SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO AND CHEMICAL ENGINEERING

DEPARTMENT OF BIOTECHNOLOGY

Unit 1 – Introduction to Bioinformatics (Elective) – SBB1609

I HISTORY OF BIOINFORMATICS

Bioinformatics is an interdisciplinary field that develops methods and software tools for understanding biologicaldata. As an interdisciplinary field of science, bioinformatics combines computer science, statistics, mathematics, and engineering to analyze and interpret biological data. Bioinformatics has been used for in silico analyses of biological queries using mathematical and statistical techniques. Bioinformatics derives knowledge from computer analysis of biological data. These can consist of the information stored in the genetic code, but also experimental results from various sources, patient statistics, and scientific literature. Research in bioinformatics includes method development for storage, retrieval, and analysis of the data. Bioinformatics is a rapidly developing branch of biology and is highly interdisciplinary, using techniques and concepts from informatics, statistics, mathematics, chemistry, biochemistry, physics, and linguistics. It has many practical applications in different areas of biology and medicine.

Bioinformatics: Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data.

Computational Biology: The development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems.

"Classical" bioinformatics: "The mathematical, statistical and computing methods that aim to solve biological problems using DNA and amino acid sequences and related information."

The National Center for Biotechnology Information (NCBI 2001) defines bioinformatics as: "Bioinformatics is the field of science in which biology, computer science, and information technology merge into a single discipline. There are three important sub-disciplines within bioinformatics: the development of new algorithms and statistics with which to assess relationships among members of large data sets; the analysis and interpretation of various types of data including nucleotide and amino acid sequences, protein domains, and protein structures; and the development and implementation of tools that enable efficient access and management of different types of information

Even though the three terms: bioinformatics, computational biology and bioinformation infrastructure are often times used interchangeably, broadly, the three may be defined as follows:

1. bioinformatics refers to database-like activities, involving persistent sets of data that are maintained in a consistent state over essentially indefinite periods of time;

2. computational biology encompasses the use of algorithmic tools to facilitate biological analyses; while

3. bioinformation infrastructure comprises the entire collective of information management systems, analysis tools and communication networks supporting biology. Thus, the latter may be viewed as a computational scaffold of the former two.

There are three important sub-disciplines within bioinformatics:

- the development of new algorithms and statistics with which to assess relationships among members of large data sets;
- the analysis and interpretation of various types of data including nucleotide and amino acid sequences, protein domains, and protein structures;
- and the development and implementation of tools that enable efficient access and management of different types of information

Bioinformatics definition - other sources

- Bioinformatics or computational biology is the use of mathematical and informational techniques, including statistics, to solve biological problems, usually by creating or using computer programs, mathematical models or both. One of the main areas of bioinformatics is the data mining and analysis of the data gathered by the various genome projects. Other areas are sequence alignment, protein structure prediction, systems biology, protein-protein interactions and virtual evolution. (source: www.answers.com)
- Bioinformatics is the science of developing computer databases and algorithms for the purpose of speeding up and enhancing biological research. (source: www.whatis.com)
- "Biologists using computers, or the other way around. Bioinformatics is more of a tool

than a discipline.(source: An Understandable Definition of Bioinformatics , The O'Reilly Bioinformatics Technology Conference, 2003) (4)

- The application of computer technology to the management of biological information. Specifically, it is the science of developing computer databases and algorithms to facilitate and expedite biological research.(source: Webopedia)
- Bioinformatics: a combination of Computer Science, Information Technology and Genetics to determine and analyze genetic information. (Definition from BitsJournal.com)
- Bioinformatics is the application of computer technology to the management and analysis of biological data. The result is that computers are being used to gather, store, analyse and merge biological data.(EBI - 2can resource)
- Bioinformatics is concerned with the creation and development of advanced information and computational technologies to solve problems in biology.
- Bioinformatics uses techniques from informatics, statistics, molecular biology and high-performance computing to obtain information about genomic or protein sequence data.

Bioinformaticist versus a Bioinformatician

A bioinformaticist is an expert who not only knows how to use bioinformatics tools, but also knows how to write interfaces for effective use of the tools.

A bioinformatician , on the other hand, is a trained individual who only knows to use bioinformatics tools without a deeper understanding.

Aims of Bioinformatics

In general, the aims of bioinformatics are three-fold.

1. The first aim of bioinformatics is to store the biological data organized in form of a database. This allows the researchers an easy access to existing information and submit new entries. These data must be annoted to give a suitable meaning or to assign its functional characteristics. The databases must also be able to correlate between different hierarchies of information. For example: GenBank for nucleotide and protein sequence information, Protein Data Bank for 3D macromolecular structures, etc.

2. The second aim is to develop tools and resources that aid in the analysis of data. For example: BLAST to find out similar nucleotide/amino-acid sequences, ClustalW to align two or more nucleotide/amino-acid sequences, Primer3 to design primers probes for PCR techniques, etc.

3. The third and the most important aim of bioinformatics is to exploit these computational tools to analyze the biological data interpret the results in a biologically meaningful manner.

Goals

The goals of bioinformatics thus is to provide scientists with a means to explain

1. Normal biological processes
2. Malfunctions in these processes which lead to diseases
3. Approaches to improving drug discovery

To study how normal cellular activities are altered in different disease states, the biological data must be combined to form a comprehensive picture of these activities. Therefore, the field of bioinformatics has evolved such that the most pressing task now involves the analysis and interpretation of various types of data. This includes nucleotide and amino acid sequences, protein domains, and protein structures. The actual process of analyzing and interpreting data is referred to as computational biology.

Important sub-disciplines within bioinformatics and computational biology include:

- Development and implementation of computer programs that enable efficient access to, use and management of, various types of information
- Development of new algorithms (mathematical formulas) and statistical measures that assess relationships among members of large data sets. For example, there are methods to locate a gene within a sequence, to predict protein structure and/or function, and to cluster protein sequences into families of related sequences.

The primary goal of bioinformatics is to increase the understanding of biological processes. What sets it apart from other approaches, however, is its focus on developing and applying

computationally intensive techniques to achieve this goal. Examples include: pattern recognition, data mining, machine learning algorithms, and visualization. Major research efforts in the field include sequence alignment, gene finding, genome assembly, drug design, drug discovery, protein structure alignment, protein structure prediction, prediction of gene expression and protein–protein interactions, genome-wide association studies, the modeling of evolution and cell division/mitosis.

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

Tools: Used in three areas

- Molecular Sequence Analysis
- Molecular Structural Analysis
- Molecular Functional Analysis

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

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

Bioinformatics encompasses the use of tools and techniques from three separate disciplines; molecular biology (the source of the data to be analyzed), computer science (supplies the hardware for running analysis and the networks to communicate the results), and the data analysis algorithms which strictly define bioinformatics. For this reason, the editors have decided to incorporate events from these areas into a brief history of the field.

A SHORT HISTORY OF BIOINFORMATICS

- 1933 A new technique, electrophoresis, is introduced by Tiselius for separating proteins in solution.

- 1951 Pauling and Corey propose the structure for the alpha-helix and beta-sheet (Proc. Natl. Acad. Sci. USA, 27: 205-211, 1951; Proc. Natl. Acad. Sci. USA, 37: 729- 740, 1951).
- 1953 Watson and Crick propose the double helix model for DNA based on x-ray data obtained by Franklin and Wilkins (Nature, 171: 737-738, 1953).
- 1954 Perutz's group develop heavy atom methods to solve the phase problem in protein crystallography.
- 1955 The sequence of the first protein to be analyzed, bovine insulin, is announced by F. Sanger.
- 1969 The ARPANET is created by linking computers at Stanford and UCLA.
- 1970 The details of the Needleman-Wunsch algorithm for sequence comparison are published.
- 1972 The first recombinant DNA molecule is created by Paul Berg and his group.
- 1973 The Brookhaven Protein Data Bank is announced (Acta. Cryst. B, 1973, 29: 1746).
- Robert Metcalfe receives his Ph.D. from Harvard University. His thesis describes Ethernet.
- 1974 Vint Cerf and Robert Kahn develop the concept of connecting networks of computers into an "internet" and develop the Transmission Control Protocol (TCP).
- 1975 Microsoft Corporation is founded by Bill Gates and Paul Allen.
- Two-dimensional electrophoresis, where separation of proteins on SDS polyacrylamide gel is combined with separation according to isoelectric points, is announced by P. H. O'Farrell (J. Biol. Chem., 250: 4007-4021, 1975).
- E. M. Southern published the experimental details for the Southern Blot technique of specific sequences of DNA (J. Mol. Biol., 98: 503-517, 1975).
- 1977 The full description of the Brookhaven PDB (http://www.pdb.bnl.gov) is published (Bernstein, F.C.; Koetzle, T.F.; Williams, G.J.B.; Meyer, E.F.; Brice, M.D.; Rodgers, J.R.; Kennard, O.; Shimanouchi, T.; Tasumi, M.J.; J. Mol. Biol., 1977, 112:, 535).
- Allan Maxam and Walter Gilbert (Harvard) and Frederick Sanger (U.K. Medical Research Council), report methods for sequencing DNA.
- 1980 The first complete gene sequence for an organism (FX174) is published. The gene consists of 5,386 base pairs which code nine proteins.

- Wuthrich et. al. publish paper detailing the use of multi-dimensional NMR for protein structure determination (Kumar, A.; Ernst, R.R.; Wuthrich, K.; Biochem. Biophys. Res. Comm., 1980, 95:, 1).
- IntelliGenetics, Inc. founded in California. Their primary product is the IntelliGenetics Suite of programs for DNA and protein sequence analysis.
- 1981 The Smith-Waterman algorithm for sequence alignment is published.
- IBM introduces its Personal Computer to the market.
- 1982 Genetics Computer Group (GCG) created as a part of the University of Wisconsin of Wisconsin Biotechnology Center. The company's primary product is The Wisconsin Suite of molecular biology tools.
- 1983 The Compact Disk (CD) is launched.
- 1984 Jon Postel's Domain Name System (DNS) is placed on-line.
- The Macintosh is announced by Apple Computer.
- 1985 The FASTP algorithm is published.
- The PCR reaction is described by Kary Mullis and co-workers.
- 1986 The term "Genomics" appeared for the first time to describe the scientific discipline of mapping, sequencing, and analyzing genes. The term was coined by Thomas Roderick as a name for the new journal.
- Amoco Technology Corporation acquires IntelliGenetics.
- NSFnet debuts.
- The SWISS-PROT database is created by the Department of Medical Biochemistry of the University of Geneva and the European Molecular Biology Laboratory (EMBL).
- 1987 The use of yeast artifical chromosomes (YAC) is described (David T. Burke, et. al., Science, 236: 806-812).
- The physical map of E. coli is published (Y. Kohara, et. al., Cell 51: 319-337).
- 1988 The National Center for Biotechnology Information (NCBI) is established at the National Cancer Institute.
- The Human Genome Initiative is started (Commission on Life Sciences, National Research Council. Mapping and Sequencing the Human Genome, National Academy Press: Washington, D.C.), 1988.
- The FASTA algorithm for sequence comparison is published by Pearson and Lupman.
- A new program, an Internet computer virus designed by a student, infects 6,000 military computers in the US.

- 1989 The Genetics Computer Group (GCG) becomes a private company.
- Oxford Molecular Group, Ltd. (OMG) founded in Oxford, UK by Anthony Marchington, David Ricketts, James Hiddleston, Anthony Rees, and W. Graham Richards. Primary products: Anaconda, Asp, Cameleon and others (molecular modeling, drug design, protein design).
- 1990 The BLAST program (Altschul, et. al.) is implemented.
- Molecular Applications Group is founded in California by Michael Levitt and Chris Lee. Their primary products are Look and SegMod which are used for molecular modeling and protein design.
- InforMax is founded in Bethesda, MD. The company's products address sequence analysis, database and data management, searching, publication graphics, clone construction, mapping and primer design.
- 1991 The research institute in Geneva (CERN) announces the creation of the protocols which make-up the World Wide Web.
- The creation and use of expressed sequence tags (ESTs) is described (J. Craig Venter, et. al., Science, 252: 1651-1656).
- Incyte Pharmaceuticals, a genomics company headquartered in Palo Alto California, is formed.
- Myriad Genetics, Inc. is founded in Utah. The company's goal is to lead in the discovery of major common human disease genes and their related pathways. The Company has discovered and sequenced, with its academic collaborators, the following major genes: BRCA1, BRCA2, CHD1, MMAC1, MMSC1, MMSC2, CtIP, p16, p19, and MTS2.
- 1992 Human Genome Systems, Gaithersburg Maryland, is formed by William Haseltine.
- The Institute for Genomic Research (TIGR) is established by Craig Venter.
- Genome Therapeutics announces its incorporation.
- Mel Simon and coworkers announce the use of BACs for cloning.
- 1993 CuraGen Corporation is formed in New Haven, CT.
- Affymetrix begins independent operations in Santa Clara, California
- 1994
- Netscape Comminications Corporation founded and releases Navigator, the commercial version of NCSA's Mozilla.
- Gene Logic is formed in Maryland.

- The PRINTS database of protein motifs is published by Attwood and Beck.
- Oxford Molecular Group acquires IntelliGenetics.
- 1995 The Haemophilus influenzea genome (1.8 Mb) is sequenced.
- The Mycoplasma genitalium genome is sequenced.
- 1996 Oxford Molecular Group acquires the MacVector product from Eastman Kodak.
- The genome for Saccharomyces cerevisiae (baker's yeast, 12.1 Mb) is sequenced.
- The Prosite database is reported by Bairoch, et.al.
- Affymetrix produces the first commercial DNA chips.
- 1997 The genome for E. coli (4.7 Mbp) is published.
- Oxford Molecular Group acquires the Genetics Computer Group.
- LION bioscience AG founded as an integrated genomics company with strong focus on bioinformatics. The company is built from IP out of the European Molecular Biology Laboratory (EMBL), the European Bioinformatics Institute (EBI), the German Cancer Research Center (DKFZ), and the University of Heidelberg.
- Paradigm Genetics Inc., a company focussed on the application of genomic technologies to enhance worldwide food and fiber production, is founded in Research Triangle Park, NC.
- deCode genetics publishes a paper that described the location of the FET1 gene, which is responsible for familial essential tremor, on chromosome 13 (Nature Genetics).
- 1998 The genomes for Caenorhabditis elegans and baker's yeast are published.
- The Swiss Institute of Bioinformatics is established as a non-profit foundation.
- Craig Venter forms Celera in Rockville, Maryland.
- PE Informatics was formed as a Center of Excellence within PE Biosystems. This center brings together and leverages the complementary expertise of PE Nelson and Molecular Informatics, to further complement the genetic instrumentation expertise of Applied Biosystems.
- Inpharmatica, a new Genomics and Bioinformatics company, is established by University College London, the Wolfson Institute for Biomedical Research, five leading scientists from major British academic centers and Unibio Limited.
- GeneFormatics, a company dedicated to the analysis and prediction of protein structure and function, is formed in San Diego.

- Molecular Simulations Inc. is acquired by Pharmacopeia
- 1999 deCode genetics maps the gene linked to pre-eclampsia as a locus on chromosome 2p13.
- 2000 The genome for Pseudomonas aeruginosa (6.3 Mbp) is published.
- The A. thaliana genome (100 Mb) is secquenced.
- The D. melanogaster genome (180Mb) is sequenced.
- Pharmacopeia acquires Oxford Molecular Group.
- 2001 The human genome (3,000 Mbp) is published.
- 2002 Chang Gung Genomic Research Center established.
- -Bioinformatics Center, -Proteomics Center, -Microarray Center

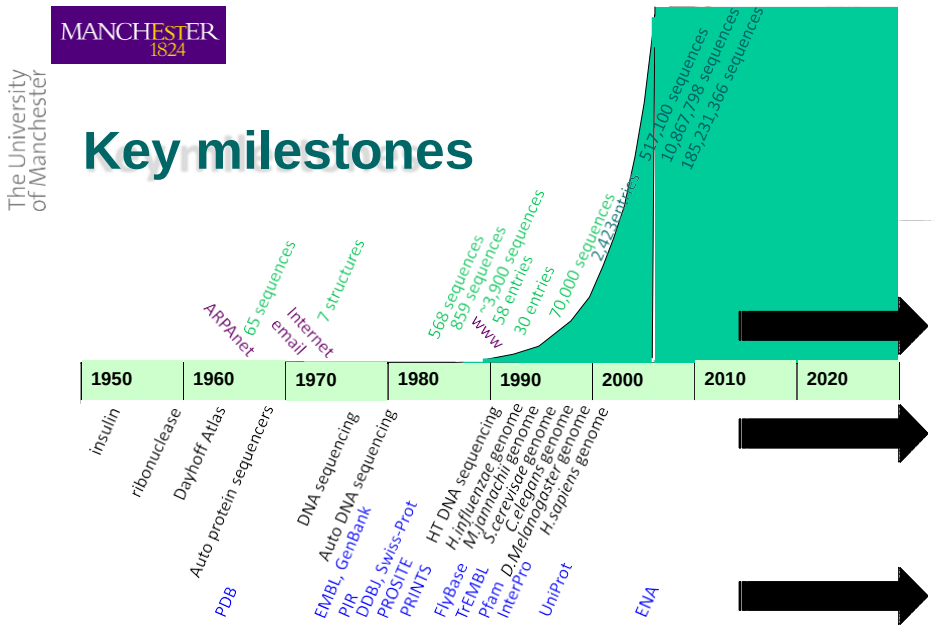


Figure 1

Applications

Bioinformatics joins mathematics, statistics, and computer science and information technology to solve complex biological problems. These problems are usually at the molecular level which cannot be solved by other means. This interesting field of science has many applications and research areas where it can be applied.

All the applications of bioinformatics are carried out in the user level. Here is the biologist including the students at various level can use certain applications and use the output in their research or in study. Various bioinformatics application can be categorized under following groups:

- Sequence Analysis
- Function Analysis
- Structure Analysis

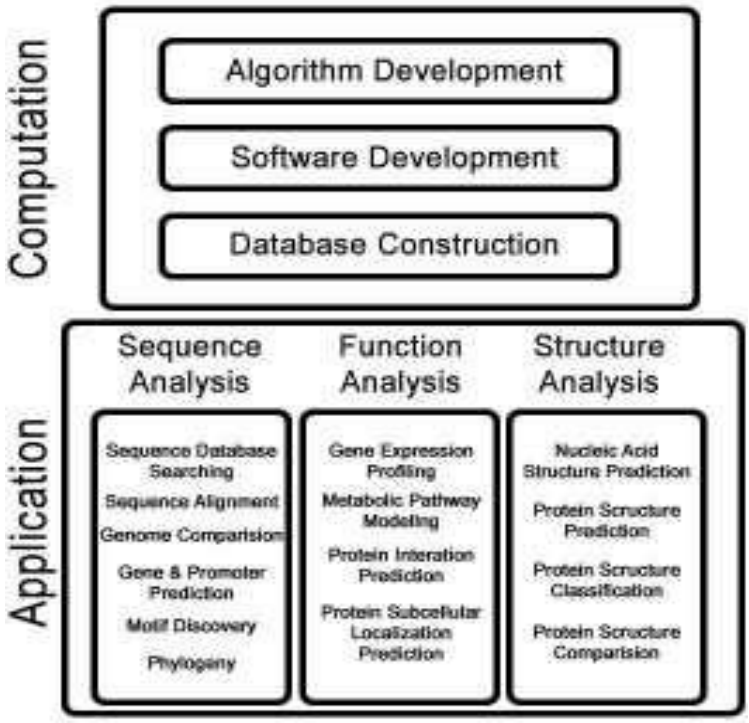


Figure 2

Sequence Analysis: All the applications that analyzes various types of sequence information and can compare between similar types of information is grouped under Sequence Analysis.

Function Analysis: These applications analyze the function engraved within the sequences and helps predict the functional interaction between various proteins or genes. Also expressional analysis of various genes is a prime topic for research these days.

Structure Analysis: When it comes to the realm of RNA and Proteins, its structure plays a vital role in the interaction with any other thing. This gave birth to a whole new branch termed

Structural Bioinformatics with is devoted to predict the structure and possible roles of these structures of Proteins or RNA

Sequence Analysis:

The application of sequence analysis determines those genes which encode regulatory sequences or peptides by using the information of sequencing. For sequence analysis, there are many powerful tools and computers which perform the duty of analyzing the genome of various organisms. These computers and tools also see the DNA mutations in an organism and also detect and identify those sequences which are related. Shotgun sequence techniques are also used for sequence analysis of numerous fragments of DNA. Special software is used to see the overlapping of fragments and their assembly.

Prediction of Protein Structure:-

It is easy to determine the primary structure of proteins in the form of amino acids which are present on the DNA molecule but it is difficult to determine the secondary, tertiary or quaternary structures of proteins. For this purpose either the method of crystallography is used or tools of bioinformatics can also be used to determine the complex protein structures.

Genome Annotation:-

In genome annotation, genomes are marked to know the regulatory sequences and protein coding. It is a very important part of the human genome project as it determines the regulatory sequences.

Comparative Genomics:-

Comparative genomics is the branch of bioinformatics which determines the genomic structure and function relation between different biological species. For this purpose, intergenomic maps are constructed which enable the scientists to trace the processes of evolution that occur in genomes of different species. These maps contain the information about the point mutations as well as the information about the duplication of large chromosomal segments.

Health and Drug discovery:

The tools of bioinformatics are also helpful in drug discovery, diagnosis and disease

management. Complete sequencing of human genes has enabled the scientists to make medicines and drugs which can target more than 500 genes. Different computational tools and drug targets has made the drug delivery easy and specific because now only those cells can be targeted which are diseased or mutated. It is also easy to know the molecular basis of a disease.

Application of Bioinformatics in various Fields

Molecular medicine

The human genome will have profound effects on the fields of biomedical research and clinical medicine. Every disease has a genetic component. This may be inherited (as is the case with an estimated 3000-4000 hereditary disease including Cystic Fibrosis and Huntingtons disease) or a result of the body's response to an environmental stress which causes alterations in the genome (eg. cancers, heart disease, diabetes.). The completion of the human genome means that we can search for the genes directly associated with different diseases and begin to understand the molecular basis of these diseases more clearly. This new knowledge of the molecular mechanisms of disease will enable better treatments, cures and even preventative tests to be developed.

Personalised medicine

Clinical medicine will become more personalised with the development of the field of pharmacogenomics. This is the study of how an individual's genetic inheritence affects the body's response to drugs. At present, some drugs fail to make it to the market because a small percentage of the clinical patient population show adverse affects to a drug due to sequence variants in their DNA. As a result, potentially life saving drugs never make it to the marketplace. Today, doctors have to use trial and error to find the best drug to treat a particular patient as those with the same clinical symptoms can show a wide range of responses to the same treatment. In the future, doctors will be able to analyse a patient's genetic profile and prescribe the best available drug therapy and dosage from the beginning.

Preventative medicine

With the specific details of the genetic mechanisms of diseases being unravelled, the development of diagnostic tests to measure a persons susceptibility to different diseases may become a distinct reality. Preventative actions such as change of lifestyle or having treatment

at the earliest possible stages when they are more likely to be successful, could result in huge advances in our struggle to conquer disease.

Gene therapy

In the not too distant future, the potential for using genes themselves to treat disease may become a reality. Gene therapy is the approach used to treat, cure or even prevent disease by changing the expression of a persons genes. Currently, this field is in its infantile stage with clinical trials for many different types of cancer and other diseases ongoing.

Drug development

At present all drugs on the market target only about 500 proteins. With an improved understanding of disease mechanisms and using computational tools to identify and validate new drug targets, more specific medicines that act on the cause, not merely the symptoms, of the disease can be developed. These highly specific drugs promise to have fewer side effects than many of today's medicines.

Microbial genome applications

Microorganisms are ubiquitous, that is they are found everywhere. They have been found surviving and thriving in extremes of heat, cold, radiation, salt, acidity and pressure. They are present in the environment, our bodies, the air, food and water. Traditionally, use has been made of a variety of microbial properties in the baking, brewing and food industries. The arrival of the complete genome sequences and their potential to provide a greater insight into the microbial world and its capacities could have broad and far reaching implications for environment, health, energy and industrial applications. For these reasons, in 1994, the US Department of Energy (DOE) initiated the MGP (Microbial Genome Project) to sequence genomes of bacteria useful in energy production, environmental cleanup, industrial processing and toxic waste reduction. By studying the genetic material of these organisms, scientists can begin to understand these microbes at a very fundamental level and isolate the genes that give them their unique abilities to survive under extreme conditions.

Waste cleanup

Deinococcus radiodurans is known as the world's toughest bacteria and it is the most radiation resistant organism known. Scientists are interested in this organism because of its potential

usefulness in cleaning up waste sites that contain radiation and toxic chemicals.

Climate change Studies

Increasing levels of carbon dioxide emission, mainly through the expanding use of fossil fuels for energy, are thought to contribute to global climate change. Recently, the DOE (Department of Energy, USA) launched a program to decrease atmospheric carbon dioxide levels. One method of doing so is to study the genomes of microbes that use carbon dioxide as their sole carbon source.

Alternative energy sources

Scientists are studying the genome of the microbe Chlorobium tepidum which has an unusual capacity for generating energy from light

Biotechnology

The archaeon Archaeoglobus fulgidus and the bacterium Thermotoga maritima have potential for practical applications in industry and government-funded environmental remediation. These microorganisms thrive in water temperatures above the boiling point and therefore may provide the DOE, the Department of Defence, and private companies with heat-stable enzymes suitable for use in industrial processes Other industrially useful microbes include, Corynebacterium glutamicum which is of high industrial interest as a research object because it is used by the chemical industry for the biotechnological production of the amino acid lysine. The substance is employed as a source of protein in animal nutrition. Lysine is one of the essential amino acids in animal nutrition. Biotechnologically produced lysine is added to feed concentrates as a source of protein, and is an alternative to soybeans or meat and bonemeal. Xanthomonas campestris pv. is grown commercially to produce the exopolysaccharide xanthan gum, which is used as a viscosifying and stabilising agent in many industries. Lactococcus lactis is one of the most important micro-organisms involved in the dairy industry, it is a non-pathogenic rod-shaped bacterium that is critical for manufacturing dairy products like buttermilk, yogurt and cheese. This bacterium, Lactococcus lactis ssp., is also used to prepare pickled vegetables, beer, wine, some breads and sausages and other fermented foods. Researchers anticipate that understanding the physiology and genetic make- up of this bacterium will prove invaluable for food manufacturers as well as the pharmaceutical industry,

which is exploring the capacity of L. lactis to serve as a vehicle for delivering drugs.

Antibiotic resistance

Scientists have been examining the genome of Enterococcus faecalis-a leading cause of bacterial infection among hospital patients. They have discovered a virulence region made up of a number of antibiotic-resistant genes that may contribute to the bacterium's transformation from harmless gut bacteria to a menacing invader. The discovery of the region, known as a pathogenicity island, could provide useful markers for detecting pathogenic strains and help to establish controls to prevent the spread of infection in wards.

Forensic analysis of microbes

Scientists used their genomic tools to help distinguish between the strain of Bacillus anthryacis that was used in the summer of 2001 terrorist attack in Florida with that of closely related anthrax strains.

The reality of bioweapon creation

Scientists have recently built the virus poliomyelitis using entirely artificial means. They did this using genomic data available on the Internet and materials from a mail-order chemical supply. The research was financed by the US Department of Defence as part of a biowarfare response program to prove to the world the reality of bioweapons. The researchers also hope their work will discourage officials from ever relaxing programs of immunisation. This project has been met with very mixed feeelings

Evolutionary studies

The sequencing of genomes from all three domains of life, eukaryota, bacteria and archaea means that evolutionary studies can be performed in a quest to determine the tree of life and the last universal common ancestor.

Crop improvement

Comparative genetics of the plant genomes has shown that the organisation of their genes has remained more conserved over evolutionary time than was previously believed. These findings

suggest that information obtained from the model crop systems can be used to suggest improvements to other food crops. At present the complete genomes of Arabidopsis thaliana (water cress) and Oryza sativa (rice) are available.

Insect resistance

Genes from Bacillus thuringiensis that can control a number of serious pests have been successfully transferred to cotton, maize and potatoes. This new ability of the plants to resist insect attack means that the amount of insecticides being used can be reduced and hence the nutritional quality of the crops is increased.

Improve nutritional quality

Scientists have recently succeeded in transferring genes into rice to increase levels of Vitamin A, iron and other micronutrients. This work could have a profound impact in reducing occurrences of blindness and anaemia caused by deficiencies in Vitamin A and iron respectively. Scientists have inserted a gene from yeast into the tomato, and the result is a plant whose fruit stays longer on the vine and has an extended shelf life.

Development of Drought resistance varieties

Progress has been made in developing cereal varieties that have a greater tolerance for soil alkalinity, free aluminium and iron toxicities. These varieties will allow agriculture to succeed in poorer soil areas, thus adding more land to the global production base. Research is also in progress to produce crop varieties capable of tolerating reduced water conditions.

Veterinary Science

Sequencing projects of many farm animals including cows, pigs and sheep are now well under way in the hope that a better understanding of the biology of these organisms will have huge impacts for improving the production and health of livestock and ultimately have benefits for human nutrition.

Comparative Studies

Analysing and comparing the genetic material of different species is an important method for studying the functions of genes, the mechanisms of inherited diseases and species evolution.

Bioinformatics tools can be used to make comparisons between the numbers, locations and biochemical functions of genes in different organisms.
Organisms that are suitable for use in experimental research are termed model organisms. They have a number of properties that make them ideal for research purposes including short life spans, rapid reproduction, being easy to handle, inexpensive and they can be manipulated at the genetic level.

An example of a human model organism is the mouse. Mouse and human are very closely related (>98%) and for the most part we see a one to one correspondence between genes in the two species. Manipulation of the mouse at the molecular level and genome comparisons between the two species can and is revealing detailed information on the functions of human genes, the evolutionary relationship between the two species and the molecular mechanisms of many human diseases.

Table 1

Data source	Data size	Bioinformatics topics
Raw DNA sequence	11.5 million sequences (12.5 billion bases)	Separating coding and non-coding regions Identification of introns and exons Gene product prediction Forensic analysis
Protein sequence	400,000 sequences (~300 amino acids each)	Sequence comparison algorithms Multiple sequence alignments algorithms Identification of conserved sequence motifs
Macromolecular structure	15,000 structures (~1,000 atomic coordinates each)	Secondary, tertiary structure prediction 3D structural alignment algorithms Protein geometry measurements Surface and volume shape calculations Intermolecular interactions Molecular simulations (force-field calculations, molecular movements, docking predictions)
Genomes	300 complete genomes (1.6 million – 3 billion bases each)	Characterisation of repeats Structural assignments to genes Phylogenetic analysis Genomic-scale censuses (characterisation of protein content, metabolic pathways) Linkage analysis relating specific genes to diseases
Gene expression	largest: ~20 time point measurements for ~6,000 genes in yeast	Correlating expression patterns Mapping expression data to sequence, structural and biochemical data
Other data		
Literature	11 million citations	Digital libraries for automated bibliographical searches Knowledge databases of data from literature
Metabolic pathways		Pathway simulations

Definitions of Fields Related to Bioinformatics

Bioinformatics has various applications in research in medicine, biotechnology, agriculture etc.

Following research fields has integral component of Bioinformatics

1. **Computational Biology:** The development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems.
2. **Genomics:** Genomics is any attempt to analyze or compare the entire genetic complement of a species or species (plural). It is, of course possible to compare genomes by comparing more-or-less representative subsets of genes within genomes.
3. **Proteomics:** Proteomics is the study of proteins - their location, structure and function. It is the identification, characterization and quantification of all proteins involved in a particular pathway, organelle, cell, tissue, organ or organism that can be studied in concert to provide accurate and comprehensive data about that system. Proteomics is the study of the function of all expressed proteins. The study of the proteome, called proteomics, now evokes not only all the proteins in any given cell, but also the set of all protein isoforms and modifications, the interactions between them, the structural description of proteins and their higher-order complexes, and for that matter almost everything 'post-genomic'."
4. **Pharmacogenomics:** Pharmacogenomics is the application of genomic approaches and technologies to the identification of drug targets. In Short, pharmacogenomics is using genetic information to predict whether a drug will help make a patient well or sick. It Studies how genes influence the response of humans to drugs, from the population to the molecular level.
5. **Pharmacogenetics:** Pharmacogenetics is the study of how the actions of and reactions to drugs vary with the patient's genes. All individuals respond differently to drug treatments; some positively, others with little obvious change in their conditions and yet others with side effects or allergic reactions. Much of this variation is known to have a genetic basis. Pharmacogenetics is a subset of pharmacogenomics which uses genomic/bioinformatic methods to identify genomic correlates, for example SNPs (Single Nucleotide Polymorphisms), characteristic of particular patient response profiles and use those markers to inform the administration and development of therapies. Strikingly such approaches have been used to "resurrect" drugs thought previously to be ineffective, but subsequently found to work with in subset of patients

or in optimizing the doses of chemotherapy for particular patients.

6. **Cheminformatics:**

   Chemical informatics: 'Computer-assisted storage, retrieval and analysis of chemical information, from data to chemical knowledge.' This definition is distinct from Chemoinformatics which focus on drug design. ***chemometrics:*** The application of statistics to the analysis of chemical data (from organic, analytical or medicinal chemistry) and design of chemical experiments and simulations. ***computational chemistry:*** A discipline using mathematical methods for the calculation of molecular properties or for the simulation of molecular behavior. It also includes, e.g., synthesis planning, database searching, combinatorial library manipulation

7. **Structural genomics or structural bioinformatics** refers to the analysis of macromolecular structure particularly proteins, using computational tools and theoretical frameworks. One of the goals of structural genomics is the extension of idea of genomics, to obtain accurate three-dimensional structural models for all known protein families, protein domains or protein folds Structural alignment is a tool of structural genomics.
8. **Comparative genomics:** The study of human genetics by comparisons with model organisms such as mice, the fruit fly, and the bacterium E. coli.
9. **Biophysics:** The British Biophysical Society defines biophysics as: "an interdisciplinary field which applies techniques from the physical sciences to understanding biological structure and function".
10. **Biomedical informatics / Medical informatics:** "Biomedical Informatics is an emerging discipline that has been defined as the study, invention, and implementation of structures and algorithms to improve communication, understanding and management of medical information."
11. **Mathematical Biology:** Mathematical biology also tackles biological problems, but the methods it uses to tackle them need not be numerical and need not be implemented in software or hardware. It includes things of theoretical interest which are not necessarily algorithmic, not necessarily molecular in nature, and are not necessarily useful in analyzing collected data.
12. **Computational chemistry:** Computational chemistry is the branch of theoretical chemistry whose major goals are to create efficient computer programs that calculate the properties of molecules (such as total energy, dipole moment, vibrational

frequencies) and to apply these programs to concrete chemical objects. It is also sometimes used to cover the areas of overlap between computer science and chemistry.

13. **Functional genomics:** Functional genomics is a field of molecular biology that is attempting to make use of the vast wealth of data produced by genome sequencing projects to describe genome function. Functional genomics uses high-throughput techniques like DNA microarrays, proteomics, metabolomics and mutation analysis to describe the function and interactions of genes.
14. **Pharmacoinformatics:** Pharmacoinformatics concentrates on the aspects of bioinformatics dealing with drug discovery
15. **In silico ADME-Tox Prediction:** Drug discovery is a complex and risky treasure hunt to find the most efficacious molecule which do not have toxic effects but at the same time have desired pharmacokinetic profile. The hunt starts when the researchers look for the binding affinity of the molecule to its target. Huge amount of research requires to be done to come out with a molecule which has the reliable binding profile. Once the molecules have been identified, as per the traditional methodologies, the molecule is further subjected to optimization with the aim of improving efficacy. The molecules which show better binding is then evaluated for its toxicity and pharmacokinetic profiles. It is at this stage that most of the candidates fail in the race to become a successful drug.
16. **Agroinformatics / Agricultural informatics:** Agroinformatics concentrates on the aspects of bioinformatics dealing with plant genomes.

INTERNET

The **Internet** is a global system of interconnected computer networks that use the standard Internet protocol suite (TCP/ IP) to serve billions of users worldwide. It is a *network of networks* that consists of millions of private, public, academic, business, and government networks, of local to global scope, that are linked by a broad array of electronic, wireless and optical networking technologies. The Internet carries a vast range of information resources and services, such as the inter- linked hypertext documents of the World Wide Web (WWW) and the infrastructure to support electronic mail.

Uses of Internet

Internet has been the most useful technology of the modern times which helps us not only in our daily lives, but also our personal and professional lives developments. The internet helps us achieve this in several different ways.

For the students and educational purposes the internet is widely used to gather information so as to do the research or add to the knowledge of various subjects. Even the business professionals and the professionals like doctors, access the internet to filter the necessary information for their use. The internet is therefore the largest encyclopedia for everyone, in all age categories. The internet has served to be more useful in maintaining contacts with friends and relatives who live abroad permanently.

Advantages of Internet:

E-mail: Email is now an essential communication tools in business. With e-mail you can send and receive instant electronic messages, which works like writing letters. Your messages are delivered instantly to people anywhere in the world, unlike traditional mail that takes a lot of time. Email is free, fast and very cheap when compared to telephone, fax and postal services.

24 hours a day - 7 days a week: Internet is available, 24x7 days for usage.

Information: Information is probably the biggest advantage internet is offering. There is a huge amount of information available on the internet for just about every subject, ranging from government law and services, trade fairs and conferences, market information, new ideas and technical support. You can almost find any type of data on almost any kind of subject that you are looking for by using search engines like google, yahoo, msn, etc.

Online Chat: You can access many 'chat rooms' on the web that can be used to meet new people, make new friends, as well as to stay in touch with old friends. You can chat in MSN and yahoo websites.

Services: Many services are provided on the internet like net banking, job searching, purchasing tickets, hotel reservations, guidance services on array of topics engulfing every aspect of life.

Communities: Communities of all types have sprung up on the internet. Its a great way to meet up with people of similar interest and discuss common issues.

E-commerce: Along with getting information on the Internet, you can also shop online. There are many online stores and sites that can be used to look for products as well as buy them using your credit card. You do not need to leave your house and can do all your shopping from the convenience of your home. It has got a real amazing and wide range of products from household needs, electronics to entertainment.

Entertainment: Internet provides facility to access wide range of Audio/ Video songs, plays films. Many of which can be downloaded. One such popular website is YouTube.

Software Downloads: You can freely download innumerable, softwares like utilities, games, music, videos, movies, etc from the Internet.

Limitations of Internet

Theft of Personal information: Electronic messages sent over the Internet can be easily snooped and tracked, revealing who is talking to whom and what they are talking about. If you use the Internet, your personal information such as your name, address, credit card, bank details and other information can be accessed by unauthorized persons. If you use a credit card or internet banking for online shopping, then your details can also be 'stolen'.

Negative effects on family communication: It is generally observed that due to more time spent on Internet, there is a decrease in communication and feeling of togetherness among the family members.

Internet addiction: There is some controversy over whether it is possible to actually be addicted to the Internet or not. Some researchers, claim that it is simply people trying to escape their problems in an online world.

Children using the Internet has become a big concern. Most parents do not realize the dangers involved when their children log onto the Internet. When children talk to others online, they do not realize they could actually be talking to a harmful person. Moreover, pornography is also a very serious issue concerning the Internet, especially when it comes to young children. There are thousands of pornographic sites on the Internet that can be easily found and can be a detriment to letting children use the Internet.

Virus threat: Today, not only are humans getting viruses, but computers are also. Computers are mainly getting these viruses from the Internet. Virus is is a program which disrupts the

normal functioning of your computer systems. Computers attached to internet are more prone to virus attacks and they can end up into crashing your whole hard disk.

Spamming: It is often viewed as the act of sending unsolicited email. This multiple or vast emailing is often compared to mass junk mailings. It needlessly obstruct the entire system. Most spam is commercial advertising, often for dubious products, get-rich-quick schemes, or quasi-legal services. Spam costs the sender very little to send — most of the costs are paid for by the recipient or the carriers rather than by the sender

SERVICES OF INTERNET - E-mail, FTP, Telnet

Email, discussion groups, long-distance computing, and file transfers are some of the important services provided by the Internet. Email is the fastest means of communication. With email one can also send software and certain forms of compressed digital image as an attachment. News groups or discussion groups facilitate Internet user to join for various kinds of debate, discussion and news sharing. Long-distance computing was an original inspiration for development of ARPANET and does still provide a very useful service on Internet. Programmers can maintain accounts on distant, powerful computers and execute programs. File transfer service allows Internet users to access remote machines and retrieve programs, data or text.

E-Mail (Electronic Mail)

E-mail or Electronic mail is a paperless method of sending messages, notes or letters from one person to another or even many people at the same time via Internet. E-mail is very fast compared to the normal post. E-mail messages usually take only few seconds to arrive at their destination. One can send messages anytime of the day or night, and, it will get delivered immediately. You need not to wait for the post office to open and you don't have to get worried about holidays. It works 24 hours a day and seven days a week. What's more, the copy of the message you have sent will be available whenever you want to look at it even in the middle of the night. You have the privilege of sending something extra such as a file, graphics, images etc. along with your e-mail. The biggest advantage of using e- mail is that it is cheap, especially when sending messages to other states or countries and at the same time it can be delivered to a number of people around the world.

It allows you to compose note, get the address of the recipient and send it. Once the mail is received and read, it can be forwarded or replied. One can even store it for later use, or delete. In e-mail even the sender can request for delivery receipt and read receipt from the recipient.

Features of E-mail:

- One-to-one or one-to-many communications
- Instant communications
- Physical presence of recipient is not required
- Most inexpensive mail services, 24-hours a day and seven days a week
- Encourages informal communications

Components of an E-mail Address

As in the case of normal mail system, e-mail is also based upon the concept of a recipient address. The email address provides all of the information required to get a message to the recipient from any where in the world. Consider the e-mail ID.

john@hotmail.com

In the above example john is the username of the person who will be sending/ receiving the email. Hotmail is the mail server where the username john has been registered and com is the type of organization on the internet which is hosting the mail server.

FTP (File Transfer Protocol)

File Transfer Protocol, is an Internet utility software used to uploaded and download files. It gives access to directories or folders on remote computers and allows software, data and text files to be transferred between different kinds of computers. FTP works on the basis of same principle as that of Client/ Server. FTP "Client" is a program running on your computer that enables you to communicate with remote computers. The FTP client takes FTP command and sends these as requests for information from the remote computer known as FTP servers. To access remote FTP server it is required, but not necessary to have an account in the FTP server. When the FTP client gets connected, FTP server asks for the identification in terms of User Login name and password of the FTP client. If one does not have an account in the remote FTP server, still he can connect to the server using anonymous login.

Using anonymous login anyone can login in to a FTP server and can access public archives; anywhere in the world, without having an account. One can easily Login to the FTP site with the username anonymous and e-mail address as password.

Objectives of FTP:

- Provide flexibility and promote sharing of computer programs, files and data
- Transfer data reliably and more efficiently over network
- Encourage implicit or indirect use of remote computers using Internet
- Shield a user from variations in storage systems among hosts.

The basic steps in an FTP session

Start up your FTP client, by typing ftp on your system's command line/ 'C>' prompt (or, if you are in a Windows, double-click on the FTP icon).

Give the FTP client an address to connect. This is the FTP server address to which the FTP client will get connected

Identify yourself to the FTP remote site by giving the Login Name

Give the remote site a password

Remote site will verify the Login Name/ Password to allow the FTP client to access its files

Look directory for files in FTP server

Change Directories if required

Set the transfer mode (optional);

Get the file(s) you want, and

Quit.

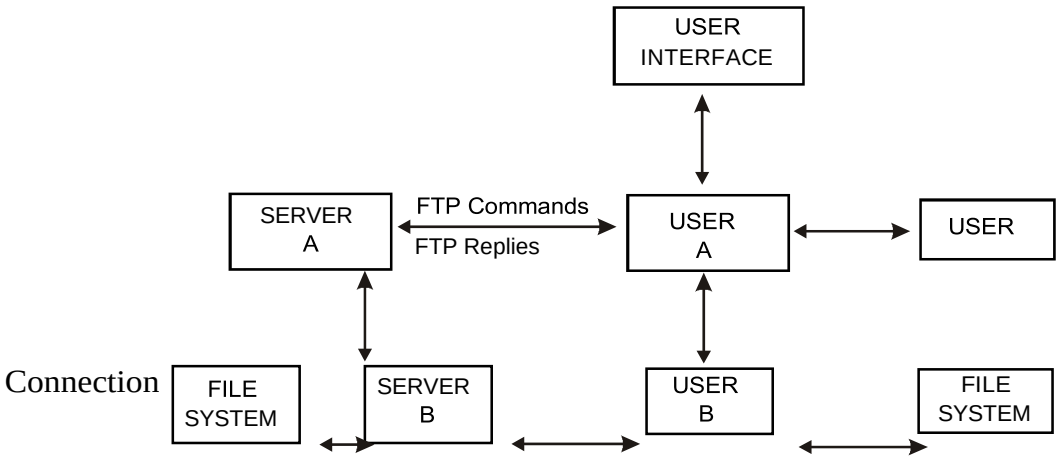


Figure 3

Telnet (Remote Computing)

Telnet or remote computing is telecommunication utility software, which uses available telecommunication facility and allows you to become a user on a remote computer. Once you gain access to remote computer, you can use it for the intended purpose. The TELNET works in a very step by step procedure. The commands typed on the client computer are sent to the local Internet Service Provider (ISP), and then from the ISP to the remote computer that you have gained access. Most of the ISP provides facility to TELENET into your own account from another city and check your e-mail while you are travelling or away on business.

The following steps are required for a TELNET session

- Start up the TELNET program
- Give the TELNET program an address to connect (some really nifty TELNET packages allow you to combine steps 1 and 2 into one simple step)
- Make a note of what the "escape character" is
- Log in to the remote computer,
- Set the "terminal emulation"

- Play around on the remote computer, and
- Quit.

TYPES OF INTERNET CONNECTIONS

There are five types of internet connections which are as follows:

(i) Dial up Connection

(ii) Leased Connection

(iii) DSL connection

(iv) Cable Modem Connection

(v) VSAT

Dial up connection

Dial-up refers to an Internet connection that is established using a modem. The modem connects the computer to standard phone lines, which serve as the data transfer medium. When a user initiates a dial-up connection, the modem dials a phone number of an Internet Service Provider (ISP) that is designated to receive dial-up calls. The ISP then establishes the connection, which usually takesabout ten seconds and is accompanied by several beepings and a buzzing sound. After the dial-up connection has been established, it is active until the user disconnects from the ISP. Typically, this is done by selecting the "Disconnect" option using the ISP's software or a modem utility program. However, if a dial-up connection is interrupted by an incoming phone call or someone picking up a phone in the house, the service may also be disconnected.

Advantages

Low Price

Secure connection – your IP address continually changes

Offered in rural areas – you need a phone line

Disadvantages

Slow speed.

Phone line is required.

Busy signals for friends and family members.

Leased Connection

Leased connection is a permanent telephone connection between two points set up by a telecommunications common carrier. Typically, leased lines are used by businesses to connect geographically distant offices. Unlike normal dial- up connections, a leased line is always active. The fee for the connection is a fixed monthly rate. The primary factors affecting the monthly fee are distance between end points and the speed of the circuit. Because the connection doesn't carry anybody else's communications, the carrier can assure a given level of quality.

For example, a T-1 channel is a type of leased line that provides a maximum transmission speed of 1.544 Mbps. You candivide the connection into different lines for data and voice communication or use the channel for one high speed data circuit. Dividing the connection is called multiplexing.

Increasingly, leased lines are being used by companies, and even individuals, for Internet access because they afford faster data transfer rates and are cost-effective if the Internet is used heavily.

Advantages

- Secure and private: dedicated exclusively to the customer
- Speed: symmetrical and direct
- Reliable: minimum down time
- Wide choice of speeds: bandwidth on demand, easily upgradeable
- Leased lines are suitable for in-house office web hosting

Disadvantages

- Leased lines can be expensive to install and rent.

- Not suitable for single or home workers
- Lead times can be as long as 65 working days
- Distance dependent to nearest POP
- Leased lines have traditionally been the more expensive access option. A Service Level Agreement (SLA) confirms an ISP's contractual requirement in ensuring the service is maintained. This is often lacking in cheaper alternatives.

DSL connection

Digital Subscriber Line (**DSL**) is a family of technologies that provides digital data transmission over the wires of a local telephone network. DSL originally stood for *digital subscriber loop*. In telecommunications marketing, the term DSL is widely understood to mean Asymmetric Digital Subscriber Line (ADSL), the most commonly installed DSL technology. DSL service is delivered simultaneously with wired telephone service on the same telephone line. This is possible because DSL uses higher frequency bands for data separated by filtering. On the customer premises, a DSL filter on each outlet removes the high frequency interference, to enable simultaneous use of the telephone and data.

The data bit rate of consumer DSL services typically ranges from 256 kbit/ s to 40 Mbit/ s in the direction to the customer (downstream), depending on DSL technology, line conditions, and service-level implementation. In ADSL, the data throughput in the upstream direction, (the direction to the service provider) is lower, hence the designation of *asymmetric* service. In Symmetric Digital Subscriber Line (SDSL) services, the downstream and upstream data rates are equal.

Advantages:

Security: Unlike cable modems, each subscriber can be configured so that it will not be on the same network. In some cable modem networks, other computers on the cable modem network are left visibly vulnerable and are easily susceptible to break in as well as data destruction.

Integration: DSL will easily interface with ATM and WAN technology.

High bandwidth

Cheap line charges from the phone company.

Good for "bursty" traffic patterns

Disadvantages

No current standardization: A person moving from one area to another might find that their DSL modem is just another paperweight. Customers may have to buy new equipment to simply change ISPs.

Expensive: Most customers are not willing to spend more than $20 to $25 per month for Internet access. Current installation costs, including the modem, can be as high as $750. Prices should come down within 1-3 years. As with all computer technology, being first usually means an emptier wallet.

Distance Dependence: The farther you live from the DSLAM (DSL Access Multiplexer), the lower the data rate. The longest run lengths are 18,000 feet, or a little over 3 miles.

Cable Modem Connection

A **cable modem** is a type of Network Bridge and modem that provides bi-directional data communication via radio frequency channels on a HFC and RFoG infrastructure. Cable modems are primarily used to deliver broadband Internet access in the form of cable Internet, taking advantage of the high bandwidth of a HFC and RFoG network. They are commonly deployed in Australia, Europe, Asia and Americas.

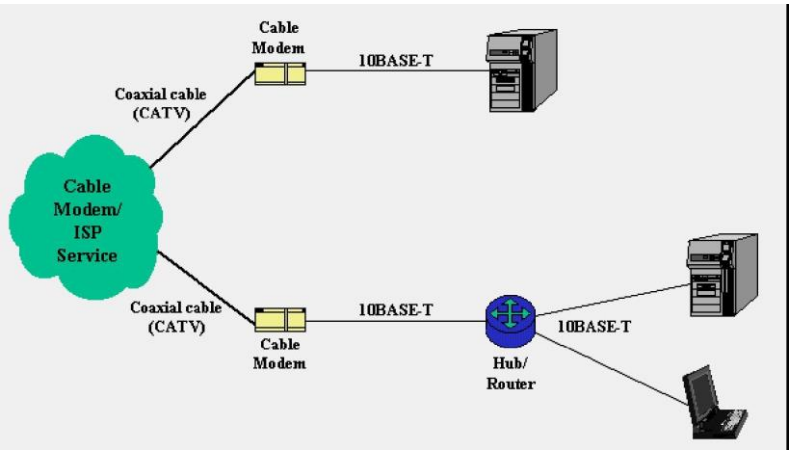


Figure 4

Figure shows the most common network connection topologies when using cable modems. The cable TV company runs a coaxial cable into the building to deliver their Internet service. Although fed from the same coax that provides cable TV service, most companies place a splitter outside of the building and runs two cables in, rather than using a splitter at the set-top box. The coax terminates at the cable modem.

The cable modem itself attaches to the SOHO computing equipment via its 10BASE-T port. In most circumstances, the cable modem attaches directly to a user's computer. If a LAN is present on the premises (something many cable companies frown upon), some sort of router can be connected to the cable modem.

Advantages

Always Connected: A cable modem connection is always connected to the Internet. This is advantageous because you do not have to wait for your computer to "log on" to the Internet; however, this also has the disadvantage of making your computer more vulnerable to hackers. Broadband: Cable modems transmit and receive data as digital packets, meaning they provide high-speed Internet access. This makes cable modem connections much faster than traditional dial-up connections.

Bandwidth: Cable modems have the potential to receive data from their cable provider at speeds greater than 30 megabits per second; unfortunately, this speed is rarely ever realized. Cable lines are shared by all of the cable modem users in a given area; thus, the connection speed varies depending upon the number of other people using the Internet and the amount of data they are receiving or transmitting.

File Transfer Capabilities: Downloads may be faster, but uploads are typically slower. Since the same lines are used to transmit data to and from the modem, priority is often given to data traveling in one direction.

Signal Integrity: Cable Internet can be transmitted long distances with little signal degradation. This means the quality of the Internet signal is not significantly decreased by the distance of the modem from the cable provider.

Routing: Cable routers allow multiple computers to be hooked up to one cable modem, allowing several devices to be directly connected through a single modem. Wireless routers can also be attached to your cable modem.

Rely on Existing Connections: Cable modems connect directly to preinstalled cable lines. This is advantageous because you do not need to have other services, such as telephone or Internet, in order to receive Internet through your cable modem. The disadvantage is that you cannot have cable internet in areas where there are no cable lines.

Disadvantages

Cable internet technology excels at maintaining signal strength over distance. Once it is delivered to a region, however, such as a neighborhood, it is split among that regions subscribers. While increased capacity has diminished the effect somewhat, it is still possible that users will see significantly lower speeds at peak times when more people are using the shared connection.

Bandwidth equals money, so cable's advantage in throughput comes with a price. Even in plans of similar speeds compared with DSL, customers spend more per Mb with cable than they do with DSL.

It's hard to imagine, but there are still pockets of the United States without adequate cable television service. There are far fewer such pockets without residential land-line service meaning cable internet is on balance less accessible in remote areas.

VSAT

Short for very small aperture terminal, an earthbound station used in satellite communications of data, voice and video signals, excluding broadcast television. A VSAT consists of two parts, a transceiver that is placed outdoors in direct line of sight to the satellite and a device that is placed indoors to interface the transceiver with the end user's communications device, such as a PC. The transceiver receives or sends a signal to a satellite transponder in the sky. The satellite sends and receives signals from a ground station computer that acts as a hub for the system. Each end user is interconnected with the hub station via the satellite, forming a star topology. The hub controls the entire operation of the network. For one end user to communicate with another, each transmission has to first go to the hub station that then retransmits it via the satellite to the other end user's VSAT.

Advantages

Satellite communication systems have some advantages that can be exploited for the provision

of connectivity. These are:

- Costs Insensitive to Distance
- Single Platform service delivery (one-stop-shop)
- Flexibility
- Upgradeable
- Low incremental costs per unit

Disadvantages

However like all systems there are disadvantages also. Some of these are

- High start-up costs (hubs and basic elements must be in place before the services can be provided)
- Higher than normal risk profiles
- Severe regulatory restrictions imposed by countries that prevent VSAT networks and solutions from reaching critical mass and therefore profitability
- Some service quality limitations such the high signal delays (latency)
- Natural availability limitsthat cannot be mitigated against
- Lack of skills required in the developing world to design, install and maintain satellite communication systems adequately

DOWNLOADING FILES

Downloading is the process of copying a file (such as a game or utility) from one computer to another across the internet. When you download a game from our web site, it means you are copying it from the author or publisher's web server to your own computer. This allows you to install and use the program on your own machine.

Here's how to download a file using Internet Explorer and Windows XP. (This example shows a download of the file "dweepsetup.exe" from Dexterity Games.) If you're using a different browser such as Netscape Navigator or a different version of Windows, your screen may look a little different, but the same basic steps should work.

Click on the download link for the program you want to download. Many sites offer multiple download links to the same program, and you only need to choose one of these links.

You may be asked if you want to save the file or run it from its current location. If you are asked this question, select "Save." If not, don't worry — some browsers will automatically choose "Save" for you.

You will then be asked to select the folder where you want to save the program or file, using a standard "Save As" dialog box. Pay attention to which folder you select before clicking the "Save" button. It may help you to create a folder like "C:\ Download" for all of your downloads, but you can use any folder you'd like.

The download will now begin. Your web browser will keep you updated on the progress of the download by showing a progress bar that fills up as you download. You will also be reminded where you're saving the file. The file will be saved as "C:\ Download\ dweepsetup.exe" in the picture below.

Note: You may also see a check box labeled "Close this dialog box when download completes." If you see this check box, it helps to **uncheck** this box. You don't have to, but if you do, it will be easier to find the file after you download it.

Depending on which file you're downloading and how fast your connection is, it may take anywhere from a few seconds to a few minutes to download. When your download is finished, if you left the "Close this dialog box when download completes" option unchecked, you'll see a dialog box as shown in fig. :

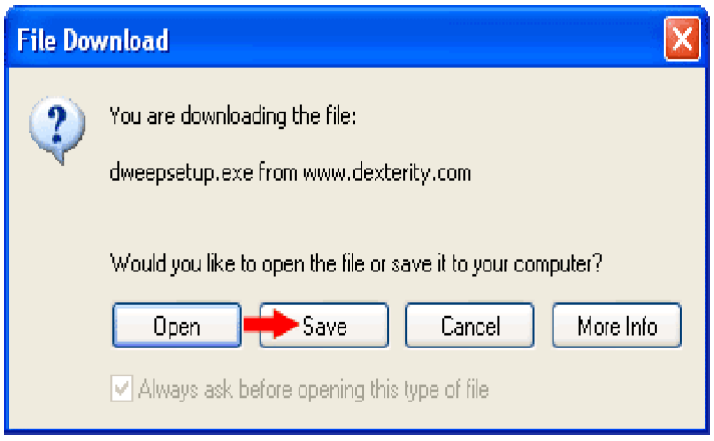


Figure 5 *a*

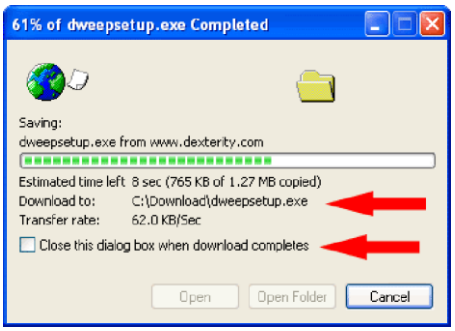


Figure 5b

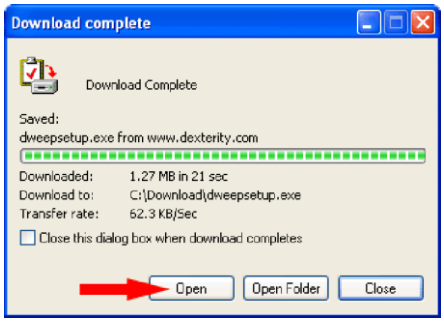


Figure 5c

Now click the "Open" button to run the file you just downloaded. If you don't see the "Download complete" dialog box, open the folder where you saved the file and double-click on the icon for the file there.

What happens next will depend on the type of file you downloaded. The files you'll download most often will end in one of two extensions. (An extension is the last few letters of the filename, after the period.) They are:

.EXE files: The file you downloaded is a program. Follow the on-screen instructions from there to install the program to your computer and to learn how to run the program after it's installed.

.ZIP files: ZIP is a common file format used to compress and combine files to make them download more quickly. Some versions of Windows (XP and sometimes ME) can read ZIP fi les without extra software. Otherwise, you will need an unzipping program to read these ZIP files. Common unzipping programs are WinZip, PKZIP, and Bit Zipper, but there are also many others. Many unzipping programs are shareware, which means you will need to purchase them if you use them beyond their specified trial period.

World Wide Web

What is the Internet? What is the World Wide Web? How are they related?

The Internet is an international network (a collection of connected, in this case, computers) – networked for the purpose of communication of information. The Internet offers many software services for this purpose, including:

- World Wide Web
- E-mail
- Instant messaging, chat
- Telnet (a service that lets a user login to a remote computer that the user has login privileges for)
- FTP (File Transfer Protocol) – a service that lets one use the Internet to copy files from one computer to another

The Web was originally designed for the purpose of displaying "public domain" data to anyone who could view it. Although this is probably the most popular use of the Web today, other uses of the Web include:

- Research, using tools such as "search engines" to find desired information.
- A variety of databases are available on the Web (this is another "research" tool). One example of such a database: a library's holdings.
- Shopping – most sizable commercial organizations have Web sites with forms you can fill out to specify goods or services you wish to purchase. Typically, you must include your credit card information in this form. Typically, your credit card information is safe – the system is typically automated so no human can see (and steal) your credit card number.
- We can generalize the above: Web forms can be filled out and submitted to apply for admission to a university, to give a donation to a charity, to apply for a job, to become a member of an organization, do banking chores, pay bills, etc.
- Listen to music or radio-like broadcasts, view videos or tv-like broadcasts.
- Some use the Web to access their e-mail or bulletin board services such as Blackboard.
- Most "browsers" today are somewhat like operating systems, in that they can enable a variety of application programs. For example, a Word, Excel, PowerPoint document can be placed on the Web and viewed in its "native" application.

Some terminology you should know:

- **Browser:** A program used to view Web documents. Popular browsers include Microsoft Internet Explorer (IE), Netscape, Opera; an old text-only browser called *Lynx* is still around on some systems; etc. The browsers of Internet Service Providers (ISPs) like AOL, Adelphia, Juno, etc., are generally one of the above, with the ISP's logo displayed. Most browsers work alike, today. There may be minor (for example, what IE calls "Favorites," Netscape calls "Bookmarks").

- **A Web document is called a "page."** A collection of related pages is a "site." A Web site typically has a "home page" designed to be the first, introductory, page a user of the site views.
- A Web page typically has an "address" or URL (Universal Reference Locator). You can view a desired page by using any of several methods to inform your browser of the URL whose page you wish to view. The home page of a site typically has a URL of the form http://www.DomainName.suffix

  where the "DomainName" typically tells you something about the identity of the "host" or "owner" of the site, and the "suffix" typically tells either the type of organization of the owner or its country. Some common suffixes include:
  - ✓ edu – An educational institution, usually a college or university.
  - ✓ com – A commercial site – a company
  - ✓ gov – a government site
  - ✓ org – an organization that's non-profit
  - ✓ net – an alternative to "com" for network service providers

  Also, the Internet originally was almost entirely centered in the US. As it spread to other countries, it became common for sites outside the US to use a suffix that's a 2-letter country abbreviation: "ca" (without quotation marks) for Canada; "it" for Italy; "mx" for Mexico; etc.

  A page that isn't a home page will typically have an address that starts with its site's home page address, and has appended further text to describe the page. For example, the Niagara University home page is at http://www.niagara.edu/ and the Niagara University Academics page is at http://www.niagara.edu/academic.htm.

Navigating:

- One way to reach a desired page is to enter its URL in the "Address" textbox.
- You can click on a *link* (usually underlined text, or a graphic may also serve as a link; notice that the mouse cursor changes its symbol, typically to a hand, when hovering over a Web link) to get to the page addressed by the link.
- The Back button may be used to retrace your steps, revisiting pages recently visited.
- You can click the Forward button to retrace your steps through pages recently Backed out of.

- Notice the drop-down button at the right side of the Address textbox. This reveals a menu of URLs recently visited by users of the browser on the current computer. You may click one of these URLs to revisit its page.
- Favorites (what Netscape calls "Bookmarks") are URLs saved for the purpose of making revisits easy. If you click a Favorite, you can easily revisit the corresponding page.

How do we find information on the Web? Caution: Don't believe everything you see on the Web. Many Web sites have content made up of hate literature, political propaganda, unfounded opinions, and other content of dubious reliability. Therefore, you should try to use good judgment about the sites you use for research.

Strategies for finding information on the Web include:

- Often, you can make an intelligent guess at the URL of a desired site. For example, you might guess the UB Web site is http://www.ub.edu (turned out to be the University of Barcelona) or http://www.buffalo.edu (was correct); similarly, if you're interested in the IRS Web site, you might try http://www.irs.gov – and it works. Similarly, you might try, for Enron, we might try http://www.enron.com – and this redirected us to the page http://www.enron.com/corp/.
- "Search engines" are Web services provided on a number of Web sites, allowing you to enter a keyword or phrase describing the topic you want information for. You may then click a button to activate the search. A list of links typically appears, and you may explore these links to find (you hope) the information you want. Note: if you use a phrase of multiple words, and don't place that phrase in quotation marks, you may get links by virtue of matching all the words separately – *e.g.*, "Diane" and "Pilarski" separately appeared in a document that matched the phrase "Diane Pilarski" without quotation marks; but the same link did not appear when we searched for "Diane Pilarski" with quotation marks. Also, you may find if the phrase you enter is someone's name, that many people have the same name.
- Another strategy: Some Web sites (including some that offer search engines) have "Web directories" or "indices" – classifications of Web pages. A good example: The Yahoo! site at http://www.yahoo.com has such a Web directory. You can work your way through the directory, often, to find desired information.

WEB BROWSER

A web browser (commonly referred to as a browser) is a software application for accessing information on the World Wide Web. Each individual web page, image, and video is identified by a distinct Uniform Resource Locator (URL), enabling browsers to retrieve these resources from a web server and display them on a user's device.

A web browser is not the same thing as a search engine, though the two are often confused. For a user, a search engine is just a website, such as google.com, that stores searchable data about other websites. But to connect to a website's server and display its web pages, a user must have a web browser installed on their device.

As of March 2019, more than 4.3 billion people use a browser, which is about 55% of the world’s population.

The most popular browsers are Chrome, Firefox, Safari, Internet Explorer, and Edge.

History

The first web browser, called WorldWideWeb, was created in 1990 by Sir Tim Berners-Lee. He then recruited Nicola Pellow to write the Line Mode Browser, which displayed web pages on dumb terminals; it was released in 1991.

Nicola Pellow and Tim Berners-Lee in their office at CERN.

Marc Andreessen, lead developer of Mosaic and Navigator

1993 was a landmark year with the release of Mosaic, credited as "the world's first popular browser". Its innovative graphical interface made the World Wide Web system easy to use and thus more accessible to the average person. This, in turn, sparked the Internet boom of the 1990s when the Web grew at a very rapid rate. Marc Andreessen, the leader of the Mosaic team, soon started his own company, Netscape, which released the Mosaic-influenced Netscape Navigator in 1994. Navigator quickly became the most popular browser.

Microsoft debuted Internet Explorer in 1995, leading to a browser war with Netscape. Microsoft was able to gain a dominant position for two reasons: it bundled Internet Explorer

with its popular Microsoft Windows operating system and did so as freeware with no restrictions on usage. Eventually the market share of Internet Explorer peaked at over 95% in 2002.

WorldWideWeb was the first web browser.
In 1998, desperate to remain competitive, Netscape launched what would become the Mozilla Foundation to create a new browser using the open source software model. This work evolved into Firefox, first released by Mozilla in 2004. Firefox reached a 28% market share in 2011.

Apple released its Safari browser in 2003. It remains the dominant browser on Apple platforms, though it never became a factor elsewhere.

The last major entrant to the browser market was Google. Its Chrome browser, which debuted in 2008, has been a huge success. It steadily took market share from Internet Explorer and became the most popular browser in 2012. Chrome has remained dominant ever since.

In terms of technology, browsers have greatly expanded their HTML, CSS, JavaScript, and multimedia capabilities since the 1990s. One reason has been to enable more sophisticated websites, such as web applications. Another factor is the significant increase of broadband connectivity, which enables people to access data-intensive web content, such as YouTube streaming, that was not possible during the era of dial-up modems.

Function
The purpose of a web browser is to fetch information resources from the Web and display them on a user's device.

This process begins when the user inputs a URL, such as https://en.wikipedia.org/, into the browser. Virtually all URLs on the Web start with either http: or https: which means the browser will retrieve them with the Hypertext Transfer Protocol. In the case of https:, the communication between the browser and the web server is encrypted for the purposes of security and privacy. Another URL prefix is file: which is used to display local files already stored on the user's device.

Once a web page has been retrieved, the browser's rendering engine displays it on the user's

device. This includes image and video formats supported by the browser.

Web pages usually contain hyperlinks to other pages and resources. Each link contains a URL, and when it is clicked, the browser navigates to the new resource. Thus the process of bringing content to the user begins again.

Settings

Web browsers can typically be configured with a built-in menu. Depending on the browser, the menu may be named Settings, Options, or Preferences.

The menu has different types of settings. For example, users can change their home page and default search engine. They also can change default web page colors and fonts. Various network connectivity and privacy settings are also usually available.

Privacy

During the course of browsing, cookies received from various websites are stored by the browser. Some of them contain login credentials or site preferences. However, others are used for tracking user behavior over long periods of time, so browsers typically provide settings for removing cookies when exiting the browser.Finer-grained management of cookies requires a browser extension.

Features

The most popular browsers have a number of features in common. They allow users to set bookmarks and browse in a private mode. They also can be customized with extensions, and some of them provide a sync service.

Most browsers have these user interface features:

Allow the user to open multiple pages at the same time, either in different browser windows or in different tabs of the same window.

Back and forward buttons to go back to the previous page visited or forward to the next one.

A refresh or reload button to reload the current page.

A stop button to cancel loading the page. (In some browsers, the stop button is merged with the reload button.)

A home button to return to the user's home page.

An address bar to input the URL of a page and display it.
A search bar to input terms into a search engine. (In some browsers, the search bar is merged with the address bar.)
There are also niche browsers with distinct features. One example is text-only browsers that can benefit people with slow Internet connections or those with visual impairments.

Security
Web browsers are popular targets for hackers, who exploit security holes to steal information, destroy files, and other malicious activity. Browser vendors regularly patch these security holes, so users are strongly encouraged to keep their browser software updated. Other protection measures are antivirus software and avoiding known-malicious websites.

EMBnet

The European Molecular Biology network (EMBnet) is an international scientific network and interest group that aims to enhance bioinformatics services by bringing together bioinformatics expertises and capacities. On 2011 EMBnet has 37 nodes spread over 32 countries. The nodes include bioinformatics related university departments, research institutes and national service providers.

Operations
The main task of most EMBnet nodes is to provide their national scientific community with access to bioinformatics databanks, specialised software and sufficient computing resources and expertise. EMBnet is also working in the fields of bioinformatics training and software development. Examples of software created by EMBnet members are: EMBOSS, wEMBOSS, UTOPIA.
EMBnet represents a wide user group and works closely together with the database producers such as EMBL's European Bioinformatics Institute (EBI), the Swiss Institute of Bioinformatics (Swiss-Prot), the Munich Information Center for Protein Sequences (MIPS), in order to provide a uniform coverage of services throughout Europe. EMBnet is registered in the Netherlands as a public foundation (Stichting).
Since its creation in 1988, EMBnet has evolved from an informal network of individuals in

charge of maintaining biological databases into the only worldwide organization bringing bioinformatics professionals to work together to serve the expanding fields of genetics and molecular biology. Although composed predominantly of academic nodes, EMBnet gains an important added dimension from its industrial members. The success of EMBnet is attracting increasing numbers of organizations outside Europe to join.

EMBnet has a tried-and-tested infrastructure to organise training courses, give technical help and help its members effectively interact and respond to the rapidly changing needs of biological research in a way no single institute is able to do.

In 2005 the organization created additional types of node to allow more than one member per country. The new category denomination is "associated node".

Coordination and organization

EMBnet is governed by the Annual General Meetings (AGM), and is coordinated by an Executive Board (EB) that oversees the activities of three project committees:

Education and Training committee (E&T). Educational support includes a series of courses organised in the member countries and languages, the committee works as well on the continued development of on-line accessible education materials.

Publicity and Public Relations committee (P&PR). This committee is responsible for promoting any type of EMBnet activities, for the advertisement of products and services provided by the EMBnet community, as well as for proposing and developing new strategies aiming to enhance EMBnet's visibility, and to take care of public relationships with EMBnet communities and related networks/societies.

Technical Manager committee (TM). The TM PC provides assistance and practical help to the participating nodes and their users.

THE NATIONAL CENTER FOR BIOTECHNOLOGY INFORMATION (NCBI)

The National Center for Biotechnology Information (NCBI) is part of the United States National Library of Medicine (NLM), a branch of the National Institutes of Health (NIH). The NCBI is located in Bethesda, Maryland and was founded in 1988 through legislation sponsored by Senator Claude Pepper.

The NCBI houses a series of databases relevant to biotechnology and biomedicine and is an important resource for bioinformatics tools and services. Major databases include GenBank for DNA sequences and PubMed, a bibliographic database for the biomedical literature. Other databases include the NCBI Epigenomics database. All these databases are available online through the Entrez search engine. NCBI was directed by David Lipman, one of the original authors of the BLAST sequence alignment program and a widely respected figure in bioinformatics. He also led an intramural research program, including groups led by Stephen Altschul (another BLAST co-author), David Landsman, Eugene Koonin, John Wilbur, Teresa Przytycka, and Zhiyong Lu. David Lipman stood down from his post in May 2017.

GenBank

NCBI has had responsibility for making available the GenBank DNA sequence database since 1992.GenBank coordinates with individual laboratories and other sequence databases such as those of the European Molecular Biology Laboratory (EMBL) and the DNA Data Bank of Japan (DDBJ).

Since 1992, NCBI has grown to provide other databases in addition to GenBank. NCBI provides Gene, Online Mendelian Inheritance in Man, the Molecular Modeling Database (3D protein structures), dbSNP (a database of single-nucleotide polymorphisms), the Reference Sequence Collection, a map of the human genome, and a taxonomy browser, and coordinates with the National Cancer Institute to provide the Cancer Genome Anatomy Project. The NCBI assigns a unique identifier (taxonomy ID number) to each species of organism.

The NCBI has software tools that are available by WWW browsing or by FTP. For example, BLAST is a sequence similarity searching program. BLAST can do sequence comparisons against the GenBank DNA database in less than 15 seconds.

NCBI Bookshelf

The "NCBI Bookshelf is a collection of freely accessible, downloadable, on-line versions of selected biomedical books. The Bookshelf covers a wide range of topics including molecular biology, biochemistry, cell biology, genetics, microbiology, disease states from a molecular and cellular point of view, research methods, and virology. Some of the books are online versions of previously published books, while others, such as Coffee Break, are written and edited by NCBI staff. The Bookshelf is a complement to the Entrez PubMed repository of peer-

reviewed publication abstracts in that Bookshelf contents provide established perspectives on evolving areas of study and a context in which many disparate individual pieces of reported research can be organized.

Basic Local Alignment Search Tool (BLAST)

BLAST is an algorithm used for calculating sequence similarity between biological sequences such as nucleotide sequences of DNA and amino acid sequences of proteins. BLAST is a powerful tool for finding sequences similar to the query sequence within the same organism or in different organisms. It searches the query sequence on NCBI databases and servers and post the results back to the person's browser in chosen format. Input sequences to the BLAST are mostly in FASTA or Genbank format while output could be delivered in variety of formats such as HTML, XML formatting and plain text. HTML is the default output format for NCBI's web-page. Results for NCBI-BLAST are presented in graphical format with all the hits found, a table with sequence identifiers for the hits having scoring related data, along with the alignments for the sequence of interest and the hits received with analogous BLAST scores for these

Entrez

The Entrez Global Query Cross-Database Search System is used at NCBI for all the major databases such as Nucleotide and Protein Sequences, Protein Structures, PubMed, Taxonomy, Complete Genomes, OMIM, and several others. Entrez is both indexing and retrieval system having data from various sources for biomedical research. NCBI distributed the first version of Entrez in 1991, composed of nucleotide sequences from PDB and GenBank, protein sequences from SWISS-PROT, translated GenBank, PIR, PRF , PDB and associated abstracts and citations from PubMed. Entrez is specially designed to integrate the data from several different sources, databases and formats into a uniform information model and retrieval system which can efficiently retrieve that relevant references, sequences and structures.

Gene

Gene has been implemented at NCBI to characterize and organize the information about genes. It serves as a major node in the nexus of genomic map, expression, sequence, protein function, structure and homology data. A unique GeneID is assigned to each gene record that can be followed through revision cycles. Gene records for known or predicted genes are established here and are demarcated by map positions or nucleotide sequence. Gene has several advantages

over its predecessor, LocusLink, including, better integration with other databases in NCBI, broader taxonomic scope, and enhanced options for query and retrieval provided by Entrez system.

Protein

Protein database maintains the text record for individual protein sequences, derived from many different resources such as NCBI Reference Sequence (RefSeq) project, GenbBank, PDB and UniProtKB/SWISS-Prot. Protein records are present in different formats including FASTA and XML and are linked to other NCBI resources. Protein provides the relevant data to the users such as genes, DNA/RNA sequences, biological pathways, expression and variation data and literature. It also provides the pre-determined sets of similar and identical proteins for each sequence as computed by the BLAST. The Structure database of NCBI contains 3D coordinate sets for experimentally-determined structures in PDB that are imported by NCBI. The Conserved Domain database (CDD) of protein contains sequence profiles that characterize highly conserved domains within protein sequences. It also has records from external resources like SMART and Pfam. There is another database in protein known as Protein Clusters database which contains sets of proteins sequences that are clustered according to the maximum alignments between the individual sequences as calculated by BLAST.

Pubchem database

PubChem database of NCBI is a public resource for molecules and their activities against biological assays. PubChem is searchable and accessible by Entrez information retrieval system.

FILE TRANSFER PROTOCOL

The File Transfer Protocol (FTP) is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

FTP is built on a client-server model architecture using separate control and data connections between the client and the server. FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the

username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).

The first FTP client applications were command-line programs developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems. Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as HTML editors.

History of FTP servers

The original specification for the File Transfer Protocol was written by Abhay Bhushan and published as RFC 114 on 16 April 1971. Until 1980, FTP ran on NCP, the predecessor of TCP/IP. The protocol was later replaced by a TCP/IP version, RFC 765 (June 1980) and RFC 959 (October 1985), the current specification. Several proposed standards amend RFC 959, for example RFC 1579 (February 1994) enables Firewall-Friendly FTP (passive mode), RFC 2228 (June 1997) proposes security extensions, RFC 2428 (September 1998) adds support for IPv6 and defines a new type of passive mode.

Protocol overview

Communication and data transfer

Illustration of starting a passive connection using port 21

FTP may run in active or passive mode, which determines how the data connection is established. In both cases, the client creates a TCP control connection from a random, usually an unprivileged, port N to the FTP server command port 21.

In active mode, the client starts listening for incoming data connections from the server on port M. It sends the FTP command PORT M to inform the server on which port it is listening. The server then initiates a data channel to the client from its port 20, the FTP server data port.

In situations where the client is behind a firewall and unable to accept incoming TCP connections, passive mode may be used. In this mode, the client uses the control connection to send a PASV command to the server and then receives a server IP address and server port number from the server, which the client then uses to open a data connection from an arbitrary client port to the server IP address and server port number received

Both modes were updated in September 1998 to support IPv6. Further changes were introduced

to the passive mode at that time, updating it to extended passive mode.

The server responds over the control connection with three-digit status codes in ASCII with an optional text message. For example, "200" (or "200 OK") means that the last command was successful. The numbers represent the code for the response and the optional text represents a human-readable explanation or request (e.g. <Need account for storing file>) An ongoing transfer of file data over the data connection can be aborted using an interrupt message sent over the control connection.

While transferring data over the network, four data representations can be used:

ASCII mode: Used for text. Data is converted, if needed, from the sending host's character representation to "8-bit ASCII" before transmission, and (again, if necessary) to the receiving host's character representation. As a consequence, this mode is inappropriate for files that contain data other than plain text.
Image mode (commonly called Binary mode): The sending machine sends each file byte by byte, and the recipient stores the bytestream as it receives it. (Image mode support has been recommended for all implementations of FTP).
EBCDIC mode: Used for plain text between hosts using the EBCDIC character set.
Local mode: Allows two computers with identical setups to send data in a proprietary format without the need to convert it to ASCII.
For text files, different format control and record structure options are provided. These features were designed to facilitate files containing Telnet or ASA.

Data transfer can be done in any of three modes:
Stream mode: Data is sent as a continuous stream, relieving FTP from doing any processing. Rather, all processing is left up to TCP. No End-of-file indicator is needed, unless the data is divided into records.
Block mode: FTP breaks the data into several blocks (block header, byte count, and data field) and then passes it on to TCP.
Compressed mode: Data is compressed using a simple algorithm (usually run-length encoding). Some FTP software also implements a DEFLATE-based compressed mode, sometimes called "Mode Z" after the command that enables it. This mode was described in an Internet Draft, but not standardized

Login

FTP login uses normal username and password scheme for granting access. The username is sent to the server using the USER command, and the password is sent using the PASS command. This sequence is unencrypted "on the wire", so may be vulnerable to a network sniffing attack. If the information provided by the client is accepted by the server, the server will send a greeting to the client and the session will commence. If the server supports it, users may log in without providing login credentials, but the same server may authorize only limited access for such sessions.

Anonymous FTP

A host that provides an FTP service may provide anonymous FTP access. Users typically log into the service with an 'anonymous' (lower-case and case-sensitive in some FTP servers) account when prompted for user name. Although users are commonly asked to send their email address instead of a password, no verification is actually performed on the supplied data. Many FTP hosts whose purpose is to provide software updates will allow anonymous logins.

NAT and firewall traversal

FTP normally transfers data by having the server connect back to the client, after the PORT command is sent by the client. This is problematic for both NATs and firewalls, which do not allow connections from the Internet towards internal hosts. For NATs, an additional complication is that the representation of the IP addresses and port number in the PORT command refer to the internal host's IP address and port, rather than the public IP address and port of the NAT.

There are two approaches to solve this problem. One is that the FTP client and FTP server use the PASV command, which causes the data connection to be established from the FTP client to the server. This is widely used by modern FTP clients. Another approach is for the NAT to alter the values of the PORT command, using an application-level gateway for this purpose.

Differences from HTTP

HTTP essentially fixes the bugs in FTP that made it inconvenient to use for many small ephemeral transfers as are typical in web pages.

FTP has a stateful control connection which maintains a current working directory and other flags, and each transfer requires a secondary connection through which the data are transferred. In "passive" mode this secondary connection is from client to server, whereas in the default "active" mode this connection is from server to client. This apparent role reversal when in active mode, and random port numbers for all transfers, is why firewalls and NAT gateways have such a hard time with FTP. HTTP is stateless and multiplexes control and data over a single connection from client to server on well-known port numbers, which trivially passes through NAT gateways and is simple for firewalls to manage.

Setting up an FTP control connection is quite slow due to the round-trip delays of sending all of the required commands and awaiting responses, so it is customary to bring up a control connection and hold it open for multiple file transfers rather than drop and re-establish the session afresh each time. In contrast, HTTP originally dropped the connection after each transfer because doing so was so cheap. While HTTP has subsequently gained the ability to reuse the TCP connection for multiple transfers, the conceptual model is still of independent requests rather than a session.

When FTP is transferring over the data connection, the control connection is idle. If the transfer takes too long, the firewall or NAT may decide that the control connection is dead and stop tracking it, effectively breaking the connection and confusing the download. The single HTTP connection is only idle between requests and it is normal and expected for such connections to be dropped after a time-out.

Web browser support

Most common web browsers can retrieve files hosted on FTP servers, although they may not support protocol extensions such as FTPS.When an FTP—rather than an HTTP—URL is supplied, the accessible contents on the remote server are presented in a manner that is similar to that used for other web content. A full-featured FTP client can be run within Firefox in the form of an extension called FireFTP.

Syntax

FTP URL syntax is described in RFC 1738, taking the form: ftp://[user[:password]@]host[:port]/url-path (the bracketed parts are optional).

For example, the URL ftp://public.ftp-servers.example.com/mydirectory/myfile.txt represents the file myfile.txt from the directory mydirectory on the server public.ftp-servers.example.com as an FTP resource. The URL ftp://user001:secretpassword@private.ftp-servers.example.com/mydirectory/myfile.txt adds a specification of the username and password that must be used to access this resource.

More details on specifying a username and password may be found in the browsers' documentation (e.g., Firefox and Internet Explorer). By default, most web browsers use passive (PASV) mode, which more easily traverses end-user firewalls.

Some variation has existed in how different browsers treat path resolution in cases where there is a non-root home directory for a user.

Security

FTP was not designed to be a secure protocol, and has many security weaknesses. In May 1999, the authors of RFC 2577 listed a vulnerability to the following problems:

Brute force attack

FTP bounce attack

Packet capture

Port stealing (guessing the next open port and usurping a legitimate connection)

Spoofing attack

Username enumeration

DoS or DDoS

FTP does not encrypt its traffic; all transmissions are in clear text, and usernames, passwords, commands and data can be read by anyone able to perform packet capture (sniffing) on the network. This problem is common to many of the Internet Protocol specifications (such as SMTP, Telnet, POP and IMAP) that were designed prior to the creation of encryption mechanisms such as TLS or SSL.

Common solutions to this problem include:

Using the secure versions of the insecure protocols, e.g., FTPS instead of FTP and TelnetS instead of Telnet.

Using a different, more secure protocol that can handle the job, e.g. SSH File Transfer Protocol or Secure Copy Protocol.

Using a secure tunnel such as Secure Shell (SSH) or virtual private network (VPN).

FTP over SSH

FTP over SSH is the practice of tunneling a normal FTP session over a Secure Shell connection. Because FTP uses multiple TCP connections (unusual for a TCP/IP protocol that is still in use), it is particularly difficult to tunnel over SSH. With many SSH clients, attempting to set up a tunnel for the control channel (the initial client-to-server connection on port 21) will protect only that channel; when data is transferred, the FTP software at either end sets up new TCP connections (data channels) and thus have no confidentiality or integrity protection.

Otherwise, it is necessary for the SSH client software to have specific knowledge of the FTP protocol, to monitor and rewrite FTP control channel messages and autonomously open new packet forwardings for FTP data channels. Software packages that support this mode include:

Tectia ConnectSecure (Win/Linux/Unix) of SSH Communications Security's software suite

Derivatives

FTPS

Explicit FTPS is an extension to the FTP standard that allows clients to request FTP sessions to be encrypted. This is done by sending the "AUTH TLS" command. The server has the option of allowing or denying connections that do not request TLS. This protocol extension is defined in RFC 4217. Implicit FTPS is an outdated standard for FTP that required the use of a SSL or TLS connection. It was specified to use different ports than plain FTP.

SSH File Transfer Protocol

The SSH file transfer protocol (chronologically the second of the two protocols abbreviated SFTP) transfers files and has a similar command set for users, but uses the Secure Shell protocol (SSH) to transfer files. Unlike FTP, it encrypts both commands and data, preventing passwords and sensitive information from being transmitted openly over the network. It cannot interoperate with FTP software.

Trivial File Transfer Protocol

Trivial File Transfer Protocol (TFTP) is a simple, lock-step FTP that allows a client to get a file from or put a file onto a remote host. One of its primary uses is in the early stages of booting from a local area network, because TFTP is very simple to implement. TFTP lacks security and most of the advanced features offered by more robust file transfer protocols such as File

Transfer Protocol. TFTP was first standardized in 1981 and the current specification for the protocol can be found in RFC 1350.

Simple File Transfer Protocol

Simple File Transfer Protocol (the first protocol abbreviated SFTP), as defined by RFC 913, was proposed as an (unsecured) file transfer protocol with a level of complexity intermediate between TFTP and FTP. It was never widely accepted on the Internet, and is now assigned Historic status by the IETF. It runs through port 115, and often receives the initialism of SFTP. It has a command set of 11 commands and support three types of data transmission: ASCII, binary and continuous. For systems with a word size that is a multiple of 8 bits, the implementation of binary and continuous is the same. The protocol also supports login with user ID and password, hierarchical folders and file management (including rename, delete, upload, download, download with overwrite, and download with append).

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO AND CHEMICAL ENGINEERING

DEPARTMENT OF BIOTECHNOLOGY

Unit 2 – Introduction to Bioinformatics (Elective) – SBB1609

II BROWSERS AND SEARCH ENGINES

DATABASE BROWSER

Database Browser is a Google GSuite add-on that makes Google Sheet an awesome new way of browsing your database (RDBMS like MySQL, Oracle, MS SQL Server and noSQL services like mLab - MongoDB).

You can connect to the database, list the tables, query records of table into Google Sheet with an intuitive and vibrant GUI.

Database Browser is created for making things easier for developers, admin, testers, customers or any one who wants to query/edit database without any coding and display/ store results into Google Sheet.

This “Database Browser” add-on provides an easy GUI for establishing connection with databases, browsing tables and querying records out to Google Sheet.

Key Features are

Manage Connections - Create, test and save connections to databases on the Internet and cloud.

Manage Queries - Build, execute, save queries and query results into Google Sheet.

Edit - Edit the data in the database directly from Google Sheet and saving into database

This “Database Browser” add-on provides an easy GUI for establishing connection with databases, browsing tables and querying records out to Google Sheet.

User can prepare the query by selecting connection, tables, and fields of the table

User can prepare the query where clause by visually selecting fields and forming conditions

User can provide a name to the query

User can execute a query and display/ store results into Google Sheet

User can list, add, modify, delete queries

User can edit data directly from Google sheet

SEARCH ENGINES

A web search engine is a software system that is designed to search for information on the World Wide Web. It uses the keywords to search for documents that relate to these key words and then puts the result in order of relevance to the topic that was searched for.

HOW DOES A SEARCH ENGINE WORK

A search engine is a website, but generally speaking, a search engine wouldn't normally provide answers straight away. Search engines crawl through websites using computers to make an electronic copy of website. When we enter a search term and it brings up a number of pages from its database which it thinks are applicable to your search terms.

IMPORTANCE OF SEARCH ENGINE

A website is something you already know how to get on and where to go, which is typing the URL into the little space provided. A search engine helps you find an appropriate website for something you are looking for but don't know the URL but still need to find what you are looking for. After you type it into a search engine a bunch of URLs will pop up and you click on the 1 you think is most helpful.

Search engines are important because with over 8 billion web pages available, it would be impossible to search for the information that is specifically needed. This is why search engines are used to filter the information that is on the internet and transform it into results that each individual can easily access and use within the matter of seconds.

TYPES OF SEARCH ENGINES

Crawler based Search Engines

Directories Search Engines

Hybrid Search Engines

Meta Search Engines

CRAWLER BASED SEARCH ENGINES

This search engines use a "spider" or a "crawler" to search the internet.

The crawler digs through individual web pages, pulls out keywords and then adds the pages to the search engine's database.

Google and Yahoo are examples of crawler search engines.

Crawler-based search engines are good when search topic is specific

GOOGLE

Google was founded by Larry Page and Sergey Brin in 1998. The homepage of Google has a button labeled as “I’m feeling lucky”. When a user types in a search and clicks on the button, the user is directly taken to the first search result. Google has various “special features” which include weather, unit conversion, currency conversion, time, calculator, maps etc.

YAHOO

Yahoo was founded by David Filo and Jerry Yang Yahoo operates a portal that provides the latest news, entertainment, and sports information. The portal also gives users access to other Yahoo services like Yahoo Mail, Yahoo Maps, Yahoo Finance, Yahoo Groups and Yahoo Messenger. The first Yahoo company started in 1995

DIRECTORIES

Directories depend on human editors to create their listings or the database.

Yahoo directory, Open directory and Look smart are few examples.

Human-powered directories are good when you are interested in a general topic of search

HYBRID SEARCH ENGINES

Hybrid search engines are search engines that use both crawler based searches and directory searches to obtain their results.

Examples: yahoo.com and google.com

META SEARCH ENGINES

These transmit user-supplied keywords simultaneously to several individual search engines to actually carry out the search.

Search results returned from all the search engines can be integrated, duplicates can be eliminated and additional features such as clustering by subjects within the search results can be implemented by meta-search engines.

Examples: Dogpile, Metacrawler

DOGPILE: Dogpile is a search engine that fetches results from Google, Yahoo! and includes results from several other popular search engines. Dogpile began operation in November 1996. The site was created and developed by Aaron Flin and later sold to Go2net.

MS ACCESS

Microsoft Access is a Database Management System offered by Microsoft. It uses the Microsoft Jet Database Engine and comes as a part of the Microsoft Office suite of application.

Microsoft Access offers the functionality of a database and the programming capabilities to create easy to navigate screens (forms). It helps you analyze large amounts of information, and manage data efficiently.

Database File:

It is a file which stores the entire database. The database file is saved to your hard drive or other storage devices.

Datatypes:

Datatypes are the properties of each field. Every field has one datatype like text, number, date, etc.

Table

- A Table is an object which stores data in Row & Column format to store data.
- A Table is usually related to other tables in the database file.
- Each column must have Unique name
- We can also define Primary Key in a table.

Query

- Queries answer a question by selecting and sorting and filtering data based on search criteria.
- Queries show a selection of data based on criteria (limitations) you provide.
- Queries can pull from one or more related Tables and other Queries.
- Types of Query can be SELECT, INSERT, UPDATE, DELETE.

Form

- A form is a database object that you can use to create a user interface for a database application.
- Forms help you to display live data from the table. It mainly used to ease the process of data entry or editing.

Report

- A report is an object in desktop databases primarily used for formatting, calculating, printing, and summarizing selected data.
- You can even customize the report's look and feel.

Macros

Macros are mini computer programming constructs. They allow you to set up commands and processes in your forms, like, searching, moving to another record, or running a formula.

Modules:

Modules are procedures(functions) which you can write using Visual Basic for Applications (VBA).

Differences between Access and Excel

Microsoft Access and Excel are very similar yet very different. Here, are some important difference points between both of them:

Table 1

Access	Excel
Deals with text, numbers, files and all kinds of data	Microsoft Excel generally deals with numerical data
All the data is stored one time, in one place.	Lots of worksheets or documents are a store with similar, repeated data.
Helps you to build highly functional data entry forms and report templates.	Only the primary data entry screen is available.
Users will be able to enter the data more efficiently and accurately.	Data accuracy and speed is not much because of the format.

Advantages of MS Access

- Access offers a fully functional, relational database management system in minutes.
- Easy to import data from multiple sources into Access
- You can easily customize Access according to personal and company needs
- Microsoft Access online works well with many of the development languages that work on Windows OS
- It is robust and flexible, and it can perform any challenging office or industrial database tasks.
- MS-Access allows you to link to data in its existing location and use it for viewing, updating, querying, and reporting.
- Allows you to create tables, queries, forms, and reports, and connect with the help of Macros
- Macros in Access is a simple programming construct with which you can use to add functionality to your database.

- Microsoft Access online can perform heterogeneous joins between various data sets stored across different platforms

Disadvantages of MS Access

- Microsoft Access database is useful for small-to-medium business sectors. However, it is not useful for large-sized organizations
- Lacks robustness compared to dbms systems like MS SQL Server or Oracle
- All the information from your database is saved into one file. This can slow down reports, queries, and forms
- Technical limit is 255 concurrent users. However, the real-world limit is only 10 to 80 (depending on the type of application which you are using)
- It requires a lot more learning and training compares with other Microsoft programs

Summary

- Microsoft Access is a Database Management System offered by Microsoft.
- Allows you to create tables, queries, forms, and reports, and connect with the help of Macros
- MS-Access will enable you to link to data in its existing location and use it for viewing, updating, querying, and reporting.
- Access consists of **four main database objects**: Tables, Queries, Forms, and Reports.
- There are **two ways** to **create Database in SQL Access**:
    - Create Database from **Template**
    - Create a **Blank Database**
- There are **two ways** to **create Database in MS Access**
    - Create a Table from **Design View**
    - Create a Table from **Datasheet View**
- You can switch between the datasheet and the design view by just clicking the 'View' button in the top-left hand corner of the Access program.
- A form can be created using Form Wizard, Form, Multiple Item, Split Form
- Macro in MS Access database is a time-saving feature that allows you to add functionality or automate simple tasks.

- A report is an object in MS Access that is designed for formatting, calculating and printing selected data in an organized way.
- A Module is a collection of user-defined functions, declarations, statements, and procedures that are stored together as a unit.

MAKING QUERIES

The real power of a relational database lies in its ability to quickly **retrieve** and **analyze** your data by running a query. **Queries** allow you to **pull information** from one or more tables based on a set of search conditions you define.

What are queries?

Queries are a way of **searching** for and **compiling** data from one or more tables. Running a query is like asking a **detailed question** of your database. When you build a query in Access, you are **defining specific search conditions** to find exactly the data you want.

How are queries used?

Queries are far more powerful than the simple searches or filters you might use to find data within a table. This is because queries can draw their information from **multiple** tables. For example, while you could use a **search** in the customers table to find the name of one customer at your business or a **filter** on the orders table to view only orders placed within the past week, neither would let you view both customers and orders at once. However, you could easily run a **query** to find the name and phone number of every customer who's made a purchase within the past week. A well-designed query can give information you might not be able to find out just by examining the data in your tables.

When you run a query, the results are presented to you in a table, but when you design one you use a different view. This is called **Query Design view**, and it lets you see how your query is put together.

Click the buttons in the interactive below to learn how to navigate the **Query Design view**.

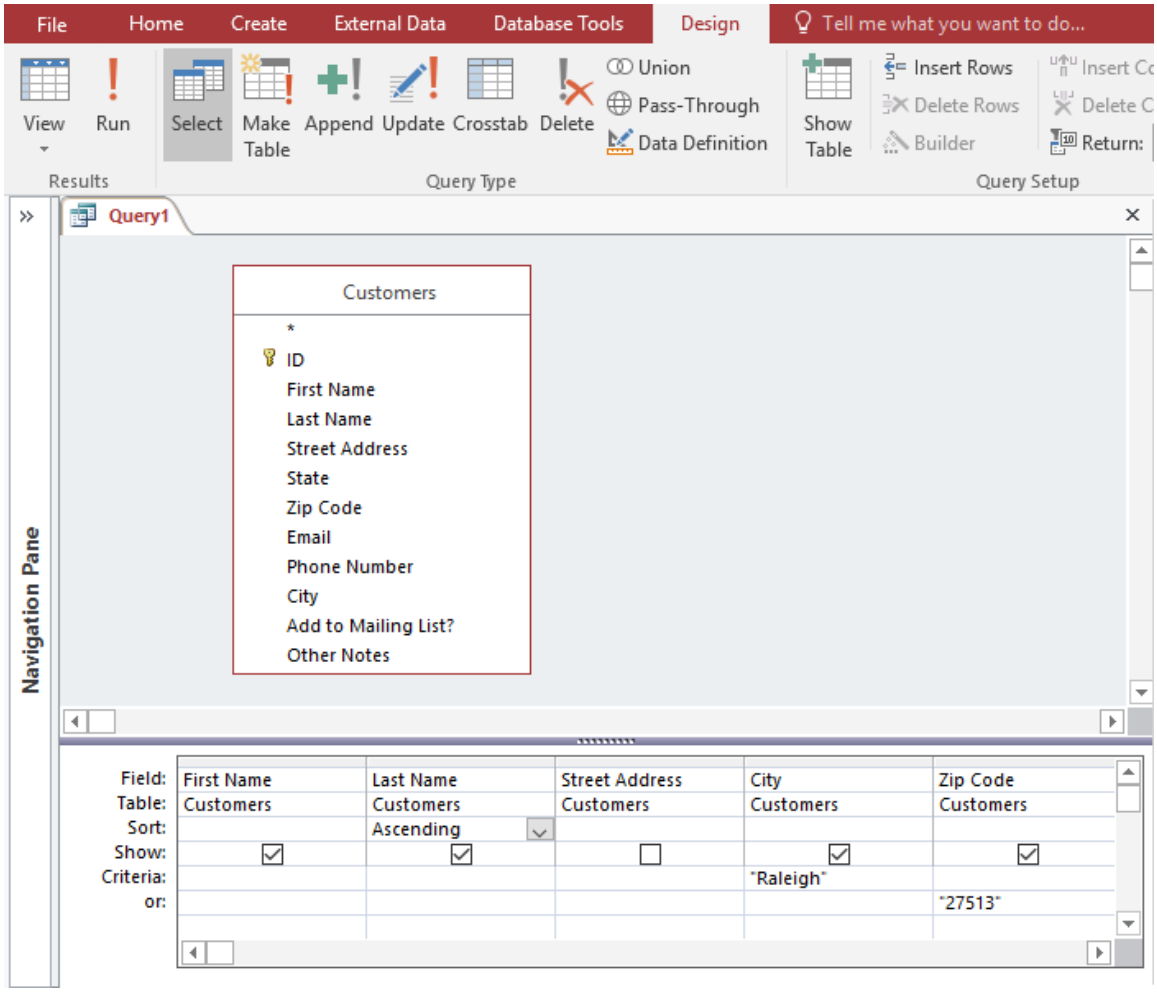


Figure 1

One-table queries

Let's familiarize ourselves with the query-building process by building the **simplest** query possible: a one-table query.

We will run a query on the **Customers** table of our bakery database. Let's say our bakery is having a special event, and we want to invite our customers who live nearby because they are the most likely to come. This means we need to see a list of all customers who live close by, and **only** those customers.

We want to find our customers who live in the city of **Raleigh**, so we'll search for **"Raleigh"** in the City field. Some customers who live in the suburbs live fairly close by, and we'd like to invite them as well. We'll add their zip code, **27513**, as another criteria.

If you think this sounds a little like applying a filter, you're right. A one-table query is actually just an **advanced filter** applied to a table.

To create a simple one-table query:

1. Select the **Create** tab on the Ribbon, and locate the **Queries** group.
2. Click the **Query Design** command.

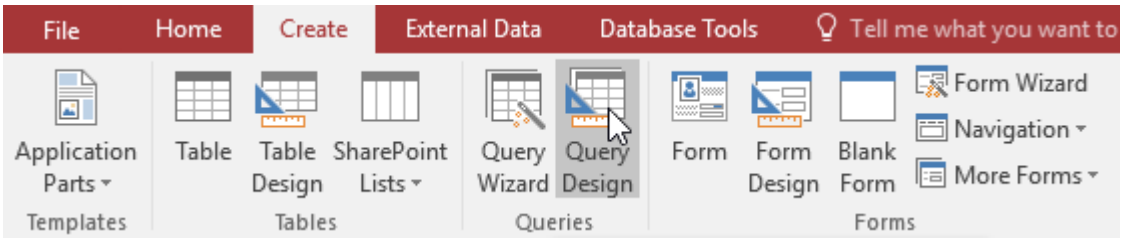


Figure 2

3. Access will switch to **Query Design view**. In the **Show Table** dialog box that appears, select the table you want to run a query on. We are running a query on our customers, so we'll select the **Customers** table.

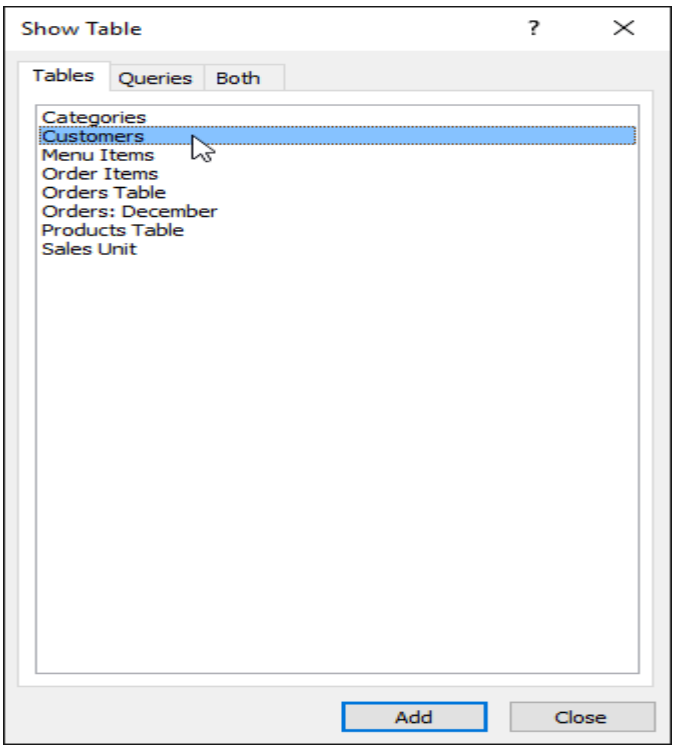


Figure 3

4. Click **Add**, then click **Close**.

5. The selected table will appear as a small window in the **Object Relationship pane**. In the table window, double-click the **field names** you want to include in your query. They will be added to the **design grid** in the bottom part of the screen. In our example, we want to mail invitations to customers who live in a certain area, so we'll include the **First Name**, **Last Name**, **Street Address**, **City**, and **Zip Code** fields.

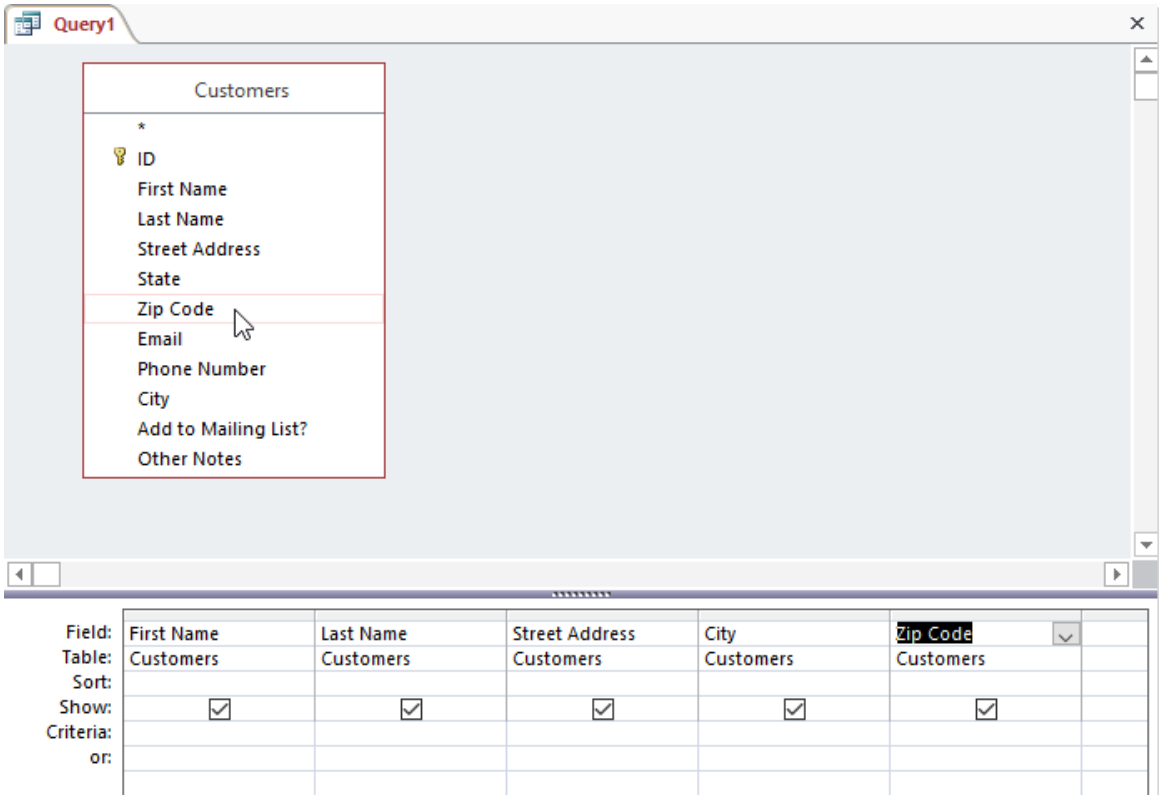


Figure 4

6. Set the **search criteria** by clicking the cell in the **Criteria:** row of each field you want to filter. Typing criteria into more than one field in the Criteria: row will set your query to include only results that meet all criteria. If you want to set multiple criteria but don't need the records shown in your results to meet all of them, type the first criteria in the Criteria: row and additional criteria in the **or:** row and the rows beneath it. Because we want to find customers who either live in Raleigh **or** in the 27513 zip code, we'll type "Raleigh" in

the **City** field and "27513" into the **or:** row of the **Zip Code** field. The **quotation marks** will search these fields for an **exact match**.

Field:	City	Zip Code	
Table:	Customers	Customers	
Sort:			
Show:	☑	☑	☐
Criteria:	"Raleigh"		
or:		"27513"	

Figure 5

7. After you have set your criteria, **run** the query by clicking the **Run** command on the **Design** tab.

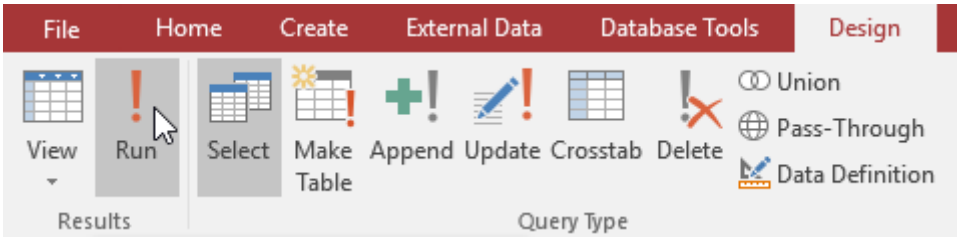


Figure 6

8. The query results will be displayed in the query's **Datasheet view**, which looks like a table. If you want, **save** your query by clicking the **Save** command in the Quick Access Toolbar. When prompted to name it, type the desired name, then click **OK**.

Query1

First Name	Last Name	Street Address	City	Zip Code
Tracey	Beckham	7 East Walker Dr.	Raleigh	27612
Lucinda	George	789 Brewer St.	Cary	27513
Jerrod	Smith	211 St. George Ave.	Raleigh	27610
Brett		h St.	Raleigh	27608
Chloe			Raleigh	27609
Alex			Cary	27513
Nisha		h St.	Raleigh	27612
Hillary			Raleigh	27606
Katy	Jones	456 Denver Rd.	Cary	27513
Beatrix	Joslin	85 North West St.	Raleigh	27606
Mariah	Allen	12 Jupe	Raleigh	27605
Jennifer	Hill	2100 Field Ave.	Raleigh	27609
Cody	Hayes	65 North St.	Raleigh	27609
Amaya	Gibson	5 West St.	Raleigh	27612

Save As ? ×

Query Name:

Nearby Customers

OK Cancel

Figure 7

DESIGNING FORMS

Forms in Access are like display cases in stores that make it easier to view or get the items that you want. Since forms are objects through which you or other users can add, edit, or display the data stored in your Access desktop database, the design of your form is an important aspect. If your Access desktop database is going to be used by multiple users, well-designed forms is essential for efficiency and data entry accuracy.

Create a form from an existing table or query in Access

To create a form from a table or query in your database, in the Navigation Pane, click the table or query that contains the data for your form, and on the **Create** tab, click **Form.**

Access creates a form and displays it in Layout view. You can make design changes like adjusting the size of the text boxes to fit the data, if necessary.

Create a blank form in Access

1. To create a form with no controls or preformatted elements: On the **Create** tab, click **Blank Form**. Access opens a blank form in Layout view, and displays the **Field List** pane.
2. In the **Field List** pane, click the plus sign (+) next to the table or tables that contain the fields that you want to see on the form.
3. To add a field to the form, double-click it or drag it onto the form. To add several fields at once, hold down CTRL and click several fields, and then drag them onto the form at the same time.
4. Use the tools in the **Controls** group on the **Form Layout Tools** tab to add a logo, title, page numbers, or the date and time to the form.
5. If you want to add a wider variety of controls to the form, click **Design** and use the tools in the **Controls** group.

Create a split form in Access

A split form gives you two views of the data at the same time — a Form view and a Datasheet view. Working with split forms gives you the benefits of both types of forms in a single form. For example, you can use the datasheet portion of the form to quickly locate a record, and then

use the form portion to view or edit the record. The two views are connected to the same data source and are synchronized with each other at all times.

Figure 8

To create a new split form by using the Split Form tool, in the Navigation Pane, click the table or query that contains the data, and then on the **Create** tab, click **More Forms**, and then click **Split Form**.

Access creates the form and you can make design changes to the form. For example, you can adjust the size of the text boxes to fit the data, if necessary. For more information on working with a split form, see the article on creating a split form.

Create a form that displays multiple records in Access

A multiple item form, also known as a continuous form, and is useful if you want a form that displays multiple records but is more customizable than a datasheet, you can use the Multiple Items tool.

1. In the Navigation Pane, click the table or query that contains the data you want to see on your form.
2. On the **Create** tab and click **More Forms** > **Multiple Items**.

Access creates the form and displays it in Layout view. In Layout view, you can make design

changes to the form while it is displaying data. For example, you can adjust the size of the text boxes to fit the data. For more details, see Create a form by using the Multiple Items tool.

Create a form that contains a subform in Access

When you are working with related data that is stored in separate tables, you often need to view data from multiple tables or queries on the same form and subforms are a convenient way to do this. Since there are several ways of adding a subform depending on your needs, for more information, see the article Create a form that contains a subform (a one-to-many form).

Create a Navigation form in Access

A navigation form is simply a form that contains a Navigation Control. Navigation forms are a great addition to any database, but creating a navigation form is particularly important if you plan to publish a database to the Web, because the Access Navigation Pane does not display in a browser.

1. Open the database to which you want to add a navigation form.
2. On the **Create** tab, in the **Forms** group, click **Navigation**, and then select the style of navigation form that you want.

   Access creates the form, adds the Navigation Control to it, and displays the form in Layout view.

   There are several options that you can use to customize your forms, see if some the following fit your needs:

Table 2

Options	See this resource
If you want to be able to select which fields appear on the form, use the Form Wizard to create your form.	Create a form by using the Form Wizard
Adding tabs to a form can make the form appear more organized and easier to use, especially if the form contains many controls.	Create a tabbed form
A Web Browser Control lets you to display Web pages on a form. You can link the control to fields or controls in your database to dynamically determine which page is displayed. For example, you can use address information in a database to create custom maps on an Internet mapping site, or you can use product information in a database to search for items on a supplier's Web site.	Add Web browsing to a form
Access places controls in guides called layouts to help you align controls on a form. Find out how to move and size controls independently.	Move and resize controls independently

REPORT DESIGN

Access provides you with a number of tools that help you to quickly build attractive, easy-to-read reports that present the data in a way that best suits the needs of its users. You can use the commands on the **Create** tab to create a simple report with a single click. You can use the Report Wizard to create a more complicated report, or you can create a report by adding all the data and formatting elements yourself. Whichever method that you choose, you will probably make at least a few changes to the design of the report to make it display the data the way that you want.

Decide how to lay out your report

When you design a report, you must first consider how you want the data arranged on the page and how the data is stored in the database. During the design process, you might even discover that the arrangement of data in the tables will not allow you to create the report that you want. This can be an indication that the tables are not normalized — this means that the data is not stored in the most efficient manner.

Make a sketch of your report

This step is not required — you might find that the Access Report Wizard or the Report tool (both of which are available on the **Create** tab, in the **Reports** group) provide a sufficient starting design for your report. However, if you decide to design your report without using these tools, you might find it helpful to make a rough sketch of your report on a piece of paper by drawing a box where each field goes and writing the field name in each box. Alternatively, you can use programs such as Word or Visio to create a mockup of the report. Whichever method that you use, be sure to include enough rows to indicate how the data repeats.

For example, you can use a row for product information, then several repeating rows for that product's sales, and finally a row of sales totals for the product. Then, the sequence repeats for the next product and so on until the end of the report. Or, perhaps your report is a simple listing of the data in the table, in which case your sketch can contain just a series of rows and columns.

After you create your sketch, determine which table or tables contain the data that you want to display on the report. If all the data is contained in a single table, you can base your report directly on that table. More often, the data that you want is stored in several tables that you must pull together in a query, before you can display it on the report. The query can be embedded in the **RecordSource** property of the report, or you can create a separate, saved query and base the report on that.

Decide which data to put in each report section

Each report has one or more report sections. The one section that is present in every report is the Detail section. This section repeats once for each record in the table or query that the report

is based on. Other sections are optional and repeat less often and are usually used to display information that is common to a group of records, a page of the report, or the entire report.

The following table describes where each section is located and how the section is typically used.

Table 3

Section	Location	Typical contents
Report header section	Appears only once, at the top of the first page of the report.	Report title Logo Current date
Report footer section	Appears after the last line of data, above the Page Footer section on the last page of the report.	Report totals (sums, counts, averages, and so on)
Page header section	Appears at the top of each page of the report.	Report title Page number
Page footer section	Appears at the bottom of each page of the report.	Current date Page number
Group header section	Appears just preceding of a group of records.	The field that is being grouped on
Group footer section	Appears just after a group of records.	Group totals (sums, counts, averages, and so on)

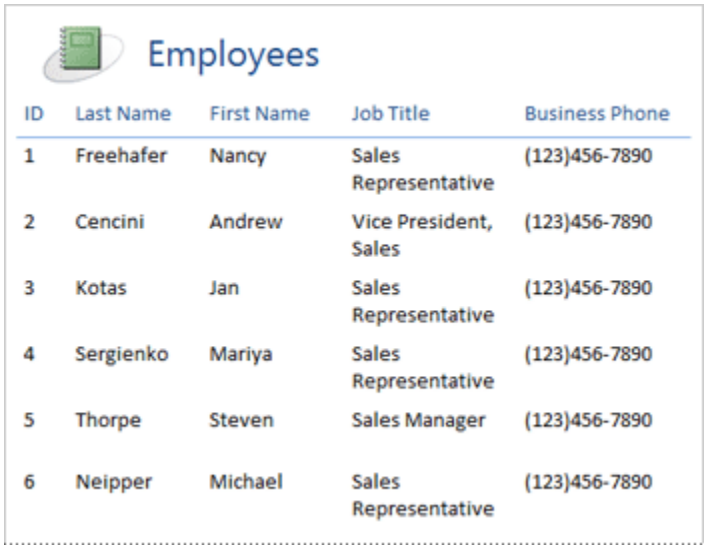
For information about adding or removing report header and footer sections or page header and footer sections, see the section Add or remove report or page header and footer sections in this article. You can add group header and footer sections by using the **Group, Sort, and**

Total pane in Layout view or Design view.

Decide how to arrange the detail data

Most reports are arranged in either a tabular or a stacked layout, but Access gives you the flexibility to use just about any arrangement of records and fields that you want.

Tabular layout A tabular layout is similar to a spreadsheet. Labels are across the top, and the data is aligned in columns below the labels. Tabular refers to the table-like appearance of the data. This is the type of report that Access creates when you click **Report** in the **Reports** group of the **Create** tab. The tabular layout is a good one to use if your report has a relatively small number of fields that you want to display in a simple list format. The following illustration shows an employee report that was created by using a tabular layout.



Employees

ID	Last Name	First Name	Job Title	Business Phone
1	Freehafer	Nancy	Sales Representative	(123)456-7890
2	Cencini	Andrew	Vice President, Sales	(123)456-7890
3	Kotas	Jan	Sales Representative	(123)456-7890
4	Sergienko	Mariya	Sales Representative	(123)456-7890
5	Thorpe	Steven	Sales Manager	(123)456-7890
6	Neipper	Michael	Sales Representative	(123)456-7890

Figure 9

Stacked layout

A stacked layout resembles a form that you fill out when you open a bank account or make a purchase from an online retailer. Each piece of data is labeled, and the fields are stacked on top of each other. This layout is good for reports that contain too many fields to display in a tabular format — that is, the width of the columns would exceed the width of the report. The following illustration shows an employee report that was created by using a stacked layout.

Figure 10

Mixed layout You can mix elements of tabular and stacked layouts. For example, for each record, you can arrange some of the fields in a horizontal row at the top of the Detail section and arrange other fields from the same record in one or more stacked layouts beneath the top row. The following illustration shows an employee report that was created by using a mixed layout. The ID, Last Name, and First Name fields are arranged in a tabular control layout, and the Job Title and Business Phone fields are arranged in a stacked layout. In this example, gridlines are used to provide a visual separation of fields for each employee.

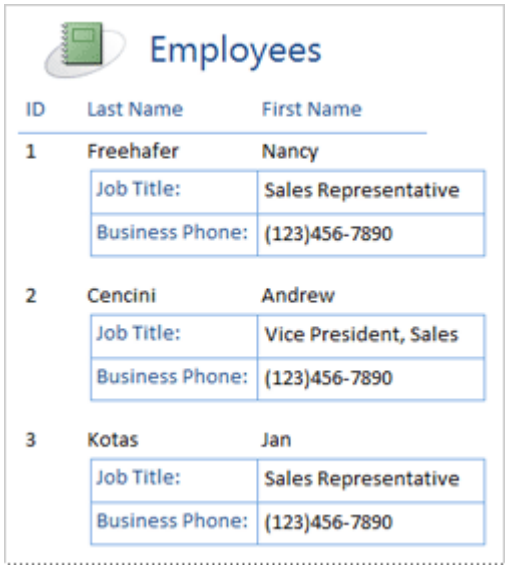


Figure 11

Justified layout If you use the Report Wizard to create your report, you can choose to use a justified layout. This layout uses the full width of the page to display the records as compactly as possible. Of course, you can achieve the same results without using the Report Wizard, but it can be a painstaking process to align the fields exactly. The following illustration shows an employee report that was created by using the Report Wizard's justified layout.

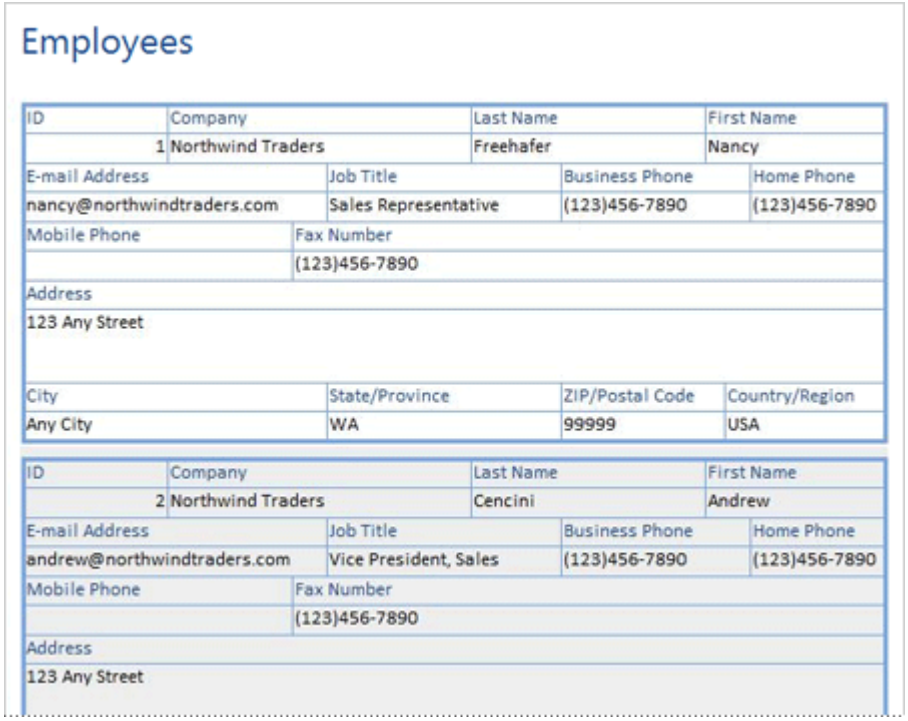


Figure 12

The justified layout is a good layout to use if you are displaying a large number of fields on the report. In the preceding example, if you use a tabular layout to display the same data, the fields extend off the edge of the page. If you use a stacked layout, each record takes up much more vertical space, which wastes paper and makes the report more difficult to read.

Use control layouts to align your data

Control layouts are guides that you can add to a report while it is open in Layout view or Design view. Access adds control layouts automatically when you use the Report Wizard to build a report, or when you create a report by clicking **Report** in the **Reports** group of the **Create** tab. A control layout is like a table, each cell of which can contain a label, a text box, or any other type of control. The following illustration shows a tabular control layout on a report.

Last Name	First Name	Business Phone
Cencini	Andrew	(123)456-7890
Freehafer	Nancy	(123)456-7890
Giussani	Laura	(123)456-7890
Hellung-Larsen	Anne	(123)456-7890
Kotas	Jan	(123)456-7890
Neipper	Michael	(123)456-7890
Sergienko	Mariya	(123)456-7890
Thorpe	Steven	(123)456-7890
Zare	Robert	(123)456-7890

Figure 13

The orange lines indicate the rows and columns of the control layout, and they are visible only when the report is open in Layout view or Design view. Control layouts help you achieve a uniform alignment of data in rows and columns, and they make it easier to add, resize, or remove fields. By using the tools in the **Table** and **Position** groups on the **Arrange** tab (available in Layout view or Design view), you can change one type of control layout to another, and you can remove controls from layouts so that you can position the controls wherever you want on the report.

Add or remove report or page header and footer sections

As mentioned earlier in this article, headers and footers are report sections that you can use to display information that is common to the entire report, or to each page of a report. For example, you can add a Page Footer section to display a page number at the bottom of each page, or you can add a Report Header section to display a title for the entire report.

Add report or page header and footer sections

1. In the Navigation Pane, right-click the report that you want to change, and then click **Design View** on the shortcut menu.
2. Verify which sections are already on the report. The sections are separated by shaded horizontal bars called **section selectors**. The label on each section selector indicates what the section directly below it is.

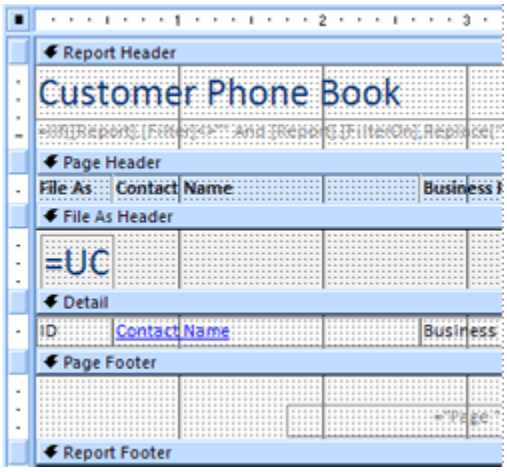


Figure 14

Every report has a Detail section and can also contain Report Header, Page Header, Page Footer, and Report Footer sections. In addition, if there are grouping levels in the report, you might see group headers or footers (such as the **File As Header** shown in the preceding illustration). By default, group headers and footers are named by using the field name or expression that is the basis of the group. In this case, the name of the grouping field is "File As."

3. To add page header and footer sections or report header and footer sections to your report, right-click any section selector and then click **Page Header/Footer** or **Report Header/Footer** on the shortcut menu.

   You can now move existing controls or add new controls to the new sections.

   Access always adds page and report header and footer sections in pairs. That is, you cannot add a page or report header section without also adding the corresponding footer section. If you do not need both sections, you cannot delete a section, but you can resize the unused section to a height of zero (0) to avoid adding extra vertical spacing to your report. Position the pointer at the bottom of the unused section until it turns into a double-headed arrow , and then drag upward until the section is hidden. If there are any controls in the section, you must delete them before you can fully hide the section.

Remove report or page header and footer sections

1. In the Navigation Pane, right-click the report that you want to change, and then click **Design View** on the shortcut menu.
2. Right-click any section selector and then click **Page Header/Footer** or **Report Header/Footer** on the shortcut menu.

If you are removing a header and footer pair and those sections contain controls, Access warns you that deleting the sections will also delete the controls and that you will not be able to undo the action. Click **Yes** to remove the sections and delete the controls, or click **No** to cancel the operation.

Set formatting styles for a text box that displays a rich text field

1. Right-click the report in the Navigation Pane, and then click **Layout View** on the shortcut menu.
2. Click the text box that displays the rich text field, and then, on the **Format** tab, in the **Font** group, click the formatting style that you want to apply.

Access applies the formatting to all text in the rich text field that has not already had that type (but not value) of formatting applied in a view that supports data entry, such as Datasheet view for a table or query, or Form view for a form. For example, if a portion of the text in the field is formatted with a red font color, and you apply a blue font color to the text box, Access turns all of the text blue except for that which was individually formatted as red. As another example, if a portion of the text in the field is formatted with an 11-point font size, and you apply a 14-point font size to the text box, Access applies the 14-point font size to all of the text except for that which was individually formatted at 11 points.

Attachment fields Attachment fields use a special control that is not used for any other data type. You can attach multiple files to a record by using a single Attachment field, but the field can only display information about one attachment at a time. By default, the attachment control displays either an icon or an image, depending on the file type of the attachment that is currently displayed by the control. If you want, you can set the properties for the attachment control so that all attached files are displayed as icons, or so that the field simply displays a paperclip icon and the number of attachments. Assuming that you already use an attachment control on your

report, you can use the following procedure to adjust the control's properties for different uses of the control.

Set the display properties for an Attachment field

1. Right-click the report in the Navigation Pane, and then click **Layout View** on the shortcut menu.
2. Click the attachment control. If the property sheet is not already displayed, press F4 to display it. On the property sheet, click the **Format** tab.

Use the following table as a guide for setting the attachment control's properties.

Table 4

Property	Setting
Display As	**Image/Icon** displays graphics as images and all other files as icons. This is the default setting. **Icon** displays all files as icons. **Paperclip** displays a paperclip icon followed by the number of attachments in parentheses.
Default Picture	To make a default picture appear in the attachment control when there are no attached files, click ... in the property box, browse to the picture that you want, and then click **Open**. **Note:** The default picture is not displayed if the **Display As** property is set to **Paperclip**.
Picture Alignment	Select the alignment that you want from the list. The default setting is **Center**. Adjusting this setting can produce unexpected results, depending on the setting of the **Picture Size Mode** property.
Picture Size Mode	This setting is available only if the **Display As** property is set to **Image/Icon**.

Property	Setting
	Clip displays the image in its actual size. The image is clipped if it is too big to fit inside the control. **Stretch** stretches the image so that it fills the entire control. **Note:** Unless the attachment control is the same exact size as the image, using this setting will distort the image, making it appear stretched either vertically or horizontally. **Zoom** displays the image as large as possible without clipping or distorting the image. This is the default setting.

3. If you are using the control to display graphics, adjust the size of the attachment control so that you can see the amount of detail that you want.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO AND CHEMICAL ENGINEERING

DEPARTMENT OF BIOTECHNOLOGY

Unit 3 – Introduction to Bioinformatics (Elective) – SBB1609

III GENE AND PROTEIN DATABASES

Database - definition

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

DATABASE-MANAGEMENT SYSTEM

A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The DBMS is a general purpose software system that facilitates the process of defining constructing and manipulating databases for various applications.

Goals of DBMS:

The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient

1. Manage large bodies of information
2. Provide convenient and efficient ways to store and access information
3. Secure information against system failure or tampering
4. Permit data to be shared among multiple users

Properties of DBMS:

1. A Database represents some aspect of the real world. Changes to the real world reflected in the database.
2. A Database is a logically coherent collection of data with some inherent meaning.
3. A Database is designed and populated with data for a specific purpose.

Need of DBMS:

1. Before the advent of DBMS, organizations typically stored information using a "File Processing Systems". Example of such systems is File Handling in High Level Languages like C, Basic and COBOL etc., these systems have Major disadvantages to perform the Data Manipulation. So to overcome those drawbacks now we are using the DBMS.
2. Database systems are designed to manage large bodies of information.

3. In addition to that the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results.

ADVANTAGES OF A DBMS OVER FILE SYSTEM:

Using a DBMS to manage data has many advantages:

Data Independence: Application programs should be as independent as possible from details of data representation and storage. The DBMS can provide an abstract view of the data to insulate application code from such details.

Efficient Data Access: A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is stored on external storage devices.

Data Integrity and Security: If data is always accessed through the DBMS, the DBMS can enforce integrity constraints on the data. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, the DBMS can enforce access controls that govern what data is visible to different classes of users. Concurrent Access and

Crash Recovery: A database system allows several users to access the database concurrently. Answering different questions from different users with the same (base) data is a central aspect of an information system. Such concurrent use of data increases the economy of a system. An example for concurrent use is the travel database of a bigger travel agency. The employees of different branches can access the database concurrently and book journeys for their clients. Each travel agent sees on his interface if there are still seats available for a specific journey or if it is already fully booked. A DBMS also protects data from failures such as power failures and crashes etc. by the recovery schemes such as backup mechanisms and log files etc.

Data Administration: When several users share the data, centralizing the administration of data can offer significant improvements. Experienced professionals, who understand the nature of the data being managed, and how different groups of users use it, can be responsible for organizing the data representation to minimize redundancy and fine-tuning the storage of the data to make retrieval efficient.

Reduced Application Development Time: DBMS supports many important functions that are common to many applications accessing data stored in the DBMS. This, in conjunction with the high-level interface to the data, facilitates quick

development of applications. Such applications are also likely to be more robust than applications developed from scratch because many important tasks are handled by the DBMS instead of being implemented by the application

DBMS FUNCTIONS:

DBMS performs several important functions that guarantee the integrity and consistency of the data in the database. Those functions transparent to end users and can be accessed only through the use of DBMS.

They include:

Data Dictionary Management

Data Storage Management

Data transformation and Presentation

Security Management

Multiple Access Control

Backup and Recovery Management

Data Integrity Management

Database Access Languages

Databases Communication Interfaces

Table 1 Difference between File system & DBMS

File system	DBMS
File system is a collection of data. Any 1.management with the file system, user has to write the procedures	1. DBMS is a collection of data and user is not required to write the procedures for managing the database.
File system gives the details of 2. the data representation and Storage of data.	2. DBMS provides an abstract view of data that hides the details.
In File system storing and retrieving of data 3.cannot be done efficiently.	3. DBMS is efficient to use since there are wide varieties of sophisticated techniques to store and retrieve the data.
Concurrent access to the data in the file system 4.has many problems like : Reading the file while other deleting some information, updating some information	4. DBMS takes care of Concurrent access using some form of locking.
File system doesn't provide crash 5. recovery mechanism. Eg. While we are entering some data into the file if System crashes then content of the file is lost	5. DBMS has crash recovery mechanism, DBMS protects user from the effects of system failures.
6.Protecting a file under file system is very difficult.	6. DBMS has a good protection mechanism.

What is Schema?

A database schema is the skeleton structure that represents the logical view of the entire database. (or) The logical structure of the database is called as Database Schema. (or) The overall design of the database is the database schema. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

Biological databases

Biological databases are libraries of life sciences information, collected from scientific experiments, published literature, high-throughput experiment technology, and computational analysis. 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.

Why databases?

- Means to handle and share large volumes of biological data
- Support large-scale analysis efforts
- Make data access easy and updated
- Link knowledge obtained from various fields of biology and medicine

Features

- Most of the databases have a web-interface to search for data
- Common mode to search is by Keywords
- User can choose to view the data or save to your computer
- Cross-references help to navigate from one database to another easily

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.

A biological database is a collection of data that is organized so that its contents can easily be accessed, managed, and updated. There are two main functions of biological databases:

- **Make biological data available to scientists.**
    - As much as possible of a particular type of information should be available in one single place (book, site, and database). Published data may be difficult to find or access and collecting it from the literature is very time-consuming. And not all data is actually published explicitly in an article (genome sequences!).
- **To make biological data available in computer-readable form.**
    - Since analysis of biological data almost always involves computers, having the data in computer-readable form (rather than printed on paper) is a necessary first step.

Data Domains

- Types of data generated by molecular biology research:
  – Nucleotide sequences (DNA and mRNA)
  – Protein sequences
  – 3-D protein structures
  – Complete genomes and maps

Sequence Databases

Nucleic acid sequence databases

EMBL • GenBank • DDBJ

Main protein sequence databases

Swiss Prot

also TREMBL, GenPept

Often integrated with other databases

Structure databases

- NDB, wwPDB, BMRB, CSD, EMDB

Biological databases can be broadly classified into sequence and structure databases. Sequence databases are applicable to both nucleic acid sequences and protein sequences, whereas structure database is applicable to only Proteins. The first database was created within a short period after the Insulin protein sequence was made available in 1956. Incidentally, Insulin is the first protein to be sequenced. The sequence of Insulin consisted of just 51 residues (analogous to alphabets in a sentence) which characterize the sequence. Around mid nineteen sixties, the first nucleic acid sequence of Yeast tRNA with 77 bases (individual units of nucleic acids) was found out. During this period, three dimensional structures of proteins were studied and the well known Protein Data Bank was developed as the first protein structure database with only 10 entries in 1972. This has now grown in to a large database with over 10,000 entries. While the initial databases of protein sequences were maintained at the individual laboratories, the development of a consolidated formal database known as SWISS-PROT protein sequence database was

Primary databases

I. Primary database

1. It is also known as archival database
2. Databases consisting of data derived experimentally such as nucleotide sequences and three dimensional structures are known as primary databases.
3. Experimental results are directly submitted into database by researchers across the globe
4. Example: Gen bank, DDBJ, SWISS-PROT

Contain sequence data such as nucleic acid or protein

Example of primary databases include :

Protein Databases

- SWISS-PROT
- TREMBL
- PIR

Nucleic Acid Databases

- EMBL
- Genbank
- DDBJ

Secondary databases

II. Secondary database

1. It is also known as curated database
2. Databases consisting of data derived from the analysis of primary data such as sequences, secondary structures etc
3. It contains results of analysis of primary databases and significant data in the form of conserved sequences, signature sequences, active site residues of proteins etc.

Or sometimes known as pattern databases

Contain results from the analysis of the sequences in the primary databases

Example of secondary databases include : PROSITE, Pfam, BLOCKS, PRINTS

Composite databases

Combine different sources of primary databases.

Make querying and searching efficient and without the need to go to each of the primary databases.

Example of composite databases include : NRDB – Non-Redundant DataBase, OWL

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. Its legacy is the FASTA format which is now ubiquitous in bioinformatics.

The original FASTP 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. There are several programs in this package that allow the alignment of protein sequences and DNA sequences.

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 SEARCH, 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.).

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 kmer, controls the sensitivity and speed of the program. Increasing the kmer 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 kmer 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 kmer value determines how many consecutive identities are required for a match to be declared. Thus the lesser the kmer value: the more sensitive the search. kmer=2 is frequently taken by users for protein sequences and kmer=4 or 6 for nucleotide sequences. Short oligonucleotides are usually run with kmer= 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 kmer 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 kmer 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.

Rescan the regions taken using the scoring matrices. trimming the ends of the region to include only those contributing to the highest score.

Rescan the 10 regions taken. This time use the relevant scoring matrix while rescoring to allow runs of identities shorter than the kmer value. Also while rescoring conservative replacements that contribute to the similarity score are taken. Though protein sequences use the BLOSUM50 matrix, scoring matrices based on the minimum number of base changes required for a specific replacement, on identities alone, or on an alternative measure of similarity such as PAM, can also be used with the program. For each of the diagonal regions rescanned this way, a subregion with the maximum score is identified. The initial scores found in step1 are used to rank the library sequences. The highest score is referred to as init1 score.

In an alignment if several initial regions with scores greater than a CUTOFF value are found, check whether the trimmed initial regions can be joined to form an approximate alignment with gaps. Calculate a similarity score that is the sum of the joined regions penalising for each gap 20 points. This initial similarity score (initn) is used to rank the library sequences. The score of the single best initial region found in step 2 is reported (init1). Here the program calculates an optimal alignment of initial regions as a combination of compatible regions with maximal score. This optimal alignment of initial regions can be rapidly calculated using a dynamic programming algorithm. The resulting score initn is used to rank the library sequences. This joining process increases sensitivity but decreases selectivity. A carefully calculated cut-off value is thus used to control where this step is implemented, a value that is approximately one standard deviation above the average score expected from unrelated sequences in the library. A 200-residue query sequence with kmer 2 uses a value 28.

This step uses a banded Smith-Waterman algorithm to create an optimised score (opt) for each alignment of query sequence to a database (library) sequence. It takes a band of 32 residues centered on the init1 region of step2 for calculating the optimal alignment. After all sequences are searched the program plots the initial scores of

each database sequence in a histogram, and calculates the statistical significance of the "opt" score. For protein sequences, the final alignment is produced using a full Smith-Waterman alignment. For DNA sequences, a banded alignment is provided.

FASTA cannot remove low complexity regions before aligning the sequences as it is possible with BLAST. This might be problematic as when the query sequence contains such regions, e.g. mini- or microsatellites repeating the same short sequence frequent times, this increases the score of not familiar sequences in the database which only match in this repeats, which occur quite frequently. Therefore the program PRSS is added in the FASTA distribution package. PRSS shuffles the matching sequences in the database either on the one-letter level or it shuffles short segments which length the user can determine. The shuffled sequences are now aligned again and if the score is still higher than expected this is caused by the low complexity regions being mixed up still mapping to the query. By the amount of the score the shuffled sequences still attain PRSS now can predict the significance of the score of the original sequences. The higher the score of the shuffled sequences the less significant the matches found between original database and query sequence.

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.

BLAST - Basic Local Alignment Search Tool

The BLAST algorithm was developed by Altschul, Gish, Miller, Myers and Lipman in 1990. The motivation for the development of BLAST was the need to increase the speed of FASTA by finding fewer and better hot spots during the algorithm. The idea was to integrate the substitution matrix in the first stage of finding the hot spots. The BLAST algorithm was developed for protein alignments in comparison to FASTA, which was developed for DNA sequences.

Different types of BLAST

blastn compares your query nucleotide sequence with database nucleotide sequences

blastp compares your query protein sequence with database of protein sequences that were derived form cDNA of interest blastx first translates your query sequence into amino acids in six reading frames (three forward and three back) then compares the protein sequences with protein databases

tblastn compares your query protein sequence with the database after translating each nucleotide sequence into protein using all six reading frames (This algorithm takes a long time, but is more likely to find distantly related sequences than the blastn, blastx, and blastp.)

tblastx translates both the query nucleotide sequence and the database sequences in all six reading frames and then compares the protein sequences (This, like tblastn, is very time consuming, but finds more results).

Algorithm

1. Remove low-complexity region or sequence repeats in the query sequence.
2. Make a k-letter word list of the query sequence.
3. List the possible matching words.
4. Organize the remaining high-scoring words into an efficient search tree.
5. Repeat step 3 to 4 for each k-letter word in the query sequence.
6. Scan the database sequences for exact matches with the remaining high-scoring words.
7. Extend the exact matches to high-scoring segment pair (HSP).
8. List all of the HSPs in the database whose score is high enough to be considered.
9. Evaluate the significance of the HSP score.
10. Make two or more HSP regions into a longer alignment.
11. Show the gapped Smith-Waterman local alignments of the query and each of the matched database sequences.
12. Report every match whose expect score is lower than a threshold parameter

E.

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 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 National Institutes of Health and was published in the Journal of Molecular Biology in 1990 and cited over 50,000 times.

Background

BLAST is one of the most widely used bioinformatics programs for sequence searching. 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 available, although subsequent algorithms can be even faster.

Before BLAST, FASTA was developed by David J. Lipman and William R. Pearson in 1985.

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 any Smith-Waterman implementation for most cases,

it cannot "guarantee the optimal alignments of the query and database sequences" as Smith-Waterman algorithm 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.

Examples of other questions that researchers use BLAST to answer are:

- Which bacterial species have a protein that is related in lineage to a certain protein with known amino-acid sequence
- What other genes encode proteins that exhibit structures or motifs such as ones that have just been determined

BLAST is also often used as part of other algorithms that require approximate sequence matching.

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 (last updated in 2006), and ScalaBLAST.

Input

Input sequences (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, by locating short matches between the two sequences. This process of finding similar sequences 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 BLAST was being conducted under normal 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. This result will then be used to build an alignment. After making words for the sequence of interest, the rest of the 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 BLAST 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. If this score is higher than a pre-determined T, the alignment will be included in the results given by BLAST. However, if this score is lower than this pre-determined T, the alignment will cease to extend, preventing the 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.

Program

The BLAST program can either be downloaded and run as a command-line utility "blastall" or accessed for free over the web. The BLAST web server, hosted by the NCBI, allows anyone with a web browser to perform similarity searches against constantly updated databases of proteins and DNA that include most of the newly sequenced organisms.

The BLAST program is based on an open-source format, giving everyone access to it and enabling them to have the ability to change the program code. This has led to the creation of several BLAST "spin-offs".

There are now a handful of different BLAST programs available, which can be used depending on what one is attempting to do and what they are working with. These different programs vary in query sequence input, the database being searched, and what is being compared. These programs and their details are listed below:

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

These include:

Nucleotide-nucleotide BLAST (blastn)

This program, given a DNA query, returns the most similar DNA sequences from the DNA database that the user specifies.

Protein-protein BLAST (blastp)

This program, given a protein query, returns the most similar protein sequences from the protein database that the user specifies.

Position-Specific Iterative BLAST (PSI-BLAST) (blastpgp)

The program is used to find distant relatives of a protein. First, a list of all closely related proteins is created. These proteins are combined into a general "profile" sequence, which summarises significant features present in these sequences. A query against the protein database is then run using this profile, and a larger group of proteins is found. This larger group is used to construct another profile, and the process is repeated.

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

Nucleotide 6-frame translation-protein (blastx)

This program compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database.

Nucleotide 6-frame translation-nucleotide 6-frame translation (tblastx)

This program is the slowest of the BLAST family. It translates the query nucleotide sequence in all six possible frames and compares it against the six-frame translations of a nucleotide sequence database. The purpose of tblastx is to find very distant relationships between nucleotide sequences.

Protein-nucleotide 6-frame translation (tblastn)

This program compares a protein query against the all six reading frames of a nucleotide sequence database.

Large numbers of query sequences (megablast)

When comparing large numbers of input sequences via the command-line BLAST, "megablast" is much faster than running BLAST multiple times. It concatenates many input sequences together to form a large sequence before searching the BLAST database, then post-analyzes the search results to glean individual alignments and statistical values.

BLASTn and BLASTp are the most commonly used programs for direct comparisons, and do not require translations. However, since protein sequences are better conserved evolutionarily than nucleotide sequences, tBLASTn, tBLASTx, and BLASTx, produce more reliable and accurate results when dealing with coding DNA. They also enable one to be able to directly see the function of the protein sequence, since by translating the sequence of interest before searching often gives you annotated protein hits.

Genbank

GenBank, the National Institutes of Health (NIH) genetic sequence database, is an annotated collection of all publicly available nucleotide and protein sequences. The records within GenBank represent, in most cases, single, contiguous stretches of DNA or RNA with annotations. GenBank files are grouped into divisions; some of these divisions are phylogenetically based, whereas others are based on the technical approach that was used to generate the sequence information. Presently, all records in GenBank are generated from direct submissions to the DNA sequence databases from the original authors, who volunteer their records to make the data publicly available or do so as part of the publication

process. GenBank, which is built by the National Center for Biotechnology Information (NCBI), is part of the International Nucleotide Sequence Database Collaboration, along with its two partners, the DNA Data Bank of Japan (DDBJ, Mishima, Japan) and the European Molecular Biology Laboratory (EMBL) nucleotide database from the European Bioinformatics Institute (EBI, Hinxton, UK). All three centers provide separate points of data submission, yet all three centers exchange this information daily, making the same database (albeit in slightly different format and with different information systems) available to the community at-large.

Only original sequences can be submitted to GenBank. Direct submissions are made to GenBank using BankIt, which is a Web-based form, or the stand-alone submission program, Sequin. Upon receipt of a sequence submission, the GenBank staff examines the originality of the data and assigns an accession number to the sequence and performs quality assurance checks. The submissions are then released to the public database, where the entries are retrievable by Entrez or downloadable by FTP. Bulk submissions of Expressed Sequence Tag (EST), Sequence-tagged site (STS), Genome Survey Sequence (GSS), and High- Throughput Genome Sequence (HTGS) data are most often submitted by large-scale sequencing centers. The GenBank direct submissions group also processes complete microbial genome sequences.

THE GENBANK FLATFILE: A DISSECTION The GenBank flatfile (GBFF) is the elementary unit of information in the GenBank database. It is one of the most commonly used formats in the representation of biological sequences. At the time of this writing, it is the format of exchange from GenBank to the DDBJ and EMBL databases and vice versa. The DDBJ flat file format and the GBFF format are now nearly identical to the GenBank format. Subtle differences exist in the formatting of the definition line and the use of the gene feature. EMBL uses line-type prefixes, which indicate the type of information present in each line of the record.

The GBFF can be separated into three parts: the header, which contains the information (descriptors) that apply to the whole record; the features, which are the annotations on the record; and the nucleotide sequence itself. All major nucleotide database flat files end with // on the last line of the record. The header is the most database-specific part of the record. The various databases are not obliged to carry the same information in this segment, and

minor variations exist, but some effort is made to ensure that the same information is carried from one to the other.

The first line of all GBFFs is the Locus line:

Locus name

The locus name was originally designed to help group entries with similar sequences: the first three characters usually designated the organism; the fourth and fifth characters were used to show other group designations, such as gene product; for segmented entries, the last character was one of a series of sequential integers.

Sequence length

Number of nucleotide base pairs (or amino acid residues) in the sequence record.

Molecule Type

The type of molecule that was sequenced Genbank division

The GenBank division to which a record belongs is indicated with a three letter abbreviation. In this example, GenBank division is PRI.

The GenBank database is divided into 18 divisions:

1. PRI - primate sequences
2. ROD - rodent sequences
3. MAM - other mammalian sequences
4. VRT - other vertebrate sequences
5. INV - invertebrate sequences
6. PLN - plant, fungal, and algal sequences
7. BCT - bacterial sequences
8. VRL - viral sequences
9. PHG - bacteriophage sequences
10. SYN - synthetic sequences
11. UNA - unannotated sequences
12. EST - EST sequences (expressed sequence tags)
13. PAT - patent sequences
14. STS - STS sequences (sequence tagged sites)

15. GSS - GSS sequences (genome survey sequences)
16. HTG - HTG sequences (high-throughput genomic sequences)
17. HTC - unfinished high-throughput cDNA sequencing
18. ENV - environmental sampling sequences

Modification date

The date in the LOCUS field is the **date of last modification**. The sample record shown here was last modified on

Definition

Brief description of sequence; includes information such as source organism, gene name/protein name, or some description of the sequence's function

Accession

The unique identifier for a sequence record. An accession number applies to the complete record and is usually a combination of a letter(s) and numbers, such as a single letter followed by five digits (e.g., U12345) or two letters followed by six digits (e.g., AF123456). Accession numbers do not change, even if information in the record is changed at the author's request.

Version

If there is any change to the sequence data (even a single base), the version number will be increased, e.g., U12345.1 → U12345.2, but the accession portion will remain stable.

GI

"GenInfo Identifier" sequence identification number, in this case, for the nucleotide sequence. If a sequence changes in any way, a new GI number will be assigned. GI sequence identifiers run parallel to the new **accession.version** system of sequence identifiers

Keywords

Word or phrase describing the sequence. If no keywords are included in the entry, the field contains only a period.

Source

Free-format information including an abbreviated form of the organism name, sometimes followed by a molecule type.

Features

Information about genes and gene products, as well as regions of biological significance reported in the sequence. These can include regions of the sequence that code for proteins and RNA molecules, as well as a number of other features.

The **location of each feature** is provided as well, and can be a single base, a contiguous span of bases, a joining of sequence spans, and other representations. If a feature is located on the complementary strand, the word "complement" will appear before the base span

Source: Mandatory feature in each record that summarizes the length of the sequence, scientific name of the source organism, and Taxon ID number. Can also include other information such as map location, strain, clone, tissue type, etc., if provided by submitter.

Taxon: A stable unique identification number for the taxon of the source organism. A taxonomy ID number is assigned to each taxon

CDS:

Coding sequence; region of nucleotides that corresponds with the sequence of amino acids in a protein (location includes start and stop codons). The CDS feature includes an amino acid translation <1…206 Base span of the biological feature indicated to the left, in this case, a CDS feature Gene

A region of biological interest identified as a gene and for which a name has been assigned. The base span for the gene feature is dependent on the furthest 5' and 3' features.

Origin

The ORIGIN may be left blank, may appear as "Unreported," or may give a local pointer to the sequence start, usually involving an experimentally determined restriction cleavage site or the genetic locus (if available). This information is present only in older records.

The sequence data begin on the line immediately below ORIGIN.

DNA Data Bank of Japan

The DNA Data Bank of Japan (DDBJ) is a biological database that collects DNA sequences. It is located at the National Institute of Genetics (NIG) in the Shizuoka prefecture of Japan. It is also a member of the International Nucleotide Sequence Database Collaboration or INSDC. It exchanges its data with European Molecular Biology Laboratory at the European Bioinformatics Institute and with GenBank at the National Center for Biotechnology Information on a daily basis. Thus these three databanks contain the same data at any given time.

DDBJ began data bank activities in 1986 at NIG and remains the only nucleotide sequence data bank in Asia. Although DDBJ mainly receives its data from Japanese researchers, it can accept data from contributors from any other country. DDBJ is primarily funded by the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT). DDBJ has an international advisory committee which consists of nine members, 3 members each from Europe, US, and Japan. This committee advises DDBJ about its maintenance, management and future plans once a year. Apart from this DDBJ also has an international collaborative committee which advises on various technical issues related to international collaboration and consists of working-level participants.

The format of DDBJ is similar to that of Genbank.

EMBL

The European Molecular Biology Laboratory (EMBL) is a molecular biology

research institution supported by 21 member states, three prospect and two associate member states. EMBL was created in 1974 and is an intergovernmental organisation funded by public research money from its member states. Research at EMBL is conducted by approximately 85 independent groups covering the spectrum of molecular biology.

The Laboratory operates from five sites: the main laboratory in Heidelberg, and outstations inHinxton (the European Bioinformatics Institute (EBI), in England), Grenoble (France), Hamburg (Germany), and Monterotondo (near Rome). EMBL groups and laboratories perform basic research in molecular biology and molecular medicine as well as training for scientists, students and visitors. The organization aids in the development of services, new instruments and methods, and technology in its member states. Each of the different EMBL sites have a specific research field. The EMBL-EBI is a hub for bioinformatics research and services, developing and maintaining a large number of scientific databases, which are free of charge. At Grenoble and Hamburg, research is focused on structural biology. EMBL's dedicated Mouse Biology Unit is located in Monterotondo. Many scientific breakthroughs have been made at EMBL, most notably the first systematic genetic analysis of embryonic development in the fruit fly by Christiane Nüsslein-Volhard and Eric Wieschaus, for which they were awarded the Nobel Prize in Physiology or Medicine in 1995

EMBL format

A sequence file in EMBL format can contain several sequences. One sequence entry starts with an identifier line ("ID"), followed by further annotation lines. The start of the sequence is marked by a line starting with "SQ" and the end of the sequence is marked by two slashes ("//").

UniProt

UniProt is a comprehensive, high-quality and freely accessible database of protein sequence and functional information, many entries being derived from genome sequencing projects. It contains a large amount of information about the biological function of proteins derived from the research literature. Universal Protein resource, a central repository of protein data created by combining the Swiss-Prot, TrEMBL

and PIR-PSD databases.

The UniProt consortium comprises the European Bioinformatics Institute (EBI), the Swiss Institute of Bioinformatics (SIB), and the Protein Information Resource (PIR). EBI, located at the Welcome Trust Genome Campus in Hinxton, UK, hosts a large resource of bioinformatics databases and services. SIB, located in Geneva, Switzerland, maintains the ExPASy(Expert Protein Analysis System) servers that are a central resource for proteomics tools and databases. PIR, hosted by the National Biomedical Research Foundation (NBRF) at the Georgetown University Medical Center in Washington, DC, USA, is heir to the oldest protein sequence database, Margaret Dayhoff's Atlas of Protein Sequence and Structure, first published in 1965.[2] In 2002, EBI, SIB, and PIR joined forces as the UniProt consortium

SWISSPROT

SWISS-PROT is an annotated protein sequence database, which was created at the Department of Medical Biochemistry of the University of Geneva and has been a collaborative effort of the Department and the European Molecular Biology Laboratory (EMBL), since 1987. SWISS-PROT is now an equal partnership between the EMBL and the Swiss Institute of Bioinformatics (SIB). The EMBL activities are carried out by its Hinxton Outstation, the European Bioinformatics Institute (EBI). The SWISS-PROT protein sequence database consists of sequence entries. Sequence entries are composed of different line types, each with their own format. For standardisation purposes the format of SWISS-PROT (see http://www.expasy. ch/txt/userman.txt) follows as closely as possible that of the EMBL Nucleotide Sequence Database.

The SWISS-PROT database distinguishes itself from other protein sequence databases by three distinct criteria: (i) annotations, (ii) minimal redundancy and (iii) integration with other databases (Cross references).

Annotation

In SWISS-PROT two classes of data can be distinguished: the core data and the annotation. For each sequence entry the core data consists of the sequence data; the

citation information (bibliographical references) and the taxonomic data (description of the biological source of the protein), while the annotation consists of the description of the following items:

- Function(s) of the protein
- Post-translational modification(s). For example carbohydrates, phosphorylation, acetylation, GPI-anchor, etc.
- Domains and sites. For example calcium binding regions, ATP-binding sites, zinc fingers, homeoboxes, SH2 and SH3 domains, etc.
- Secondary structure. For example alpha helix, beta sheet, etc.
- Quaternary structure. For example homodimer, heterotrimer, etc.
- Similarities to other proteins
- Disease(s) associated with deficiencie(s) in the protein
- Sequence conflicts, variants, etc.

Minimal redundancy

Many sequence databases contain, for a given protein sequence, separate entries which correspond to different literature reports. In SWISS-PROT we try as much as possible to merge all these data so as to minimise the redundancy of the database

Integration with other databases

It is important to provide the users of biomolecular databases with a degree of integration between the three types of sequence-related databases (nucleic acid sequences, protein sequences and protein tertiary structures) as well as with specialised data collections. Cross-references are provided in the form of pointers to information related to SWISS-PROT entries and found in data collections other than SWISS-PROT. For example the sample sequence mentioned above contains, among others, DR (Databank Reference) lines that point to EMBL, PDB, OMIM, Pfam and PROSITE.

TREMBL: A COMPUTER ANNOTATED SUPPLEMENT TO SWISS-PROT

Maintaining the high quality of sequence and annotation in SWISS-PROT requires careful sequence analysis and detailed annotation of every entry. This is the rate-

limiting step in the production of SWISS-PROT. On one hand we do not wish to relax the high editorial standards of SWISS-PROT and it is clear that there is a limit to how much we can accelerate the annotation procedures. On the other hand, it is also vital that we make new sequences available as quickly as possible. To address this concern, we introduced in 1996 TrEMBL (Translation of EMBL nucleotide sequence database). TrEMBL consists of computer-annotated entries derived from the translation of all coding sequences (CDSs) in the EMBL database, except for CDSs already included in SWISS-PROT.

We have split TREMBL into two main sections, SP-TREMBL and REM-TREMBL. SP- TREMBL (SWISS-PROT TREMBL) contains entries (~55 000) which should be incorporated into SWISS-PROT. SWISS-PROT accession numbers have been assigned to these entries. SP- TREMBL is partially redundant against SWISS-PROT, since ~30 000 of these SP-TREMBL entries aie only additional sequence reports of proteins already in SWISS-PROT. REM-TREMBL (REMaining TREMBL) contains those entries (~15 000) that we do not wish to include in SWISS- PROT. This section is organized into four subsections. Most REM-TREMBL entries are immunoglobulins and T-cell receptors. We have stopped entering immunoglobulins and T-cell receptors into SWISS-PROT, because we want to keep only germ line gene-derived translations of these proteins in SWISS-PROT and not all known somatic recombinant variations of these proteins. Another category of data which will not be included in SWISS-PROT is synthetic sequences. A third subsection consists of fragments with less than seven amino acids. The last subsection consists of CDS translations where we have strong evidence to believe that these CDS are not coding for real proteins.

The creation of TREMBL as a supplement to SWISS-PROT was not only for the purpose of producing a more complete and up to date protein sequence collection. Also to achieve a deeper integration of the EMBL nucleotide sequence database with SWISS-PROT + TREMBL.

Structure of a sequence entry

The entries in the SWISS-PROT data bank are structured so as to be usable by

human readers as well as by computer programs. The explanations, descriptions, classifications and other comments are in ordinary English. Wherever possible, symbols familiar to biochemists, protein chemists and molecular biologists are used. Each sequence entry is composed of lines. Different types of lines, each with their own format, are used to record the various data which make up the entry.

Each line begins with a two-character line code, which indicates the type of data contained in the line. The current line types and line codes and the order in which they appear in an entry, are shown below:

ID - Identification.
AC - Accession number(s). DT - Date.
DE - Description. GN - Gene name(s).
OS - Organism species. OG - Organelle.
OC - Organism classification. RN - Reference number.
RP - Reference position. RC - Reference comments.
RX - Reference cross-references. RA - Reference authors.
RL - Reference location. CC - Comments or notes.
DR - Database cross-references. KW Keywords.
FT - Feature table data. SQ - Sequence header.
(blanks) sequence data. // - Termination line.

Protein Information Resource

The Protein Information Resource (PIR), located at Georgetown University Medical Center (GUMC), is an integrated public bioinformatics resource to support genomic and proteomic research, and scientific studies. PIR was established in 1984 by the National Biomedical Research Foundation (NBRF) as a resource to assist researchers and costumers in the identification and interpretation of protein sequence information. Prior to that, the NBRF compiled the first comprehensive collection of macromolecular sequences in the Atlas of Protein Sequence and Structure, published from 1964-1974 under the editorship of Margaret Dayhoff.

Dr. Dayhoff and her research group pioneered in the development of computer methods for the comparison of protein sequences, for the detection of distantly

related sequences and duplications within sequences, and for the inference of evolutionary histories from alignments of protein sequences.

The Protein Information Resource (PIR) produces the largest, most comprehensive, annotated protein sequence database in the public domain, the PIR-International Protein Sequence Database, in collaboration with the Munich Information Center for Protein Sequences (MIPS) and the Japan International Protein Sequence Database (JIPID).

PIR, MIPS and JIPID constitute the PIR-International consortium that maintains the PIR- International Protein Sequence Database (PSD), the largest publicly distributed and freely available protein sequence database. The database has the following distinguishing features.

- It is a comprehensive, annotated, and non-redundant protein sequence database, containing over 142 000 sequences as of September 1999. Included are sequences from the completely sequenced genomes of 16 prokaryotes, six archaebacteria, 17 viruses and phages, >100 eukaryote organelles and Saccharomyces cerevisiae.

- The collection is well organized with >99% of entries classified by protein family and >57% classified by protein superfamily.

- PSD annotation includes concurrent cross-references to other sequence, structure, genomic and citation databases, including the public nucleic acid sequence databases ENTREZ, MEDLINE, PDB, GDB, OMIM, FlyBase, MIPS/Yeast, SGD/Yeast, MIPS/Arabidopsis and TIGR. Where these databases are publicly and freely accessible and provide suitable WWW access, the cross-references presented on the PIR WWW site are hot-linked so that searchers can consult the most current data.

- The PIR is the only sequence database to provide context cross-references between its own database entries. These cross-references assist searchers in exploring relationships such as subunit associations in molecular complexes,

enzyme–substrate interactions, activation and regulation cascades, as well as in browsing entries with shared features and annotations.

• Interim updates are made publicly available on a weekly basis, and full releases have been published quarterly since 1984.

It is split into 4 distinct section (PIR1-PIR4).
PIR1: contains fully classified and annotated entries.
PIR2: includes preliminary entries not been thoroughly reviewed, contain redundancy. PIR3: contains unverified entries.
PIR4: fall into one of four categories.

- Conceptual translations of art factual sequences.
- Conceptual translations of sequences that are not transcribed or translated.
- Protein sequences or conceptual translations that are extensively genetically engineered
- Sequences that are not genetically encoded and not produced on ribosomes.

Protein data bank

The Protein Data Bank (PDB) is a crystallographic database for the three-dimensional structural data of large biological molecules, such as proteins and nucleic acids. The data, typically obtained by X-ray crystallography, NMR spectroscopy, or, increasingly, cryo-electron microscopy, and submitted by biologists and biochemists from around the world, are freely accessible on the Internet via the websites of its member organisations (PDBe, PDBj, and RCSB). The PDB is overseen by an organization called the Worldwide Protein Data Bank, wwPDB.

The PDB is a key resource in areas of structural biology, such as structural genomics. Most major scientific journals, and some funding agencies, now require scientists to submit their structure data to the PDB. Many other databases use protein structures deposited in the PDB. For example, SCOP and CATH classify protein structures, while PDBsum provides a graphic overview of PDB entries using

information from other sources, such as Gene ontology

The Protein Data Bank (PDB) at Brookhaven National Laboratory (BNL), is a database containing experimentally determined three-dimensional structures of proteins, nucleic acids and other biological macromolecules. The archives contain atomic coordinates, citations, primary and secondary structure information, crystallographic structure experimental data, as well as hyperlinks to many other scientific databases.

Protein Data Bank (PDB) format is a standard for files containing atomic coordinates. Structures deposited in the Protein Data Bank at the Research Collaboratory for Structural Bioinformatics (RCSB) are written in this standardized format. The complete PDB file specification provides for a wealth of information, including authors, literature references, and the identification of substructures such as disulfide bonds, helices, sheets, and active sites.

Protein Data Bank format consists of lines of information in a text file. Each line of information in the file is called a record. A file generally contains several different types of records, which are arranged in a specific order to describe a structure.

Table 2 PDB Record Types

Selected Protein Data Bank Record Types

Record Type	
ATOM	atomic coordinate record containing the x,y,z orthogonal Angstrom coordinates for atoms in standard residues (amino acids and nucleic acids).
HETATM	atomic coordinate record containing the x,y,z orthogonal Angstrom coordinates for atoms in nonstandard residues. Nonstandard residues include inhibitors, cofactors, ions, and solvent. The only functional difference from ATOM records is that HETATM residues are by default not connected to other residues. Note that water residues should be in HETATM records.
TER	indicates the end of a chain of residues. For example, a hemoglobin molecule consists of four subunit chains which are not connected. TER indicates the end of a chain and prevents the display of a connection to the next chain.
SSBOND	defines disulfide bond linkages between cysteine residues.
HELIX	indicates the location and type (right-handed alpha, *etc.*) of helices. One record per helix.
SHEET	indicates the location, sense (anti-parallel, *etc.*) and registration with respect to the previous strand in the sheet (if any) of each strand in the model. One record per strand.

The Protein Data Bank (pdb) file format is a textual file format describing the three-dimensional structures of molecules held in the Protein Data Bank. The pdb format accordingly provides for description and annotation of protein and nucleic acid structures including atomic coordinates, observed sidechain rotamers, secondary structure assignments, as well as atomic connectivity. Structures are often deposited with other molecules such as water, ions, nucleic acids, ligands and so on, which can be described in the pdb format as well. The Protein Data Bank also keeps data on biological macromolecules in the newer mmCIF file format.

A typical PDB file describing a protein consists of hundreds to thousands of lines like the following (taken from a file describing the structure of a synthetic collagen-like peptide):

HEADER, TITLE and AUTHOR records provide information about the researchers who defined the structure; numerous other types of records are available to provide other types of information.

REMARK records can contain free-form annotation, but they also accommodate standardized information; for records describe how to compute the coordinates of the experimentally observed multimer from those of the explicitly specified ones of a single repeating unit.

SEQRES records give the sequences of the three peptide chains (named A, B and C), which are very short in this example but usually span multiple lines.

ATOM records describe the coordinates of the atoms that are part of the protein. For example, the first ATOM line above describes the alpha-N atom of the first residue of peptide chain A, which is a proline residue; the first three floating point numbers are its x, y and z coordinates and are in units of Ångströms. The next three columns are the occupancy, temperature factor, and the element name, respectively.

HETATM records describe coordinates of hetero-atoms which are not part of the protein molecule.

Secondary Databases

- A biological database is a large, organized body of persistent data, usually associated with computerized software designed to update, query, and retrieve components of the data stored within the system.
- The chief objective of the development of a database is to organize data in a set of structured records to enable easy retrieval of information.
- Based on their contents, biological databases can be either primary database or secondary databases.
- Among the two, secondary databases have become a biologist's reference library over the past decade or so, providing a wealth of information on just any research or research product that has been investigated by the research community.

- Sequence annotation information in the primary database is often minimal.
- To turn the raw sequence information into more sophisticated biological knowledge, much post-processing of the sequence information is needed.
- This begs the need for secondary databases, which contain computationally processed sequence information derived from the primary databases.
- Thus, secondary databases comprise data derived from the results of analyzing primary data.
- Secondary databases often draw upon information from numerous sources, including other databases (primary and secondary), controlled vocabularies and the scientific literature.
- They are highly curated, often using a complex combination of computational algorithms and manual analysis and interpretation to derive new knowledge from the public record of science.
- The amount of computational processing work, however, varies greatly among the secondary databases; some are simple archives of translated sequence data from identified open reading frames in DNA, whereas others provide additional annotation and information related to higher levels of information regarding structure and functions.

Importance of secondary databases

- Secondary databases contain information derived from primary sequence data which are in the form of regular expressions (patterns), Fingerprints, profiles blocks or Hidden Markov Models.
- The type of information stored in each of the secondary databases is different. But in secondary databases, homologous sequences may be gathered together in multiple alignments.
- In multiple alignments, there are conserved regions that show little or no variation between the constituent sequences. These conserved regions are called motifs.
- Motifs reflect some vital biological role and are crucial to the structure of the function of the protein. This is the importance of the secondary database.
- So by concentrating on motifs, we can find out the common conserved

regions in the sequences and study the functional and evolutionary details or organisms.

Some of the common secondary databases include:

Prosite

- It was the first secondary database developed.
- Protein families usually contain some most conserved motifs which can be encoded to find out various biological functions.
- So by using such a database tool, we can easily find out the family of proteins when a new sequence is searched. This is the importance of PROSITE.
- Within PROSITE motifs are encoded as a regular expression (called patterns).
- Entries are deposited in PROSITE in two distant files. The first file gives the pattern and lists all matches of pattern, whereas the second one gives the details of family, description of the biological role, etc.
- The process used to derive patterns involves the construction of multiple alignment and manual inspection.
- So PROSITE contains documentation entries describing protein domains, families and functional sites as well as associated patterns and profiles to identify them.
- A set of databases collects together patterns found in protein sequences rather than the complete sequences. PROSITE is one such pattern database.
- The protein motif and pattern are encoded as “regular expressions”.
- The information corresponding to each entry in PROSITE is of the two forms – the patterns and the related descriptive text.

Prints

- Most protein families are characterized by several conserved motifs.
- All of these motifs can be an aid in constructing the `signatures" of different families. This principle is highlighted in constructing PRINT database.
- Within PRINTS motifs are encoded as unweighted local alignments. So small initial multiple alignments are taken to identify conserved motifs.
- Then these regions are searched in the database to find out similarities.
- Results are analyzed to find out the sequences which matched all the

motifs within the fingerprint.

- PROSITE and PRINTS are the only manually annotated secondary databases. The print is a diagnostic collection of protein fingerprints.
- In the PRINTS database, the protein sequence patterns are stored as "fingerprints". Afingerprint is a set of motifs or patterns rather than a single one.
- The information contained in the PRINT entry may be divided into three sections. In addition to entry name, accession number and number of motifs, the first section contains cross-links to other databases that have more information about the characterized family.
- The second section provides a table showing how many of the motifs that make up the fingerprint occurs in the how many of the sequences in that family.
- The last section of the entry contains the actual fingerprints that are stored as multiple aligned sets of sequences, the alignment is made without gaps. There is, therefore, one set of aligned sequences for each motif.

Blocks

- The limitations of the above two databases led to the formation of Block database.
- In this database, the motifs (here called Blocks) are created automatically by highlighting and detecting the most conserved regions of each family of proteins.
- Block databases are fully automated.
- Keyword and sequence searching are the two important features of this type of database.
- Blocks are ungapped Multiple Sequence Alignment representing conserved protein regions.

Pfam

- Pfam contains the profiles used using Hidden Markov models.
- HMMs build the model of the pattern as a series of the match, substitute, insert or delete states, with scores assigned for alignment to go from one state to another.
- Each family or pattern defined in the Pfam consists of the four elements. The first is the annotation, which has the information on the source to

make the entry, the method used and some numbers that serve as figures of merit.

- The second is the seed alignment that is used to bootstrap the rest of the sequences into the multiple alignments and then the family.
- The third is the HMM profile.
- The fourth element is the complete alignment of all the sequences identified in that family.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO AND CHEMICAL ENGINEERING

DEPARTMENT OF BIOTECHNOLOGY

Unit 4 – Introduction to Bioinformatics (Elective) – SBB1609

IV PATHWAY DATABASES

File formats

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

What is a file format?

A *file format* is a way for computers (and humans) to standardize how data is organized. For example, this page was written on an .html extension. HTML files contain special *tags* that tell the browser what each block of text is, and how to display it on the page.

Additionally, computers are able to check file formats and immediately determine whether it should be opened in a text editor (for editing), a modern browser (for viewing) or some other software.

File types can also indicate which algorithm to use to view (or open) that file. For example, .gif, .jpg and .png all display images, but the level of compression, size and resolution differ.

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

- **EMBL** - similar in form to the Genbank file, the EMBL format is used by public databases such as European Molecular Biology Laboratory. The Genbank file format is quite flexible

and allows annotations, comments, and references to be included within the file. The file is plain text and thus can be read with a text editor. Genbank files often have the file extension '.gb' or '.genbank'.

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

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

- **FASTA format**

  File format : FASTA

  File extensions : file.fa, file.fasta,

  file.fsa Example :

  >XR_002086427.1 Candida albicans SC5314 uncharacterized ncRNA (SCR1), ncRNA
  TGGCTGTGATGGCTTTTAGCGGAAGCGCGCTGTTCGCGTACCTGCTGTTTGTTGAA
  AATTTAAGAGCAAAGTGTCCGGCTCGATCCCTGCGAATTGAATTCTGAACGCTAG
  AGTAATCAGTGTCTTTCAAGTTCTGGTAATGTTTAGCATAACCACTGGAGGGAAG
  CAATTCAGCACAGTAATGCTAATCGTGGTGGAGGCGAATCCGGATGGCACCTTGT
  TTGTTGATAAATAGTGCGGTATCTAGTGTTGCAACTCTATTTTT

  Fasta format is a simple way of representing nucleotide or amino acid sequences of nucleic acids and proteins. This is a very basic format with two minimum lines. First line referred as comment line starts with '>' and gives basic information about sequence. There is no set format for comment line. Any other line that starts with ';' will be ignored. Lines with ';' are not a common feature of fasta files. After comment line, sequence of nucleic acid or protein is included in standard one letter code. Any tabulators, spaces, asterisks etc in

sequence will be ignored.

Plain sequence format

A sequence in plain format may contain only IUPAC characters and spaces (no numbers!).

Note: A file in plain sequence format may only contain **one** sequence, while most other formats accept several sequences in one file.

An example sequence in plain format is:

```
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC
CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGAAAAGCAGC
CTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCCCCTCATAGGAGAGG
AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGCGCCGGGACAGAATGCC
CTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCACGCAAG
TTTAATTACAGACCTGAA
```

FASTQ format

A sequence file in FASTQ format can contain several sequences.
FASTQ is a text-based format for storing both a biological sequence (usually nucleotide sequence) and its corresponding quality scores. It is mainly used for storing the output of high-throughput sequencing instruments.
A FASTQ file usually uses four lines per sequence.

1. a '@' character, followed by a sequence identifier and an optional description
2. the raw sequence letters.
3. a '+' character, optionally followed by the same sequence identifier (and any description)
4. quality values for the sequence in Line 2

An example sequence in FASTQ format is:

@SEQUENCE_ID
GTGGAAGTTCTTAGGGCATGGC

AAAGAGTCAGAATTTGAC

+

FAFFADEDGDBGEGGB

CGGHE>EEBA@@=

EMBL format

A sequence file in EMBL format can contain several sequences.
One sequence entry starts with an identifier line ("ID"), followed by further annotation lines. The start of the sequence is marked by a line starting with "SQ" and the end of the sequence is marked by two slashes ("//").

```
ID   AB000263 standard; RNA; PRI; 368 BP.
XX
AC   AB000263;
XX
DE   Homo sapiens mRNA for prepro cortistatin like peptide, complete cds.
XX
SQ   Sequence 368 BP;
     acaagatgcc attgtccccc ggcctcctgc tgctgctgct ctccggggcc acggccaccg        60
     ctgccctgcc cctggagggt ggccccaccg gccgagacag cgagcatatg caggaagcgg       120
     caggaataag gaaaagcagc ctcctgactt tcctcgcttg gtggtttgag tggacctccc       180
     aggccagtgc cgggcccctc ataggagagg aagctcggga ggtggccagg cggcaggaag       240
     gcgcaccccc ccagcaatcc gcgcgccggg acagaatgcc ctgcaggaac ttcttctgga       300
     agaccttctc ctcctgcaaa taaaacctca cccatgaatg ctcacgcaag tttaattaca       360
     gacctgaa                                                               368
//
```

FASTA format

A sequence file in FASTA format can contain several sequences.
Each sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line must begin with a greater-than (">") symbol in the first column.

An example sequence in FASTA format is:

```
>AB000263 |acc=AB000263|descr=Homo sapiens mRNA for prepro cortistatin
like peptide, complete cds.|len=368
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC
CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGAAAAGCAGC
CTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCCCCTCATAGGAGAGG
AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGCGCCGGGACAGAATGCC
CTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCACGCAAG
TTTAATTACAGACCTGAA
```

GCG format

A sequence file in GCG format contains exactly one sequence, begins with annotation lines and the start of the sequence is marked by a line ending with two dot ("..") characters. This line also contains the sequence identifier, the sequence length and a checksum. This format should only be used if the file was created with the GCG package.

An example sequence in GCG format is:

An example sequence in GCG format is:

```
ID   AB000263 standard; RNA; PRI; 368 BP.
XX
AC   AB000263;
XX
DE   Homo sapiens mRNA for prepro cortistatin like peptide, complete cds.
XX
SQ   Sequence 368 BP;
AB000263  Length: 368  Check: 4514 ..
        1  acaagatgcc attgtccccc ggcctcctgc tgctgctgct ctccggggcc acggccaccg
       61  ctgccctgcc cctggagggt ggccccaccg gccgagacag cgagcatatg caggaagcgg
      121  caggaataag gaaaagcagc ctcctgactt tcctcgcttg gtggtttgag tggacctccc
      181  aggccagtgc cgggcccctc ataggagagg aagctcggga ggtggccagg cggcaggaag
      241  gcgcaccccc ccagcaatcc gcgcgccggg acagaatgcc ctgcaggaac ttcttctgga
      301  agaccttctc ctcctgcaaa taaaacctca cccatgaatg ctcacgcaag tttaattaca
      361  gacctgaa
```

GCG-RSF (rich sequence format)

The new GCG-RSF can contain several sequences in one file. This format should only be used if the file was created with the GCG package.

GenBank format

A sequence file in GenBank format can contain several sequences.

One sequence in GenBank format starts with a line containing the word LOCUS and a number of annotation lines. The start of the sequence is marked by a line containing "ORIGIN" and the end of the sequence is marked by two slashes ("//").

```
LOCUS       AB000263                368 bp    mRNA    linear   PRI 05-FEB-1999
DEFINITION  Homo sapiens mRNA for prepro cortistatin like peptide, complete
            cds.
ACCESSION   AB000263
ORIGIN
        1 acaagatgcc attgtccccc ggcctcctgc tgctgctgct ctccggggcc acggccaccg
       61 ctgccctgcc cctggagggt ggccccaccg gccgagacag cgagcatatg caggaagcgg
      121 caggaataag gaaaagcagc ctcctgactt tcctcgcttg gtggtttgag tggacctccc
      181 aggccagtgc cgggcccctc ataggagagg aagctcggga ggtggccagg cggcaggaag
      241 gcgcacccCC ccagcaatcc gcgcgccggg acagaatgcc ctgcaggaac ttcttctgga
      301 agaccttctc ctcctgcaaa taaaacctca cccatgaatg ctcacgcaag tttaattaca
      361 gacctgaa
//
```

IG format

A sequence file in IG format can contain several sequences, each consisting of a number of comment lines that must begin with a semicolon (";"), a line with the sequence name (it may not contain spaces!) and the sequence itself terminated with the termination character '1' for linear or '2' for circular sequences.

```
; comment
; comment
AB000263
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC
CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGAAAAGCAGC
CTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCCCCTCATAGGAGAGG
AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGCGCCGGGACAGAATGCC
CTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCACGCAAG
TTTAATTACAGACCTGAA1
```

Carbohydrate Databases

Carbohydrate Structure Database (CSDB) is a free database and service platform in glycoinformatics, launched in 2005 by a group of Russian scientists from N.D. Zelinsky Institute of Organic Chemistry, Russian Academy of Sciences. CSDB stores published structural, taxonomical, bibliographic and NMR-spectroscopic data on natural carbohydrates and carbohydrate-related molecules.

Overview

The main data stored in CSDB are carbohydrate structures of bacterial, fungal, and plant origin. Each structure is assigned to an organism and is provided with the link(s) to the corresponding scientific publication(s), in which it was described. Apart from structural data, CSDB also stores NMR spectra, information on methods used to decipher a particular structure, and some other data. CSDB provides access to several carbohydrate-related research tools:

Simulation of 1D and 2D NMR spectra of carbohydrates (GODESS: glycan-oriented dual empirical spectrum simulation). Automated NMR-based structure elucidation (GRASS: generation, ranking and assignment of saccharide structures). Statistical analysis of structural feature distribution in glycomes of living organisms. Generation of optimized atomic coordinates for an arbitrary saccharide, Taxon clustering based on similarities of glycomes (carbohydrate-based tree of life), Glycosyltransferase subdatabase (GT-explorer)

History and funding

Until 2015, Bacterial Carbohydrate Structure Database (BCSDB) and Plant&Fungal Carbohydrate Structure Database (PFCSDB) databases existed in parallel. In 2015, they were joined into the single Carbohydrate Structure Database (CSDB). The development and maintenance of CSDB have been funded by International Science and Technology Center (2005-2007), Russian Federation President grant program (2005-2006), Russian Foundation for Basic Research (2005-2007,2012-2014,2015-2017), and Deutsches Krebsforschungszentrum (short-term in 2006-2010).

Data sources and coverage

The main sources of CSDB data are:

Scientific publications indexed in the dedicated citation databases, including NCBI Pubmed and Thomson Reuters Web of Science (approx. 14000 records).
CCSD (Carbbank) database (approx. 3000 records).

The data are selected and added to CSDB manually by browsing original scientific publications. The data originating from other databases are subject to error-correction and approval procedures. As of the beginning of 2017, the coverage on bacteria and archaea is ca. 80% of carbohydrate structures published in scientific literature in the years 1943 - 2015. The time lag between the publication of relative data and their deposition into CSDB is about 18 months. Plants are covered up to 1997, and fungi up to 2005. CSDB does not cover data from the animalia domain, except unicellular metazoa. There is a number of dedicated databases on animal carbohydrates, e.g. UniCarbKB or GLYCOSCIENCES.de.

CSDB is reported as one of the biggest projects in glycoinformatics. It is employed in structural studies of natural carbohydrates and in glyco-profiling. The content of CSDB has been used as a data source in other glycoinformatics projects.

Interrelation with other databases

CSDB is cross-linked to other glycomics databases, such as MonosaccharideDB, Glycosciences.DE, NCBI Pubmed, NCBI Taxonomy, NLM catalog, etc. Structures are presented in multiple carbohydrate notations (SNFG, SweetDB, GlycoCT, WURCS, GLYCAM, etc.). CSDB is exportable as a Resource Description Framework (RDF) feed according to the GlycoRDF ontology.

ENZYME DATABASES

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

General Enzyme Databases

Enzyme Nomenclature Web Sites

The classification of enzymes according to the rules of enzyme nomenclature is the responsibility of the Enzyme Commission of the International Union for Biochemistry and Molecular Biology (IUBMB). The outcome of the decisions made by the commission is deposited in the enzyme list, which is made accessible by several Web sites (IUBMB website, ExplorEnz, SIB-ENZYME, IntEnz). They provide forms for searching the enzyme's accepted name, the systematic name, some synonyms, the reaction, cofactors, and literature references.

IUBMB Nomenclature and ExplorEnz

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

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

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

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

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

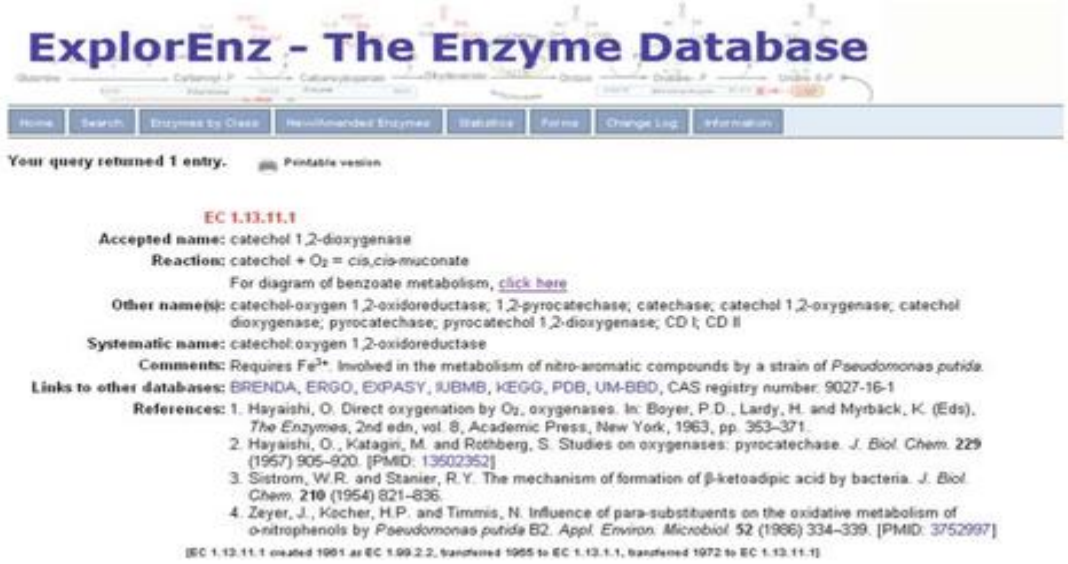


Figure 1

Figure 2

SIB-ENZYME Nomenclature Database

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

A special feature is the links for the protein sequences, which are deposited in UniProt enabling a direct access to individual enzyme proteins

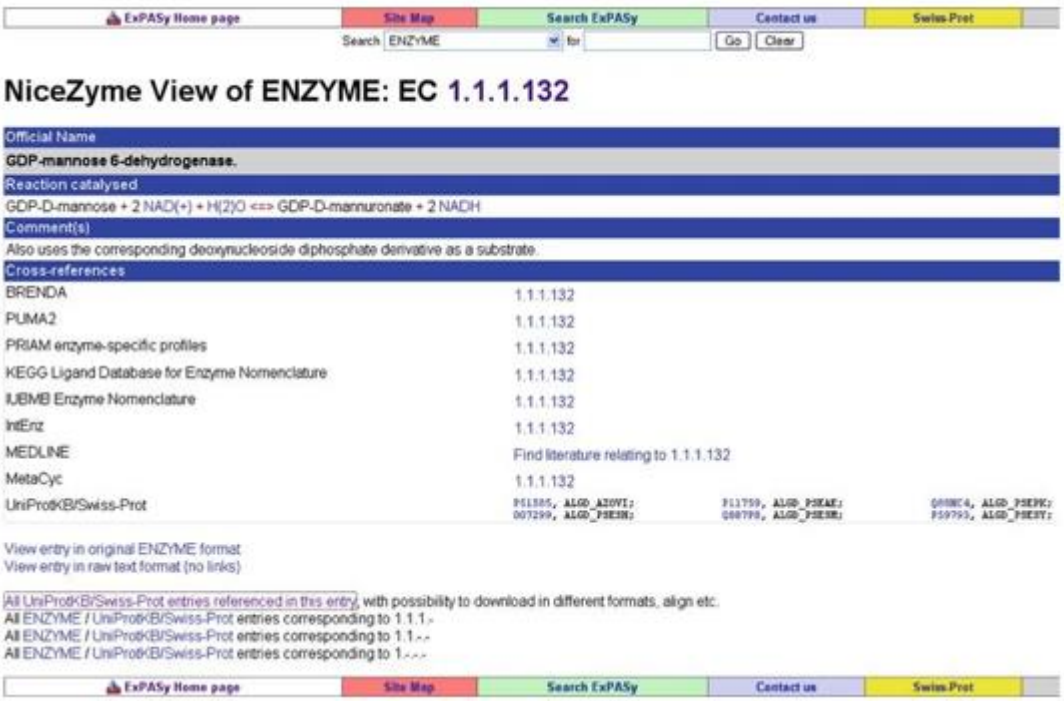


Figure 3

IntEnz

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

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

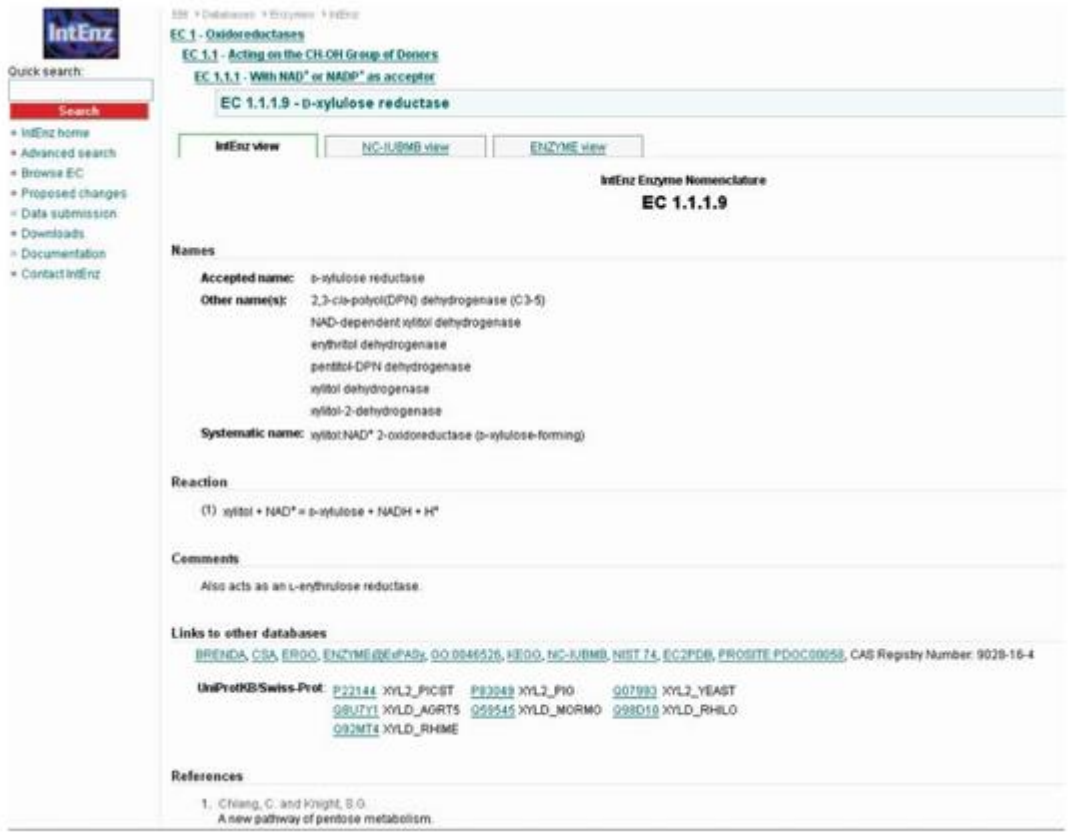


Figure 4

Enzyme-Functional Databases

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

BRENDA

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

Classification and nomenclature

Reaction and specificity

Functional parameters

Organism-related information

Enzyme structure

Isolation and preparation

Literature references

Application and engineering

Enzyme–disease relationships

The section on Classification and Nomenclature is based on the enzyme names as defined

~

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

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

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

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

Quick search can be used for a direct search in one of the 54 data fields providing a fast and direct access, e.g., via enzyme names or metabolites.

Fulltext search performs a search in all sections of the database, including commentaries.

Advanced search allows a combinatorial search for text or numerical data fields.

Figure 5

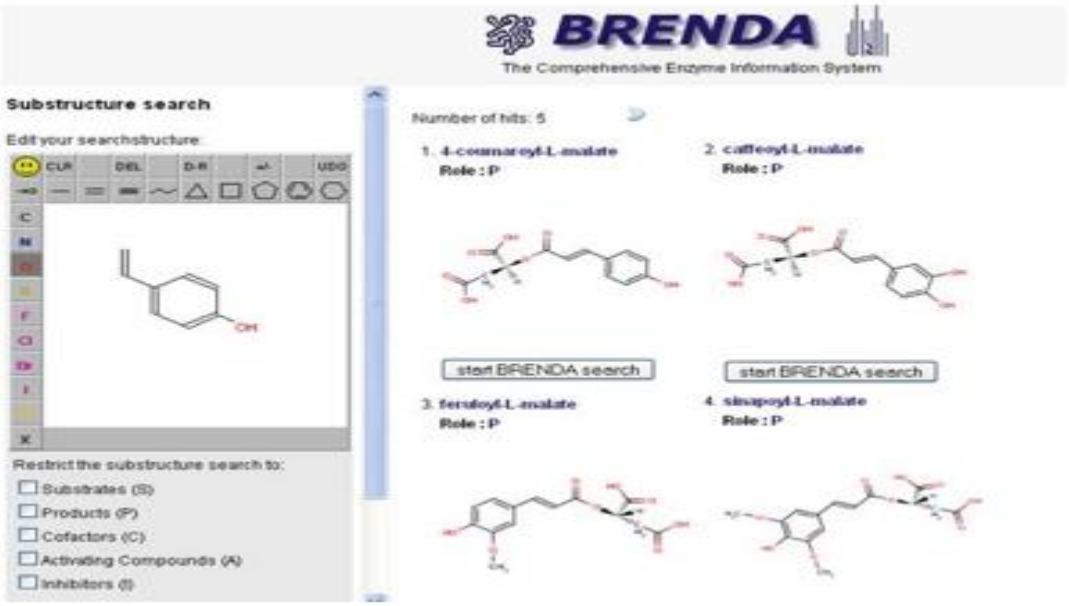


Figure 6

Substructure search is a tool for drawing a molecule which then is searched in the database. The results are exact matches or any molecule containing the plotted structure.

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

EC explorer can be used to browse or search the hierarchical tree of enzymes.

Sequence search is useful for enzymes with a known protein sequence.

Genome explorer connects enzymes to genome sequences. The location of classified enzymes is displayed in their genomic context.

Ontology explorer allows to simultaneously search in all bio- chemically relevant ontologies, among them Brenda Tissue Ontology (BTO).

About 1.4 million functional and property data describing enzymes are stored in the database covering 50 datafields. All data in BRENDA are linked to the original paper reference.

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

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

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

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

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

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

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

Different isoenzymes in different tissues may be found. Some- times enzymes restricted to

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

AMENDA/FRENDA

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

KEGG

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

Compound
Drug
Glycan
Reaction
Repair
Enzyme

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

Enzyme-catalyzed reactions are stored in the KEGG REACTION database containing all reactions from KEGG ENZYME and additional reactions from the KEGG metabolic

pathways, the latter without an EC classification. Each reaction is identified by the R number, such as R06466 for the iso- merization of (S)-2,3-epoxysqualene to lupeol. Reactions are linked to ortholog groups of enzymes as defined by the KEGG ORTHOLOGY database, enabling integrated analysis of genomic (enzyme genes) and chemical (compound pairs) information. Figure shows the entry for mannitol dehydrogenase as an example.

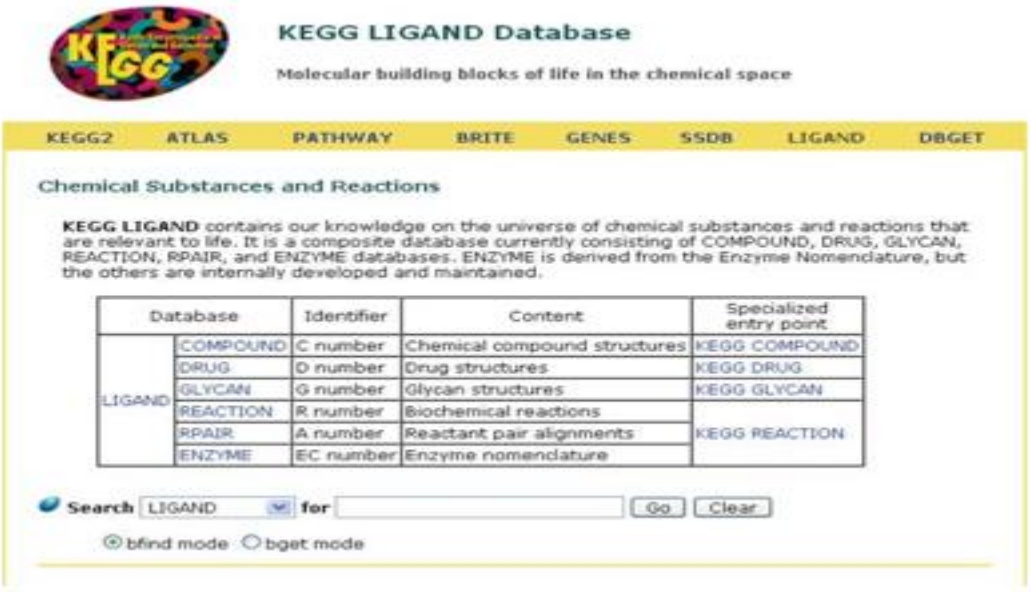


Figure 7

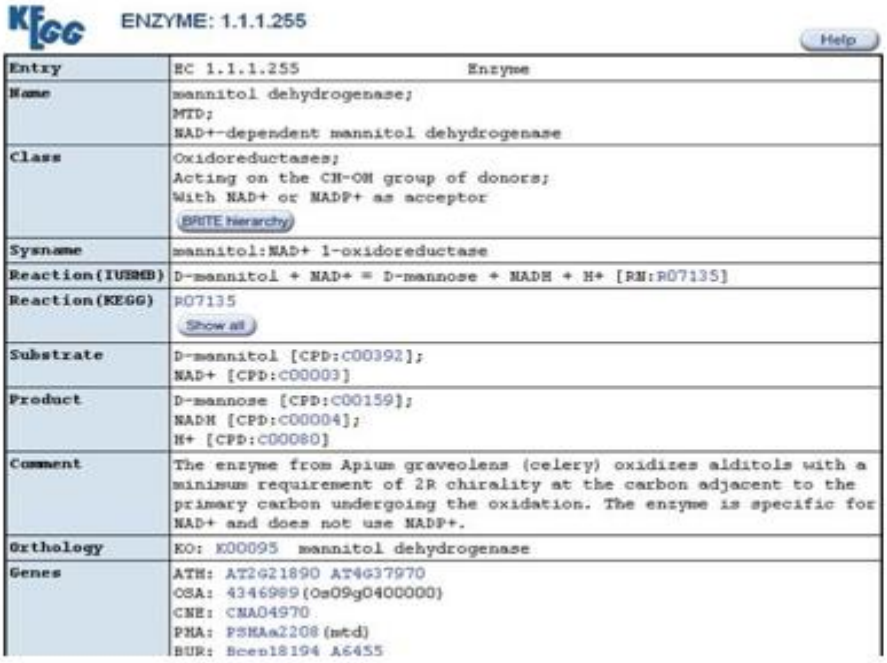


Figure 8

Special Enzyme Databases

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

MEROPS

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

The substrate specificity is described in two ways:

For any peptidase with more than ten known cleavages, a display is presented that gives an indication of the amino acids preferred at its substrate binding sites. This display uses the WebLogo software Details of the amino acid sequences around the cleavage sites are displayed in the ‘‘Spe- cificity Matrix.’’

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

Artificial or model substrates are summarized in text-sheets, including literature references.

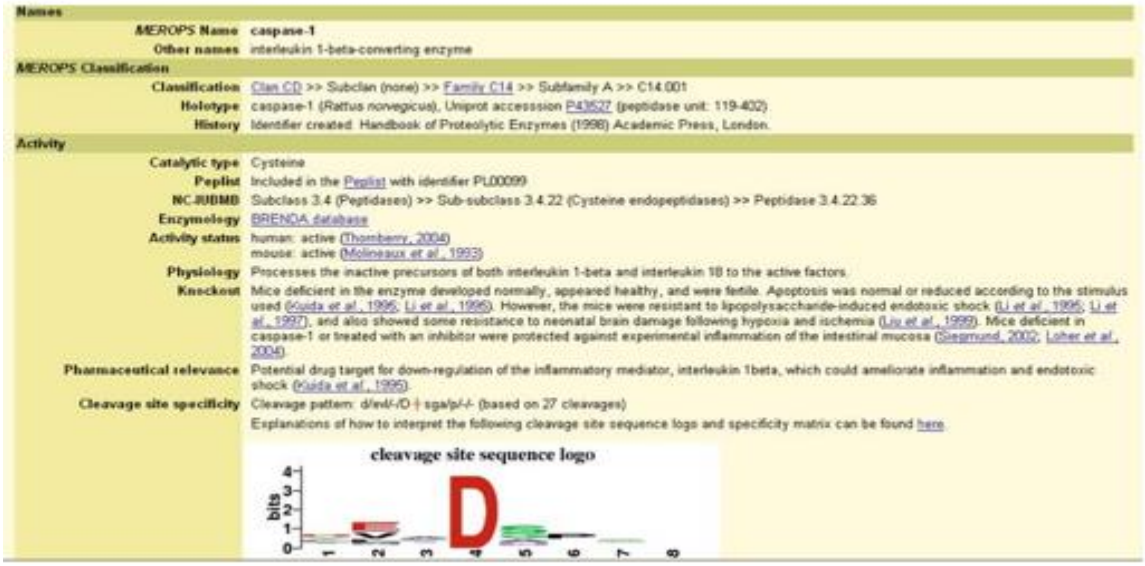


Figure 9

MetaCyc

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

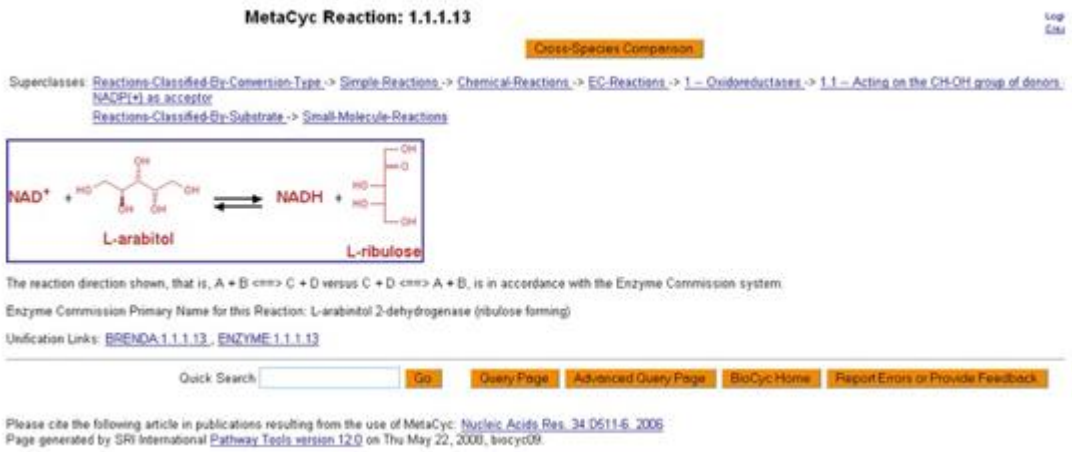


Figure 10

REBASE

REBASE is a comprehensive database of information about restric- tion enzymes, DNA methyltransferases, and related proteins involved in the biological process of restriction-modification. It contains fully referenced information about recognition and cleavage sites, isoschizomers, neoschizomers, commercial availability, methylation sensitivity, crystal and sequence data. Experimentally characterized homing endonucleases are also included. All newly sequenced genomes are analyzed for the presence of putative restriction systems and these data are included within the REBASE. The contents or REBASE may be browsed from the Web (http://rebase.neb.com/rebase/rebase. ftp.html) and selected compilations can be downloaded by ftp.

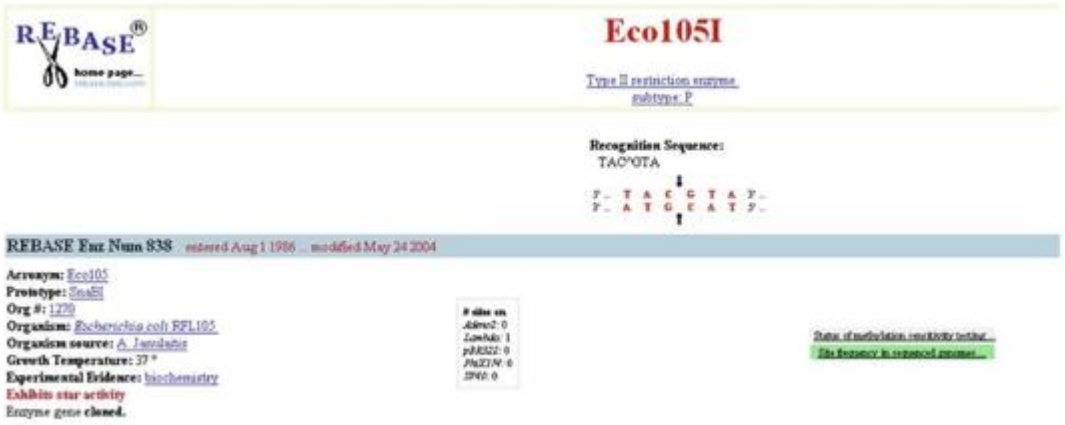


Figure 11

CARBOHYDRATE-ACTIVE Enzymes (CAzy)

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

The biosynthesis of disaccharides, oligosaccharides, and polysaccharides involves the action of hundreds of different glycosyl- transferases (EC 2.4.x.y), enzymes which

catalyze the transfer of sugar moieties from activated donor molecules to specific acceptor molecules, forming glycosidic bonds. In similar manners, classifications for polysaccharide lyases and carbohydrate esterases are presented.

Because there is a direct relationship between sequence and folding similarities, these classifications reflect the structural features of these enzymes better than their sole substrate specificity help to reveal the evolutionary relationships between these enzymes provide a convenient tool to derive mechanistic information.

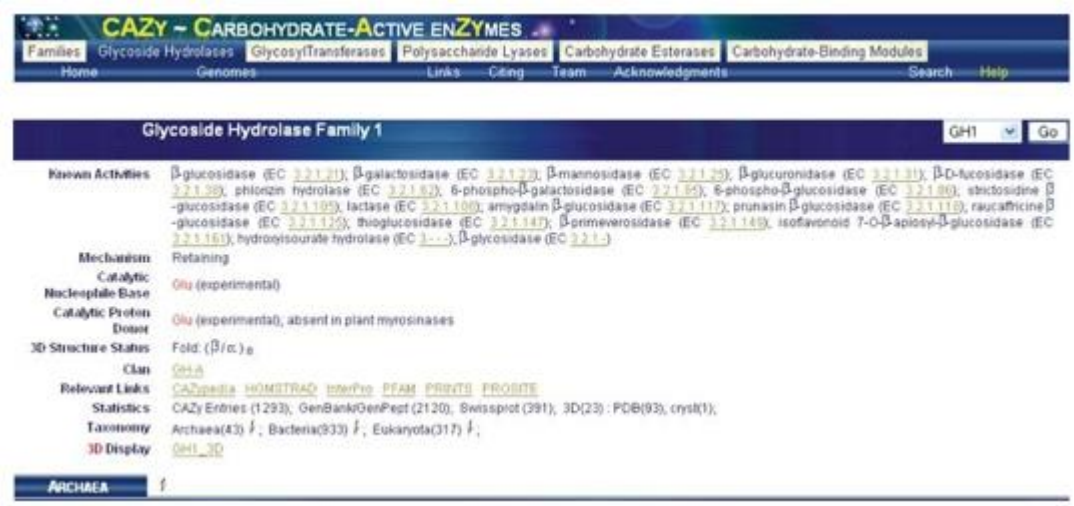


Figure 12

Databases Based on Sequence Homologies

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

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

PeroxiBase is curated in collaboration with the Swiss Institute of Bioinformatics (SIB). The goal of this peroxidase database is to centralize most of the peroxidase superfamilies encoding sequences, to follow the evolution of peroxidase among living organism and compile the information concerning putative functions and transcription regulation (http://peroxibase.isb-sib.ch/ index.php).

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

PATHWAY DATABASE - KEGG

KEGG (Kyoto Encyclopedia of Genes and Genomes) is a collection of databases dealing with genomes, biological pathways, diseases, drugs, and chemical substances. KEGG is utilized for bioinformatics research and education, including data analysis in genomics, metagenomics, metabolomics and other omics studies, modeling and simulation in systems biology, and translational research in drug development.

Introduction

The KEGG database project was initiated in 1995 by Minoru Kanehisa, Professor at the Institute for Chemical Research, Kyoto University, under the then ongoing Japanese Human Genome Program. Foreseeing the need for a computerized resource that can be used for biological interpretation of genome sequence data, he started developing the KEGG PATHWAY database. It is a collection of manually drawn KEGG pathway maps representing experimental knowledge on metabolism and various other functions of the cell and the organism. Each pathway map contains a network of molecular interactions and reactions and is designed to link genes in the genome to gene products (mostly proteins) in the pathway. This has enabled the analysis called KEGG pathway mapping, whereby the gene content in the genome is compared with the KEGG PATHWAY database to examine which pathways and associated functions are likely to be encoded in the genome.

According to the developers, KEGG is a "computer representation" of the biological system. It integrates building blocks and wiring diagrams of the system — more specifically, genetic building blocks of genes and proteins, chemical building blocks of small molecules and reactions, and wiring diagrams of molecular interaction and reaction networks. This concept is realized in the following databases of KEGG, which are categorized into systems, genomic, chemical, and health information.

Systems information
PATHWAY — pathway maps for cellular and organismal functions
MODULE — modules or functional units of genes
BRITE — hierarchical classifications of biological entities
Genomic information
GENOME — complete genomes
GENES — genes and proteins in the complete genomes

ORTHOLOGY — ortholog groups of genes in the complete genomes
Chemical information
COMPOUND, GLYCAN — chemical compounds and glycans
REACTION, RPAIR, RCLASS — chemical reactions
ENZYME — enzyme nomenclature
Health information
DISEASE — human diseases
DRUG — approved drugs
ENVIRON — crude drugs and health-related substances
Databases
Systems information

The KEGG PATHWAY database, the wiring diagram database, is the core of the KEGG resource. It is a collection of pathway maps integrating many entities including genes, proteins, RNAs, chemical compounds, glycans, and chemical reactions, as well as disease genes and drug targets, which are stored as individual entries in the other databases of KEGG. The pathway maps are classified into the following sections:

Metabolism
Genetic information processing (transcription, translation, replication and repair, etc.)
Environmental information processing (membrane transport, signal transduction, etc.)
Cellular processes (cell growth, cell death, cell membrane functions, etc.)
Organismal systems (immune system, endocrine system, nervous system, etc.)
Human diseases

Drug development
The metabolism section contains aesthetically drawn global maps showing an overall picture of metabolism, in addition to regular metabolic pathway maps. The low-resolution global maps can be used, for example, to compare metabolic capacities of different organisms in genomics studies and different environmental samples in metagenomics studies. In contrast, KEGG modules in the KEGG MODULE database are higher-resolution, localized wiring diagrams, representing tighter functional units within a pathway map, such as subpathways conserved among specific organism groups and molecular complexes. KEGG modules are defined as characteristic gene sets that can be linked to specific metabolic capacities and other phenotypic

features, so that they can be used for automatic interpretation of genome and metagenome data.

Another database that supplements KEGG PATHWAY is the KEGG BRITE database. It is an ontology database containing hierarchical classifications of various entities including genes, proteins, organisms, diseases, drugs, and chemical compounds. While KEGG PATHWAY is limited to molecular interactions and reactions of these entities, KEGG BRITE incorporates many different types of relationships.

Genomic information

Several months after the KEGG project was initiated in 1995, the first report of the completely sequenced bacterial genome was published. Since then all published complete genomes are accumulated in KEGG for both eukaryotes and prokaryotes. The KEGG GENES database contains gene/protein-level information and the KEGG GENOME database contains organism-level information for these genomes. The KEGG GENES database consists of gene sets for the complete genomes, and genes in each set are given annotations in the form of establishing correspondences to the wiring diagrams of KEGG pathway maps, KEGG modules, and BRITE hierarchies.

These correspondences are made using the concept of orthologs. The KEGG pathway maps are drawn based on experimental evidence in specific organisms but they are designed to be applicable to other organisms as well, because different organisms, such as human and mouse, often share identical pathways consisting of functionally identical genes, called orthologous genes or orthologs. All the genes in the KEGG GENES database are being grouped into such orthologs in the KEGG ORTHOLOGY (KO) database. Because the nodes (gene products) of KEGG pathway maps, as well as KEGG modules and BRITE hierarchies, are given KO identifiers, the correspondences are established once genes in the genome are annotated with KO identifiers by the genome annotation procedure in KEGG.

Chemical information

The KEGG metabolic pathway maps are drawn to represent the dual aspects of the metabolic network: the genomic network of how genome-encoded enzymes are connected to catalyze consecutive reactions and the chemical network of how chemical structures of substrates and products are transformed by these reactions. A set of enzyme genes in the genome will identify enzyme relation networks when superimposed on the KEGG pathway maps, which in turn

characterize chemical structure transformation networks allowing interpretation of biosynthetic and biodegradation potentials of the organism. Alternatively, a set of metabolites identified in the metabolome will lead to the understanding of enzymatic pathways and enzyme genes involved.

The databases in the chemical information category, which are collectively called KEGG LIGAND, are organized by capturing knowledge of the chemical network. In the beginning of the KEGG project, KEGG LIGAND consisted of three databases: KEGG COMPOUND for chemical compounds, KEGG REACTION for chemical reactions, and KEGG ENZYME for reactions in the enzyme nomenclature. Currently, there are additional databases: KEGG GLYCAN for glycansand two auxiliary reaction databases called RPAIR (reactant pair alignments) and RCLASS (reaction class). KEGG COMPOUND has also been expanded to contain various compounds such as xenobiotics, in addition to metabolites.

Health information

In KEGG, diseases are viewed as perturbed states of the biological system caused by perturbants of genetic factors and environmental factors, and drugs are viewed as different types of perturbants. The KEGG PATHWAY database includes not only the normal states but also the perturbed states of the biological systems. However, disease pathway maps cannot be drawn for most diseases because molecular mechanisms are not well understood. An alternative approach is taken in the KEGG DISEASE database, which simply catalogs known genetic factors and environmental factors of diseases. These catalogs may eventually lead to more complete wiring diagrams of diseases.

The KEGG DRUG database contains active ingredients of approved drugs in Japan, the USA, and Europe. They are distinguished by chemical structures and/or chemical components and associated with target molecules, metabolizing enzymes, and other molecular interaction network information in the KEGG pathway maps and the BRITE hierarchies. This enables an integrated analysis of drug interactions with genomic information. Crude drugs and other health-related substances, which are outside the category of approved drugs, are stored in the KEGG ENVIRON database. The databases in the health information category are collectively called KEGG MEDICUS, which also includes package inserts of all marketed drugs in Japan.

RDBMS

● RDBMS stands for Relational Database Management System.

RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

● A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd.

● The data in RDBMS is stored in database objects called tables.

● DBMS applications store data as file.

● DBMS does not support client/server architecture.

● DBMS does not allow normalization.

● DBMS does not impose integrity constraints.

● RDBMS applications store data in a tabular form.

● RDBMS supports client/server architecture.

● RDBMS allows normalization.

● RDBMS imposes integrity constraints.

● The data in RDBMS is stored in database objects called tables. The table is a collection of related data entries and it consists of columns and rows.

● Remember, a table is the most common and simplest form of data storage in a relational database.

● Every table is broken up into smaller entities called fields. The fields in the CUSTOMERS table consist of ID, NAME, AGE, ADDRESS and SALARY.

● A record, also called a row of data, is each individual entry that exists in a table. For example there are 7 records in the above CUSTOMERS table.

● A column is a vertical entity in a table that contains all information associated with a specific field in a table.

● A NULL value in a table is a value in a field that appears to be blank, which means a field with a NULL value is a field with no value.

SQL - Structured Query Language

SQL (Structured Query Language) is a computer language aimed to store, manipulate, and retrieve data stored in relational databases. IBM implemented the language, originally called Sequel, as part of the System R project in the early 1970s. The first commercial relational database was released by Relational Software later becoming oracle.

SQL language has several parts:

Data-definition language (DDL)- provides commands for defining relation schemas, deleting relations, and modifying relation schemas.

Interactive data-manipulation language (DML). It includes also commands to insert tuples into, delete tuples from, and modify tuples in the database. View definition-includes commands for defining views. Transaction control-includes commands for specifying the beginning and ending of transactions.

Embedded SQL and dynamic SQL- define how SQL statements can be embedded within general-purpose programming languages, such as C, C++, Java, PL/I, Cobol, Pascal, and Fortran.

Integrity- The SQL DDL includes commands for specifying integrity constraints that the data stored in the database must satisfy. Updates that violate integrity constraints are disallowed.

Authorization (DCL)-The SQL DDL includes commands for specifying access rights to relations and views.

DDL DDL - Data Definition Language:

Statements used to define the database structure or schema.

Some examples: CREATE - to create objects in the database

ALTER - alters the structure of the database

DROP - delete objects from the database

RENAME - rename an object

Schema in SQL

Create table : Example : create table branch (branch-name char(15), branch-city char(30),assets integer, primary key (branch-name), check (assets >= 0)) □ Delete : Delete table : drop table r Delete tuples : Delete from r □ Alter table : Add Attribute : alter table r add AD Drop attribute : alter table r drop A

Data Manipulation Language DML- Data Manipulation Language: Statements used for managing data within schema objects.

- SELECT - retrieve data from the a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - deletes all records from a table, the space for the records remain
- CALL - call a PL/SQL or Java subprogram

Example Setting Schema & Attributes

- Branch-schema = (branch-name, branch-city, assets)
- Customer-schema = (customer-name, customer- street, customer-city)
- Loan-schema = (loan-number, branch-name, amount)
- Borrower-schema = (customer-name, loan-number)
- Account-schema = (account-number, branch-name, balance)
- Depositor-schema = (customer-name, account- number)

- Basic SQL Query
  - Select Clause: 1. select branch-name from loan Retain Duplicates 2. select distinct branch-name from loan Remove duplicates 3. select all branch-name from loan Retain Duplicates
  - Where Clause : 1. select loan-number from loan where branch-name = 'Perryridge' and amount > 1200
  - From Clause : select customer-name, borrower.loan-number, amount from borrower, loan where borrower.loan-number = loan.loan-number
- Basic SQL Query
  - Rename Operation : select customer-name, borrower.loan-number as loan-id, amount from borrower, loan where borrower.loan- number = loan.loan-number
  - Tuple variables : select customer-name, T.loan-number, S.amount from borrower as T, loan as S where T.loan-number = S.loan- number Tuple variables are most useful for comparing two tuples in the same relation.
  - String Operations : select customer-name from customer where customer- street like '%Main%'
- Basic SQL Query
  - Ordering the Display of Tuples : select distinct customer-name from borrower, loan where borrower.loan-number = loan.loan-number and branch- name = 'Perryridge' order by customer-name select * from loan order by amount desc, loan-number asc
- Set Operations
  - Union Operation : find all customers having a loan, an account, or both (select customer-name from depositor) union (select customer-name from borrower) Removes Duplicates
  - Intersect Operation : find all customers who have both a loan and an account (select distinct customer-name from depositor) intersect (select distinct

customer-name from borrower) Removes Duplicates

- Except Operation : find all customers who have an account but no loan (select distinct customer-name from depositor) except (select customer-name from borrower)

Aggregate Functions avg, min, max, sum, count Example : “Find the average balance for each customer who lives in Harrison and has at least three accounts.” select depositor.customer-name, avg (balance) from depositor, account, customer where depositor.account-number = account.account- number and depositor.customer-name = customer.customer-name and customer-city = ’Harrison’ group by depositor.customer-name having count (distinct depositor.account-number) >= 3

Nested Subqueries

- Set Membership : Example : find those customers who are borrowers from the bank and who appear in the list of account holders select distinct customer-name from borrower where customer-name in (select customer-name from depositor)
- Set Comparison : Example : Find the names of all branches that have assets greater than those of at least one branch located in Brooklyn. select branch-name from branch where assets > some (select assets from branch where branch-city = ’Brooklyn’) >some : greater than at least one member

Views - Find for each branch the sum of the amounts of all the loans at the branch. create view branch-total-loan(branch-name, total-loan) as select branch-name, sum(amount) from loan group by branch-name

Complex Queries

- Derived Relations : Example : “find the maximum across all branches of the total balance at each branch.” select max(tot-balance) from (select branch-name, sum(balance) from account group by branch-name) as branch-total (branch-name, tot-balance)
- with Clause : Example : find accounts with the maximum balance; if there are many accounts with the same maximum balance, all of them are selected. with max-balance (value) as select max(balance) from account select account-number from account, max-balance where account.balance = max-balance.value

- Modification of the Database
  - Deletion : delete from account where branch-name = ’Perryridge’
  - Insertion : insert into account (account-number, branch-name, balance) values

('A-9732', 'Perryridge', 1200)

- Updates : Example : "Pay 5 percent interest on accounts whose balance is greater than average" update account set balance = balance * 1.05 where balance > select avg (balance) from account

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF BIO AND CHEMICAL ENGINEERING

DEPARTMENT OF BIOTECHNOLOGY

Unit 5 – Introduction to Bioinformatics (Elective) – SBB1609

V – APPLICATION OF BIOINFORMATICS

GENE PREDICTION

Gene prediction by computational methods for finding the location of protein coding regions is one of the essential issues in bioinformatics.

Gene prediction basically means locating genes along a genome. Also called gene finding, it refers to the process of identifying the regions of genomic DNA that encode genes.

This includes protein coding genes, RNA genes and other functional elements such as the regulatory genes.

Importance of Gene Prediction

Helps to annotate large, contiguous sequences

Aids in the identification of fundamental and essential elements of genome such as functional genes, intron, exon, splicing sites, regulatory sites, gene encoding known proteins, motifs, EST, ACR, etc.

Distinguish between coding and non-coding regions of a genome

Predict complete exon – intron structures of protein coding regions

Describe individual genes in terms of their function

It has vast application in structural genomics, functional genomics, metabolomics, transcriptomics, proteomics, genome studies and other genetic related studies including genetics disorders detection, treatment and prevention.

Bioinformatics and the Prediction of Genes

With databases of human and model organism DNA sequences increasing quickly with time, it has become almost impossible to carry out the conventional painstaking experimentation on living cells and organisms to predict genes.

Formerly, statistical analysis of the rates of homologous recombination of several different genes could determine their order on a certain chromosome, and information from many such experiments could be combined to create a genetic map specifying the rough location of known genes relative to each other.

However, today, the frontiers of bioinformatics research are making it increasingly possible to predict the function of such a deluge of genes based on its sequence alone.

Methods of Gene Prediction

Two classes of methods are generally adopted:

A. Similarity based searches

It is a method based on sequence similarity searches.

It is a conceptually simple approach that is based on finding similarity in gene sequences between ESTs (expressed sequence tags), proteins, or other genomes to the input genome. This approach is based on the assumption that functional regions (exons) are more conserved evolutionarily than nonfunctional regions (intergenic or intronic regions).

Once there is similarity between a certain genomic region and an EST, DNA, or protein, the similarity information can be used to infer gene structure or function of that region.

Local alignment and global alignment are two methods based on similarity searches. The most common local alignment tool is the BLAST family of programs, which detects sequence similarity to known genes, proteins, or ESTs.

Two more types of software, PROCRUSTES and GeneWise, use global alignment of a homologous protein to translated ORFs in a genomic sequence for gene prediction.

A new heuristic method based on pairwise genome comparison has been implemented in the software called CSTfinder.

B. Ab- initio prediction

It is a method based on gene structure and signal-based searches.

It uses gene structure as a template to detect genes

Ab initio gene predictions rely on two types of sequence information: signal sensors and content sensors.

Signal sensors refer to short sequence motifs, such as splice sites, branch points, polypyrimidine tracts, start codons and stop codons.

On the other hand content sensors refer to the patterns of codon usage that are unique to a species, and allow coding sequences to be distinguished from the surrounding non-coding sequences by statistical detection algorithms. Exon detection must rely on the content sensors.

The search by this method thus relies on the major feature present in the genes.

Many algorithms are applied for modeling gene structure, such as Dynamic Programming, linear discriminant analysis, Linguist methods, Hidden Markov Model and Neural Network.

Based on these models, a great number of ab initio gene prediction programs have been developed. Some of the frequently used ones are GeneID, FGENESH, GeneParser, GlimmerM, GENSCAN etc.

DRUG DESIGNING

"In Silico" is an expression used to mean "performed on computer or via computer simulation."

In Silico drug designing is thus the identification of the drug target molecule by employing bioinformatics tools.

The inventive process of finding new medications based on the knowledge of a biological target is called as drug designing. It can be accomplished in two ways:

Ligand based drug design

Relies on knowledge of other molecules that bind to the biological target of interest.

Structure-based drug design

Relies on knowledge of the three-dimensional structure of the biological target obtained through methods such as homology modeling, NMR spectroscopy, X-ray crystallography etc.

Drug discovery process is a critical issue in the pharmaceutical industry since it is a very costly and time-consuming process to produce new drug potentials and enlarge the scope of diseases incurred.

In both methods of designing drugs, computers and various bioinformatics tool come handy. Thus, in silico drug designing today is very crucial means to allay the arduous task of manual and experimental designing of drugs.

In silico technology alone, however, cannot guarantee the identification of new, safe and effective lead compound but more realistically future success depends on the proper integration of new promising technologies with the experience and strategies of classical medicinal chemistry.

DRUG TARGETS

• Target - Molecular recognition site to which drug binds

• Target may be – Protein molecule ♣ A receptor ♣ Enzyme ♣ Transport molecule ♣ Ion channel ♣ Tubulin ♣ Immunophilin

TYPES

There are four different methodologies commonly used in the drug designing:

1) Ligand-Based Drug Design or Indirect Drug Design 2) Structure-Based Drug Design or Direct Drug Design 3) Rational Drug Design 4) Computer-Assisted Drug Design

Mechanism Based Drug Design

• When the disease process is understood at the molecular level and the target molecule(s) are defined • Drugs can be designed specifically to interact with the target molecule in such a way as to disrupt the disease.

Structure-Based Drug Design

• First techniques to be used in drug design. • Helped in the discovery process of new drugs. • Information about the structural dynamics and electronic properties about ligands are obtained from calculations. • Structure-based drug design can be divided roughly into two categories: I. Ligand based II. Receptor Based

The first category is about "finding" ligands for a given receptor. • A large number of potential ligand molecules are screened • This method is usually referred as ligand-based drug design. • It saves synthetic effort to obtain new lead compounds.

• Docking attempts to find the "best" matching between two molecules • It includes finding the Right Key for the Lock • Given two biological molecules determine: - Whether the two molecules "interact" - If so, what is the orientation that maximizes the "interaction" while minimizing the total "energy" of the complex Goal: To be able to search a database of molecular structures and retrieve all molecules that can interact with the query structure

Receptor Based Drug Design

• Another category is about "building" ligands, which is usually referred as receptor-based drug design. • Ligand molecules are built up within the constraints of the binding pocket by assembling small pieces in a stepwise manner. • These pieces can be either individual atoms or molecular fragments. • The key advantage of such a method is that novel structures, not contained in any database, can be suggested.

X-Ray Crystallography

• Starting point for gathering information from mechanistic drug design. • Determine structural information about a molecule. • Provides the critically important coordinates needed for the handling of data by computer modeling system.

Nuclear Magnetic Resonance (NMR)

• NMR uses much softer radiation • Examine molecules in the more mobile liquid phase • Three-dimensional information will be obtained. • Examines small molecule-macromolecule complexes.

Homology Modeling

• Homology modeling, also known as comparative modeling of protein. • Constructing an atomic-resolution model of the "target" and an experimental three-dimensional structure of a related homologous protein.

The Process of Drug Designing

The drug discovery process involves the identification of the lead structure followed by the synthesis of its analogs, their screening to get candidate molecules for drug development.

In the traditional drug discovery process, the steps include:

Identification of the suitable drug target which are biomolecules mainly including DNA, RNA and proteins (such as receptors, transporters, enzymes and ion channels).

Validation of such targets is necessary to exhibit a sufficient level of ‘confidence’ and to know their pharmacological relevance to the disease under investigation. This can be performed from very basic levels such as cellular, molecular levels to the whole animal level.

Identification of effective compounds such as inhibitors, modulators or antagonists for such target is called lead identification where the design and development of a suitable assay is done to monitor the effect on the target under study.

Compounds showing dose-dependent target modulation in terms of a certain degree of confidence are processed further as lead compounds.

Subsequently, the experiments are performed on the animal models in the laboratories and the positive results are then optimized in terms of potency and selectivity.

Assessing of the physicochemical properties, pharmacokinetic and safety features are also assessed before they become candidates for drug development.

Even though most of the processes depend on experimental tasks, in silico approaches are playing important roles in every stage of this drug discovery pipeline which are described below:

In silico Methods in Drug Discovery and the role of Bioinformatics

In silico drug design represents computational methods and resources that are used to facilitate the opportunities for future drug lead discovery.

The explosion of bioinformatics, cheminformatics, genomics, proteomics, and structural information has provided hundreds of new targets as well as new ligands.

The Role of Bioinformatics

Bioinformatics techniques hold a lot of prospective in target identification (generally proteins/enzymes), target validation, understanding the protein, evolution and phylogeny and protein modeling.

Bioinformatics analysis can not only accelerate drug target identification and drug candidate screening and refinement, but also facilitate characterization of side effects and predict drug resistance.

One of the major thrusts of current bioinformatics approaches is the prediction and identification of biologically active candidates, and mining and storage of related information.

It also provides strategies and algorithm to predict new drug targets and to store and manage available drug target information.

In molecular docking:

Docking is an automated computer algorithm that attempts to find the best matching between two molecules which is a computational determination of binding affinity between molecules.

This includes determining the orientation of the compound, its conformational geometry, and the scoring. The scoring may be a binding energy, free energy, or a qualitative numerical measure.

In some way, every docking algorithm automatically tries to put the compound in many different orientations and conformations in the active site, and then computes a score for each.

Some bioinformatics programs store the data for all of the tested orientations, but most only keep a number of those with the best scores.

Docking can be done using bioinformatics tools which are able to search a database containing molecular structures and retrieve the molecules that can interact with the query structure.

It also aids in the building up chemical and biological information databases about ligands and targets/proteins to identify and optimize novel drugs.

It is involved in devising in silico filters to calculate drug likeness or pharmacokinetic properties for the chemical compounds prior to screening to enable early detection of the compounds which are more likely to fail in clinical stages and further to enhance detection of promising entities.

Bioinformatics tools help in the identification of homologs of functional proteins such as motif, protein families or domains.

It helps in the identification of targets by cross species examination by the use of pairwise or multiple alignments.

The tools help in the visualization of molecular models.

It allows identifying drug candidates from a large collection of compound libraries by means of virtual high-throughput screening (VHTS).

Homology modeling is extensively used for active site prediction of candidate drugs.

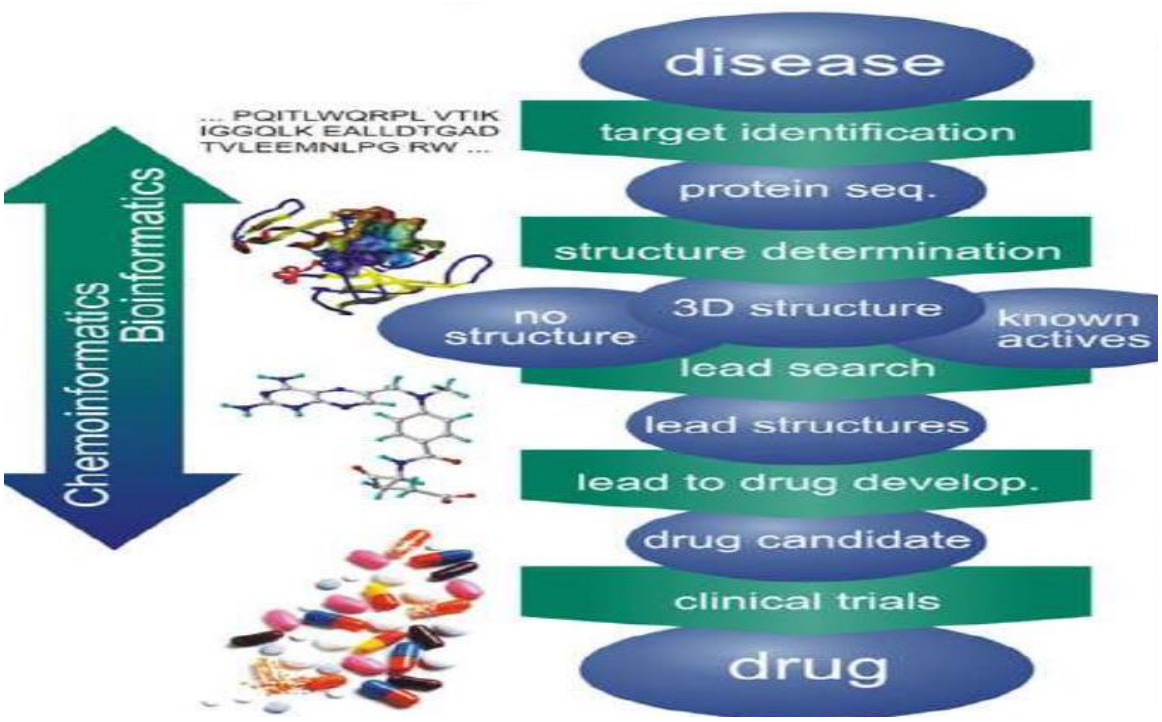


Figure 1

E – CELL

The E-CELL system is, in essence, a rule-based simulation system and is written in C++, an object-oriented programming language. The model consists of three lists, and is loaded at runtime. The substance list defines all objects which make up the cell and the culture medium. The rule list defines all of the reactions which can take place within the cell, and the system list defines spatial and/or functional structure of the cell and its environment. The state of the cell at each time frame is expressed as a list of concentration values of all substances within the cell, along with global values for cell volume, pH and temperature. The simulator engine generates the next state in time by computing all of the functions defined in the reaction rule list. In addition to using the sample models provided with the system, the user can create user-defined models by writing original substance and rule lists. Graphical interfaces are provided to allow observation and interaction throughout the simulation process. A substance can be a substrate, product or catalyst of a reaction. Typical substances include proteins, protein complexes, DNA (genes), RNA and small molecules. The list of substance concentrations is updated with the new values computed by the simulator engine after each time interval. In a single time interval, each rule in the rule list is called upon by the simulator engine to compute the change in concentration of each substance. The net change in concentration for each substance is added to the present concentration at the end of each time interval to update the set of state variables, i.e. to generate the next state of the cell. By encapsulating numerical integration methods into object classes, virtually any integration algorithm can be used for simulation of an E-CELL model. Furthermore, E-CELL allows the assignment of any numerical integration algorithm for each compartment of the cell model, facilitating the optimization of the simulation for the user's purpose (e.g. simulation accuracy or speed). Different time intervals (Δt) can also be defined for each spatial or functional compartment and they can be redefined through the control panel at runtime by the user. In the present version, the system defaults to 1 ms for Δt and the user can select between the first-order Euler [error is $O(\Delta t\ 2)$] or fourth-order Runge–Kutta [$O(\Delta t\ 5)$] methods for the numerical integration in each compartment. The Euler method is used in compartments with discrete, stochastic reactions such as DNA–protein binding, and the Runge–Kutta method is used for compartments with deterministic reactions defined by continuous rate functions. The simulation of our present whole-cell model runs at ~1/20 of real time on a laptop computer with Pentium-II 200 MHz, and about four times faster on a DEC alpha 21264A 533 MHz

with 1 ms integration step and monolithic integration model. A single pathway such as glycolysis runs ~30 times faster under the same conditions.

The E-CELL interfaces provide a means of conducting 'experiments in silico'. For example, we can 'starve' the cell by draining glucose from the culture medium. The cell would eventually 'die', running out of ATP. If glucose is added back, it may or may not recover, depending on the duration of starvation. We can also 'kill' the cell by knocking out an essential gene for, for example, protein synthesis. The cell would become unable to synthesize proteins, and all enzymes would eventually disappear due to spontaneous degradation.

Application to genome engineering

One of our ultimate goals is to model the real cell of M.genitalium, the organism having the smallest known chromosome. Because of the small number of genes (470 proteins, 37 RNAs), M.genitalium is a prime candidate for exhaustive functional (proteome) analysis. Because there are still many genes whose functions are not yet known, it will probably be necessary to hypothesize putative proteins to complement missing metabolic functions, in order for the model cell to work in silico.

Metabolic requirements

The assessment of the metabolic requirements of the cell is an excellent example of a potential application for E-CELL. At present, M.genitalium is grown in a complex medium containing several chemically undefined components including fetal bovine serum, and also extracts of yeast and beef. The problem of designing a chemically defined growth medium could be addressed through a purely empirical approach. However, a more interesting approach is one that is informed by knowledge of the complete genome sequence. By combining knowledge of the metabolic enzymes present in the cell with information concerning protein transporters of metabolites across the cell membrane, it should be possible to evaluate whether a particular defined medium can support growth, by using the E-CELL model. The main difficulty in this approach is that identification of gene function solely on the basis of sequence is uncertain. Comparison of laboratory results with E-CELL predictions should help to overcome this difficulty. Agreement between the model and laboratory growth experiments will be evaluated for a large number of different chemically defined media. Differences between experimental observations and the E-CELL predictions will be used to refine the model. This could lead to the identification of

new enzymes or transporters among genes with previously unassigned roles, or to the removal of a questionable role assignment based on a marginal level of sequence similarity.

Gene expression

Another area to apply the E-CELL software is in the deciphering of gene regulatory networks. Gene expression patterns of M.genitalium are currently being determined at TIGR under a variety of growth conditions. We expect that these results will suggest specific mechanisms for control of transcript levels which can be modeled by rules in the E-CELL system. We will conduct parallel experiments in the laboratory and in silico with the E-CELL system; given an appropriate model of the cell, we can change initial values of ingredients of the culture medium and observe increases and decreases of mRNA levels. The results of those in silico experiments should be consistent with results of biological and biochemical experiments. The computer model will then be refined as necessary.

Minimal gene set

We expect that the E-CELL system will be useful in defining the minimal set of genes required for a self-replicating cell under a specific set of laboratory conditions. At TIGR, work is under way to identify the genes of M.genitalium which are non-essential, by gene disruption experiments using transposons. If the E-CELL model is sufficiently detailed and accurate, then these gene disruption experiments can be modeled in silico to predict a minimal gene set. The laboratory experiments will lead to the prediction of a reduced gene set which should be a close approximation to the truly minimal Mycoplasma genome. Alternative predictions of a minimal gene set can also be proposed on theoretical grounds, or by deducing a core set of genes conserved between M.genitalium and other microbial genomes. The E-CELL system should be useful in modeling cells based on these alternative proposals for a minimal cellular genome. We expect that a combination of laboratory experiments and in silico modeling using the E-CELL system will lead to a more reliable prediction of the minimal gene complement for a self-replicating cell than could be obtained by either method alone.

Comparison of the models with the results of laboratory experiments will allow further refinement of the computer models. This, in turn, will lead to a better understanding of the experimental results, and hence a better understanding of the essential requirements of a minimal living cell.

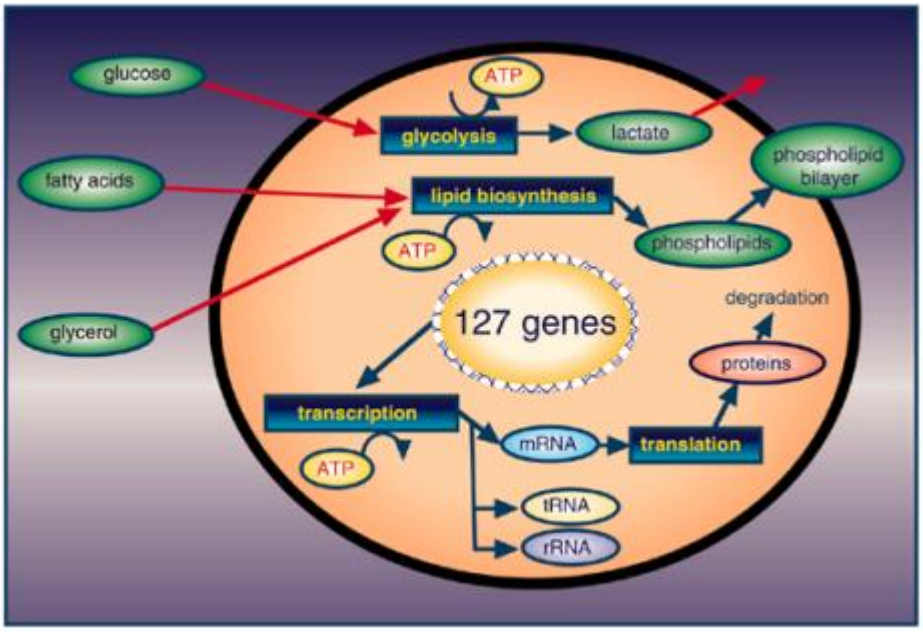


Figure 2 Metabolism overview of the model cell. It has pathways for glycolysis and phospholipid biosynthesis, as well as transcription and translation metabolisms.

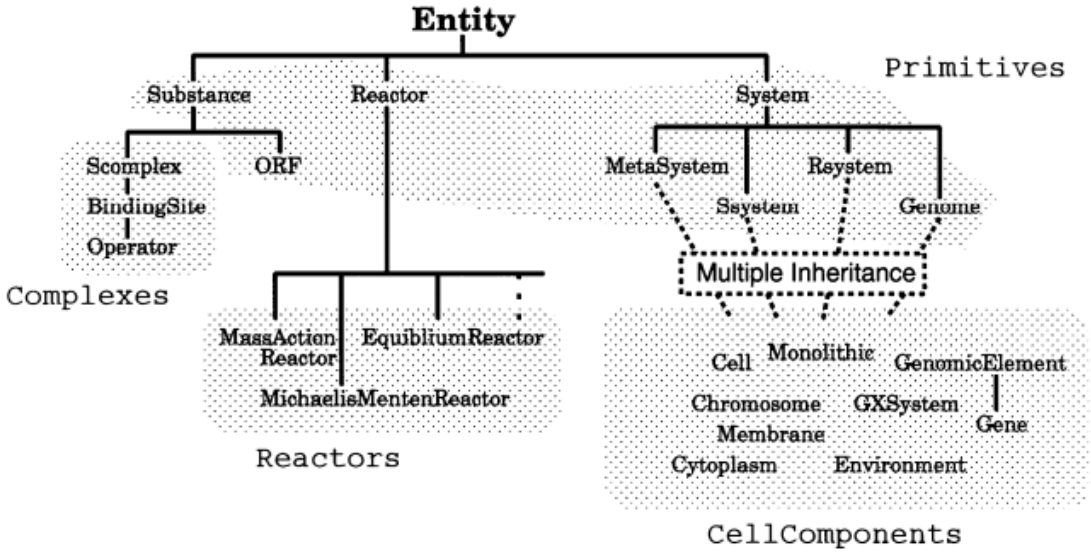


Figure 3 Ontology structure of the E-CELL system. There are three fundamental classes: Substance, Reactor and System. Reactors and Cell Components are the user-definable classes

PHYLOGENETIC ANALYSIS

How to construct a Phylogenetic tree?

- A phylogenetic tree is a visual representation of the relationship between different organisms, showing the path through evolutionary time from a common ancestor to different descendants.
- Similarities and divergence among related biological sequences revealed by sequence alignment often have to be rationalized and visualized in the context of phylogenetic trees. Thus, molecular phylogenetics is a fundamental aspect of bioinformatics.
- Molecular phylogenetics is the branch of phylogeny that analyzes genetic, hereditary molecular differences, predominately in DNA sequences, to gain information on an organism's evolutionary relationships.
- The similarity of biological functions and molecular mechanisms in living organisms strongly suggests that species descended from a common ancestor. Molecular phylogenetics uses the structure and function of molecules and how they change over time to infer these evolutionary relationships.
- From these analyses, it is possible to determine the processes by which diversity among species has been achieved. The result of a molecular phylogenetic analysis is expressed in a phylogenetic tree.

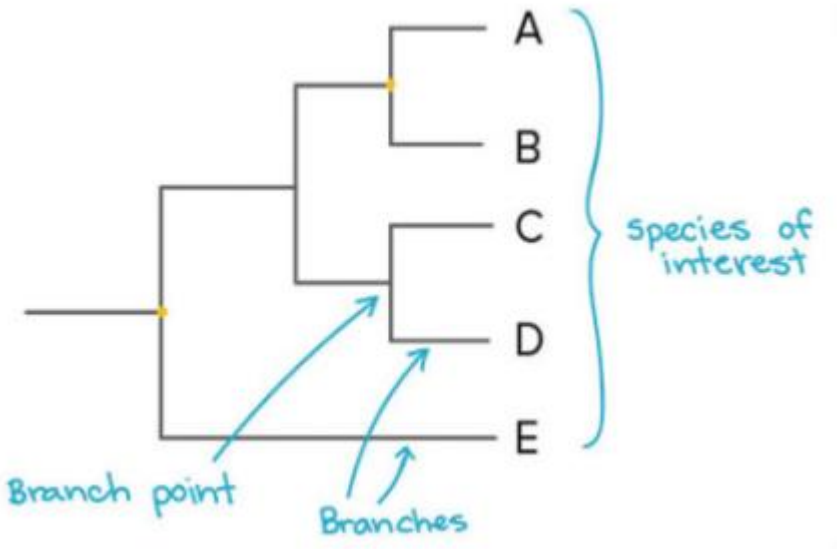


Figure 4

Phylogenetic Analysis and the Role of Bioinformatics

Molecular data that are in the form of DNA or protein sequences can also provide very useful evolutionary perspectives of existing organisms because, as organisms evolve, the genetic materials accumulate mutations over time causing phenotypic changes. Because genes are the medium for recording the accumulated mutations, they can serve as molecular fossils. Through comparative analysis of the molecular fossils from a number of related organisms, the evolutionary history of the genes and even the organisms can be revealed.

However, phylogeny inference are notoriously difficult endeavours because the number of solutions increases explosively with the number of taxa and the tremendous number of new questions in evolutionary biology that could be investigated through the use of larger taxon samplings.

But with the development and use of computational and an array of bioinformatics tools, the ability to analyze large data sets in practical computing times, and yielding an optimal or near-optimal solutions with high probability are being possible. In response to this trend, much of the current research in phyloinformatics (i.e., computational phylogenetics) concentrates on the development of more efficient heuristic approaches.

Steps in Phylogenetic Analysis

The basic steps in any phylogenetic analysis include:

1. **Assemble and align a dataset**

- The first step is to identify a protein or DNA sequence of interest and assemble a dataset consisting of other related sequences.
- DNA sequences of interest can be retrieved using NCBI BLAST or similar search tools.
- Once sequences are selected and retrieved, multiple sequence alignment is created.
- This involves arranging a set of sequences in a matrix to identify regions of homology.
- There are many websites and software programs, such as ClustalW, MSA, MAFFT, and T-Coffee, designed to perform multiple sequence on a given set of molecular data.

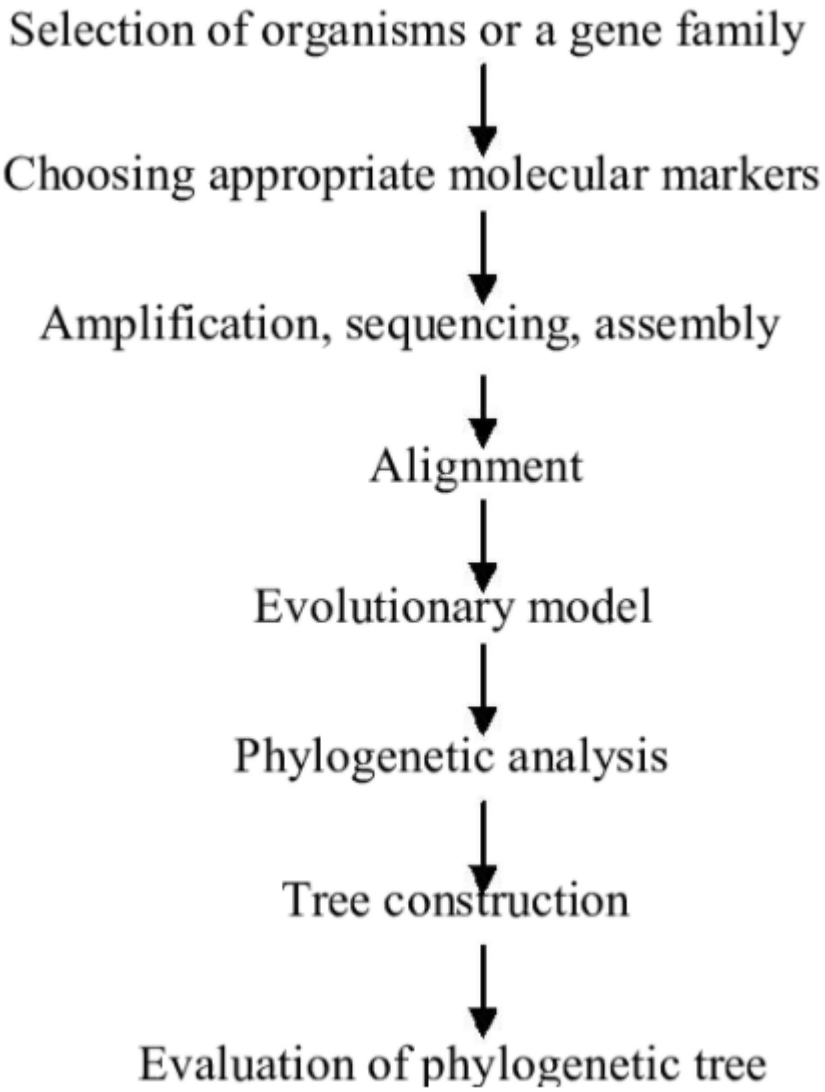


Figure 5

2. **Build (estimate) phylogenetic trees** from sequences using computational methods and stochastic models

- To build phylogenetic trees, statistical methods are applied to determine the tree topology and calculate the branch lengths that best describe the phylogenetic relationships of the aligned sequences in a dataset.
- The most common computational methods applied include distance-matrix methods, and discrete data methods, such as maximum parsimony and maximum likelihood.
- There are several software packages, such as Paup, PAML, PHYLIP, that apply these most popular methods.

3. **Statistically test and assess the estimated trees.**

- Tree estimating algorithms generate one or more optimal trees.

- This set of possible trees is subjected to a series of statistical tests to evaluate whether one tree is better than another – and if the proposed phylogeny is reasonable.
- Common methods for assessing trees include the Bootstrap and Jackknife Resampling methods, and analytical methods, such as parsimony, distance, and likelihood.

Bioinformatics Tools for Phylogenetic Analysis

- There are several bioinformatics tools and databases that can be used for phylogenetic analysis.
- These include **PANTHER, P-Pod, PFam, TreeFam, and the PhyloFacts structural phylogenomic encyclopedia**.
- Each of these databases uses different algorithms and draws on different sources for sequence information, and therefore the trees estimated by PANTHER, for example, may differ significantly from those generated by P-Pod or PFam.
- As with all bioinformatics tools of this type, it is important to test different methods, compare the results, then determine which database works best (according to consensus results) for studies involving different types of datasets.

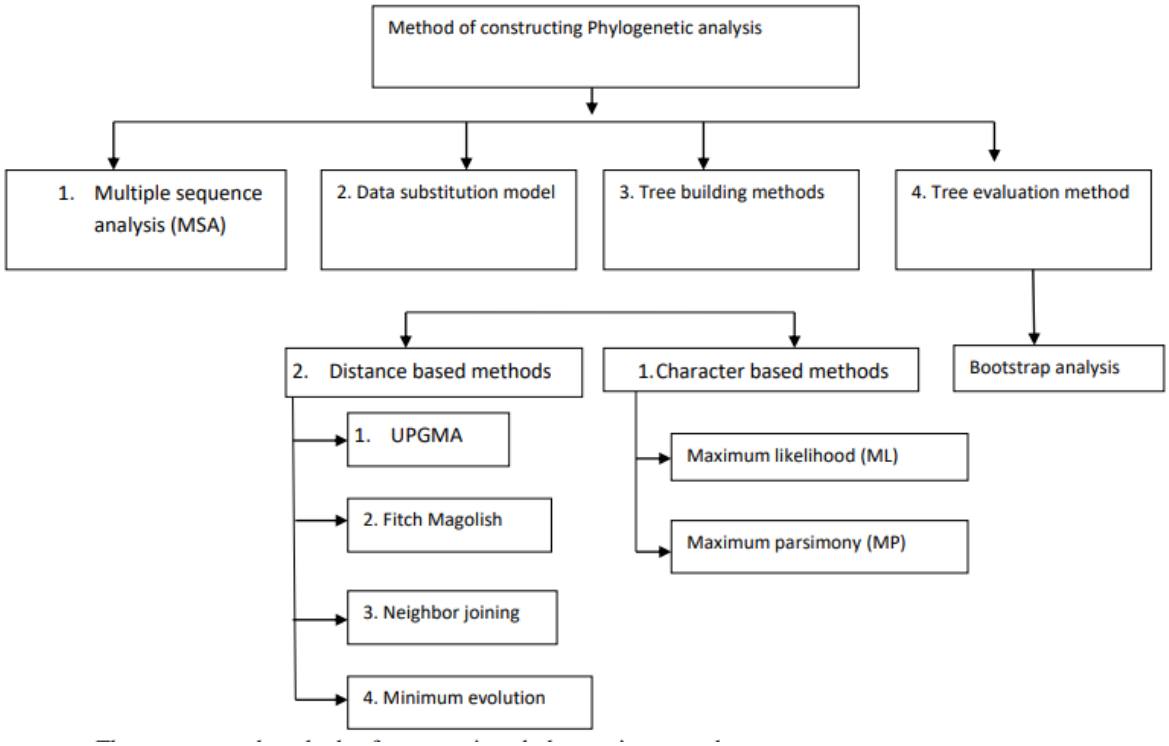


Figure 6

There are several methods of constructing phylogenetic trees - the most common are:

- distance methods
- character based methods

All these methods can only provide estimates of what a phylogenetic tree might look like for a given set of data. Most good methods also provide an indication of how much variation there is in these estimates. Distance methods: Preferred for work with immunological data, frequency data, or data with some impreciseness in its methods. Very rapid, and easily permits statistical tests e.g. bootstrapping. Derives some measure of similarity or difference between the input sequences.

UPGMA Cluster algorithm. Links least different pairs of seqs, sequentially (so that when one pair is formed, they become a single entity). (Invalid) assumptions made: 1. Rate of change equal among all sequences. 2. Branch lengths correlate with the expected phenotypic distance between sequences, which corresponds to a proportional measure of time. o NJ Corrects several assumptions made in the UPGMA method. Yields an unrooted tree. o Fitch and Margoliash Does not try to find pairs of least different sequences, but tries to find trees that fulfil an optimum criterion. Yields an unrooted tree. Character based methods Popular for reconstructing ancestral relationships. o Maximum parsimony: Evaluates all possible trees. Infers the number of evolutionary events implied by a particular topology. The most likely tree is then one that requires the minimum number of evolutionary changes needed to explain the observed data. Problems: Most parsimonious tree may not be unique; difficult to make valid statistical statements if there are many steps in a tree; branches with particularly rapid rates of change tend to attract one another, especially when the sequence lengths are small. o Maximum likelihood: Very slow. Preferred when homoplasies (convergences of a particular character at a site) are expected to be concentrated in a few sites only, whose identities are known in advance. The method works by estimating, for all nucleotide positions in a sequence, what the probability of having a particular nucleotide at a particular site is, based on whether or not its ancestors had it (and the transition/transversion ratio). These probabilities are summed over the whole sequence, for both branches of a bifurcating tree. The product of the two probabilities gives you the likelihood of the tree up to this point. With more sequences, the estimation is done recursively at every branch point. Since each site evolves independently, the likelihood of the phylogeny can be estimated at every site. This process can only be done in a reasonable amount of time with four sequences. If there are more than four sequences, basic trees can be made for sets of four sequences, and then extra sequences

added to the tree and the process of finding the maximum likelihood re-estimated. The order in which the sequences are added and the initial sequences chosen to start the process critically influences the resulting tree. To prevent any bias, the whole process is done multiple times with random choices for the order of the sequences. A majority rule consensus tree is then chosen as the final tree. To create a phylogenetic tree, you must first have an alignment. This can be created using ClustalW. ClustalW can also create a tree file for you (if you choose 'nj', 'phylip', or 'dist' from the "Tree type" pull-down menu.) However, you have more control over the tree if you simply choose to create an alignment in ClustalW (do not choose a tree type in this case, because then the alignment itself will not be presented). Copy the alignment (including the title, so that the PHYLIP programs recognise the alignment format as ClustalW), and paste it into the text-entry box provided for alignments in one of the following programs in the PHYLIP suite of programs. PHYLIP will convert the format of your alignment to Phylip format automatically. However, occasionally, especially in cases where the alignment is very large, this automatic conversion may cause errors. You can also convert the alignment yourself using SQUIZZ.

PERL

Perl is a family of two high-level, general-purpose, interpreted, dynamic programming languages. "Perl" refers to Perl 5, but from 2000 to 2019 it also referred to its redesigned "sister language", Perl 6, before the latter's name was officially changed to Raku in October 2019.

Though Perl is not officially an acronym, there are various backronyms in use, including "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. Raku, 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.

Features

The overall structure of Perl derives broadly from C. Perl is procedural in nature, with variables, expressions, assignment statements, brace-delimited blocks, control structures, and subroutines.

Perl also takes features from shell programming. All variables are marked with leading sigils, which allow variables to be interpolated directly into strings. However, unlike the shell, Perl uses sigils on all accesses to variables, and unlike most other programming languages that use sigils, the sigil doesn't denote the type of the variable but the type of the expression. So for example, to access a list of values in a hash, the sigil for an array ("@") is used, not the sigil for a hash ("%"). Perl also has many built-in functions that provide tools often used in shell programming (although many of these tools are implemented by programs external to the shell) such as sorting, and calling operating system facilities.

Perl takes lists from Lisp, hashes ("associative arrays") from AWK, and regular expressions from sed. These simplify and facilitate many parsing, text-handling, and data-management tasks. Also shared with Lisp are the implicit return of the last value in a block, and the fact that all statements have a value, and thus are also expressions and can be used in larger expressions themselves.

Perl 5 added features that support complex data structures, first-class functions (that is, closures as values), and an object-oriented programming model. These

include references, packages, class-based method dispatch, and lexically scoped variables, along with compiler directives (for example, the strict pragma). A major additional feature introduced with Perl 5 was the ability to package code as reusable modules. Wall later stated that "The whole intent of Perl 5's module system was to encourage the growth of Perl culture rather than the Perl core."

All versions of Perl do automatic data-typing and automatic memory management. The interpreter knows the type and storage requirements of every data object in the program; it allocates and frees storage for them as necessary using reference counting (so it cannot deallocate circular data structures without manual intervention). Legal type conversions — for example, conversions from number to string — are done automatically at run time; illegal type conversions are fatal errors.

Design

The design of Perl can be understood as a response to three broad trends in the computer industry: falling hardware costs, rising labor costs, and improvements in compiler technology. Many earlier computer languages, such as Fortran and C, aimed to make efficient use of expensive computer hardware. In contrast, Perl was designed so that computer programmers could write programs more quickly and easily.

Perl has many features that ease the task of the programmer at the expense of greater CPU and memory requirements. These include automatic memory management; dynamic typing; strings, lists, and hashes; regular expressions; introspection; and an eval() function. Perl follows the theory of "no built-in limits, an idea similar to the Zero One Infinity rule.

Wall was trained as a linguist, and the design of Perl is very much informed by linguistic principles. Examples include Huffman coding (common constructions should be short), good end-weighting (the important information should come first), and a large collection of language primitives. Perl favors language constructs that are concise and natural for humans to write, even where they complicate the Perl interpreter.

Perl's syntax reflects the idea that "things that are different should look different." For example, scalars, arrays, and hashes have different leading sigils. Array indices and hash keys use different kinds of braces. Strings and regular expressions have different standard delimiters. This approach can be contrasted with a language such as Lisp, where the same basic syntax, composed of simple and universal symbolic expressions, is used for all purposes.

Perl does not enforce any particular programming paradigm (procedural, object-oriented, functional, or others) or even require the programmer to choose among them.

There is a broad practical bent to both the Perl language and the community and culture that surround it. The preface to Programming Perl begins: "Perl is a language for getting your job done." One consequence of this is that Perl is not a tidy language. It includes many features, tolerates exceptions to its rules, and employs heuristics to resolve syntactical ambiguities. Because of the forgiving nature of the compiler, bugs can sometimes be hard to find. Perl's function documentation remarks on the variant behavior of built-in functions in list and scalar contexts by saying, "In general, they do what you want, unless you want consistency.

No written specification or standard for the Perl language exists for Perl versions through Perl 5, and there are no plans to create one for the current version of Perl. There has been only one implementation of the interpreter, and the language has evolved along with it. That interpreter, together with its functional tests, stands as a de facto specification of the language. Perl 6, however, started with a specification, and several projects aim to implement some or all of the specification.

Applications

Perl has many and varied applications, compounded by the availability of many standard and third-party modules.

Perl has chiefly been used to write CGI scripts. It is also an optional component of the popular LAMP technology stack for Web development, in lieu of PHP or Python. Perl is used extensively as a system programming language in the Debian GNU/Linux distribution

Perl is often used as a glue language, tying together systems and interfaces that were not specifically designed to interoperate, and for "data munging," that is, converting or processing large amounts of data for tasks such as creating reports. In fact, these strengths are intimately linked. The combination makes Perl a popular all-purpose language for system administrators, particularly because short programs, often called "one-liner programs," can be entered and run on a single command line.

Perl code can be made portable across Windows and Unix; such code is often used by suppliers of software (both COTS and bespoke) to simplify packaging and maintenance of software build- and deployment-scripts.

Graphical user interfaces (GUIs) may be developed using Perl. For example, Perl/Tk and wxPerl are commonly used to enable user interaction with Perl scripts. Such interaction may be synchronous or asynchronous, using callbacks to update the GUI.

Implementation

Perl is implemented as a core interpreter, written in C, together with a large collection of modules, written in Perl and C. As of 2010, the interpreter is 150,000 lines of C code and compiles to a 1 MB executable on typical machine architectures. Alternatively, the interpreter can be compiled to a link library and embedded in other programs. There are nearly 500 modules in the distribution, comprising 200,000 lines of Perl and an additional 350,000 lines of C code (much of the C code in the modules consists of character encoding tables).

The interpreter has an object-oriented architecture. All of the elements of the Perl language—scalars, arrays, hashes, coderefs, file handles—are represented in the interpreter by C structs. Operations on these structs are defined by a large collection of macros, typedefs, and functions; these constitute the Perl C API. The Perl API can be bewildering to the uninitiated, but its entry points follow a consistent naming scheme, which provides guidance to those who use it.

The life of a Perl interpreter divides broadly into a compile phase and a run phase. In Perl, the phases are the major stages in the interpreter's life-cycle. Each interpreter goes through each phase only once, and the phases follow in a fixed sequence.

Most of what happens in Perl's compile phase is compilation, and most of what happens in Perl's run phase is execution, but there are significant exceptions. Perl makes important use of its capability to execute Perl code during the compile phase. Perl will also delay compilation into the run phase. The terms that indicate the kind of processing that is actually occurring at any moment are compile time and run time. Perl is in compile time at most points during the compile phase, but compile time may also be entered during the run phase. The compile time for code in a string argument passed to the eval built-in occurs during the run phase. Perl is often in run time during the compile phase and spends most of the run phase in run time. Code in BEGIN blocks executes at run time but in the compile phase.

At compile time, the interpreter parses Perl code into a syntax tree. At run time, it executes the program by walking the tree. Text is parsed only once, and the syntax tree is subject to

optimization before it is executed, so that execution is relatively efficient. Compile-time optimizations on the syntax tree include constant folding and context propagation, but peephole optimization is also performed.

Perl has a Turing-complete grammar because parsing can be affected by run-time code executed during the compile phase.[85] Therefore, Perl cannot be parsed by a straight Lex/Yacc lexer/parser combination. Instead, the interpreter implements its own lexer, which coordinates with a modified GNU bison parser to resolve ambiguities in the language.

It is often said that "Only perl can parse Perl,"[86] meaning that only the Perl interpreter (perl) can parse the Perl language (Perl), but even this is not, in general, true. Because the Perl interpreter can simulate a Turing machine during its compile phase, it would need to decide the halting problem in order to complete parsing in every case. It is a long-standing result that the halting problem is undecidable, and therefore not even perl can always parse Perl. Perl makes the unusual choice of giving the user access to its full programming power in its own compile phase. The cost in terms of theoretical purity is high, but practical inconvenience seems to be rare.

Other programs that undertake to parse Perl, such as source-code analyzers and auto-indenters, have to contend not only with ambiguous syntactic constructs but also with the undecidability of Perl parsing in the general case. Adam Kennedy's PPI project focused on parsing Perl code as a document (retaining its integrity as a document), instead of parsing Perl as executable code (that not even Perl itself can always do). It was Kennedy who first conjectured that "parsing Perl suffers from the 'halting problem', which was later proved

Perl is distributed with over 250,000 functional tests for core Perl language and over 250,000 functional tests for core modules. These run as part of the normal build process and extensively exercise the interpreter and its core modules. Perl developers rely on the functional tests to ensure that changes to the interpreter do not introduce software bugs; additionally, Perl users who see that the interpreter passes its functional tests on their system can have a high degree of confidence that it is working properly.

Bioperl

BioPerl is a collection of Perl modules that facilitate the development of Perl scripts for bioinformatics applications. It has played an integral role in the Human Genome Project.BioPerl is an active open source software project supported by the Open Bioinformatics Foundation.

In order to take advantage of BioPerl, the user needs a basic understanding of the Perl programming language including an understanding of how to use Perl references, modules, objects and methods.

Influence on the Human Genome Project

The Human Genome Project faced several challenges during its lifetime. A few of these problems were solved when many of the genomics labs started to use Perl. The process of analyzing all of the DNA sequences was one such problem. Some labs built large monolithic systems with complex relational databases that took forever to debug and implement, and got surpassed by new technologies. Other labs learned to build modular, loosely-coupled systems whose parts could be swapped in and out when new technologies arose. Many of the initial results from all of the labs were mixed. It was eventually discovered that many of the steps could be implemented as loosely coupled programs that were run with a Perl shell script. Another problem that was fixed was interchange of data. Each lab usually had different programs that they ran with their scripts, resulting in several conversions when comparing results. To fix this the labs collectively started using a super-set of data. One script was used to convert from super-set to each lab's set and one was used to convert back. This minimized the number of scripts needed and data exchange became simplified with Perl.

Features

BioPerl provides software modules for many of the typical tasks of bioinformatics programming. These include:

Accessing nucleotide and peptide sequence data from local and remote databases

Searching for similar sequences

Creating and manipulating sequence alignments

Searching for genes and other structures on genomic DNA

Developing machine readable sequence annotations

In addition to being used directly by end-users

BioPerl has also provided the base for a wide variety of bioinformatic tools, including amongst others:

SynBrowse

GeneComber

TFBS

MIMOX

BioParser

Degenerate primer design

Querying the public databases

Current Comparative Table

New tools and algorithms from external developers are often integrated directly into BioPerl itself:

Dealing with phylogenetic trees and nested taxa

FPC Web tools

Advantages

BioPerl was one of the first biological module repositories that increased its usability. It has very easy to install modules, along with a flexible global repository. BioPerl uses good test modules for a large variety of processes.

Disadvantages

There are many ways to use BioPerl, from simple scripting to very complex object programming. This makes the language not clear and sometimes hard to understand. For as many modules that BioPerl has, some do not always work the way they are intended.

CHEMOINFORMATICS

Chemoinformatics/Chemiinformatics/Chemical information/Chemical informatics has been recognised in recent years as a distinct discipline in computational molecular sciences. Cheminformatics is also known as interface science as it combines Physics, Chemistry, Biology, Mathematics, Biochemistry, Statistics and in formatics.

The primary focus of cheminformatics is to analyse/simulate/ modelling/manipulate chemical information which can represented either in 2D structure or in 3D structure. Industry sectors such as, agrochemicals, food and pharmaceutical are distinct areas where cheminformatics plays significant role in the recent history of molecular sciences.

Cheminformatics has mainly dealt with small molecules, whereas bioinformatics addresses genes, proteins, and other larger chemical compounds. Chem and Bioinformatics complements each other for bimolecular process, like structure and function of proteins, the binding of a ligand to its binding site, the conversion of a substrate within its enzyme receptor, and the catalysis of a biochemical reaction by an enzyme.

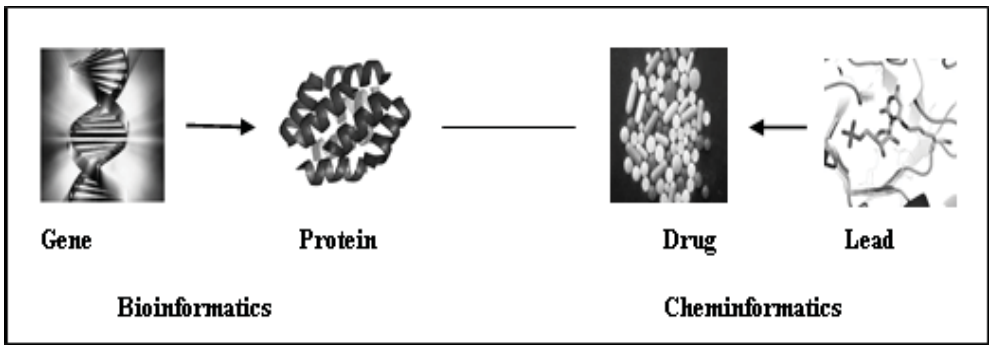


Figure 7

Different tools and methods are available to represent chemical structure, database to store chemical data, to perform searching process, Quantity Structure-Activity Relationship(QSAR), Quantity Structure-Property Relationship(QSPR), to predict physical, chemical and biological properties of a molecule

Need and Importance of Cheminformatics

Cheminformatics plays a key role to maintain and access enormous amount of chemical data, produced by chemist (more than 45 million chemical compounds are known and the number may increase in million every year,) by using a proper database.

Also, the field of chemistry needs a novel technique for knowledge extraction from data to model complex relationships between the structure of the chemical compound and biological activity or the influence of reaction condition on chemical reactivity. Cheminformatics has wider range of application.

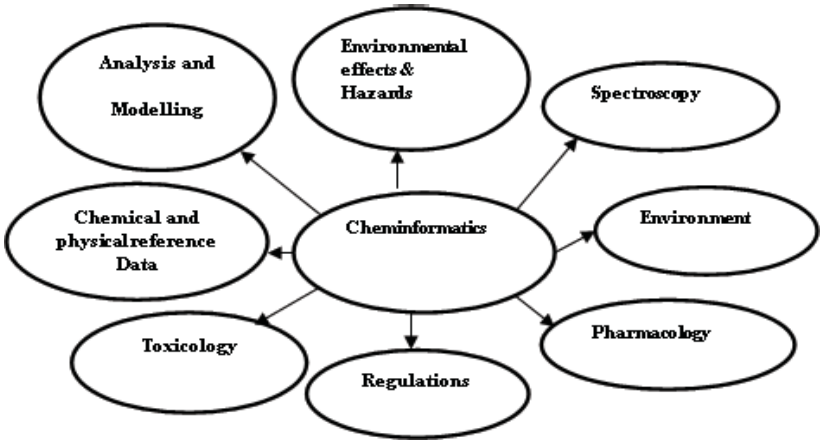


Figure 8 Need for Cheminformatics

Three major aspects of Cheminformatics are;

i) Information Acquisition, is a process of generating and collecting data empirically (experimentation) or from theory (molecular simulation)

ii) Information Management deals with storage and retrieval of information and

iii) Information use, which includes Data Analysis, correlation, and application to problems in the chemical and biochemical sciences

Cheminformatics and its Applications

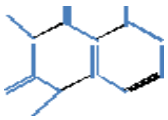
Cheminformatics is a significant application of information technology to help chemists for investigating new problems, organize, analyse, and understand scientific data in the development of novel compounds, materials and processes. Primary modules of cheminformatics are Computer-Assisted Synthesis Design, Structure representation and chemmetrics,

Computer-Assisted Synthesis Design (CASD) is applied mainly where art ificial intelligence technique can be applied. This technique is applied in various applications which included pharmaceutical, food industry, textile industry and agro industry.

Various forms of machine readable chemical representation play basic property to design chemical database where the chemical information are stored for analysis and manipulation. The chemical structure representations can be linear, 2D or in 3D format. Some of the chemical structure representations. SMILES (Simplified

Molecular Input Line Entry Specification) is one of the linear chemical notation format which is widely used among chemist for various clin ical and analysis purpose. Structure representation deals with Reaction Representation, Structure Descriptors, Molecular Modelling, Structure Searching, and Computer-Assisted Structure Elucidation (CASE).

Table 1: Some of the Chemical Structure Representation

Representation	**Name**
Caffine	Common Name
trimethylxanthine coffeine, theine, mateine,	Synonyms
$C_8H_{10}N_4O_2$	Empirical formula
3,7-dihydro-1,3,7-trimethyl-1H-purine-2,6-dione	IUPAC Name
58-08-2	CAS Registry Number
T56 BN DN FNVNVJ B1 F1 H1	WLN Notation
CN1C=NC2=C1C(=O)N(C(=O)N2C)C	SMILES
1S/C8H10N4O2/c1-10-4-9-6-5(10)7(13)12(3)8(14)11(6)2/h4H,1-3H3	Inchl
	Markush Structure
	Connection Table
[OH]c1cccc1	Fragment Code
0000100110100111	Fingerprint
5244987098423150	Hash Code

	1	2	3	4	5	6	7	8	9	10	11
1	0	1	0	0	0	0	0	0	0	2	0
2	1	0	2	0	0	0	0	0	0	0	0
3	0	2	0	1	0	0	0	1	0	0	0
4	0	0	1	0	1	0	0	0	0	0	0
5	0	0	0	1	0	2	1	0	0	0	0
6	0	0	0	0	2	0	0	0	0	0	0
7	0	0	0	0	1	0	0	0	0	0	0
8	0	0	1	0	0	0	0	0	2	0	0
9	0	0	0	0	0	0	0	2	0	1	0
10	2	0	0	0	0	0	0	0	1	0	1
11	0	0	0	0	0	0	0	0	0	1	0

Reaction Representation helps to understand the basic chemical models, quantify chemical reactivity and extract knowledge from the reaction information. Molecular modelling is a method includes a variety of computational schemes which are aimed at stimulating molecular structures, their properties

Structure Searching involves in determination of features like bond orders, rings and aromaticity. It includes searching the whole structure, substructure, structure similarity and diversity. CASE builds on information obtained from various spectroscopic methods like IR, NMR, MS, etc. Structure Descriptor used to identify the physical, chemical and biological properties of chemical compound and relationship between two structures. The descriptors fall into four classes such as, i) Topological, ii) Geo metrical, iii) Electronic and iv) Hybrid or 3D Descriptors.

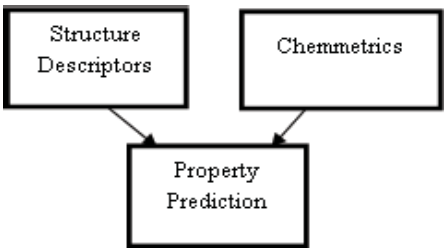


Figure 9 Structure Representation and Chemmetrics

Chemmetrics is used for quantitative analyse of the chemical data by using mathematical and statistical methods. It also deals with property prediction of chemical information.

Applications of Cheminformatics

The range of applications of cheminformatics is rich indeed; any field of chemistry can profit from its methods. The following lists different areas of chemistry and indicates some typical applications of cheminformatics.

a) Storing data generated through experiments or from molecular simulation Retrieval of chemical Structures from chemical database (Software libraries).
b) Prediction of physical, chemical and biological properties of chemical compounds.
c) Elucidation of the structure of a compound based on spectroscopic data.

Structure, Substructure, Similarity and diversity searching from chemical database

d) High Throughput Screening (HTS) is the integration of technologies (laborat ory automat ion, assay technology, micro plate based instrumentation, etc.) to quickly screen chemical

compounds in search of a desired activity.

e) Docking - Interaction between two macromolecules.

f) Drug Discovery.

g) Molecular Science, Materials Science, Food Science (nutraceuticals), Atmospheric chemistry, Polymer chemistry, Textile Industry, Combinatorial organic synthesis (COS).

2. Tools Used for cheminformatics

The development of software and tools for computer assisted organic synthesis are under vast development. This has resulted in many tools and representations for chemical structures. Some of the tools are listed below:

ISIS-Draw is a chemical structure drawing program for Windows, published by MDL Information Systems. It is the interfacial software to ISIS/Base database.

ChemDraw is a molecule editor developed by the cheminformatics company CambridgeSoft. ChemDraw is, along with Chem3D and ChemFinder, part of the ChemOffice suite of programs and is available for Macintosh and Microsoft Windows. ChemWindow, is a chemical structure drawing program with several template. The template can be created by the customer can be saved in template folder and opened in preference dialogue box.

ChemSketch, is a chemical structure drawing program with predefined temp lates are available for drawing and it is more powerful and user friendly tool for structure analysis.

PubChem is an open repository for small molecules and their experimental biological activity. It integrates and provides search, retrieval, visualization, analysis, and programmatic access tools in an effort to maximize the utility of contributed information.

Open Babel is a chemical tool box which interconverts chemical structures between different formats, over 110 formats.

Some other tools such as, CAS Draw, DIVA (Diverse Information, Visualization and Analysis), Structure Checker Accord, DS Accord Chemistry Cartridge, MarvinSketch PowerMV, TINKER, APBS, ArgusLab, Babel, ioSolveIT, ChemTK, Chimera, CLIFF, Dragon, gOpenMol, Grace, JOELib, Jmol, IA_LOGP, Lammps, MIPSIM, Mol2Mol, AMSOL, MOLCAS, Molexel, ICM-Pro, ORTEP, Packmol, Polar, XLOGP,PREMIER Biosoft, Q-chem, ALOGPS, Qmol, SageMD, ChemTK Lite, Transient, CLOGP,TURBOMOLE, UNIVIS, VM D, WHATIF, GCluto, COSMOlogic, KOWWIN are also used for similar kind of applications mentioned above.

Role of Cheminformatics in Morden Drug Discovery

Recent chemical developments for drug discovery are generating a lot of chemical data which is referred as information explosion. This has created a demand to effectively collect, organize, analyse and apply the chemical information in the process of modern drug discovery and development. The drug discovery process is aimed at discovering molecules that can be very rapidly developed for effective treatments to meet medical needs.

The entanglement of chemistry and information management started in the mid of 1970s, applying in the area of prediction of protein structure, Fourier transform of X-ray crystallography, enzyme and chemical kinetics, analyse various types of spectroscopy data and binding of chemical compoun ds. During early 1980s, computer technology is considered as the core component by the medical chemist to solve chemical problems. For example, collecting crystal structures of small molecules in Cambridge Structural Database (CSD) provides a fertile resource for geometrical data on molecular fragments for calibration of force fields and validation of results from computational chemistry. The need of storing macromolecular data results in Protein Data Base (PDB). The needs and refinement on these approaches result in several tools and upgrading the process of solving the problems.

The modern pharmaceutical drug discovery and development pipeline process, starts with Disease selection, Target identification, Lead identification, Lead Optimization, Pre -clinical trial testing, Clinical trial testing, Approval and circulation (Drug in market). In traditional drug discovery phase, the process which cost more time and money is replaced with lead identification and lead optimization process in modern drug discovery system. Each phase has an interaction component that transfers data, knowledge and information to one another.

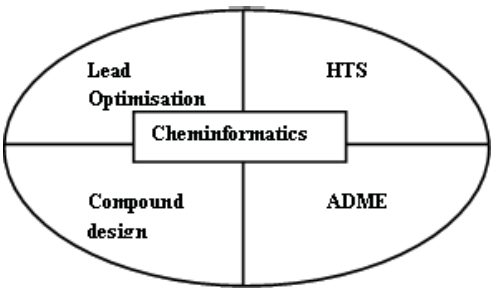


Figure 10