the web on the NCBI website. Alternative implementations include AB-BLAST (formerly known as WU-BLAST), FSA-BLAST and ScalaBLAST.

Input

Input sequences are in FASTA or Genbank format and weight matrix.

Output

BLAST output can be delivered in a variety of formats. These formats include HTML, plain text, and XML formatting. For NCBI's web-page, the default format for output is HTML. When performing a BLAST on NCBI, the results are given in a graphical format showing the hits found, a table showing sequence identifiers for the hits with scoring related data, as well as alignments for the sequence of interest and the hits received with corresponding BLAST scores for these. The easiest to read and most informative of these is probably the table.

If one is attempting to search for a proprietary sequence or simply one that is unavailable in databases available to the general public through sources such as NCBI, there is a BLAST program available for download to any computer, at no cost. This can be found at BLAST+ executables. There are also commercial programs available for purchase. Databases can be found from the NCBI site, as well as from Index of BLAST databases (FTP).

Process

Using a heuristic method, BLAST finds similar sequences, not by comparing either sequence in its entirety, but rather by locating short matches between the two sequences. This process of finding initial words is called seeding. It is after this first match that BLAST begins to make local alignments. While attempting to find similarity in sequences, sets of common letters, known as words, are very important. For example, suppose that the sequence contains the following stretch of letters, GLKFA. If a BLASTp was being conducted under default conditions, the word size would be 3 letters. In this case, using the given stretch of letters, the searched words would be GLK, LKF, KFA. The heuristic algorithm of BLAST locates all common three-letter words between the sequence of interest and the hit sequence, or sequences, from the database. These results will then be used to build an alignment. After making words for the sequence of interest, neighborhood words are also assembled. These words must satisfy a requirement of having a score of at least the threshold T , when compared by using a scoring

matrix. One commonly-used scoring matrix for BLASTp searches is BLOSUM62, although the optimal scoring matrix depends on sequence similarity. Once both words and neighborhood words are assembled and compiled, they are compared to the sequences in the database in order to find matches. The threshold score T determines whether or not a particular word will be included in the alignment. Once seeding has been conducted, the alignment, which is only 3 residues long, is extended in both directions by the algorithm used by BLAST. Each extension impacts the score of the alignment by either increasing or decreasing it. Should this score be higher than a pre-determined T , the alignment will be included in the results given by BLAST. However, should this score be lower than this pre-determined T , the alignment will cease to extend, preventing areas of poor alignment from being included in the BLAST results. Note, that increasing the T score limits the amount of space available to search, decreasing the number of neighborhood words, while at the same time speeding up the process of BLAST.

4. FASTA

FASTA is a DNA and protein sequence alignment software package first described (as FASTP) by David J. Lipman and William R. Pearson in 1985.^[1] Its legacy is the FASTA format which is now ubiquitous in bioinformatics.

History

The original FASTA program was designed for protein sequence similarity searching. FASTA added the ability to do DNA:DNA searches, translated protein:DNA searches, and also provided a more sophisticated shuffling program for evaluating statistical significance.^[2] There are several programs in this package that allow the alignment of protein sequences and DNA sequences.

Uses

FASTA is pronounced "fast A", and stands for "FAST-All", because it works with any alphabet, an extension of "FAST-P" (protein) and "FAST-N" (nucleotide) alignment.

The current FASTA package contains programs for protein:protein, DNA:DNA, protein:translated DNA (with frameshifts), and ordered or unordered peptide searches. Recent versions

of the FASTA package include special translated search algorithms that correctly handle frameshift errors (which six-frame-translated searches do not handle very well) when comparing nucleotide to protein sequence data.

In addition to rapid heuristic search methods, the FASTA package provides SSEARCH, an implementation of the optimal Smith-Waterman algorithm.

A major focus of the package is the calculation of accurate similarity statistics, so that biologists can judge whether an alignment is likely to have occurred by chance, or whether it can be used to infer homology. The FASTA package is available from fasta.bioch.virginia.edu.

The web-interface to submit sequences for running a search of the European Bioinformatics Institute (EBI)'s online databases is also available using the FASTA programs.

The FASTA file format used as input for this software is now largely used by other sequence database search tools (such as BLAST) and sequence alignment programs (Clustal, T-Coffee, etc.).

Search method

FASTA takes a given nucleotide or amino acid sequence and searches a corresponding sequence database by using local sequence alignment to find matches of similar database sequences.

The FASTA program follows a largely heuristic method which contributes to the high speed of its execution. It initially observes the pattern of word hits, word-to-word matches of a given length, and marks potential matches before performing a more time-consuming optimized search using a Smith-Waterman type of algorithm.

The size taken for a word, given by the parameter *ktup*, controls the sensitivity and speed of the program. Increasing the *ktup* value decreases number of background hits that are found. From the word hits that are returned the program looks for segments that contain a cluster of nearby hits. It then investigates these segments for a possible match.

There are some differences between fastn and fastp relating to the type of sequences used but both use four steps and calculate three scores to describe and format the sequence similarity results. These are:

- Identify regions of highest density in each sequence comparison. Taking a ktup to equal 1 or 2.

In this step all or a group of the identities between two sequences are found using a look up table. The ktup value determines how many consecutive identities are required for a match to be declared. Thus the lesser the ktup value: the more sensitive the search. ktup=2 is frequently taken by users for protein sequences and ktup=4 or 6 for nucleotide sequences. Short oligonucleotides are usually run with ktup = 1. The program then finds all similar **local regions**, represented as diagonals of a certain length in a dot plot, between the two sequences by counting ktup matches and penalizing for intervening mismatches. This way, **local regions** of highest density matches in a diagonal are isolated from background hits. For protein sequences BLOSUM50 values are used for scoring ktup matches. This ensures that groups of identities with high similarity scores contribute more to the local diagonal score than to identities with low similarity scores. Nucleotide sequences use the identity matrix for the same purpose. The best 10 local regions selected from all the diagonals put together are then saved.

The FASTA programs find regions of local or global similarity between Protein or DNA sequences, either by searching Protein or DNA databases, or by identifying local duplications within a sequence. Other programs provide information on the statistical significance of an alignment. Like BLAST, FASTA can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families.

Protein

- Protein-protein FASTA.
- Protein-protein Smith-Waterman (ssearch).
- Global Protein-protein (Needleman-Wunsch) (ggsearch)

- Global/Local protein-protein (glsearch)
- Protein-protein with unordered peptides (fasts)
- Protein-protein with mixed peptide sequences (fastf)

Nucleotide

- Nucleotide-Nucleotide (DNA/RNA fasta)
- Ordered Nucleotides vs Nucleotide (fastm)
- Unordered Nucleotides vs Nucleotide (fasts)

5. Sequence Filtering

Many nucleotide and amino acid sequences are highly repetitive in nature. If your query sequence contains regions of low complexity or repeats, you can end up with many non-related, high scoring sequences being found during BLAST (or FASTA) searches (*e.g.* hits against proline-rich regions or poly-A tails). In other cases, your sequence may contain regions of vector sequence, or repeat regions such as Alu sequences, that you either do not want included in your sequence, or at the very least, wish to have discluded in any searches you carry out based on sequence similarity.

Programs have been devised to filter out unwanted segments of sequence from within a larger sequence. For example, filtering low complexity or repeat regions out of your query sequence before searching a database can reduce the reporting of un-related sequences that match by chance. Filtering works by running programs that identify regions containing particular types of sequences. These regions are replaced with a series of N's (in the case of nucleotide sequences), or X's (in the case of peptide sequences). One N or one X replaces each residue in the region.

SEG and XNU are used to filter amino acid sequences. SEG finds areas of low compositional complexity, for example regions of biased amino acid composition like histidine-rich domains. XNU finds regions containing internal repeats of short periodicity, for example, poly-GXX of collagen tails. DUST is used to filter nucleotide sequences. You can run more than one filtering programs on a sequence before searching. But you still must choose filtering programs

appropriate to your sequence type. SEG, XNU and DUST can be run by themselves against your sequences (*i.e.* they do not have to be run as part of a BLAST search).

6. Multiple Sequence Alignment

A **Multiple Sequence Alignment (MSA)** is a sequence alignment of three or more biological sequences, generally protein, DNA, or RNA. In many cases, the input set of query sequences are assumed to have an evolutionary relationship by which they share a lineage and are descended from a common ancestor. From the resulting MSA, sequence homology can be inferred and phylogenetic analysis can be conducted to assess the sequences' shared evolutionary origins. Visual depictions of the alignment as in the image at right illustrate mutation events such as point mutations (single amino acid or nucleotide changes) that appear as differing characters in a single alignment column, and insertion or deletion mutations (indels or gaps) that appear as hyphens in one or more of the sequences in the alignment. Multiple sequence alignment is often used to assess sequence conservation of protein domains, tertiary and secondary structures, and even individual amino acids or nucleotides.

Multiple sequence alignment also refers to the process of aligning such a sequence set. Because three or more sequences of biologically relevant length can be difficult and are almost always time-consuming to align by hand, computational algorithms are used to produce and analyze the alignments. MSAs require more sophisticated methodologies than pairwise alignment because they are more computationally complex. Most multiple sequence alignment programs use heuristic methods rather than global optimization because identifying the optimal alignment between more than a few sequences of moderate length is prohibitively computationally

Dynamic programming and computational complexity

A direct method for producing an MSA uses the dynamic programming technique to identify the globally optimal alignment solution. For proteins, this method usually involves two sets of parameters: a gap penalty and a substitution matrix assigning scores or probabilities to the alignment of each possible pair of amino acids based on the similarity of the amino acids' chemical properties and the evolutionary probability of the mutation. For nucleotide sequences a similar gap penalty is used, but a much simpler substitution matrix, wherein only identical

matches and mismatches are considered, is typical. The scores in the substitution matrix may be either all positive or a mix of positive and negative in the case of a global alignment, but must be

Progressive alignment construction

The most widely used approach to multiple sequence alignments uses a heuristic search known as progressive technique (also known as the hierarchical or tree method), that builds up a final MSA by combining pairwise alignments beginning with the most similar pair and progressing to the most distantly related. All progressive alignment methods require two stages: a first stage in which the relationships between the sequences are represented as a tree, called a *guide tree*, and a second step in which the MSA is built by adding the sequences sequentially to the growing MSA according to the guide tree. The initial *guide tree* is determined by an efficient clustering method such as neighbor-joining or UPGMA, and may use distances based on the number of identical two letter sub-sequences (as in FASTA rather than a dynamic programming alignment). Progressive alignments are not guaranteed to be globally optimal. The primary problem is that when errors are made at any stage in growing the MSA, these errors are then propagated through to the final result. Performance is also particularly bad when all of the sequences in the set are rather distantly related. Most modern progressive methods modify their scoring function with a secondary weighting function that assigns scaling factors to individual members of the query set in a nonlinear fashion based on their phylogenetic distance from their nearest neighbors. This corrects for non-random selection of the sequences given to the alignment program.^[8]

Because progressive methods are heuristics that are not guaranteed to converge to a global optimum, alignment quality can be difficult to evaluate and their true biological significance can be obscure. A semi-progressive method that improves alignment quality and does not use a lossy heuristic while still running in polynomial time has been implemented in the program PSAlign.

Iterative methods

A set of methods to produce MSAs while reducing the errors inherent in progressive methods are classified as "iterative" because they work similarly to progressive methods but repeatedly realign the initial sequences as well as adding new sequences to the growing MSA. One reason

progressive methods are so strongly dependent on a high-quality initial alignment is the fact that these alignments are always incorporated into the final result — that is, once a sequence has been aligned into the MSA, its alignment is not considered further. This approximation improves efficiency at the cost of accuracy. By contrast, iterative methods can return to previously calculated pairwise alignments or sub-MSAs incorporating subsets of the query sequence as a means of optimizing a general objective function such as finding a high-quality alignment score.^[8]

A third popular iteration-based method called MUSCLE (multiple sequence alignment by log-expectation) improves on progressive methods with a more accurate distance measure to assess the relatedness of two sequences.^[16] The distance measure is updated between iteration stages (although, in its original form, MUSCLE contained only 2-3 iterations depending on whether refinement was enabled).

Hidden Markov models

Hidden Markov models are probabilistic models that can assign likelihoods to all possible combinations of gaps, matches, and mismatches to determine the most likely MSA or set of possible MSAs. HMMs can produce a single highest-scoring output but can also generate a family of possible alignments that can then be evaluated for biological significance. HMMs can produce both global and local alignments. Although HMM-based methods have been developed relatively recently, they offer significant improvements in computational speed, especially for sequences that contain overlapping regions. Typical HMM-based methods work by representing an MSA as a form of directed acyclic graph known as a partial-order graph, which consists of a series of nodes representing possible entries in the columns of an MSA. In this representation a column that is absolutely conserved (that is, that all the sequences in the MSA share a particular character at a particular position) is coded as a single node with as many outgoing connections as there are possible characters in the next column of the alignment. In the terms of a typical hidden Markov model, the observed states are the individual alignment columns and the "hidden" states represent the presumed ancestral sequence from which the sequences in the query set are hypothesized to have descended. An efficient search variant of the dynamic programming method, known as the Viterbi algorithm, is generally used to successively align the growing

MSA to the next sequence in the query set to produce a new MSA.^[17] This is distinct from progressive alignment methods because the alignment of prior sequences is updated at each new sequence addition. However, like progressive methods, this technique can be influenced by the order in which the sequences in the query set are integrated into the alignment, especially when the sequences are distantly related.

Phylogeny-aware methods

Non-homologous exon alignment by an iterative method (a), and by a phylogeny-aware method

Most multiple sequence alignment methods try to minimize the number of insertions/deletions (gaps) and, as a consequence, produce compact alignments. This causes several problems if the sequences to be aligned contain non-homologous regions, if gaps are informative in a phylogeny analysis. These problems are common in newly produced sequences that are poorly annotated and may contain frame-shifts, wrong domains or non-homologous splicedexons.

The first such method was developed in 2005 by Löytynoja and Goldman. The same authors released a software package called *PRANK* in 2008. *PRANK* improves alignments when insertions are present. Nevertheless, it runs slowly compared to progressive and/or iterative

Motif finding

Alignment of the seven *Drosophilacaspases* colored by motifs as identified by MEME. When motif positions and sequence alignments are generated independently, they often correlate well but not perfectly, as in this example.

Motif finding, also known as profile analysis, is a method of locating sequence motifs in global MSAs that is both a means of producing a better MSA and a means of producing a scoring matrix for use in searching other sequences for similar motifs. A variety of methods for isolating the motifs have been developed, but all are based on identifying short highly conserved patterns within the larger alignment and constructing a matrix similar to a substitution matrix that reflects the amino acid or nucleotide composition of each position in the putative motif. The alignment can then be refined using these matrices. In standard profile analysis, the matrix includes entries

for each possible character as well as entries for gaps. Alternatively, statistical pattern-finding algorithms can identify motifs as a precursor to an MSA rather than as a derivation. In many cases when the query set contains only a small number of sequences or contains only highly related sequences, pseudocounts are added to normalize the distribution reflected in the scoring matrix. In particular, this corrects zero-probability entries in the matrix to values that are small but nonzero.

Blocks analysis is a method of motif finding that restricts motifs to ungapped regions in the alignment. Blocks can be generated from an MSA or they can be extracted from unaligned sequences using a precalculated set of common motifs previously generated from known gene families.^[29] Block scoring generally relies on the spacing of high-frequency characters rather than on the calculation of an explicit substitution matrix. The BLOCKS server provides an interactive method to locate such motifs in unaligned sequences.

Use in phylogenetics

Multiple sequence alignments can be used to create a phylogenetic tree. This is made possible by two reasons. The first is because functional domains that are known in annotated sequences can be used for alignment in non-annotated sequences. The other is that conserved regions known to be functionally important can be found. This makes it possible for multiple sequence alignments to be used to analyze and find evolutionary relationships through homology between sequences. Point mutations and insertion or deletion events (called indels) can be detected.

7. CLUSTAL W

In Bioinformatics **Clustal** is a series of widely used computer programs for multiple sequence alignment. There have been many incarnations of Clustal that are listed below:

- **Clustal**: The original software for progressive alignment based on a phylogenetic tree. **ClustalV**: A rewrite of the original Clustal package that included phylogenetic tree reconstruction on the final alignment for the first time.

- **ClustalW**: command line interface^[6]
- **ClustalX**: This version has a graphical user interface. The more recent version of the software available for Windows, Mac OS, and Unix/Linux. This program is available from the Clustal Homepage or European Bioinformatics Institute ftp server.

Input/Output

This program accepts a wide range of input formats, including NBRF/PIR, FASTA, EMBL/Swiss-Prot, Clustal, GCC/MSF, GCG9 RSF, and GDE.

The output format can be one or many of the following: Clustal, NBRF/PIR, GCG/MSF, PHYLIP, GDE, or NEXUS.

Multiple sequence alignment

There are three main steps:

1. Do a pairwise alignment
2. Create a guide tree (or use a user-defined tree)
3. Use the guide tree to carry out a multiple alignment

These are done automatically when you select "Do Complete Alignment". Other options are "Do Alignment from guide tree" and "Produce guide tree only".

Setting

Users can align the sequences using the default setting, but occasionally it may be useful to customize one's own parameters.

The main parameters are the gap opening penalty, and the gap extension penalty.

8. Phylogenetics

In biology, **phylogenetics** is the study of evolutionary relationships among groups of organisms (e.g. species, populations), which are discovered through molecular sequencing data and

morphological data matrices. The term *phylogenetics* derives from the Greek terms *phylé* and *phylon* denoting "tribe", "clan", "race" and the adjectival form, of the word genesis "origin", "source", "birth".

Evolution is a process whereby populations are altered over time and may split into separate branches, hybridize together, or terminate by extinction. The evolutionary branching process may be depicted as a phylogenetic tree, and the place of each of the various organisms on the tree is based on a hypothesis about the sequence in which evolutionary branching events occurred. In historical linguistics, similar concepts are used with respect to relationships between languages; and in textual criticism with stemmatics.

Phylogenetic analyses have become essential to research on the evolutionary tree of life. For example, the RedToL aims at reconstructing the Red Algal Tree of Life. The National Science Foundation sponsors a project called the *Assembling the Tree of Life* (AToL) activity. The goal of this project is to determine evolutionary relationships across large groups of organisms throughout the history of life. The research on this project often involves large teams working across institutions and disciplines, and typically provides support to investigators working on computational phylogenetics and phyloinformatics tasks, including data acquisition, analysis, and algorithm development and dissemination.

.Construction of a phylogenetic tree

The scientific methods of phylogenetics are often grouped under the term cladistics. The most common ones are parsimony, maximum likelihood (ML), and MCMC-based Bayesian inference. All methods depend upon an implicit or explicit mathematical model describing the evolution of characters observed in the species included; all can be, and are, used for molecular data, wherein the characters are aligned nucleotide or amino acid sequences, and all but maximum likelihood (see below) can be, and are, used for phenotypic (morphological, chemical, and physiological) data (also called classical or traditional data).

Phenetics, popular in the mid-20th century but now largely obsolete, uses distance matrix-based methods to construct trees based on overall similarity in morphology or other observable traits (i.e. in the phenotype, not the DNA), which was often assumed to approximate phylogenetic

Continuous characters

Morphological characters that sample a continuum may contain phylogenetic signal, but are hard to code as discrete characters. Several methods have been used, one of which is gap coding, and there are variations on gap coding. In the original form of gap coding: If more taxa are added to the analysis, the gaps between taxa may become so small that all information is lost. Generalized gap coding works around that problem by comparing individual pairs of taxa rather than considering one set that contains all of the taxa.

Missing data

In general, the more data that are available when constructing a tree, the more accurate and reliable the resulting tree will be. Missing data are no more detrimental than simply having fewer data, although the impact is greatest when most of the missing data are in a small number of taxa. Concentrating the missing data across a small number of characters produces a more robust tree.^[15]

The role of fossils

Because many characters involve embryological, or soft-tissue or molecular characters that (at best) hardly ever fossilize, and the interpretation of fossils is more ambiguous than that of living taxa, extinct taxa almost invariably have higher proportions of missing data than living ones. However, despite these limitations, the inclusion of fossils is invaluable, as they can provide information in sparse areas of trees, breaking up long branches and constraining intermediate character states; thus, fossil taxa contribute as much to tree resolution as modern taxa. Fossils can also constrain the age of lineages and thus demonstrate how consistent a tree is with the stratigraphic record. Stratocladistics incorporates age information into data matrices for phylogenetic analyses.

9. Sequence annotation

DNA annotation or **genome annotation** is the process of (a) identifying the locations/segments of genes, coding regions and other specific locations that are of importance in a DNA sequence

or genome and (b) associating relevant information with those locations/segments (e.g. determining what the identified genes do). Once a genome is sequenced, it needs to be *annotated in order* to make sense of it.

Target audience/users: Molecular biologists and those who work in the field of Bioinformatics and also those enthusiastic software professionals who wish to work in the informatics part of the much larger universe of biological research.

The goal and the focus in building and publishing the first version of this tool was to demonstrate the possibilities and power of such tools. The current release has useful features, which are listed below - please download and use the tool. The current version has certain obvious limitations. For the future versions many more enhancements have been planned.

DNA & RNA Sequence Annotation Studio is a very useful sequence annotation visualization and modification tool and can be used to perform the following actions:
1. Load DNA or RNA sequences in GenBank format and visualize the complete sequence with zoom in / zoom out facility. *The current version only handles GenBank format (for details on GenBank format, please refer to GenBank Format)*, although it can be easily enhanced to support other formats. The current version only handles one sequence per file. Therefore, if an input file contains multiple sequences, only the first sequence will be loaded; in order to view/modify remaining sequence(s) in the input file, separate input files need to be created.

UNIT 5:

1. Drug discovery

Historically, drugs were discovered through identifying the active ingredient from traditional remedies or by serendipitous discovery. Later chemical libraries of synthetic small molecules, natural products or extracts were screened in intact cells or whole organisms to identify substances that have a desirable therapeutic effect in a process known as classical pharmacology. Since sequencing of the human genome which allowed rapid cloning and synthesis of large quantities of purified proteins, it has become common practice to use high throughput screening

of large compounds libraries against isolated biological targets which are hypothesized to be disease modifying in a process known as reverse pharmacology. Hits from these screens are then tested in cells and then in animals for efficacy.

Modern drug discovery involves the identification of screening hits, medicinal chemistry and optimization of those hits to increase the affinity, selectivity (to reduce the potential of side effects), efficacy/potency, metabolic stability (to increase the half-life), and oral bioavailability. Once a compound that fulfills all of these requirements has been identified, it will begin the process of drug development prior to clinical trials. One or more of these steps may, but not necessarily, involve computer-aided drug design. Modern drug discovery is thus usually a capital-intensive process that involves large investments by pharmaceutical industry corporations as well as national governments (who provide grants and loan guarantees). Despite advances in technology and understanding of biological systems, drug discovery is still a lengthy, "expensive, difficult, and inefficient process" with low rate of new therapeutic discovery. In 2010, the research and development cost of each new molecular entity (NME) was approximately US\$1.8 billion. Drug discovery is done by pharmaceutical companies, with research assistance from universities. The "final product" of drug discovery is a patent on the potential drug. The drug requires very expensive Phase I, II and III clinical trials, and most of them fail. Small companies have a critical role, often then selling the rights to larger companies that have the resources to run the clinical trials.

Discovering drugs that may be a commercial success, or a public health success, involves a complex interaction between investors, industry, academia, patent laws, regulatory exclusivity, marketing and the need to balance secrecy with communication. Meanwhile, for disorders whose rarity means that no large commercial success or public health effect can be expected, the orphan drug funding process ensures that people who experience those disorders can have some hope of pharmacotherapeutic advances.

2. Combinatorial chemistry

Combinatorial chemistry comprises chemical synthetic methods that make it possible to prepare a large number (tens to thousands or even millions) of compounds in a single process.

These compound libraries can be made as mixtures, sets of individual compounds or chemical structures generated *in silico*. Combinatorial chemistry can be used for the synthesis of small molecules and for peptides.

Strategies that allow identification of useful components of the libraries are also part of combinatorial chemistry. The methods used in combinatorial chemistry are applied outside chemistry, too.

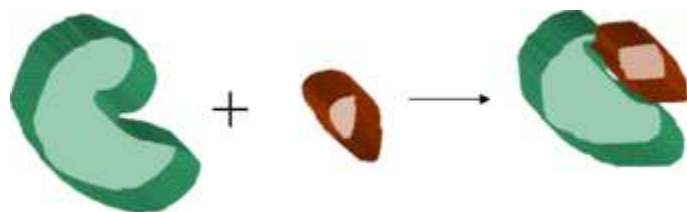
Introduction

Synthesis of molecules in a combinatorial fashion can quickly lead to large numbers of molecules. For example, a molecule with three points of diversity (R_1 , R_2 , and R_3) can generate $N_{R_1} \times N_{R_2} \times N_{R_3}$ possible structures, where N_{R_1} , N_{R_2} , and N_{R_3} are the numbers of different substituents utilized. The basic principle of combinatorial chemistry is to prepare libraries of very large number of compounds then identify the useful components of the libraries.

Although combinatorial chemistry has only really been taken up by industry since the 1990s¹ its roots can be seen as far back as the 1960s when a researcher at Rockefeller University, Bruce Merrifield, started investigating the solid-phase synthesis of peptides.

In its modern form, combinatorial chemistry has probably had its biggest impact in the pharmaceutical industry.¹ Researchers attempting to optimize the activity profile of a compound create a 'library' of many different but related compounds.¹ Advances in robotics have led to an industrial approach to combinatorial synthesis, enabling companies to routinely produce over 100,000 new and unique compounds per year. In order to handle the vast number of structural possibilities, researchers often create a 'virtual library', a computational enumeration of all possible structures of a given pharmacophore with all available reactants. Such a library can consist of thousands to millions of 'virtual' compounds. The researcher will select a subset of the 'virtual library' for actual synthesis, based upon various calculations and criteria (see ADME, computational chemistry, and QSAR).

3. Drug Docking



Small molecule docked to a protein.

In the field of molecular modeling, **docking** is a method which predicts the preferred orientation of one molecule to a second when bound to each other to form a stable complex.^[1] Knowledge of the preferred orientation in turn may be used to predict the strength of association or binding affinity between two molecules using, for example, scoring functions.

The associations between biologically relevant molecules such as proteins, nucleic acids, carbohydrates, and lipids play a central role in signal transduction. Furthermore, the relative orientation of the two interacting partners may affect the type of signal produced (e.g., agonism vs antagonism). Therefore docking is useful for predicting both the strength and type of signal produced.

Docking is frequently used to predict the binding orientation of small molecule drug candidates to their protein targets in order to in turn predict the affinity and activity of the small molecule. Hence docking plays an important role in the rational design of drugs. Given the biological and pharmaceutical significance of molecular docking, considerable efforts have been directed towards improving the methods used to predict docking.

Docking approaches

Two approaches are particularly popular within the molecular docking community. One approach uses a matching technique that describes the protein and the ligand as complementary surfaces. The second approach simulates the actual docking process in which the ligand-protein

pairwise interaction energies are calculated. Both approaches have significant advantages as well as some limitations. These are outlined below.

Shape complementarity

Geometric matching/ shape complementarity methods describe the protein and ligand as a set of features that make them dockable. These features may include molecular surface / complementary surface descriptors. In this case, the receptor's molecular surface is described in terms of its solvent-accessible surface area and the ligand's molecular surface is described in terms of its matching surface description. The complementarity between the two surfaces amounts to the shape matching description that may help finding the complementary pose of docking the target and the ligand molecules. Another approach is to describe the hydrophobic features of the protein using turns in the main-chain atoms. Yet another approach is to use a Fourier shape descriptor technique. Whereas the shape complementarity based approaches are typically fast and robust, they cannot usually model the movements or dynamic changes in the ligand/ protein conformations accurately, although recent developments allow these methods to investigate ligand flexibility. Shape complementarity methods can quickly scan through several thousand ligands in a matter of seconds and actually figure out whether they can bind at the protein's active site, and are usually scalable to even protein-protein interactions. They are also much more amenable to pharmacophore based approaches, since they use geometric descriptions of the ligands to find optimal binding.

Simulation

Simulating the docking process as such is much more complicated. In this approach, the protein and the ligand are separated by some physical distance, and the ligand finds its position into the protein's active site after a certain number of "moves" in its conformational space. The moves incorporate rigid body transformations such as translations and rotations, as well as internal changes to the ligand's structure including torsion angle rotations. Each of these moves in the conformation space of the ligand induces a total energetic cost of the system. Hence, the system's total energy is calculated after every move.

The obvious advantage of docking simulation is that ligand flexibility is easily incorporated, whereas shape complementarity techniques must use ingenious methods to incorporate flexibility in ligands. Also, it more accurately models reality, whereas shape complimentary techniques are more of an abstraction.

Clearly, simulation is computationally expensive, having to explore a large energy landscape. Grid-based techniques, optimization methods, and increased computer speed have made docking simulation more realistic.

Mechanics of docking

To perform a docking screen, the first requirement is a structure of the protein of interest. Usually the structure has been determined using a biophysical technique such as x-ray crystallography, or NMR spectroscopy. This protein structure and a database of potential ligands serve as inputs to a docking program. The success of a docking program depends on two components: the search algorithm and the scoring function.

Search algorithm

Main article: Searching the conformational space for docking

The search space in theory consists of all possible orientations and conformations of the protein paired with the ligand. However in practice with current computational resources, it is impossible to exhaustively explore the search space—this would involve enumerating all possible distortions of each molecule (molecules are dynamic and exist in an ensemble of conformational states) and all possible rotational and translational orientations of the ligand relative to the protein at a given level of granularity. Most docking programs in use account for a flexible ligand, and several attempt to model a flexible protein receptor. Each "snapshot" of the pair is referred to as a **pose**.

4.. Pharmacogenomics

Pharmacogenomics is the study of the role of genetics in drug response. It deals with the influence of acquired and inherited genetic variation on drug response in patients by correlating

gene expression or single-nucleotide polymorphisms with drug absorption, distribution, metabolism and elimination, as well as drug receptor target effects. The term pharmacogenomics is often used interchangeably with pharmacogenetics. Although both terms relate to drug response based on genetic influences, pharmacogenetics focuses on single drug-gene interactions, while pharmacogenomics encompasses a more genome-wide association approach, incorporating genomics and epigenetics while dealing with the effects of multiple genes on drug response. Pharmacogenomics aims to develop rational means to optimize drug therapy, with respect to the patients' genotype, to ensure maximum efficacy with minimal adverse effects. Through the utilization of pharmacogenomics, it is hoped that drug treatments can deviate from what is dubbed as the “one-dose-fits-all” approach. It attempts to eliminate the trial-and-error method of prescribing, allowing physicians to take into consideration their patient’s genes, the functionality of these genes, and how this may affect the efficacy of the patient’s current and/or future treatments (and where applicable, provide an explanation for the failure of past treatments).^[4] Such approaches promise the advent of "personalized medicine"; in which drugs and drug combinations are optimized for each individual's unique genetic makeup. Whether used to explain a patient’s response or lack thereof to a treatment, or act as a predictive tool, it hopes to achieve better treatment outcomes, greater efficacy, minimization of the occurrence of drug toxicities and adverse drug reactions (ADRs). For patients who have lack of therapeutic response to a treatment, alternative therapies can be prescribed that would best suit their requirements. In order to provide pharmacogenomic-based recommendations for a given drug, two possible types of input can be used: genotyping or exome or whole genomesequencing. equencing provides many more data points, including detection of mutations that prematurely terminate the synthesized protein (early stop codon)

Applications

The list below provides a few more commonly known applications of pharmacogenomics:
Improve drug safety, and reduce ADRs;

- Tailor treatments to meet patients unique genetic pre-disposition, identifying optimal dosing;
- Improve drug discovery targeted to human disease; and

- Improve proof of principle for efficacy trials.

Pharmacogenomics may be applied to several areas of medicine, including Pain Management, Cardiology, Oncology, and Psychiatry. A place may also exist in Forensic Pathology, in which pharmacogenomics can be used to determine the cause of death in drug-related deaths where no findings emerge using autopsy. In cancer treatment, pharmacogenomics tests are used to identify which patients are most likely to respond to certain cancer drugs. In behavioral health, pharmacogenomic tests provide tools for physicians and care givers to better manage medication selection and side effect amelioration. Pharmacogenomics is also known as companion diagnostics, meaning tests being bundled with drugs. Examples include KRAS test with cetuximab and EGFR test with gefitinib. Beside efficacy, germline pharmacogenetics can help to identify patients likely to undergo severe toxicities when given cytotoxics showing impaired detoxification in relation with genetic polymorphism, such as canonical 5-FU.

5.. Pharmacogenetics

Pharmacogenetics is the study of inherited genetic differences in drug metabolic pathways which can affect individual responses to drugs, both in terms of therapeutic effect as well as adverse effects. The term pharmacogenetics is often used interchangeably with the term pharmacogenomics which also investigates the role of acquired and inherited genetic differences in relation to drug response and drug behavior through a systematic examination of genes, gene products, and inter- and intra-individual variation in gene expression and function. In oncology, *pharmacogenetics* historically is the study of germline mutations (e.g., single-nucleotide polymorphisms affecting genes coding for liver enzymes responsible for drug deposition and pharmacokinetics), whereas *pharmacogenomics* refers to somatic mutations in tumoral DNA leading to alteration in drug response (e.g., KRAS mutations in patients treated

6. Toxicogenomics

Toxicogenomics is a field of science that deals with the collection, interpretation, and storage of information about gene and protein activity within particular cell or tissue of an organism in response to toxic substances. Toxicogenomics combines toxicology with genomics or other high

throughput molecular profiling technologies such as transcriptomics, proteomics and metabolomics. Toxicogenomics endeavors to elucidate molecular mechanisms evolved in the expression of toxicity, and to derive molecular expression patterns (i.e., molecular biomarkers) that predict toxicity or the genetic susceptibility to it.

In pharmaceutical research toxicogenomics is defined as the study of the structure and function of the genome as it responds to adverse xenobiotic exposure. It is the toxicological subdiscipline of pharmacogenomics, which is broadly defined as the study of inter-individual variations in whole-genome or candidate gene single-nucleotide polymorphism maps, haplotype markers, and alterations in gene expression that might correlate with drug responses. Though the term toxicogenomics first appeared in the literature in 1999 it was already in common use within the pharmaceutical industry as its origin was driven by marketing strategies from vendor companies. The term is still not universal accepted, and others have offered alternative terms such as chemogenomics to describe essentially the same area. In pharmaceutical drug discovery and development toxicogenomics is used to study adverse, i.e. toxic, effects, of pharmaceutical drugs in defined model systems in order to draw conclusions on the toxic risk to patients or the environment. Both the EPA and the U.S. Food and Drug Administration currently preclude basing regulatory decision making on genomics data alone. However, they do encourage the voluntary submission of well-documented, quality genomics data. Both agencies are considering the use of submitted data on a case-by-case basis for assessment purposes (e.g., to help elucidate mechanism of action or contribute to a weight-of-evidence approach) or for populating relevant comparative databases by encouraging parallel submissions of genomics data and traditional toxicologic test results.

7. Functional Genomics

Functional genomics is a field of molecular biology that attempts to make use of the vast wealth of data produced by genomic projects (such as genome sequencing projects) to describe gene (and protein) functions and interactions. Unlike genomics, functional genomics focuses on the dynamic aspects such as gene transcription, translation, and protein–protein interactions, as opposed to the static aspects of the genomic information such as DNA sequence or structures. Functional genomics attempts to answer questions about the function of DNA at the levels of

genes, RNA transcripts, and protein products. A key characteristic of functional genomics studies is their genome-wide approach to these questions, generally involving high-throughput methods rather than a more traditional “gene-by-gene” approach.

Goals of functional genomics

The goal of functional genomics is to understand the relationship between an organism's genome and its phenotype. The term functional genomics is often used broadly to refer to the many possible approaches to understanding the properties and function of the entirety of an organism's genes and gene products. This definition is somewhat variable; Gibson and Muse define it as "approaches under development to ascertain the biochemical, cellular, and/or physiological properties of each and every gene product",^[1] while Pevsner includes the study of nongenic elements in his definition: "the genome-wide study of the function of DNA (including genes and nongenic elements), as well as the nucleic acid and protein products encoded by DNA". Functional genomics involves studies of natural variation in genes, RNA, and proteins over time (such as an organism's development) or space (such as its body regions), as well as studies of natural or experimental functional disruptions affecting genes, chromosomes, RNA, or proteins.

The promise of functional genomics is to expand and synthesize genomic and proteomic knowledge into an understanding of the dynamic properties of an organism at cellular and/or organismal levels. This would provide a more complete picture of how biological function arises from the information encoded in an organism's genome. The possibility of understanding how a particular mutation leads to a given phenotype has important implications for human genetic diseases, as answering these questions could point scientists in the direction of a treatment or cure.

Techniques and applications

Functional genomics includes function-related aspects of the genome itself such as mutation and polymorphism (such as single nucleotide polymorphism (SNP) analysis), as well as measurement of molecular activities. The latter comprise a number of "-omics" such as transcriptomics (gene expression), proteomics (protein expression), and metabolomics. Functional genomics uses

mostly multiplex techniques to measure the abundance of many or all gene products such as mRNAs or proteins within a biological sample. Together these measurement modalities endeavor to quantitate the various biological processes and improve our understanding of gene and protein functions and interactions.

8. Metabolomics

Metabolomics is the scientific study of chemical processes involving metabolites. Specifically, metabolomics is the "systematic study of the unique chemical fingerprints that specific cellular processes leave behind", the study of their small-molecule metabolite profiles. The metabolome represents the collection of all metabolites in a biological cell, tissue, organ or organism, which are the end products of cellular processes. mRNA gene expression data and proteomic analyses reveal the set of gene products being produced in the cell, data that represents one aspect of cellular function. Conversely, metabolic profiling can give an instantaneous snapshot of the physiology of that cell. One of the challenges of systems biology and functional genomics is to integrate proteomic, transcriptomic, and metabolomic information to provide a better understanding of cellular biology.

Metabolome refers to the complete set of small-molecule metabolites (such as metabolic intermediates, hormones and other signaling molecules, and secondary metabolites) to be found within a biological sample, such as a single organism.^{[18][19]} The word was coined in analogy with transcriptomics and proteomics; like the transcriptome and the proteome, the metabolome is dynamic, changing from second to second. Although the metabolome can be defined readily enough, it is not currently possible to analyse the entire range of metabolites by a single analytical method. The first metabolite database (called METLIN) for searching m/z values from mass spectrometry data was developed by scientists at The Scripps Research Institute in 2005.^[12] In January 2007, scientists at the University of Alberta and the University of Calgary completed the first draft of the human metabolome. They catalogued approximately 2500 metabolites, 1200 drugs and 3500 food components that can be found in the human body, as reported in the literature. This information, available at the Human Metabolome Database (www.hmdb.ca) and based on analysis of information available in the current scientific literature, is far from complete. In contrast, much more is known about the metabolomes of other organisms. For

example, over 50,000 metabolites have been characterized from the plant kingdom, and many thousands of metabolites have been identified and/or characterized from single plants.

Each type of cell and tissue has a unique metabolic 'fingerprint' that can elucidate organ or tissue-specific information, while the study of biofluids can give more generalized though less specialized information. Commonly used biofluids are urine and plasma, as they can be obtained non-invasively or relatively non-invasively, respectively.^[23] The ease of collection facilitates high temporal resolution, and because they are always at dynamic equilibrium with the body, they can describe the host as a whole.^[24]

Metabolites

Metabolites are the intermediates and products of metabolism. Within the context of metabolomics, a metabolite is usually defined as any molecule less than 1 kDa in size.^[25] However, there are exceptions to this depending on the sample and detection method. For example, macromolecules such as lipoproteins and albumin are not directly involved in those processes, but usually has important ecological function. Examples include antibiotics and pigments.^[27] By contrast, in human-based metabolomics, it is more common reliably detected in NMR-based metabolomics studies of blood plasma.^[26] In plant-based metabolomics, it is common to refer to "primary" and "secondary" metabolites. A primary metabolite is directly involved in the normal growth, development, and reproduction. A secondary metabolite is to describe metabolites as being either endogenous (produced by the host organism) or exogenous.^[28] Metabolites of foreign substances such as drugs are termed xenometabolites.

9. Metabolic databases

BRENDA, the enzyme database, has comprehensive information on enzymes and enzymatic reactions. It is one of several databases nested within the metabolic pathway database set of the SRS5 sequence retrieval system at EBI.

KEGG Metabolic Pathways include graphical pathway maps for all known metabolic pathways from various organisms. Ortholog group tables, containing conserved, functional units in a molecular pathway or assembly as well comparative lists of genes for a given functional unit in different organisms, are also available.

The WIT Metabolic Reconstruction project produces metabolic reconstructions for sequenced, or partially sequenced, genomes. It currently provides a set of over 25 such reconstructions in varying states of completion. Over 2900 pathway diagrams are available, associated with functional roles and linked to ORFs.

EcoCyc describes the genome and the biochemical machinery of *E. coli*. It provides a molecular and functional catalog of the *E. coli* cell to facilitate system-level understanding. Its Pathway/Genome Navigator user interface visualizes the layout of genes, of individual biochemical reactions, or of complete pathways. It also supports computational studies of the metabolism, such as pathway design, evolutionary studies, and simulations. A related metabolic database is Metalgen.

Boehringer Mannheim - Biochemical Pathways is a searchable database of metabolic pathways, enzymes, substrates and products. Based on a given search, it produces a graphic representation of the relevant pathway(s) within the context of an enormous metabolic map. Neighboring metabolic reactions can then be viewed through links to adjacent maps.

10. Microfluidics

Microfluidics is a multidisciplinary field intersecting engineering, physics, chemistry, biochemistry, nanotechnology, and biotechnology, with practical applications to the design of systems in which small volumes of fluids will be handled. Microfluidics emerged in the beginning of the 1980s and is used in the development of inkjet printheads, DNA chips, lab-on-a-chip technology, micro-propulsion, and micro-thermal technologies. It deals with the behavior,

precise control and manipulation of fluids that are geometrically constrained to a small, typically sub-millimeter, scale. Typically, **micro** means one of the following features:

- small volumes (μL , nL, pL, fL)
- small size
- low energy consumption
- effects of the micro domain

Typically fluids are moved, mixed, separated or otherwise processed. Numerous applications employ passive fluid control techniques like capillary forces. In some applications external actuation means are additionally used for a directed transport of the media. Examples are rotary drives applying centrifugal forces for the fluid transport on the passive chips. **Active microfluidics** refers to the defined manipulation of the working fluid by active (micro) components as micropumps or micro valves. Micro pumps supply fluids in a continuous manner or are used for dosing. Micro valves determine the flow direction or the mode of movement of pumped liquids. Often processes which are normally carried out in a lab are miniaturized on a single chip in order to enhance efficiency and mobility as well as reducing sample and reagent volumes.

Key application areas

Continuous-flow microfluidics

These technologies are based on the manipulation of continuous liquid flow through microfabricated channels. Actuation of liquid flow is implemented either by external pressure sources, external mechanical pumps, integrated mechanical micropumps, or by combinations of capillary forces and electrokinetic mechanisms. Continuous-flow microfluidic operation is the mainstream approach because it is easy to implement and less sensitive to protein fouling problems. Continuous-flow devices are adequate for many well-defined and simple biochemical applications, and for certain tasks such as chemical separation, but they are less suitable for tasks requiring a high degree of flexibility or ineffect fluid manipulations. These closed-channel systems are inherently difficult to integrate and scale because the parameters that govern flow field vary along the flow path making the fluid flow at any one location dependent on the

properties of the entire system. Permanently etched microstructures also lead to limited reconfigurability and poor fault tolerance capability.

Process monitoring capabilities in continuous-flow systems can be achieved with highly sensitive microfluidic flow sensors based on MEMS technology which offer resolutions down to the nanoliter range.

Droplet-based microfluidics

Droplet-based microfluidics as a subcategory of microfluidics in contrast with continuous microfluidics has the distinction of manipulating discrete volumes of fluids in immiscible phases with low Reynolds number and laminar flow regimes. Interest in droplet-based microfluidics systems has been growing substantially in past decades. Microdroplets offer the feasibility of handling miniature volumes of fluids conveniently, provide better mixing and are suitable for high throughput experiments. Exploiting the benefits of droplet based microfluidics efficiently requires a deep understanding of droplet generation,^[19] droplet motion, droplet merging, and droplet breakup

DNA chips (microarrays)

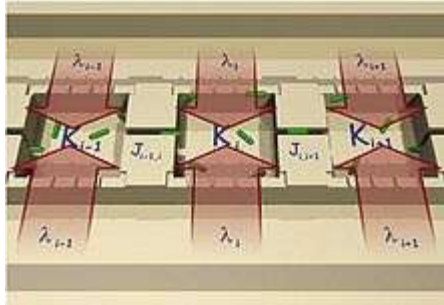
Early biochips were based on the idea of a DNA microarray, e.g., the GeneChip DNAarray from Affymetrix, which is a piece of glass, plastic or silicon substrate on which pieces of DNA (probes) are affixed in a microscopic array. Similar to a DNA microarray, a protein array is a miniature array where a multitude of different capture agents, most frequently monoclonal antibodies, are deposited on a chip surface; they are used to determine the presence and/or amount of proteins in biological samples, e.g., blood. A drawback of DNA and protein arrays is that they are neither reconfigurable nor scalable after manufacture. Digital microfluidics has been described as a means for carrying out Digital PCR.

Molecular biology

In addition to microarrays, biochips have been designed for two-dimensional electrophoresis, transcriptome analysis, and PCR amplification. Other applications include various

electrophoresis and liquid chromatography applications for proteins and DNA, cell separation, in particular blood cell separation, protein analysis, cell manipulation and analysis including cell viability analysis and microorganism capturing.

Evolutionary biology



Three Micro Habitat Patches MHPs connected by dispersal corridors (indicated here as $J_{i,j}$) into a 1D lattice. The ecosystem service (of habitat renewal) to each MHP represented here as λ_i (red arrows). Each MHP can also hold different carrying capacity K_i for its supporting local population of bacterial cells (depicted in green).

11. Nanotechnology

Nanotechnology ("nanotech") is the manipulation of matter on an atomic, molecular, and supramolecular scale. The earliest, widespread description of nanotechnology referred to the particular technological goal of precisely manipulating atoms and molecules for fabrication of macroscale products, also now referred to as molecular nanotechnology. A more generalized description of nanotechnology was subsequently established by the National Nanotechnology Initiative, which defines nanotechnology as the manipulation of matter with at least one dimension sized from 1 to 100 nanometers. This definition reflects the fact that quantum mechanical effects are important at this quantum-realm scale, and so the definition shifted from a particular technological goal to a research category inclusive of all types of research and technologies that deal with the special properties of matter that occur below the given size threshold. It is therefore common to see the plural form "nanotechnologies" as well as "nanoscale technologies" to refer to the broad range of research and applications whose common trait is size. Because of the variety of potential applications (including industrial and military), governments

have invested billions of dollars in nanotechnology research. Through its National Nanotechnology Initiative, the USA has invested 3.7 billion dollars. The European Union has invested 1.2 billion and Japan 750 million dollars. Nanotechnology as defined by size is naturally very broad, including fields of science as diverse as surface science, organic chemistry, molecular biology, semiconductor physics, microfabrication, etc. The associated research and applications are equally diverse, ranging from extensions of conventional device physics to completely new approaches based upon molecular self-assembly, from developing new materials with dimensions on the nanoscale to direct control of matter on the atomic scale.

Scientists currently debate the future implications of nanotechnology. Nanotechnology may be able to create many new materials and devices with a vast range of applications, such as in medicine, electronics, biomaterials energy production, and consumer products. On the other hand, nanotechnology raises many of the same issues as any new technology, including concerns about the toxicity and environmental impact of nanomaterials, and their potential effects on global economics, as well as speculation about various doomsday scenarios. These concerns have led to a debate among advocacy groups and governments on whether special regulation of nanotechnology is warranted.