UNIT –I

Introduction

'Bioinformatics' is an inter-disciplinary approach in which application of computer science and information technology is to the field of molecular biology. The term was coined by **Paulien Hogeweg in 1979**. 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. Mathematical and computing approaches are used to understand biological processes. Development in the field of genomics and other molecular research technologies and development in information technologies have combinedly produce large amount of information related to molecular biology giving way to this new branch of science.

Common activities in bioinformatics include mapping and analyzing DNA and protein sequences, aligning different DNA and protein sequences to compare them and creating and viewing 3D models of protein structures. The Human Genome project has increased importance of bioinformatics. The research will help development and success in sequence alignment, protein structure prediction, prediction of gene expression and protein-protein interaction, genome wise association studies and many other areas.

Bioinformatics is similar but distinct science from biological computation and computational biology. Biological computation uses bioengineering and biology to build biological computers, whereas bioinformatics uses computation to better understand biology. Bioinformatics and computational biology have similar aims and approaches, but differ in scale: bioinformatics organizes and analyzes basic biological data, whereas computational biology builds theoretical models of biological systems, just as mathematical biology does with mathematical models.

Internet

The **Internet** is a massive network of networks, a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the **Internet**.

The Internet has changed how scientists share data and made it possible for one central warehouse of information to serve an entire research community. But more importantly, experimental technologies are rapidly advancing to the point at which it's possible to imagine systematically collecting all the data of a particular type in a central "factory" and then distributing it to researchers to be interpreted.

How does computation support the whole enterprise?

Computers play many roles in modern biology:

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

WWW

The World Wide Web (abbreviated WWW or the Web) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet. English scientist Tim Berners-Lee invented the World Wide Web in 1989.

Function

The terms Internet and World Wide Web are often used without much distinction. However, the two are not the same. The Internet is a global system of interconnected computer networks. In contrast, the World Wide Web is a global collection of documents and other resources, linked by hyperlinks and

URIs. Web resources are usually accessed using HTTP, which is one of many Internet communication protocols.

Viewing a web page on the World Wide Web normally begins either by typing the URL of the page into a web browser, or by following a hyperlink to that page or resource. The web browser then initiates a series of background communication messages to fetch and display the requested page. In the 1990s, using a browser to view web pages—and to move from one web page to another through hyperlinks—came to be known as 'browsing,' 'web surfing' (after channel surfing), or 'navigating the Web'. Early studies of this new behaviour investigated user patterns in using web browsers. One study, for example, found five user patterns: exploratory surfing, window surfing, evolved surfing, bounded navigation and targeted navigation.

These are some of the important fields in bioinformatics

1. Structural Bioinformatics:

Predicting the 3D structure of a protein from its protein sequence. Homology modelling is the best method for predicting the protein structures by using already structured or crystallized protein as a template. MODELLER is one of the best software for Homology modelling. Protein Data Bank is the data base for 3D co-ordinates of a protein.

2. Drug Designing:

Drug design is the approach of finding drugs by design, based on their biological targets. Typically a drug target is a key molecule involved in a particular metabolic or signalling pathway that is specific to a disease condition or pathology, or to the infectivity or survival of a microbial pathogen. Computer-assisted drug design uses computational chemistry to discover, enhance, or study drugs and related biologically active molecules. Click to see the drug discovery softwares.

3. Phylogenetics:

Predicting the genetic or evolutionary relation of set of organisms. Mitochondrial SNPs and Microsatellites (DNA repeats) are mostly used in Phylogenetics. MEGA,PAUP are PAUP* are some of the important softwares. Maximum Parsimony and Maximum Likelyhood are mostly used methods.

4. Computational biology:

Computational biology is an interdisciplinary field that applies the techniques of computer science, applied mathematics, and statistics to address problems inspired by biology.

5. Population Genetics:

Population Genetics is a study of genotype frequency distribution and the change in the genotype frequencies under the influence of Natural selection, genetics drift, mutation and gene flow.

Coalescent theory is one of the most used theory to predict the most recent ancester. Arlequin is one of the best and most used software in population gentics.

6. Genotype Analysis:

Genotype = Genetic variation, SNP,Mutation ....

1. Studying Genotype and phenotype association.
2. Studying Genotype frequencies. There is no specific software for genotype analysis. But its called the "Generation Next Market using Bioinformatics....". Genotyping is mostly done using Illumina and Affy microarry chips.
3. Genome-wide association defines more than 30 distinct susceptibility loci for Crohn's disease.
4. Estimating coverage and power for genetic association studies using near-complete variation data.
5. Genetic diversity patterns at the human clock gene period 2 are suggestive of population-specific positive selection.
6. Environment And Genetics in Lung cancer Etiology (EAGLE) study: an integrative population-based case-control study of lung cancer.

7. Splicing Site prediction:

Splicing prediction is a very important application of Bioinformatics which is very important in Gene expression studies. Visit also Alternative Splicing site Predictior.

8. MiRNA prediction:

MiRNA = MicroRNA. MiRNA emerged as a new Gene regulatory element and gained more space in research. 20 -23 base pair RNA which regulates a gene or genes. So many methods and softwares have been developed to predicting this tiny RNAs. But still they are not precise in predicting. It means that we need some more information from experimental labs to predict.

9. RNA Structure prediction:

The functional form of single stranded RNA molecules frequently requires a specific tertiary structure. The scaffold for this structure is provided by secondary structural elements which are hydrogen bonds within the molecule. This leads to several recognizable "domains" of secondary structure like hairpin loops, bulges and internal loops. There has been a significant amount of bioinformatics research directed at the RNA structure prediction problem.

10. Gene Prediction:

Predicting the Gene by the predefined conditions. Comparative genomics is the best method for predicting the gene.

11. Transcription factor binding site prediction:

Predicting the transcription factor. Most common method is to use "Comparative genomics". And finding clusters of motifs in the noncoding part of gene.

Predicting functional transcription factor binding through alignment-free and affinity-based analysis of orthologous promoter sequences.

12. Genome Annotation:

Predicitng the genes, coding and noncoding sequences are called genome annotation.

13. Ancestry Prediction:

Predicting the Ancestry of an individual based on his/her genetic signatures or SNPs. mitochondrial SNPs are used in predicting Maternal ancestry because Mitochondria is passed ONLY through mother to the child.

Y chromosome SNPs are used in predicting paternal ancestry becuase Y chromsome is passed from Father to the child.

Ancestry is one of the successful field in Bioinformatics. Genography project by Dr. Spencer Wells is one of the finest one.

14. Mathematical Modelling:

Using mathemetics to predict the out come of some complex real time problems which cannot be done in lab or in reality. Ex: population dynamics.

15. Functional Domains prediction:

Predicting the protein domains which are functionaly important from its protein sequence like active sites in a protein.

16. Motif Prediction /Pattern matching:

Predicting the motifs or motif clusters which are functionaly important.

Ex: regulatory motifs, Binding site motifs ...miRNA motics ..repeat motis ...Microsatellites are also a kind of motifs.

17. Protein - protein interaction:

Protein folding: One of the famous and most important and still unsolved problem.

18. Database development:

In some sense Bioinformatics is called as "Comparative Method". Because Bioinformatics depends on Databases for all of its analysis. So developing data base is a very important project. Many companies surviving by devloping and updating the databases.

NCBI , PDB and UCSC genome browser are some of the very important databases.

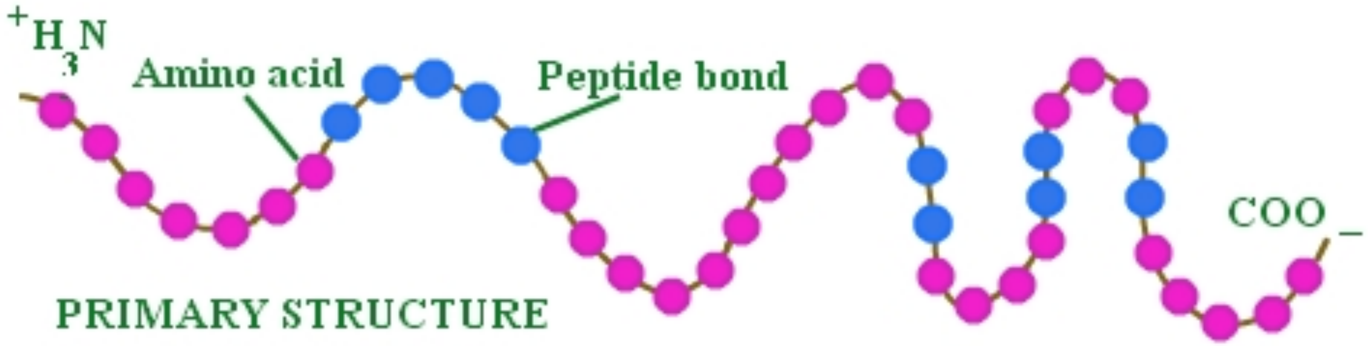
Proteins and Amino Acids

Protein: The word protein was coined by Berzelius in 1838 and was used by G. J. Mulder first time 1840. 15% of protoplasm is made up of protein. Average proteins contain 16% nitrogen, 50–55% carbon, oxygen 20–24%, hydrogen 7% and sulphur 0.3 – 0.5%. Iron, phosphorous, copper, calcium, and iodine are also present in small quantity.

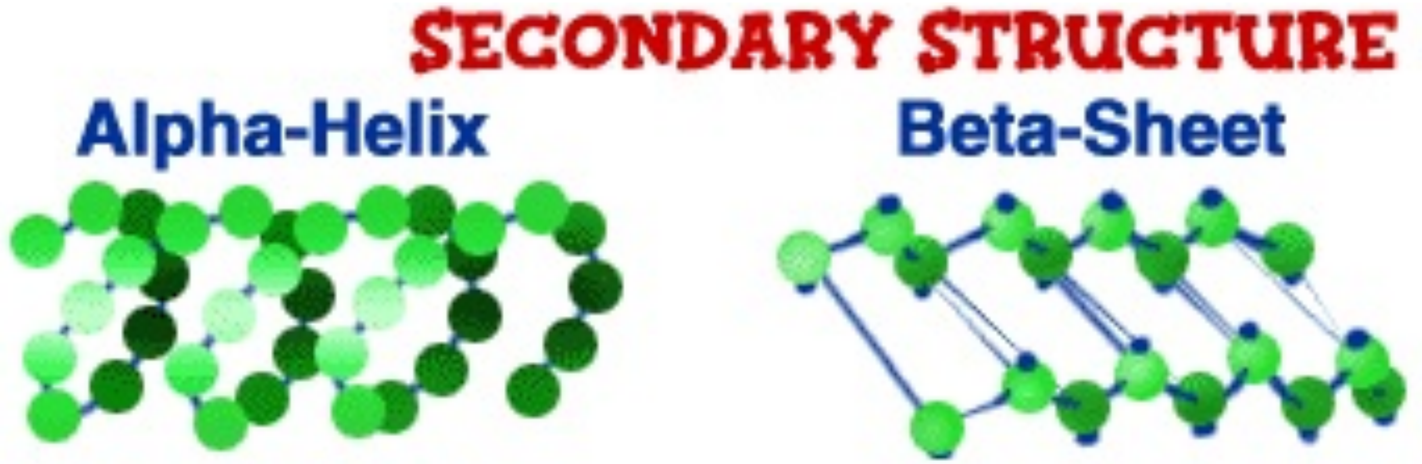
Structure of Proteins

It is due to different rearrangement of amino acids. When carboxyl group (-COOH) of one amino acid bonded with amino group (– NH_2) of another amino acid the bond is called peptide bond. A peptide may be dipeptide, tripeptide and polypeptide. The simplest protein is Insulin. According to Sanger (1953) insulin consists of 51 amino acids. A protein can have up to four level of conformation.

(i) **Primary structure :** The primary structure is the covalent connections of a protein. It refers to linear sequence, number and nature of amino acids bonded together with peptide bonds only. e.g. ribonuclease, insulin, haemoglobin, etc.

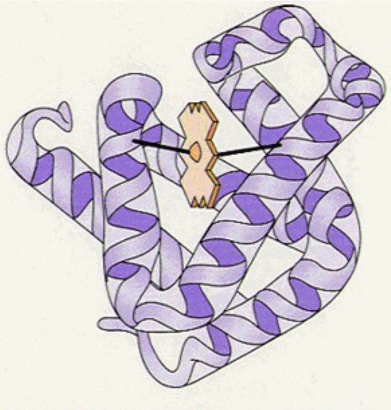


(ii) **Secondary structure :** The folding of a linear polypeptide chain into specific coiled structure (α -helix) is called secondary structure and if it is with intermolecular hydrogen bonds the structure is known as ß -pleated sheet. α -helical structure is found in protein of fur, keratin of hair claws, and feathers. ß -pleated structure is found in silk fibres.



(iii) **Tertiary structure :** The arrangement and interconnection of proteins into specific loops and bends is called tertiary structure of proteins. It is stabilized by hydrogen bond, ionic bond, hydrophobic bond and disulphide bonds. It is found in myoglobin (globular proteins).

(iv) **Quaternary structure :** It is shown by protein containing more than one peptide chain. The protein consists of identical units. It is known as homologous quaternary structure e.g. lactic dehydrogenase. If the units are dissimilar, it is called as heterogeneous quaternary structure e.g. hemoglobin which consists of two α -chains and two ß -chains.



Myoglobin, a globular protein

Classification of Proteins

Proteins are classified on the basis of their shape, constitution and function.

On the basis of shape

- **Fibrous protein/Scleroprotein :** Insoluble in water. Animal protein resistant to proteolytic enzyme is spirally coiled thread like structure form fibres. e.g. collagen (in connective tissue), actin and myosin, keratin in hairs, claws, feathers, etc.
- **Globular proteins :** Soluble in water. Polypeptides coiled about themselves to form oval or spherical molecules e.g. albumin insulin hormones like ACTH, oxytosin, etc.

On the basis of constituents

(i) **Simple proteins :** The proteins which are made up of amino acids only. e.g. albumins, globulins, prolamins, glutelins, histones, etc.

(ii) **Conjugated proteins :** These are complex proteins combined with characterstic non–amino acid substance called as prosthetic group. These are of following types :–

- **Nucleoproteins :** Combination of protein and nucleic acids, found in chromosomes and ribosomes. e.g. deoxyribonucleoproteins, ribonucleoproteins, etc.
- **Mucoproteins :** These are combined with large amount (more than 4%) of carbohydrates e.g. mucin.
- **Glycoproteins :** In this, carbohydrate content is less (about 2 – 3%) e.g. immunoglobulins or antibiotics.
- **Chromoproteins :** These are compounds of protein and coloured pigments. e.g. haemoglobin, cytochrome, etc.
- **Lipoproteins :** These are water soluble proteins and contain lipids. e.g. cholesterol and serum lipoproteins.
- **Metalloprotein :** These are metal binding proteins, AB_1–globin known as transferring is capable of combining with iron, zinc and copper e.g. chlorophyll.
- **Phosphoprotein :** They composed of protein and phosphate e.g. casein (milk) and vitellin (egg).

(iii) **Derived proteins :** When proteins are hydrolysed by acids, alkalies or enzymes, the degredation products obtained from them are called derived proteins. On the basis of progressive cleavage, derived proteins are classified as primary proteoses, secondary proteoses, peptones, polypeptides, amino acids, etc.

On the basis of nature of molecules

- **Acidic proteins :** They exist as anion and include acidic amino acids. e.g. blood groups.
- **Basic proteins :** They exist as cations and rich in basic amino acids e.g. lysine, arginine etc.

Function of Proteins

a. Proteins occur as food reserves as glutelin, globulin casein in milk.

b. Proteins are coagulated in solutions, alkaline to the isoelectric pH by positive ions such as Zn^{2+}, Cd^{2+}, Hg^{2+} etc. Casein – pH 4.6, cyt. C – 9.8, resum globulin 5.4, pepsin 2.7, lysozyme 11.0 etc.

c. Proteins are the most diverse molecule on the earth.

d. Proteins work as hormone as insulin and glucagon.

e. Antibiotics as gramicidin, tyrocidin and penicillin are peptides.

f. They are structural component of cell.

g. They are biological buffers.

h. Monellin is the sweetest substance obtained from African berry (2000 time sweeter than sucrose).

i. Proteins helps in defence, movement activity of muscles, visual pigments receptor molecules, etc.

j. Natural silk is a polyamide and artificial silk is a polysaccharide. Nitrogen is the basic constituent.

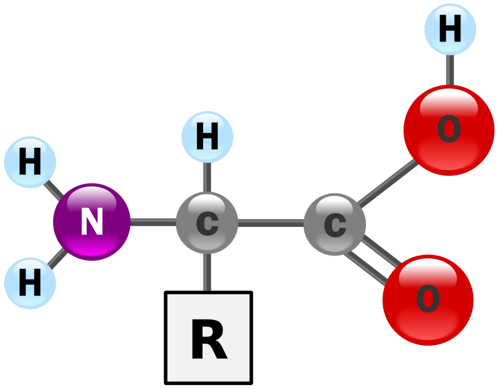
Amino Acids

Amino acids are normal components of cell proteins (called amino acid). They are 20 in number specified in genetic code and universal in viruses, prokaryotes and eukaryotes. Otherwise amino acids may be termed rare amino acids, which take part in protein synthesis e.g. hydroxyproline and non-protein amino acids do not take part in protein synthesis e.g. Ornithin, citrullin, gama-aminobutyric acid (GABA) a neurotransmitter, etc.

Structure and Composition

Amino acids are basic units of protein and made up of C, H, O, N and sometimes S. Amino acids are organic acids with a carboxyl group (–COOH) and one amino group ($-NH_2$) on the a -carbon atom. Carboxyl group attributes acidic properties and amino group gives basic ones. In solution, they serve as buffers and help to maintain pH. General formula is $R-CHNH_2.COOH$.

Amino acids are amphoteric or bipolar ions or Zueitter ions. Amino acids link with each other by peptide bond and long chains are called polypeptide chains.



Classification

Based on R-group of amino acids.

i. **Simple amino acids** : These have no functional group in the side chain. e.g. glycine, alanine , leucine, valine etc.

ii. **Hydroxy amino acids** : They have alcohol group in side chain. e.g. threonine, serine, etc.

iii. **Sulphur containing amino acids** : They have sulphur atom in side chain. e.g. methionine, cystenine.

iv. **Basic amino acids** : They have basic group (-NH_2) in side chain. e.g. lysine, arginine.

v. **Acidic amino acids** : They have carboxyl group in side chain. e.g. aspartic acid, glutamic acid.

vi. **Acid amide amino acids :** These are the derivatives of acidic amino acids. In this group, one of the carboxyl group has been converted to amide (-$CONH_2$). e.g. asparagine, glutamine.

vii. **Heterocyclic amino acids :** These are the amino acids in which the side chain includes a ring involving at least one atom other than carbon. e.g. tryptophan, histidine.

viii. **Aromatic amino acids :** They have aromatic group (benzene ring) in the side chain. e.g. phenylalanine, tyrosine, etc.

On the basis of requirements : On the basis of the synthesis amino acids in body and their requirement, they are categorized as :–

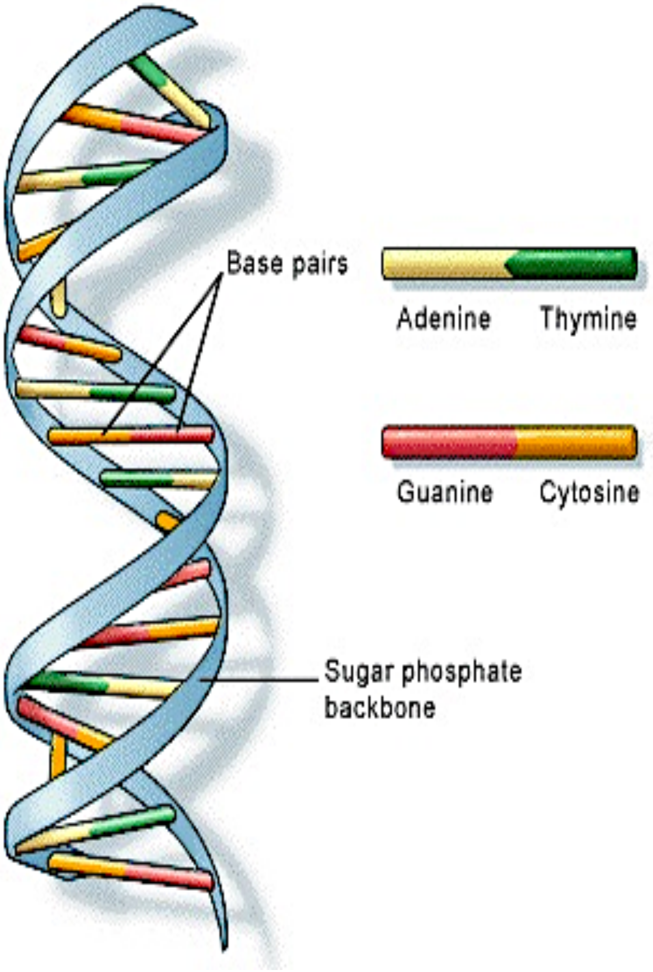
(a) **Essential amino acids :** These are not synthesized in body hence to be provided in diet e.g. valine, leucine, isoleucine, theronine ,lysine, etc.

(b) **Semi-essential amino acids :** Synthesized partially in the body but not at the rate to meet the requirement of individual. e.g., arginine and histidine.

(c) **Non-essential amino acids :** These amino acids are derived from carbon skeleton of lipids and carbohydrate metabolism. In humans there are 12 non- essential amino acids e.g. alanine, aspartic acid, cysteine, glutamic acid etc. Proline and hydroxyproline have, NH (imino group) instead of NH_2 hence are called imino acids. Tyrosine can be converted into hormone thyroxine and adrenaline and skin pigment melanin. Glycine is necessary for production of heme. Tryptophan is the precursor of vitamin nicotinamide and auxins. If amino group is removed from amino acid it can form glucose and if COOH group is removed, it forms amines e.g. histamine.

Nucleic Acids

- Nucleic acids are long chain macromolecules which are formed by end to end polymerization of large number of repeated units called nucleotides.
- Nucleic acids show a wide variety of secondary structures.
- There are two types of nucleic acids- deoxyribonucleic acid or DNA and ribonucleic acid or RNA.
- Structure
- DNA or deoxyribose nucleic acid is a helically twisted double chain polydeoxyribonucleotide macromolecule which constitutes the genetic material of all organisms with the exception of riboviruses.
- In prokaryotes it occurs in nucleoid and plasmids.this DNA is usually circular. In eucaryotes, most of the DNA is found in chromatin of nucleus.
- It is linear.
- Smaller quantities of DNA are found in mitochondria and plastids (organelle DNA).
- It may be circular or linear. Single-stranded DNA occurs as a genetic material in some viruses.
- Sense and antisense strands
- Both the strands of DNA do not take part in controlling heredity and metabolism. Only one of them does so.

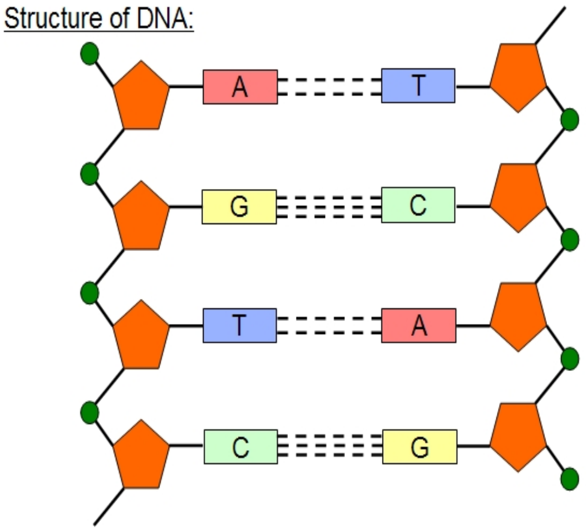


- The DNA strand which functions as template for RNA synthesis is known as template strand, minus (-) strand or antisense strand.
- Its complementary strand is named nontemplate strand, plus (+) strand, sense or coding strand.
- The latter name is given because by convention DNA genetic code is written according to its sequence.

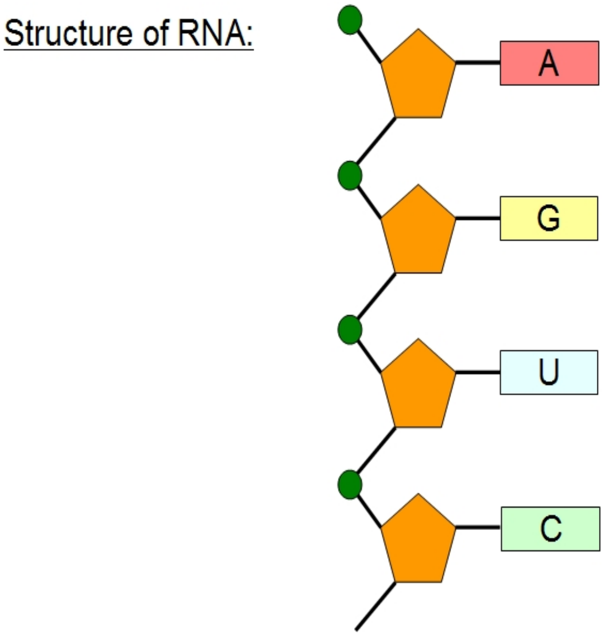
(5`) G C A T T C C G G C T A G T A A C (3') DNA Nontemplate, Sense (+) or coding Strand

(3') C G T A A G C C G A T C A T T G (5') DNA Template, Antisense, or Noncoding or (-) Strand

(5`) G C A U U C G G C U A G U A AC(3') RNA Transcript



- RNA is transcribed on 3’ -> 5’ (-) strand (template/antistrand) of DNA in 5←3 direction.
- The (+) strand of DNA is that coding strand which carries genetic information but is non-template.
- Ribonucleic acid (RNA)
- RNA is a single strand or chain (ds RNA is reported in wound tumour virus, Rice Dwarf virus) which is formed by end to end polymerization of a number of ribonucleotides or ribotides.
- The polynucleotide is characterized by the presence of ribose sugar and uracil instead of thymine.
- It is formed through transcription where one strand of DNA acts as a template.
- This is followed by processing to produce different types of RNAs.
- A strand of RNA contains 70-12000 ribonucleotides.
- At places it may appear partially double stranded due to folding of single strand.
- Four types of ribonucleotides occur in RNA. They are adenosine monophosphate, guanosine monophosphate, uridine monophosphate and cytidine monophosphate.



(1) Ribosomal RNA (rRNA).

It is the most abundant RNa (70-80% of total) which has 3-4 types. Some of its types (23S,28S, are the longest of all RNAs. As the name indicates, r RNA is a constitutuent of ribosomes. Here it lies coiled in between and over the protein molecules.

Functions: r RNAs bind protein molecules and give rise to ribosomes. 3’ end of 18S r RNA (16S in prokaryotes) has nucleotides complementary to those of cap region of m RNA. 5S r RNA and surrounding protein complex provide binding site for tRNA.

(2) Transfer RNA (t RNa).

It is also called soluble or sRNA. There are over 100 types of tRNAs. Transfer RNA constitutesabout15% of the total RNA. tRN is the smallest RNA with 70-85 nucleotides and sedimentationcoefficient of 4S.

Functions: tRNA is adapter molecule which is meant for transferring amino acids to ribosomes for synthesis of polypeptides. There are different tRNAs for different amino acids. Some amino acids can be picked up by 2-6 tRNAs. tRNAs carry specific amino acids at particular points during polypeptide synthesis as per cidons of mRNA. Codons are recognized by anticodons of tRNAs. Specific amino acids are recognized by particular activating or aminoacyl synthetase enzymes.

(3) Messenger RNA (mRNA).

It is a long RNA which constitutes 2-5% of the total RNA content. it brings instructions from the DNA for the formation of particular type of polypeptide. The instructionsare present in the base sequence of its nucleotides. It is called genetic code. Three adjacent nitrogen bases specify a particular amino acid.

(4) Genetic RNA.

It is found in some viruses called riboviruses. genetic RNA may be single stranded (e.g., Tobacco Mosaic Virus or TMV) or double stranded(e.g., Rheovirus). Genetic RNA acts as a hereditary material. It may, however, not replicate directly, but form DNA in the host cell to produce RNA of its own types.

Differences Between DNA and RNA		
S.No.	DNA	RNA
(1)	It usually occurs inside nucleus and some cell organelles.	Very little RNA occurs inside nucleus. Most of it is found in the cytoplasm.
(2)	DNA is the genetic material.	RNAis not the genetic material except in certain viruses,e.g.,Reovirus.
(3)	It isdouble stranded with the exception of some viruses (e.g., phage f × 174).	RNA is single stranded with the exception of some viruses (e.g., double stranded inReovirus).
(4)	DNA contains over a million nucleotides.	Depending upon the type, RNA contains 70-12000 nucleotides.
(5)	Molecular weight ranges from 3-4 million in Escherichia coli to 263 million in chromosome 1 of human beings.	Molecular weight ranges from 2500-2,000,000.
(6)	It is fuelgen positive.	RNa is Fuelgen negative.
(7)	It contains deoxyribose sugar.	It contains ribose sugar.
(8)	Nitrogen base thymine occurs in DNa alongwith three others – adenine, cytosine and guanine.	Thymine is replaced by uracil in RNA. The other three are similar – adenine, cytosine and guanine.
(9)	It replicate to form new DNA molecules.	It can not normally replicate itself.
(10)	DNAconrols metabolism and genetics including variations.	RNA translates the transcribed message for forming polypeptides.
(11)	Its quantity is fixed for cell.	The quantity of RNA of a cell is variable.
(12)	DNA controls metabolism and genetics including variations.	It only controls metabolism under instructios from DNA.
(13)	Purine and pyrimidine bases are in equal number.	There is no proportionality between number of purines and purimidine bases.
(14)	It is long lived.	Some RNAs are very short lived some are longer life.

Data

In computer science, data is anything in a form suitable for use with a computer. Data is often distinguished from programs. A program is a set of instructions that detail a task for the computer to perform. In this sense, data is thus everything that is not program code.

Type of data

- nucleotide sequences
- protein sequences
- proteins sequence patterns or motifs
- macromolecular 3D structure
- gene expression data
- metabolic pathways

Database

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images.

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.

Types of Biological Databases

There are two common concepts of biological databases: Primary Databases and Secondary Databases. These two differ in their archive structure. Primary databases often hold only one type of specific data which is stored in their own archive. They upload new data explored in experiments and update entries to ensure the quality of the data.

Secondary databases are databases, which use other databases as their source of information, thus they get their data by requesting other databases. They often already process or analyze the data matching the corresponding request to get new results.

NCBI

The late Senator Claude Pepper recognized the importance of computerized information processing methods for the conduct of biomedical research and sponsored legislation that established the National Center for Biotechnology Information (NCBI) on November 4, 1988, as a division of the National Library of Medicine (NLM) at the National Institutes of Health (NIH). NLM was chosen for its experience in creating and maintaining biomedical databases, and because as part of NIH, it could establish an intramural research program in computational molecular biology. The collective research components of NIH make up the largest biomedical research facility in the world.

Basic Research

As a national resource for molecular biology information, NCBI's mission is to develop new information technologies to aid in the understanding of fundamental molecular and genetic processes that control health and disease. More specifically, the NCBI has been charged with creating automated systems for storing and analyzing knowledge about molecular biology, biochemistry, and genetics; facilitating the use of such databases and software by the research and medical community; coordinating efforts to gather biotechnology information both nationally and internationally; and performing research into advanced methods of computer-based information processing for analyzing the structure and function of biologically important molecules.

To carry out its diverse responsibilities, NCBI:

- Conducts research on fundamental biomedical problems at the molecular level using mathematical and computational methods
- Maintains collaborations with several NIH institutes, academia, industry, and other governmental agencies
- Fosters scientific communication by sponsoring meetings, workshops, and lecture series
- Supports training on basic and applied research in computational biology for postdoctoral fellows through the NIH Intramural Research Program
- Engages members of the international scientific community in informatics research and training through the Scientific Visitors Program
- Develops, distributes, supports, and coordinates access to a variety of databases and software for the scientific and medical communities
- Develops and promotes standards for databases, data deposition and exchange, and biological nomenclature

EBI

In 1992, EMBL Council voted to establish the EMBL-European Bioinformatics Institute (EMBL-EBI) and locate it on the Wellcome Trust Genome Campus in Hinxton, UK, where it would be in close proximity to the major sequencing efforts at the Wellcome Trust Sanger Institute.

The transition of two major bioinformatics services from Heidelberg to Hinxton began in 1992 and in September 1994, EMBL-EBI was firmly established in the UK. The European Nucleotide Archive and the protein sequence resource UniProt (then known as Swiss-Prot–TrEMBL) were the original EMBL-EBI databases. Since then, the EMBL-EBI has played a major part in the bioinformatics revolution.

EBI now provide the world's most comprehensive range of molecular databases and offer an extensive user training programme. Our basic research programme has grown substantially, and remains closely tied with the evolution of our resources.

EMBL-EBI maintain the world's most comprehensive range of freely available molecular data resources. Developed in collaboration with our colleagues worldwide, our databases and tools help scientists share data efficiently, perform complex queries and analyse the results in different ways. Our work supports millions of researchers, who are wet-lab and computational biologists working in all areas of the life sciences, from biomedicine to biodiversity and agri-food research.

PDB

The Protein Data Bank (PDB) archive is the single worldwide repository of information about the 3D structures of large biological molecules, including proteins and nucleic acids. These are the molecules of life that are found in all organisms including bacteria, yeast, plants, flies, other animals, and humans. Understanding the shape of a molecule deduce a structure's role in human health and disease, and in drug development. The structures in the archive range from tiny proteins and bits of DNA to complex molecular machines like the ribosome.

The PDB archive is available at no cost to users. The PDB archive is updated weekly.

The PDB was established in 1971 at Brookhaven National Laboratory under the leadership of Walter Hamilton and originally contained 7 structures. After Hamilton's untimely death, Tom Koetzle began to lead the PDB in 1973, and then Joel Sussman in 1994. In 1998, the Research Collaboratory for Structural Bioinformatics (RCSB) became responsible for the management of the PDB. In 2003, the wwPDB was formed to maintain a single PDB archive of macromolecular structural data that is freely and publicly available to the global community. It consists of organizations that act as deposition, data processing and distribution centers for PDB data.

DDBJ

DDBJ Center collects nucleotide sequence data as a member of INSDC (International Nucleotide Sequence Database Collaboration) and provides freely available nucleotide sequence data and supercomputer system, to support research activities in life science.

It is generally accepted that research in biology today requires both computer and experimental equipment equally well. Information achieved from enormous exhaustive data have greatly contributed to the paradigm shift in biology. Biology or life sciences are no longer restricted to wet-bench experiments. In silico and in vitro / in vivo analyses together will push back the frontiers of life sciences. In particular, researchers in life science must rely on computers to analyze nucleotide sequence data accumulating at a remarkably rapid rate. Actually, this triggered the birth and development of information biology. DDBJ Center is to play a major role in carrying out research in information biology and to run DDBJ operation in the world.

The principal purpose of DDBJ operations is to improve the quality of INSD, as public domains. When researchers make their data open to the public through INSD and commonly shared in world wide, we at DDBJ Center make efforts to describe information on the data as rich as possible, according to the unified rules of INSD, preferably without any stress by using DDBJ.

Nucleotide sequence records organismic evolution more directly than other biological materials and thus is invaluable not only for research in life sciences but also human welfare in general. The database is, so to speak, a common treasure of human beings. With this in mind, we make the database online accessible to anyone in the world.

Asymptotic analysis

Asymptotic analysis of an algorithm refers to defining the mathematical boundation/framing of its run-time performance. Using asymptotic analysis, we can very well conclude the best case, average case, and worst case scenario of an algorithm.

Asymptotic analysis is input bound i.e., if there's no input to the algorithm, it is concluded to work in a constant time. Other than the "input" all other factors are considered constant.

Asymptotic analysis refers to computing the running time of any operation in mathematical units of computation. For example, the running time of one operation is computed as $f(n)$ and may be for another operation it is computed as $g(n^2)$. This means the first operation running time will increase linearly with the increase in **n** and the running time of the second operation will increase exponentially when **n** increases. Similarly, the running time of both operations will be nearly the same if **n** is significantly small.

Usually, the time required by an algorithm falls under three types −

- **Best Case** − Minimum time required for program execution.
- **Average Case** − Average time required for program execution.
- **Worst Case** − Maximum time required for program execution.

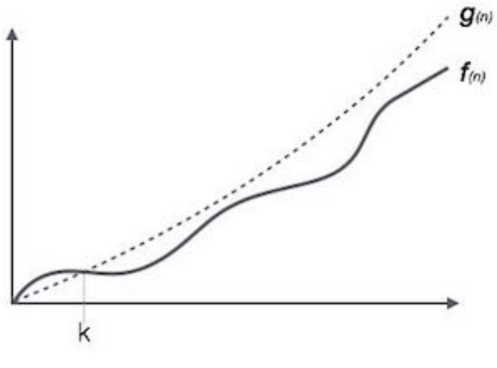
Asymptotic Notations

Following are the commonly used asymptotic notations to calculate the running time complexity of an algorithm.

- Ο Notation
- Ω Notation
- θ Notation

Big Oh Notation, Ο

The notation Ο(n) is the formal way to express the upper bound of an algorithm's running time. It measures the worst case time complexity or the longest amount of time an algorithm can possibly take to complete.

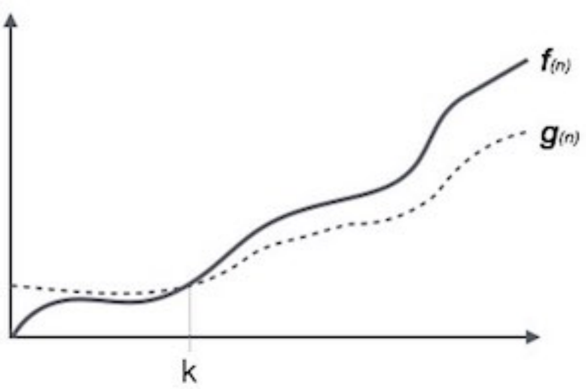


For example, for a function ***f*(n)**

$O(f(n)) = \{ g(n) : \text{there exists } c > 0 \text{ and } n_0 \text{ such that } f(n) \leq c.g(n) \text{ for all } n > n_0. \}$

Omega Notation, Ω

The notation Ω(n) is the formal way to express the lower bound of an algorithm's running time. It measures the best case time complexity or the best amount of time an algorithm can possibly take to complete.

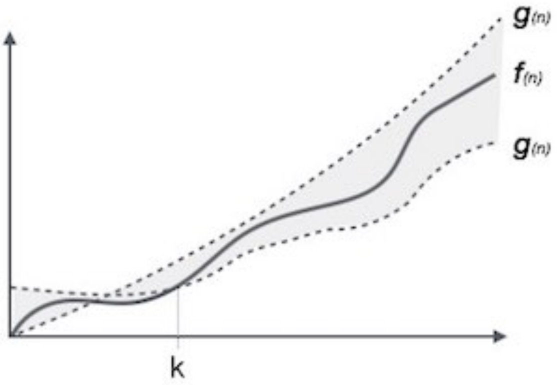


For example, for a function ***f*(n)**

$\Omega(f(n)) \geq \{\, g(n) : \text{there exists } c > 0 \text{ and } n_0 \text{ such that } g(n) \leq c.f(n) \text{ for all } n > n_0. \,\}$

Theta Notation, θ

The notation θ(n) is the formal way to express both the lower bound and the upper bound of an algorithm's running time. It is represented as follows −



$\theta(f(n)) = \{\, g(n) \text{ if and only if } g(n) = O(f(n)) \text{ and } g(n) = \Omega(f(n)) \text{ for all } n > n_0. \,\}$

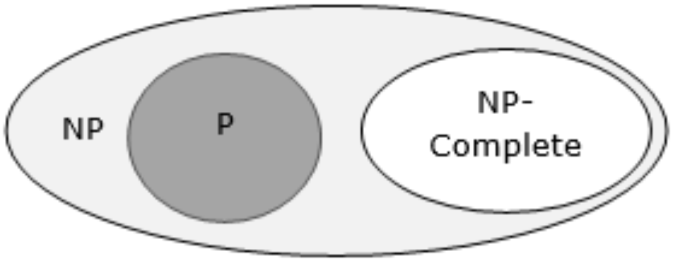
Common Asymptotic Notations

Following is a list of some common asymptotic notations −

constant	−	O(1)
logarithmic	−	O(log n)
linear	−	O(n)
n log n	−	O(n log n)
quadratic	−	$O(n^2)$
cubic	−	$O(n^3)$
polynomial	−	$n^{O(1)}$
exponential	−	$2^{O(n)}$

NP Complete

A problem is in the class NPC if it is in NP and is as **hard** as any problem in NP. A problem is **NP-hard** if all problems in NP are polynomial time reducible to it, even though it may not be in NP itself.



If a polynomial time algorithm exists for any of these problems, all problems in NP would be polynomial time solvable. These problems are called **NP-complete**. The phenomenon of NP-completeness is important for both theoretical and practical reasons.

Definition of NP-Completeness

A language **B** is ***NP-complete*** if it satisfies two conditions

- **B** is in NP
- Every **A** in NP is polynomial time reducible to **B**.

If a language satisfies the second property, but not necessarily the first one, the language **B** is known as **NP-Hard**. Informally, a search problem **B** is **NP-Hard** if there exists some **NP-Complete** problem **A** that Turing reduces to **B.**

The problem in NP-Hard cannot be solved in polynomial time, until **P = NP**. If a problem is proved to be NPC, there is no need to waste time on trying to find an efficient algorithm for it. Instead, we can focus on design approximation algorithm.

NP-Complete Problems

Following are some NP-Complete problems, for which no polynomial time algorithm is known.

- Determining whether a graph has a Hamiltonian cycle
- Determining whether a Boolean formula is satisfiable, etc.

NP-Hard Problems

The following problems are NP-Hard

- The circuit-satisfiability problem
- Set Cover

- Vertex Cover
- Travelling Salesman Problem

In this context, now we will discuss TSP is NP-Complete

TSP is NP-Complete

The traveling salesman problem consists of a salesman and a set of cities. The salesman has to visit each one of the cities starting from a certain one and returning to the same city. The challenge of the problem is that the traveling salesman wants to minimize the total length of the trip

Brute force algorithm

Brute force is a straightforward approach to solve a problem based on the problem's statement and definitions of the concepts involved. It is considered as one of the easiest approach to apply and is useful for solving small–size instances of a problem. Example 1: Computing a^n ($a > 0$, n a nonnegative integer) based on the definition of exponentiation $a^n = a^* a^* a^* \dots * a$ The brute force algorithm requires n-1 multiplications. The recursive algorithm for the same problem, based on the observation that $a^n = a^{n/2} * a^{n/2}$ requires $\Theta(\log(n))$ operations. Example 2: Computing n! based on the definition n! = 1*2*3*…*n The algorithm requires Θ (n) operations.

Brute-Force Search and Sort

Sequential search in an unordered array and simple sorts – selection sort, bubble sort are brute force algorithms. Sequential search: the algorithm simply compares successive elements of a given list with a given search key until either a match is found or the list is exhausted without finding a match.

```
Algorithm SequentialSearch (A[0..n], K)
A[n] ← K
    i ← 0
    While A [i] ≠ K do
        i ← i + 1
    if  i < n return i
    else  return -1
```

The complexity of a sequential search algorithm is $\Theta(n)$ in the worst possible case and $\Theta(1)$ in the best possible case, depending on where the desired element is situated. Selection sort: the entire given list of *n* elements is scanned to find its smallest element and exchange it with the first element. Thus, the smallest element is moved to its final position in the sorted list. Then, the list is scanned again, starting with the second element in order to find the smallest element among the *n – 1* and exchange it with the second element. The second smallest element is put in its final position in the sorted list. After *n-1* passes, the list is sorted.

```
Algorithm SelectionSort (A[0..n-1])
for i ← 0 to  n-2 do
     min ← i
     for j ← i + 1 to n-1 do
          if A[j] < A[min]
       min ← j
     swap A[i] and A[min]
```

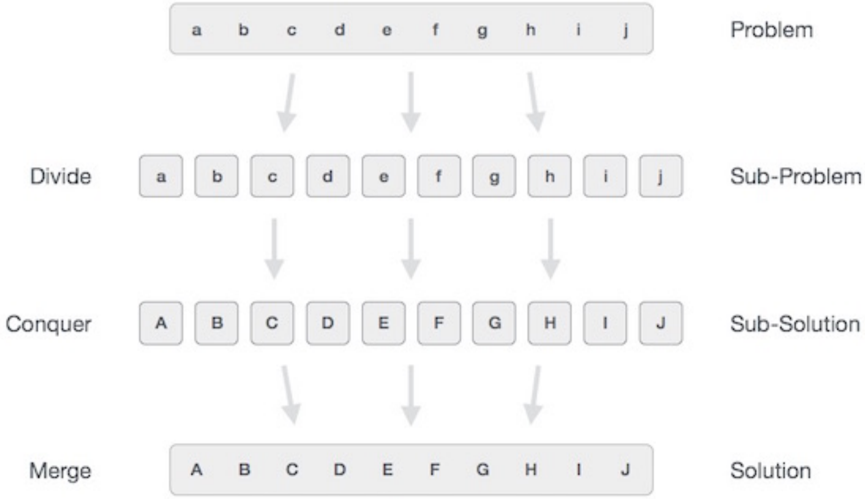
The basic operation of the selection sort is the comparison - A[j] < A[min]. The complexity of the algorithm is $\Theta(n^2)$ and the number of key swaps is $\Theta(n)$. Bubble sort is another application of a brute force. In the algorithm, adjacent elements of the list are compared and are exchanged if they are out of order. Algorithm BubbleSort (A[0..n-1])

```
for i ← 0 to  n-2 do
     for j ← 0 to n – 2 – i do
          if A[j+1] < A[j]
       swap A[j] and A[j+1]
```

The basic operation of the bubble sort is comparison - A[j+1] < A[j] and swapping - swap A[j] and A[j+1]. The number of key comparisons is the same for all arrays of size n and it is $\Theta(n^2)$. However, the number of key swaps depends on the input and in the worst case is $\Theta(n^2)$. The above implementation of Bubble sort can be slightly improved if we stop the execution of the algorithm when a pass through the list makes no exchanges (i.e. indicating that the list has been sorted). Thus, in the best case the complexity will be $\Theta(n)$ and the worst case $\Theta(n^2)$.

Divide and Conquer algorithm

In divide and conquer approach, the problem in hand, is divided into smaller sub-problems and then each problem is solved independently. When we keep on dividing the subproblems into even smaller sub-problems, we may eventually reach a stage where no more division is possible. Those "atomic" smallest possible sub-problem (fractions) are solved. The solution of all sub-problems is finally merged in order to obtain the solution of an original problem.



Broadly, we can understand **divide-and-conquer** approach in a three-step process.

Divide/Break

This step involves breaking the problem into smaller sub-problems. Sub-problems should represent a part of the original problem. This step generally takes a recursive approach to divide the problem until no sub-problem is further divisible. At this stage, sub-problems become atomic in nature but still represent some part of the actual problem.

Conquer/Solve

This step receives a lot of smaller sub-problems to be solved. Generally, at this level, the problems are considered 'solved' on their own.

Merge/Combine

When the smaller sub-problems are solved, this stage recursively combines them until they formulate a solution of the original problem. This algorithmic approach works recursively and conquer & merge steps works so close that they appear as one.

Examples

The following computer algorithms are based on **divide-and-conquer** programming approach –

- Merge Sort
- Quick Sort
- Binary Search
- Strassen's Matrix Multiplication
- Closest pair (points)

There are various ways available to solve any computer problem, but the mentioned are a good example of divide and conquer approach.

Sorting

Sorting refers to arranging data in a particular format. Sorting algorithm specifies the way to arrange data in a particular order. Most common orders are in numerical or lexicographical order.

The importance of sorting lies in the fact that data searching can be optimized to a very high level, if data is stored in a sorted manner. Sorting is also used to represent data in more readable formats. Following are some of the examples of sorting in real-life scenarios −

- **Telephone Directory** − The telephone directory stores the telephone numbers of people sorted by their names, so that the names can be searched easily.
- **Dictionary** − The dictionary stores words in an alphabetical order so that searching of any word becomes easy.

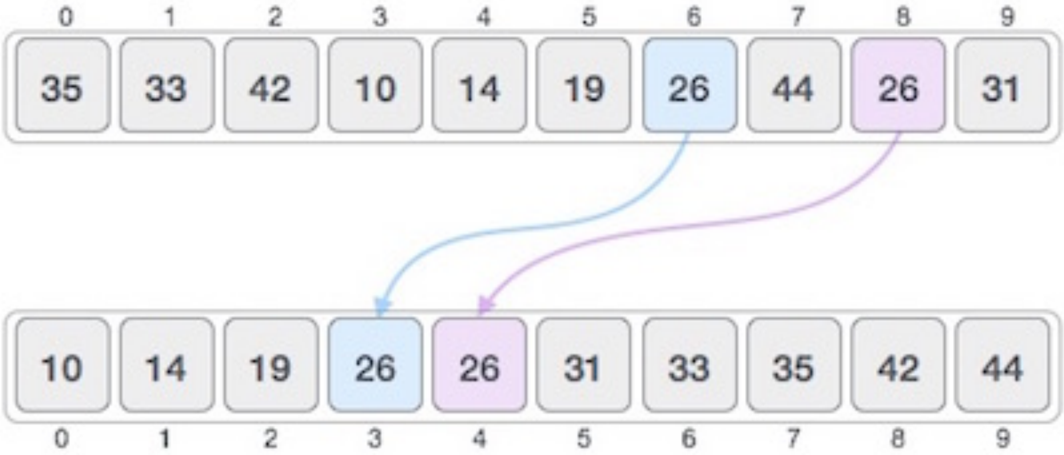
In-place Sorting and Not-in-place Sorting

Sorting algorithms may require some extra space for comparison and temporary storage of few data elements. These algorithms do not require any extra space and sorting is said to happen in-place, or for example, within the array itself. This is called **in-place sorting**. Bubble sort is an example of in-place sorting.

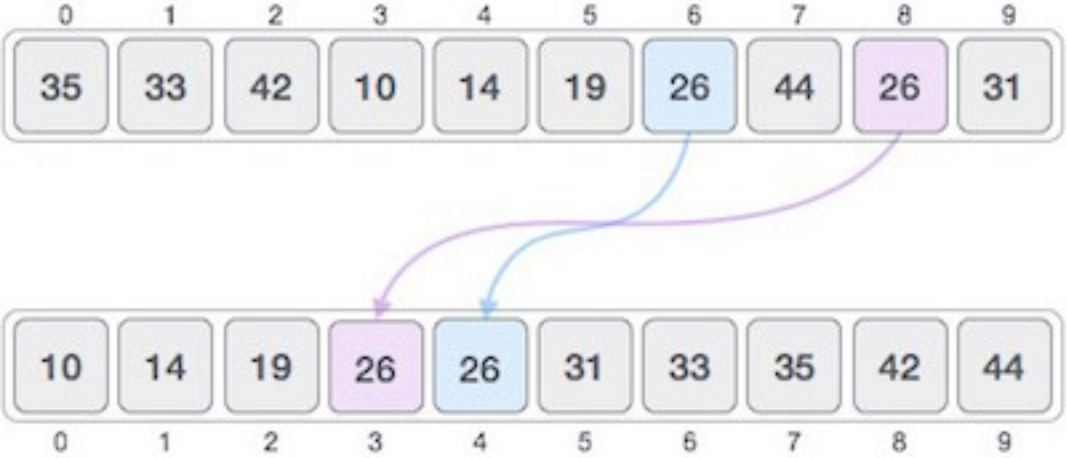
However, in some sorting algorithms, the program requires space which is more than or equal to the elements being sorted. Sorting which uses equal or more space is called **not-in-place sorting**. Merge-sort is an example of not-in-place sorting.

Stable and Not Stable Sorting

If a sorting algorithm, after sorting the contents, does not change the sequence of similar content in which they appear, it is called **stable sorting**.



If a sorting algorithm, after sorting the contents, changes the sequence of similar content in which they appear, it is called **unstable sorting**.



Stability of an algorithm matters when we wish to maintain the sequence of original elements, like in a tuple for example.

Adaptive and Non-Adaptive Sorting Algorithm

A sorting algorithm is said to be adaptive, if it takes advantage of already 'sorted' elements in the list that is to be sorted. That is, while sorting if the source list has some element already sorted, adaptive algorithms will take this into account and will try not to re-order them.

A non-adaptive algorithm is one which does not take into account the elements which are already sorted. They try to force every single element to be re-ordered to confirm their sortedness.

Important Terms

Some terms are generally coined while discussing sorting techniques, here is a brief introduction to them −

Increasing Order

A sequence of values is said to be in **increasing order**, if the successive element is greater than the previous one. For example, 1, 3, 4, 6, 8, 9 are in increasing order, as every next element is greater than the previous element.

Decreasing Order

A sequence of values is said to be in **decreasing order**, if the successive element is less than the current one. For example, 9, 8, 6, 4, 3, 1 are in decreasing order, as every next element is less than the previous element.

Non-Increasing Order

A sequence of values is said to be in **non-increasing order**, if the successive element is less than or equal to its previous element in the sequence. This order occurs when the sequence contains duplicate values. For example, 9, 8, 6, 3, 3, 1 are in non-increasing order, as every next element is less than or equal to (in case of 3) but not greater than any previous element.

Non-Decreasing Order

A sequence of values is said to be in **non-decreasing order**, if the successive element is greater than or equal to its previous element in the sequence. This order occurs when the sequence contains duplicate values. For example, 1, 3, 3, 6, 8, 9 are in non-decreasing order, as every next element is greater than or equal to (in case of 3) but not less than the previous one.

UNIT –II

Sequence similarity

Sequence similarity is a measure of an empirical relationship between sequences. A common objective of sequence similarity calculations is establishing the likelihood for sequence homology: the chance that sequences have evolved from a common ancestor.

Sequence Similarity Searching is a method of searching sequence databases by using alignment to a query sequence. By statistically assessing how well database and query sequences match one can infer homology and transfer information to the query sequence.

Homology

Similarity in the sequence of two genes, from different organisms, that share a common evolutionary origin. Often used, more loosely (as in homologous recombin- ation), to describe DNA molecules with a sequence that is sufficiently similar for complementary strands to hybridize, without evidence of common evolutionary origin or function.

Understanding Homologous Genes

Orthologous and paralogous genes are different types of homologous genes. Homologous genes are two or more genes that descend from a common ancestral deoxyribonucleic acid (DNA) sequence. An example of homologous genes are a bat wing and a bear arm; both retain similar features and are utilized in similar manners. These traits, which were passed down from their last common ancestor, have adaptive pressures that may lead to variations within the gene. The point or event in evolutionary history that accounts for the DNA sequence variation within the gene determines whether the homologous genes are considered ‘ortho’ or ‘para’.

Orthologs

Homologous sequences are said to be orthologous when they are direct descendants of a sequence in the common ancestor, i.e., without having undergone a gene duplication event.

Orthologous Genes

Orthologous genes are homologous genes that diverged after a speciation event. The genes generally maintain a similar function to that of the ancestral gene in which they evolved from. In this type of homologous gene, the ancestral gene and its function is maintained through a speciation event, though variations may arise within the gene after the point in which the species diverged.

Paralogs

Homologous sequences in two organisms A and B that are descendants of two different copies of a sequence created by a duplication event in the genome of the common ancestor. They tend to have different functions.

Paralogous Genes

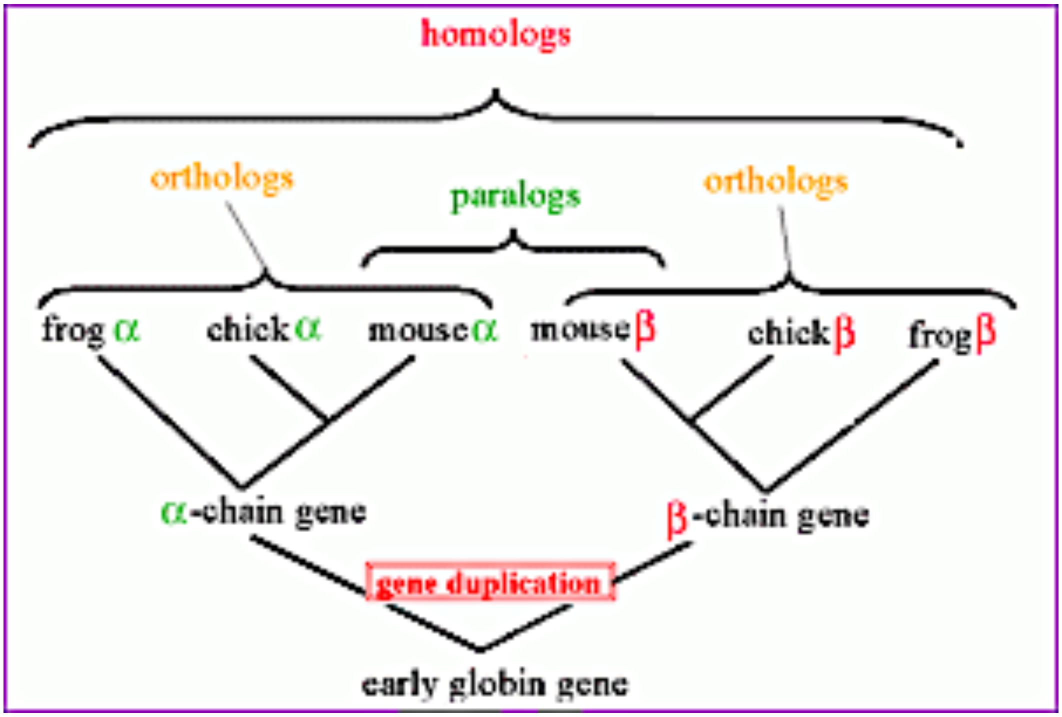
Paralogous genes are homologous genes that occur within one species and have diverged after a duplication event. Unlike orthologous genes, a paralogous gene is a new gene that holds a new function. These genes arise during gene duplication where one copy of the gene receives a mutation that gives rise to a new gene with a new function, though the function is often related to the role of the ancestral gene.

Orthologous Genes

Orthologous genes are homologous genes that diverged after a speciation event. The genes generally maintain a similar function to that of the ancestral gene in which they evolved from. In this type of homologous gene, the ancestral gene and its function is maintained through a speciation event, though variations may arise within the gene after the point in which the species diverged.

Examples of Paralogous and Orthlogous Genes

The genes that produce the hemoglobin and myoglobin proteins are homologous genes that have both orthologous and paralogous relationships. Both humans and dogs hold the genes for both hemoglobin and myglobin proteins, so we can infer that the hemoglobin and myoglobin genes evolved before human's and dog's last common ancestor. Myoglobin arose in this ancestral species as a paralogous gene to hemoglobin; myoglobin arose from a mutation in the hemoglobin gene during a duplication event and carries out a new, yet similar, function. Since divergence in human and dog hemoglobin did not occur until after speciation, these genes are orthologous. Human myoglobin and dog hemoglobin, however, are homologous genes that are neither paralogous or orthologous.



1.0Introductin

BLOSUM is stand from **BLO**cks **SU**bstitution **M**atrix. It was first calculated by Jorja G. Henikoff and Henikoff Steven in year 1992. This BLOSUM is derived from the block database which is based on comparisons of sequences Block. For the BLOSUM calculation, only amino acid sequences blocks that have small changes between them are considered. We called conserved blocks for all those blocks. The multiply aligned unggapped segments are contained in the Blocks database according to the most highly protein conserved regions.

The aim of this BLOSUM matrix is used to score the alignments between the evolutionarily divergent sequences of protein. Each BLOSUM Matrices exist using different alignment database. There are two types of alignments, which is global alignments and local alignments. For BLOSUM matrix, it based on local alignments not global alignments. PAM matrices is the one of matrices that uses global alignments which is it comparisons of closely related proteins. There are differences between the two types of alignments. Global alignments mean that it attempt to align every residue in every sequences or in other word, it align all letter (amino acid symbols) by compares one by one in sequence. While, local alignments mean that it align a part of sequence with a part of sequence.

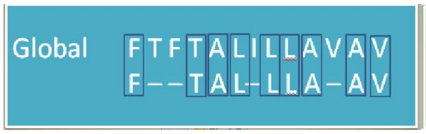


Figure 1: Global alignments

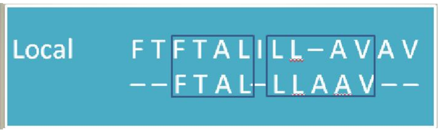


Figure 2: Local alignments

BLOSUM matrices are derived from blocks whose alignment corresponds to the BLOSUM-X. X alphabet after word BLOSUM is called numbers which mean that several set of BLOSUM matrices exist using different alignment database. For example, BLOSUM 62 is derived from Blocks that contain ungapped sequence alignment that less than 62% identity. If the BLOSUM matrices with high number, it shows that it less divergent alignments which mean, it closely related sequence. While, if the BLOSUM matrices with low numbers, it shows that it more divergent alignments which is distant related sequence. UsuallyBLAST useBLOSUM 62 as the default matrix for their standard protein.

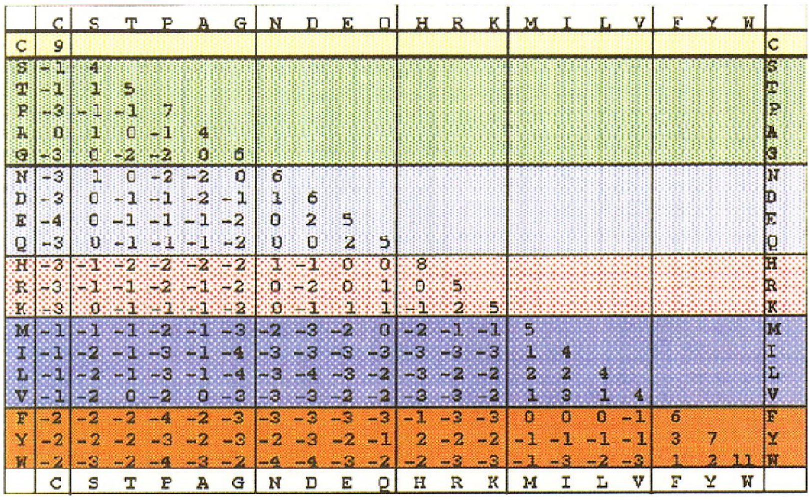
1.1 BLOSUM Version

1.1.1 BLOSUM 50

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-4	15	2	-3
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5

- This version will work better on scoring the distantly related sequences.
- For example, is good to score alignments of between species that are not closely related like butterfly and spider.

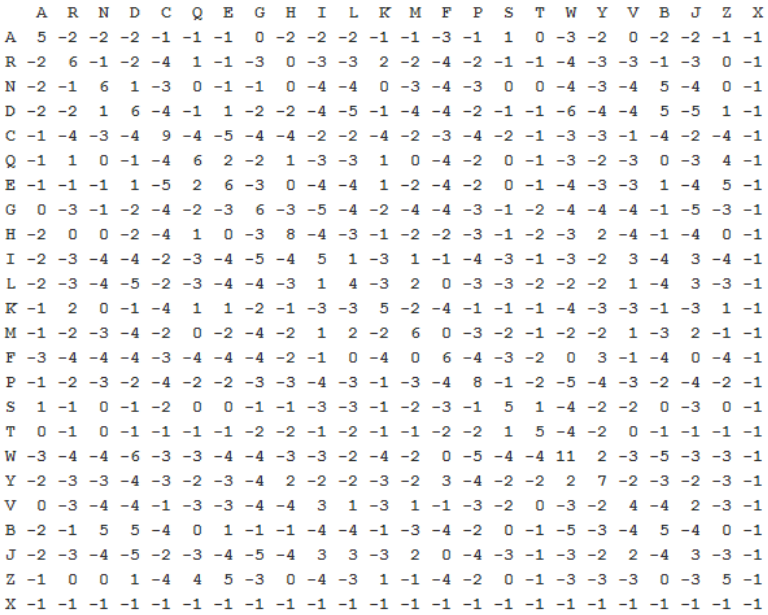
1.1.2 BLOSUM 62



	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	
C	9																				C
S	-1	4																			S
T	-1	1	5																		T
P	-3	-1	-1	7																	P
A	0	1	0	-1	4																A
G	-3	0	-2	-2	0	6															G
N	-3	1	0	-2	-2	0	6														N
D	-3	0	-1	-1	-2	-1	1	6													D
E	-4	0	-1	-1	-1	-2	0	2	5												E
Q	-3	0	-1	-1	-1	-2	0	0	2	5											Q
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8										H
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5									R
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5								K
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5							M
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4						I
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4					L
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4				V
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6			F
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7		Y
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11	W
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W	

- This version is better mainly in a function of how distant relationships are between the sequences.
- This will work better for scoring the closer relationship between sequences.
- It works better for scoring sequences that are both not too closely or distantly related ones, in between the BLOSUM 50 and BLOSUM 80.
- This is the standard version which is more commonly used.
- Is the default scoring matrix used in BLAST program.

1.1.3 BLOSUM 80



	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	J	Z	X
A	5	-2	-2	-2	-1	-1	-1	0	-2	-2	-2	-1	-1	-3	-1	1	0	-3	-2	0	-2	-2	-1	-1
R	-2	6	-1	-2	-4	1	-1	-3	0	-3	-3	2	-2	-4	-2	-1	-1	-4	-3	-3	-1	-3	0	-1
N	-2	-1	6	1	-3	0	-1	-1	0	-4	-4	0	-3	-4	-3	0	0	-4	-3	-4	5	-4	0	-1
D	-2	-2	1	6	-4	-1	1	-2	-2	-4	-5	-1	-4	-4	-2	-1	-1	-6	-4	-4	5	-5	1	-1
C	-1	-4	-3	-4	9	-4	-5	-4	-4	-2	-2	-4	-2	-3	-4	-2	-1	-3	-3	-1	-4	-2	-4	-1
Q	-1	1	0	-1	-4	6	2	-2	1	-3	-3	1	0	-4	-2	0	-1	-3	-2	-3	0	-3	4	-1
E	-1	-1	-1	1	-5	2	6	-3	0	-4	-4	1	-2	-4	-2	0	-1	-4	-3	-3	1	-4	5	-1
G	0	-3	-1	-2	-4	-2	-3	6	-3	-5	-4	-2	-4	-4	-3	-1	-2	-4	-4	-4	-1	-5	-3	-1
H	-2	0	0	-2	-4	1	0	-3	8	-4	-3	-1	-2	-2	-3	-1	-2	-3	2	-4	-1	-4	0	-1
I	-2	-3	-4	-4	-2	-3	-4	-5	-4	5	1	-3	1	-1	-4	-3	-1	-3	-2	3	-4	3	-4	-1
L	-2	-3	-4	-5	-2	-3	-4	-4	-3	1	4	-3	2	0	-3	-3	-2	-2	-2	1	-4	3	-3	-1
K	-1	2	0	-1	-4	1	1	-2	-1	-3	-3	5	-2	-4	-1	-1	-1	-4	-3	-3	-1	-3	1	-1
M	-1	-2	-3	-4	-2	0	-2	-4	-2	1	2	-2	6	0	-3	-2	-1	-2	-2	1	-3	2	-1	-1
F	-3	-4	-4	-4	-3	-4	-4	-4	-2	-1	0	-4	0	6	-4	-3	-2	0	3	-1	-4	0	-4	-1
P	-1	-2	-3	-2	-4	-2	-2	-3	-3	-4	-3	-1	-3	-4	8	-1	-2	-5	-4	-3	-2	-4	-2	-1
S	1	-1	0	-1	-2	0	0	-1	-1	-3	-3	-1	-2	-3	-1	5	1	-4	-2	-2	0	-3	0	-1
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-2	-1	-1	-2	-2	1	5	-4	-2	0	-1	-1	-1	-1
W	-3	-4	-4	-6	-3	-3	-4	-4	-3	-3	-2	-4	-2	0	-5	-4	-4	11	2	-3	-5	-3	-3	-1
Y	-2	-3	-3	-4	-3	-2	-3	-4	2	-2	-2	-3	-2	3	-4	-2	-2	2	7	-2	-3	-2	-3	-1
V	0	-3	-4	-4	-1	-3	-3	-4	-4	3	1	-3	1	-1	-3	-2	0	-3	-2	4	-4	2	-3	-1
B	-2	-1	5	5	-4	0	1	-1	-1	-4	-4	-1	-3	-4	-2	0	-1	-5	-3	-4	5	-4	0	-1
J	-2	-3	-4	-5	-2	-3	-4	-5	-4	3	3	-3	2	0	-4	-3	-1	-3	-2	2	-4	3	-3	-1
Z	-1	0	0	1	-4	4	5	-3	0	-4	-3	1	-1	-4	-2	0	-1	-3	-3	-3	0	-3	5	-1
X	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

- This version is better for scoring the closely related protein sequences.
- For example, is better to score alignments of species that are very closely related like chimpanzee and human.

1.1.4 Brief Explanations on BLOSUM Versions

Given condition like two sequences are identical at particular number of position in the sequence (conserved region) and they will be clustered or grouped in a block. Normally we will only look at the identical position within the conserved region which is ungapped where it usually gives the definition of biochemical function of a protein sequence. Every cluster has its particular level, so any other sequences that are identical at that conserved region that we are observing, will be put in the cluster with same level, believing they are similar in biochemical function without considering their species relatedness. To put it into simple, BLOSUM 62, the number 62 represents the clustering level, the level where the block is being placed. The matrix is designed from sequences with an identity level of 62%. Sequences that possess identity level of more than 62% will not be clustered into BLOSUM 62. Generally, this is same applied to other BLOSUM models (50, 80).

1.2Differences between PAM and BLOSUM

1.2.1 PAM (Point Accepted Mutation)

PAM is first introduced by Margeret Dayhoff in 1978. It is generally based on global alignmentsof closely related proteins. Global alignments in PAM includes all the letters from head to tail of both query sequence and targeted sequences, comparing the whole sequences. Normally, there are gaps in between the sequences where gap penalties are being applied. For its many matrix model used, the PAM1 is the matrix calculated from comparisons of sequences with no more than 1% divergence. One PAM of evolution means 1% of the residues or bases have changed the average over all 20 amino acids.PAM is extrapolated from comparison of closely related protein sequences.In this case, PAM assumes that the mutations that occur in one sequence can be assumed to apply in another closely related sequence as well. Hence, the PAM models can be extrapolated like PAM2 = PAM1 X PAM1. In the PAM model naming convention, high number is used for comparing distant related protein whereas low number is used for comparing closely related protein. Moreover, PAM is also based on mutations observed which includes both highly conserved and highly mutable region. Its calculations are based on 1572 observed mutations in phylogenetic trees of

71 closely related proteins. The purpose of PAM is mainly for tracing evolutionary origins of proteins. PAM takes aligned set of 71 groups closely related proteins that are at least 85%. The groups are then organized in a phylogenetic tree. Meanwhile, PAM is also a model where it is based on explicit evolutionary model such as the phylogenetic tree and Markov chain. The value assigned in each cell of the matrix is related to the probability of the column amino acid before the mutation being aligned with row amino acid afterwards.

1.2.2 BLOSUM (Block Substitution Matrix)

BLOSUM is first introduced in a paper “Amino Acid Substitution Matrices From Protein Blocks” by Steven Henikoff and Jorja Henikoff in the year of 1992. In general, BLOSUM matrices are based on local alignments unlike PAM which is based on global alignments. Local alignment aligns a substring from the query sequence to a substring from the targeted sequence where only compares at the conserved regions on both of the sequences. Normally, the conserved regions are ungapped and no gap penalties are needed to apply. For example, BLOSUM 62 is a matrix calculated from comparisons of sequences with more than 62% identity. Sequences that are more than 62% of identity level will not be put inside the BLOSUM 62. All BLOSUM matrices are based on observed alignments and are not extrapolated from closely related protein sequences as PAM does. Generally, it observes protein sequences at the BLOCKS database and searches for the conserved regions in each family where depicts very own biochemical functions. It then groups the clusters of sequences that are identical in their ungapped amino acid regions at specific level of identity. For the BLOSUM model naming convention rules, high number is used for comparing closely related protein, whereas low number is used for comparing distantly related protein. For example, BLOSUM 80 is for less divergent alignments, whereas BLOSUM 50 is for more divergent alignments.The more divergent the sequences are the more distantly related the sequences are, vice versa. It scans BLOCK databases for highly conserved region of protein family which has no gap in the sequence alignment.Meanwhile, BLOSUM is based on implicit or no evolutionary model. For the calculation part, it uses a log-odds score for each of the 210 possible substitution pair of the 20 standard amino acids. The log-odds scores measure, in an alignment, the algorithm for the ration of the likelihood of two amino acids appearing with a biological sense and the likelihood of the same amino acids appearing by chance. The main purpose of BLOSUM is to score alignment between evolutionarily divergent protein sequences. This is because; it only concentrates on finding the sequences that are identical in term of biochemical functions and ignores the relatedness between them. Hence, it is commonly good for comparison of the biological functions without consideration of evolutionary distances. Every possible identity or substitution is assigned a score based on its observed frequencies in the alignment of related proteins. A positive score is given to the more likely substitution while a negative score is given to the less likely substitution.Like AA pair is given the score 2 and BA pair is given the score of -1, AA pair is more likely to occur in the sequence pairing compared with BA pair in natural occurrence.

1.2.3 Advantages and Disadvantages of BLOSUM

Advantages of BLOSUM are that it is based on local alignments and all the matrices are calculated from the observed alignments which they are not extrapolated from closely related protein sequences. Comparing to PAM, BLOSUM is a much simpler model to understand and to be used. Since it is observation based, it is less independent of other models and concepts. In this case, BLOSUM does not require Markov chain assumption or phylogenetic trees like PAM do. Many tests suggest that BLOSUM matrices generally are more superior than PAM matrices for detecting biological relationship even if given the same amount of data to process with.

Meanwhile, disadvantages of BLOSUM are it is restricted a subset of conserved domains because it scans through the BLOCKS databases for the much conserved region of the protein which have no gaps in the sequences. BLOSUM does not use any model of evolution where closeness of relationship is ignored. Since it does not base on the evolutionary model, hence it cannot be used to calculate the evolutionary distances and also the phylogenetic trees.

1.2.4 Equivalent PAM and BLOSUM

Matrices based on relative entropy:

PAM100 ==> Blosum90

PAM120 ==> Blosum80

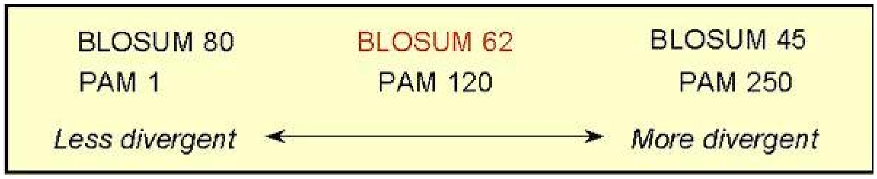
PAM160 ==> Blosum60

PAM200 ==> Blosum52

PAM250 ==> Blosum45

The picture above tells us that which models of PAM and BLOSUM that works quite the same way in scoring the alignments.

We can view the differences in a pictorial way as shown below :



BLOSUM 80	BLOSUM 62	BLOSUM 45
PAM 1	PAM 120	PAM 250
Less divergent	↔	*More divergent*

As you can view from the picture above, both PAM and BLOSUM are in a huge contradiction in using the numbers to score the alignments. BLOSUM uses higher number to score the less divergent and lower number to score the more divergent ones. Whereas PAM works totally in a different way as BLOSUM does. The picture also tells us that BLOSUM 62 and PAM 120 are the most commonly used scoring matrix in most alignment programs like BLAST.

Question 2

What are the differences between BLOSUM and PAM? List three differences.

Solution:

PAM(Point Accepted Mutation)	**BLOSUM(Block Substitution Matrix)**
By Margeret Dayhoff in 1978.	By Steven Henikoff and Jorja Henikoff in 1992.
Based on global alignments	Based on local alignments.
Is extrapolated from comparison of closely related protein sequence.	Is based on observed alignments.
To trace evolutionary origins of proteins	To score alignment between evolutionarily divergent protein sequences.
One PAM of evolution means 1% of the residues or bases have changed the average over all 20 amino acids.	BLOSUM 62 is a matrix calculated from comparisons of sequences with more than 62% identity
High number is used for comparing distant related protein.	High number is used for comparing closely related protein.
Low number is used for comparing closely related protein.	Low number is used for comparing distantly related protein.
Calculations are based on 1572 observed mutations in phylogenetic trees of 71 closely related proteins.	Calculation uses a log-odds score for each of the 210 possible substitution pair of the 20 standard amino acids.
Based on mutations observed which includes both highly conserved and highly mutable region.	Based on BLOCK databases for highly conserved region of protein family which has no gap in the sequence alignment.

Question 3

What are the advantages of the BLOSUM?

Solution:

- Simpler model
- Observation based, less independent of other models
- Superior in detecting biological relationship

UNIT –III

BLAST

In the field of bioinformatics, a sequence database is a type of biological database that is composed of a large collection of computerized ("digital") nucleic acid sequences, protein sequences, or other polymer sequences stored on a computer.

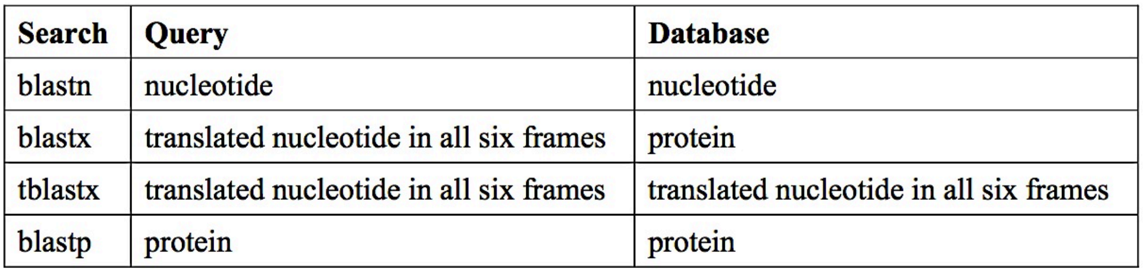
As of 2013 it contained over 40 million sequences and is growing at an exponential rate. Historically, sequences were published in paper form, but as the number of sequences grew, this storage method became unsustainable.

Database Searching:

Search: Sequence databases can be searched using a variety of methods. The most common usage is probably searching for sequences similar to a certain target protein or gene whose sequence is already known to the user. The BLAST program is a popular method of this type.

The Basic Local Alignment Search Tool (BLAST) finds regions of local similarity between sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches. BLAST can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families.

BLAST search you actually need to specify the type of search that you will perform. The following table outlines each algorithm and the nature of the query and database used.



Search	Query	Database
blastn	nucleotide	nucleotide
blastx	translated nucleotide in all six frames	protein
tblastx	translated nucleotide in all six frames	translated nucleotide in all six frames
blastp	protein	protein

This is the common procedure for any BLAST program.

Step 1: Select the BLAST program.

Step 2: Enter a query sequence or upload a file containing sequence.

Step 3: Select the database to search.

Step 4: Select the algorithm and the parameters of the algorithm for the search.

Step 5: Run the BLAST program.

Step 1: Select the BLAST program

User have to specify the type of BLAST programs from the database like BLASTp, BLASTn, BLASTx, tBLASTn, tBLASTx.

Step 2: Enter a query sequence or upload a file containing sequence

Enter a query sequence by pasting the sequence in the query box or uploading a FASTA file which is having the sequence for similarity search. This step is similar for all BLAST programs. The user can give the accession number or gi number or even a raw FASTA sequence. Go to simulator tab to know more about how to retrieve query sequence.

Step 3: Select database to search

User first has to know what all databases are available and what type of sequences are present in those databases. Sequence similarity search involves searching of similar sequences of the query sequence from the selected databases

Step 4*: Select the algorithm and the parameters of the algorithm for the search*

There are different algorithms for some of the BLAST program. User has to specify the algorithm for the BLAST program. Nucleotide BLAST uses algorithms like MegaBLAST which searches for highly similar sequences, discontiguous MegaBLAST which searches for more dissimilar sequences and BLASTn which searches for somewhat similar sequences. Meanwhile for protein BLAST algorithms like BLASTp, searches for similarity between protein query and protein database, PSI-BLAST performs position specific search iteratively, PHI-BLAST searches for a particular pattern (user has to enter the pattern to search in the PHI pattern box provided) that is present in the sequence against the sequences in the database, DELTA-BLAST is Domain Enhanced Lookup Time Accelerated BLAST. It searches multiple sequence and aligns them to find protein homology. The different algorithmic parameters are, Target sequences, Short queries, E-value, Word size, Query range, scoring parameters (Match/Mismatch scores, and Gap penalties) and filters (Filter and Mask) which are required to run BLAST programs. Default values are provided but the user can adjust the values accordingly.

Step 5: Run the BLAST program

Submission of the BLAST program can be done by clicking the BLAST button at the end of the page.

BLAST Result:

After submitting the query sequence for sequence similarity search, the result page will appear along with the information like Query id, Description, Molecule type, Length of sequence, Database name and BLAST program. It shows the putative conserved domains that have been detected while undergoing sequence similarity search.

The query sequence represented as a numbered red bar below the color key. Database hits are shown below the query (red) bar according to the alignment score. Among the aligned sequences, the most related sequences are kept near to the query sequence. User can find more description about these alignments, by dragging the mouse to the each colored bar.

The alignment is preceded by the sequence identities, along with the definition line, length of the matched sequence, followed by the score and E-value. The line also contains the information about the identical residues in alignment (identities), number of positivity's, number of gaps used in the alignment. Finally it shows the actual alignment, along with the query sequence on the top and database sequence below the query. The number on either sides of the alignment indicates the position of amino acids/nucleotides in sequence

BLAST Statistics

Score (bits)

- A statistical conversion of the score derived by summing using the substitution matrix

Expect (e) Value

Function of the S value and the database size

An e value of 1

One alignment using a query of this size will by chance produce a S score of this value in a database of this size

E value of –10 (=1x10-10)

- Unlikely that random chance lead to this current alignment compared to an alignment with an e value of 1
- Often considered to be a probability

Procedure to run FASTA program

There are four steps require to run FASTA program.

Step 1: Specify the tool input (sequence and database).

Step 2: Entering of input sequence.

Step 3: Set up the parameters.

Step 4: Submit the query for processing.

Step 1: Specify the tool input

Select the database to search : Databases are required to run the sequence similarity search. Multiple databases can be used at the same time. The different databases are

Uniprot Knoweldge base
Uniprot KB/swiss-prot
Uniprot KB/ Swissprot isoforms
Uniprot KB /Trembl
UniProtKB Taxonomic Subsets
UniProt Clusters
Patents
Structure

Step 2 Entering of input sequence

The query sequence can be entered directly in GCG, FASTA, EMBL, GenBank, PIR, NBRF, PHYLIP or UniProtKB/Swiss-Prot formats.

Sequence file upload

A file containing the valid sequence in any format mentioned above can be used as a query for sequence similarity search. Sequence type indicates the type of sequence (PROTEIN / DNA / RNA) for similarity search.Go to simulator tab to know more about how to retrieve the query sequence.

Step 3: Setting up parameters

User has to specify the type of program and the matrix for scoring. FASTA, FASTX, FASTY, SSEARCH, GGSEARCH and GLSEARCH are the different programs used. Substitution matrix are used for scoring alignments. The matrices are BLOSUM50, BLOSUM62, BLASTP62, PAM120, PAM250, MDM10, MDM20, and MDM40. BLOSUM50 is set as a default substitution matrix. Parameters include.

GAP open and GAP extended penalty: Common and regular cause for GAP is mutation, if gap penalty is low we can get high scoring sequence similarity search. Also gaps will increase uncertainty in alignment.

Ktup: It is a value given as the word size for comparison.

Expectation value (E-value): It decreases exponentially with the score that is assigned to an alignment between two sequences.

Strands, Histograms, Filter: It filters the low complex regions in sequence similarity search. Histogram will give graphical representation of scores.

Statistical estimates, Scores, alignments, sequence range and database range: specify the range of the query for search in database.

HSPs, Score format, Transition table score format: are the different score formats. Transition table gives the genetic codes used in translation.

Step 4: Submission

The result page can be seen in another window by clicking submit. This is an interactive process, when the process is complete the result will be displayed in the browser. Result can be sent to a valid email address which has to be specified in the text box.

FASTA Result Analysis

Summary Table

Result page appears by giving the information like aligned sequences from the sequence similarity search, database id, source of the sequence, Gene-expression, molecule type, Nucleotide sequence, Genomics, Protein sequences, Ontologies, Enzymes, protein families, and Literature, which is followed by the length of sequence, score, identities, positives and E-value.

Tool output:

Tool output gives complete statistical details of the sequence similarity search.

BLAST algorithm:

Dynamic programming algorithms are recursive algorithms modified to store intermediate results, which improves efficiency for certain problems. The Smith-Waterman (Needleman-Wunsch) algorithm uses a dynamic programming algorithm to find the optimal local (global) alignment of two sequences -- a and b. The alignment algorithm is based on finding the elements of a matrix H where the element $H_{i,j}$ is the optimal score for aligning the sequence (a_1, a_2, a_i..) with (b_1, b_2, b_i.......). Two similar amino acids (e.g. arginine and lysine) receive a high score, two dissimilar amino acids (e.g. arginine and glycine) receive a low score. The higher the score of a path through the matrix, the better the alignment.

Applications:

Sequence Alignment is almost the most useful tool in **Bioinlformatics**, it helps almost in every application of **Bioinformatics** (predicting protein structure, predicting protein function, phylogenetic analysis...etc).

The main applications of **Sequence Alignment are**:

1- Structure Prediction: a **Multiple Sequence Alignment** can give you the almost perfect protein or RNA secondary structure, some times it helps even with the 3D structure.

2- Protein Family: a **Multiple Sequence Alignment** can help you to decide that your protein is a member of a known protein family or not.

3- Pattern Identification: By looking at conserved regions or sites, you can identify which region is responsible for a functional site.

4- Domain Identification: By looking at file provided by a **Multiple Sequence Alignment**, you can extract profiles to use them against databases.

5- DNA Regulatory Elements: You can use **Multiple Sequence Alignments** to locate DNA regulatory elements such as binding sites...etc.

6- Phylogenetic Analysis: By carefully picking related sequences you can reconstruct a tree using sequences that u have used in the **Multiple Sequence Alignment** (You can use the **PHYLIP** package and you can find a post about it here).

Overview of Help-Pages

DNA Sequence formats

[Plain] [FASTQ] [EMBL] [FASTA] [GCG] [GenBank] [IG] [IUPAC]
[How Genomatix represents sequence annotation]

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
GTGGAAGTTCTTAGGGCATGGCAAAGAGTCAGAATTTGAC
+
FAFFADEDGDBGEGGBCGGHE>EEBA@@=
```

For a detailed decription please see the Wikipedia entry.

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 ("//").

An example sequence in EMBL 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;
     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:

```
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 ("//").

An example sequence in GenBank format is:

```
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.

An example sequence in IG format is:

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

Genomatix annotation syntax

Some Genomatix tools, e.g. Gene2Promoter or GPD allow the extraction of sequences. Genomatix uses the following syntax to annotate sequence information: each information item is denoted by a keyword, followed by a "=" and the value. These information items are separated by a pipe symbol "|". The keywords are the following:

loc	The **Genomatix Locus Id**, consisting of the string "GXL_" followed by a number.
sym	The **gene symbol**. This can be a (comma-separated) list.
geneid	The **NCBI Gene Id**. This can be a (comma-separated) list.
acc	A **unique identifier** for the sequence. E.g. for Genomatix promoter regions, the Genomatix Promoter Id is listed in this field.
taxid	The organism's **Taxon Id**
spec	The **organism name**
chr	The **chromosome** within the organism.
ctg	The **NCBI contig** within the chromosome.
str	**Strand**, (+) for sense, (-) for antisense strand.
start	**Start position** of the sequence (relative to the contig).
end	**End position** of the sequence (relative to the contig).
len	**Length** of the sequence in basepairs.
tss	A (comma-separated list of) **UTR-start/TSS position(s)**. If there are several TSS/UTR-starts, this means that several transcripts share the same promoter (e.g. when they are splice variants). The positions are relative to the promoter region.
probe	A (comma-separated list of) **Affymetrix Probe Id(s)**.
unigene	A (comma-separated list of) **UniGene Cluster Id(s)**.
homgroup	An identifier (a number) for the **homology group** (available for promoter sequences only). Orthologously related sequences have the same value in this field.
promset	If the sequence is a promoter region, the **promoter set** is denoted here.
descr	The **gene description**. If several genes (i.e. NCBI gene ids) are associated with the sequence, the descriptions for all of the genes are note, separated by ";"
comm	A **comment** field, used for additional annotation. For promoter sequences, this field contains information about the transcripts associated with the promoter. For each transcript the Genomatix Transcript Id, accession number, TSS position and quality is listed, separated by "/". For Genomatix CompGen promoters no transcripts are assigned, in this case the string "CompGen promoter" is denoted.

This syntax is currently used only for sequences in the FASTA and GenBank formats.

Example (a promoter sequence in GenBank format):

```
LOCUS        GXP_170357    743 bp    DNA
DEFINITION   loc=GXL_141619|sym=TPH2|geneid=121278|acc=GXP_170357|
             taxid=9606|spec=Homo sapiens|chr=12|ctg=NC_000012|str=(+)|
             start=70618393|end=70619135|len=743|tss=501,632|
             homgroup=4612|promset=1|descr=tryptophan hydroxylase 2|
             comm=GXT_2756574/AK094614/632/gold;
             GXT_2799672/NM_173353/501/bronze
```

```
ACCESSION   GXP_170357
BASE COUNT    216 a  180 c  147 g  200 t
ORIGIN
        1 TTGATTACCT TATTTGATCA TTACACATTG TACGCTTGTG TCAAAATATC ACATGTGCCT
       61 TATAAATGTG TACAACTATT AGTTATCCAT AAAAATTAAA AATTAAAAAA TCCGTAAAAT
      121 GGTTTAAGCA TTCAGCAGTG CTGATCTTTC TTAAATTATT TTTCTAATTT TGGAAAGAAA
      181 GCACAAAATC TTTGAATTCA CAATTGCTTA AAGACTGAGG TTAACTTGCC AGTGGCAGGC
      241 TTGAGAGATG AGAGAACTAA CGTCAGAGGA TAGATGGTTT CTTGTACAAA TAACACCCCC
      301 TTATGTATTG TTCTCCACCA CCCCCGCCCA AAAAGCTACT CGACCTATGA AACAAATCAC
      361 ACTATGAGCA CAGATAACCC CAGGCTTCAG GTCTGTAATC TGACTGTGGC CATCGGCAAC
      421 CAGAAATGAG TTTCTTTCTA ATCAGTCTTG CATCAGTCTC CAGTCATTCA TATAAAGGAG
      481 CCCGGGGATG GGAGGATTCG CATTGCTCTT CAGCACCAGG GTTCTGGACA GCGCCCCAAG
      541 CAGGCAGCTG ATCGCACGCC CCTTCCTCTC AATCTCCGCC AGCGCTGCTA CTGCCCCTCT
      601 AGTACCCCCT GCTGCAGAGA AAGAATATTA CACCGGGATC CATGCAGCCA GCAATGATGA
      661 TGTTTTCCAG TAAATACTGG GCACGGAGAG GGTTTTCCCT GGATTCAGCA GTGCCCGAAG
      721 AGCATCAGCT ACTTGGCAGC TCA
//
```

IUPAC nucleic acid codes

To represent ambiguity in DNA sequences the following letters can be used (following the rules of the *International Union of Pure and Applied Chemistry* (IUPAC)):

```
A = adenine
C = cytosine
G = guanine
T = thymine
U = uracil
R = G A (purine)
Y = T C (pyrimidine)
K = G T (keto)
M = A C (amino)
S = G C
W = A T
B = G T C
D = G A T
H = A C T
V = G C A
N = A G C T (any)
```

UNIT –IV

ClustalW2: Introduction

ClustalW2 is a general purpose multiple sequence alignment program for DNA or proteins. It attempts to calculate the best match for the selected sequences, and lines them up so that the identities, similarities and differences can be seen.

How to use this tool

Running a tool from the web form is a simple multiple steps process, starting at the top of the page and following the steps to the bottom.

Each tool has at least 2 steps, but most of them have more:

- The first steps are usually where the user sets the tool input (e.g. sequences, databases...)
- In the following steps, the user has the possibility to change the default tool parameters
- And finally, the last step is always the tool submission step, where the user can specify a title to be associated with the results and an email address for email notification. Using the submit button will effectively submit the information specified previously in the form to launch the tool on the server

Note that the parameters are validated prior to launching the tool on the server and in the event of a missing or wrong combination of parameters, the user will be notified directly in the form.

Step 1 - Sequence

Sequence Input Window

Three or more sequences to be aligned can be entered directly into this form. Sequences can be in GCG, FASTA, EMBL, PIR, NBRF or UniProtKB/Swiss-Prot format. Partially formatted sequences are not accepted. There is a limit of 500 sequences or 1MB of data.

Sequence File Upload

A file containing three or more valid sequences in any format (GCG, FASTA, EMBL, PIR, NBRF or UniProtKB/Swiss-Prot) can be uploaded and used as input for the multiple sequence alignment. There is a limit of 500 sequences or 1MB of data.

Sequence Type

Indicates if the sequences to align are protein or nucleotide (DNA/RNA).

Type	Abbreviation
Protein	protein
DNA	dna

Default value is: Protein [protein]

Step 2 - Pairwise Alignment Options

Alignment Type: The alignment method used to perform the pairwise alignments used to generate the guide tree.

Output Format	Description	Abbreviation
slow	Slow, but accurate	slow
fast	Fast, but approximate	fast

Default value is: slow

Protein Weight Matrix (PW)

Slow pairwise alignment protein sequence comparison matrix series used to score alignment.

Matrix (Protein Only)	Abbreviation
BLOSUM	blosum
PAM	pam
Gonnet	gonnet
ID	id

Default value is: Gonnet [gonnet]

Step 3 - Multiple Sequence Alignment Options

Protein Weight Matrix

Multiple alignment protein sequence comparison matrix series used to score the alignment.

Matrix (Protein Only)	Abbreviation
BLOSUM	blosum
PAM	pam
Gonnet	gonnet
ID	id

Default value is: Gonnet [gonnet]

DNA Weight Matrix

Multiple alignment nucleotide sequence comparison matrix used to score the alignment.

Matrix (Protein Only)	Abbreviation
IUB	iub
ClustalW	clustalw

Default value is: IUB [iub]

Gap Open: Multiple alignment penalty for the first residue in a gap. *Default value is: 10*

Gap Extension :Multiple alignment penalty for each additional residue in a gap. *Default value is: 0.20*

Gap Distances: Multiple alignment gaps that are closer together than this distance are penalised. *Default value is: 5*

Output

Format for generated multiple sequence alignment.

Order	Description	Abbreviation
Clustal w/ numbers	Clustal alignment format with base/residue numbering	aln1
Clustal w/o numbers	Clustal alignment format without base/residue numbering	aln2
GCG MSF	GCG Multiple Sequence File (MSF) alignment format	gcg
PHYLIP	PHYLIP interleaved alignment format	phylip
NEXUS	NEXUS alignment format	nexus
NBRF/PIR	NBRF or PIR sequence format	pir
GDE	GDE sequence format	gde
Pearson/FASTA	Pearson or FASTA sequence format	fasta

Default value is: Clustal w/ numbers [aln1]

Step 4 – Submission: Job title

It's possible to identify the tool result by giving it a name. This name will be associated to the results and might appear in some of the graphical representations of the results.

Email Notification

Running a tool is usually an interactive process, the results are delivered directly to the browser when they become available. Depending on the tool and its input parameters, this may take quite a long time. It's possible to be notified by email when the job is finished by simply ticking the box "Be notified by email". An email with a link to the results will be sent to the email address specified in the corresponding text box. Email notifications require valid email addresses.

Email Address

If email notification is requested, then a valid Internet email address in the form joe@example.org must be provided. This is not required when running the tool interactively (The results will be delivered to the browser window when they are ready).

T-Coffee

Introduction

T-Coffee is a multiple sequence alignment program. The main characteristic of T-Coffee is that it will allow you to combine results obtained with several alignment methods. By default, T-Coffee will compare all you sequences two by two, producing a global alignment and a series of local alignments (using lalign). The program will then combine all these alignments into a multiple alignment.

How to use this tool

Running a tool from the web form is a simple multiple steps process, starting at the top of the page and following the steps to the bottom.

Each tool has at least 2 steps, but most of them have more:

- The first steps are usually where the user sets the tool input (e.g. sequences, databases...)
- In the following steps, the user has the possibility to change the default tool parameters
- And finally, the last step is always the tool submission step, where the user can specify a title to be associated with the results and an email address for email notification. Using the submit button will effectively submit the information specified previously in the form to launch the tool on the server

Note that the parameters are validated prior to launching the tool on the server and in the event of a missing or wrong combination of parameters, the user will be notified directly in the form.

Step 1 - Sequence

Sequence Input Window

Three or more sequences to be aligned can be entered directly into this form. Sequences can be in GCG, FASTA, EMBL, GenBank, PIR, NBRF, PHYLIP or UniProtKB/Swiss-Prot format. Partially formatted sequences are not accepted. There is currently a sequence input limit of 500 sequences and 1MB of data.

Sequence File Upload

A file containing three or more valid sequences in any format (GCG, FASTA, EMBL, GenBank, PIR, NBRF, PHYLIP or UniProtKB/Swiss-Prot) can be uploaded and used as input for the multiple sequence alignment. There is currently a file upload limit of 500 sequences and 1MB of data.

STEP 2 - Set your Parameters

Matrix

Matrix series to use when generating the multiple sequence alignment. The program goes through the chosen matrix series, spanning the full range of amino acid distances.

Matrix (Protein Only)	**Description**	**Abbreviation**
BLOSUM	(Henikoff) These matrices appear to be the best available for carrying out data base similarity searches.	blosum
PAM	(Dayhoff) These have been extremely widely used since the late '70s. We use the PAM 350 matrix.	pam

Default value is: None [none]

Order

The order in which the sequences appear in the final alignment

Order	**Description**	**Abbreviation**
aligned	Determined by the guide tree	aligned
input	Same order as the input sequences	input

Default value is: aligned

Step 3 - Submission

Job title

It's possible to identify the tool result by giving it a name. This name will be associated to the results and might appear in some of the graphical representations of the results.

Email Notification

Running a tool is usually an interactive process, the results are delivered directly to the browser when they become available. Depending on the tool and its input parameters, this may take quite a long time. It's possible to be notified by email when the job is finished by simply ticking the box "Be notified by email". An email with a link to the results will be sent to the email address specified in the corresponding text box. Email notifications require valid email addresses.

Email Address

If email notification is requested, then a valid Internet email address in the form joe@example.org must be provided. This is not required when running the tool interactively (The results will be delivered to the browser window when they are ready).

Protein motifs, patterns and profiles

When aligning protein sequences it is often apparent that certain regions or specific amino acids, are more conserved than others. Such conserved regions are often conserved because they encode a part of the protein that is functionally important. The term motif is use to refer to a part of a protein sequence that is associated with a particular biological function.

For example a region of a protein that binds ATP is called an ATP binding motif. Since these regions are conserved, they may be recognisable by the presence of a particular sequence of amino acids called a pattern. A pattern is thus a qualitative description of a motif in terms of amino acid sequence.

The concept of a profile extends this concept, allowing a quantitative description of a motif, by assigning probabilities to the occurrence of a particular amino acid at each position of a motif. Thus profiles can be used to describe very divergent motifs.

The presence of a particular motif within a protein sequence can be used to suggest functions for uncharacterised proteins.

A number of databases have been constructed that attempt to describe particular protein motifs in terms of patterns and profiles. They allow you to search for patterns or profiles that are indicative of particular functional motifs within a query protein.

Some examples of such databases include:

PROSITE - a collection of patterns and profiles
Pfam - A collection of Profiles generated using hidden Markov models
PRINTS - provider of fingerprints (groups of aligned, un-weighted motifs)
BLOCKS - a database of weighted profiles or blocks

These databases all have different areas of optimum application – its difficult to tell which one will give the best results. They all have particular strengths and weaknesses. You really need to use them all.

However, a database called INTERPRO has been recently established that combines information from PRINTS, PROSITE, ProDom and Pfam. Using InterPro saves a lot of work since we can essentially search many databases in one go.

Consensus sequences

– The consensus sequence method is the simplest method to build a model from a multiple sequence alignment.
– The consensus sequence is built using the following rules:

• Majority wins. • Skip too much variation.

Advantages:

• This method is very fast and easy to implement.

Limitations:

• Models have no information about variations in the columns.
• Very dependent on the training set.
• No scoring, only binary result (YES/NO).

When I use it?

Useful to find highly conserved signatures, as for example enzyme restriction sites for DNA.

Pattern syntax

A pattern describes a set of alternative sequences, using a single expression. In computer science, patterns are known as regular expressions.

The Prosite syntax for patterns:

– uses the standard IUPAC one-letter codes for amino acids (G=Gly, P=Pro, ...),
– each element in a pattern is separated from its neighbor by a '-',
– the symbol 'X' is used where any amino acid is accepted,
– ambiguities are indicated by square parentheses '[ ]' ([AG] means Ala or Gly),
– amino acids that are not accepted at a given position are listed between a pair of curly brackets '{ }' ({AG} means any amino acid except Ala and Gly),

- repetitions are indicated between parentheses ’( )’ ([AG](2,4) means Ala or Gly between 2 and 4 times, X(2) means any amino acid twice),
- apatternisanchoredtotheN-termand/orC-termbythesymbols’<’and’>’respectively.

Pattern syntax: an example

The following pattern **<A-x-[ST](2)-x(0,1)-{V}**

means:

- An Ala in the N-term,
- Followed by any amino acid,
- Followed by a Ser or Thr twice,
- Followed or not by any residue,
- Followed by any amino acid except Val.

Generalized profiles as an extension of PSSMs

- The following information is stored in any generalized profile: • each position is called a match state. A score for every residue is defined at every match
- States, just as in the PSSM.
- Each match state can be omitted in the alignment, by what is called a deletion state and that receives a position-dependent penalty.
- Insertions of variable length are possible between any two adjacent match (or deletion) states. These insertion states are given a position-dependent penalty that might also depend upon the inserted residues.
- Every possible transition between any two states (match, delete or insert) receives a position-dependent penalty. This is primarily to model the cost of opening and closing a gap.
- A couple of additional parameters permit to finely tune the behavior of the extremities of the alignment, which can forced to be ’local’ or ’global’ at either ends of the profile and of the sequence.

BCB410 – Course Notes

Title: Sequence Alignment (Needleman-Wunsch, Smith-Waterman)
These notes are based on lecture notes taken by Gabe Musso for CSC 2427

Topics:

1. **Needleman-Wunsch** (Global Alignment)
2. **Maximum Contiguous Subsequence Sum**
3. **Smith-Waterman** (Local Alignment)

Background: Importance of Sequence Alignment
Comparative analysis is the backbone of evolutionary biology. It was phenotypic variation which allowed Darwin to compose his theory of natural selection. That theory rests on the fact that transfer of the genetic code from parent to progeny does not exist without change. It is these changes in genetic sequence which allow for divergence of species, and thus provide a backdrop for natural selection. Just as comparative analysis was key for evolutionary biology, sequence alignment is the cornerstone of modern bioinformatics. Rapid and automated sequence analysis facilitates everything from functional classification & structural determination of proteins, to studies of genetic expression and evolution.

1. Needleman-Wunsch (Global Alignment)

Dynamic programming algorithms find the best solution by breaking the original problem into smaller sub-problems and then solving. The Needleman-Wunsch algorithm is a dynamic programming algorithm for optimal sequence alignment (Needleman and Wunsch, 1970). Basically, the concept behind the Needleman-Wunsch algorithm stems from the observation that any partial sub-path that tends at a point along the true optimal path must itself be the optimal path leading up to that point. Therefore the optimal path can be determined by incremental extension of the optimal sub-paths. In a Needleman-Wunsch alignment, the optimal path must stretch from beginning to end in both sequences (hence the term 'global alignment').

In order to perform a Needleman-Wunsch alignment, a matrix is created which allows us to compare the two sequences. The score `M(i,j)` for every cell depends on the three cells corresponding to either or both sequence having 1 less letter (i.e. cells `M(i-1.j)`, `M(i,j-1)` and `M(i-1,j-1)`. It is calculated as follows:

$$\begin{aligned} M(i,j) = \mathrm{MAX}(&M_{i-1,j-1} + S(A_i, B_j) \\ &M_{i-1,\ j} + gap \\ &M_{i,j-1} + gap) \end{aligned}$$

where gap is the gap penalty and the function S returns the score/penalty for matching the two corresponding letters. Once we have computed this score for every cell, we must do a "traceback", that is to determine the actual set of operations that lead to the score.

Because when computing the score of a cell we took a max over three numbers, on the traceback we go to the location of the highest – going sideways or up corresponds to gaps, and going along the diagonal corresponds to a match. This algorithm performs alignments with a time complexity of O(mn) and a space complexity of O(mn).

Example:

Find the best alignment of these two sequences:

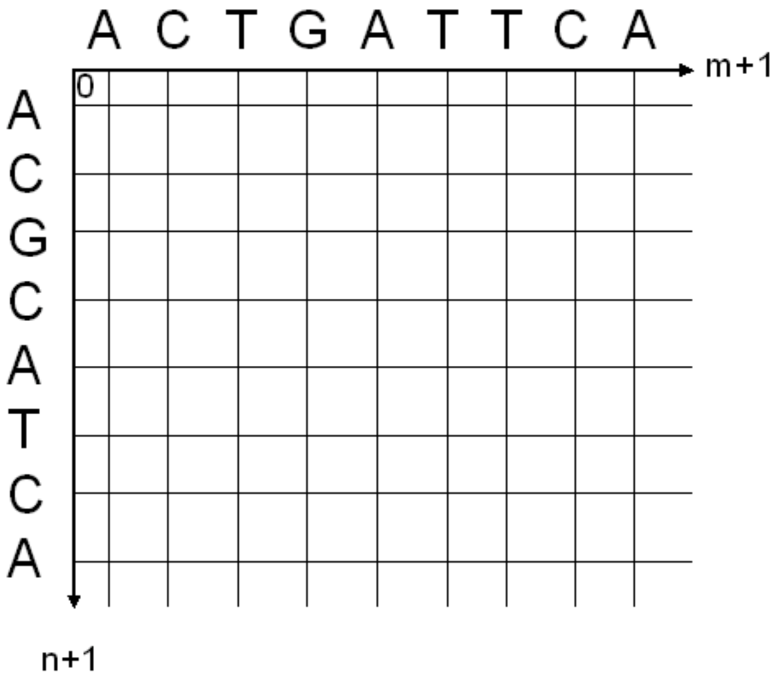
```
ACTGATTCA
 ACGCATCA
```

Using -2 as a gap penalty, -3 as a mismatch penalty, and 2 as the score for a match.

Solution:

Step 1: Draw the matrix

For 2 sequences (length m and length n) what size scoring matrix is needed for their alignment? Grid dimensions must be (m+1) × (n+1). Think of each increment as a division of the sequence members:

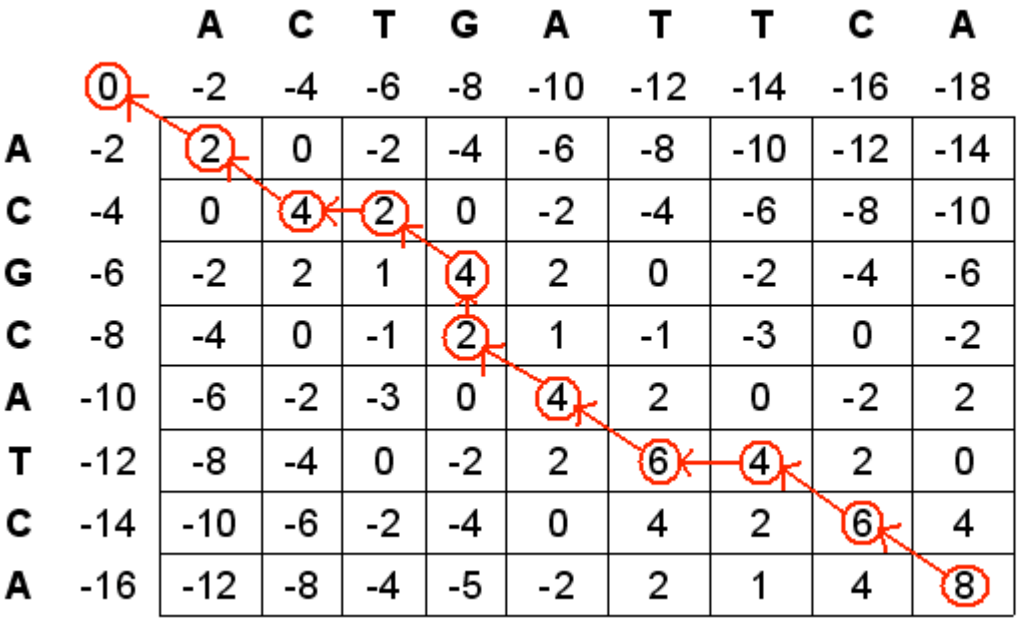


Step 2: Assign scores

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-4	0	4	2	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

Step 3: Trace back

The optimal path is traced beginning from the lower right-hand corner



		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-4	0	4	2	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

Result:

This analysis yielded the following alignment:

```
ACTG-ATTCA
||   || ||
AC-GCAT-CA
```

The alignment score is equal to the value in the lower right-hand corner of the matrix (8).

2. From Global to Local Similarity: Maximum Contiguous Subsequence Sum

When aligning two very large sequences, it is often useful to determine the locations of high similarity regions, even if there is no additional similarity inbetween the sequences. Now that we know how to calculate the *global* alignments, how can we find all local high-scoring hits, or *local* alignments above a given threshold for two large sequences? The answer is related to a programming "pearl", the 'Maximum Contiguous Subsequence Sum' (MSS).

***Problem**:*

Given integers $A_1, A_2, ..., A_N$ find (and identify the sequence corresponding to) the maximum value of:

$$\sum_{k=1}^{j} A_k$$

***Solution**:*

Can be solved in time complexity of 'n'.

```
mss(A) {
     max = 0;
     sum = 0;
     for (i=1; i ≤ n; i+1) {
          sum = sum + A[i];
          if (sum > max)
               max = sum;
          if (sum < 0)
               sum = 0;
```

```
                }
                return max;
        }
```

Analysis:

When a subsequence occurs which has a negative sum, the subsequence which will be examined next can begin after the first subsequence (the one that produced the negative sum). Basically, the entire first subsequence is regarded as not having a starting point which will generate a positive sum. For example, consider this set of numbers:

4, 6, -2, 2, -14, 9

Some sums are positive (4, 4+6, 4+6+(-2), 4+6+(-2)+2) but the sum of the first 5 terms (4+6+(-2)+2-14) is negative. Therefore it follows logically that any sequence starting between the 4 and -14 and ending with the -14 will have a negative sum.

The maximum contiguous subsequence sum searches exactly for the highest scoring local area. We now generalize this approach for sequence alignment; the only change is we do the abovealgorithm in two dimensions!

3. Smith-Waterman (Local Alignment)

Over a decade after the initial publication of the Needleman-Wunsch algorithm, a modification was made to allow for local alignments (Smith and Waterman, 1981). In this adaptation, the alignment path does not need to reach the edges of the search graph, but may begin and end internally. In order to accomplish this, 0 was added as a term in the score calculation described by Needleman and Wunsch.

Recall that for global alignments the value at any point is:

$$M(i,j) = MAX(M_{i-1,j-1} + S(A_i, B_j) \\ M_{i-1,j} + gap \\ M_{i,j-1} + gap)$$

However for local alignments:

$$M(I,j) = MAX(M_{i-1,j-1} + S(A_i, B_j) \\ M_{i-1,j} + gap \\ M_{i,j-1} + gap \\ 0)$$

The implication of this is that there are no values below zero in a local alignment scoring matrix, and the reason for the zero is exactly the same as in the MSS problem above.

Example:

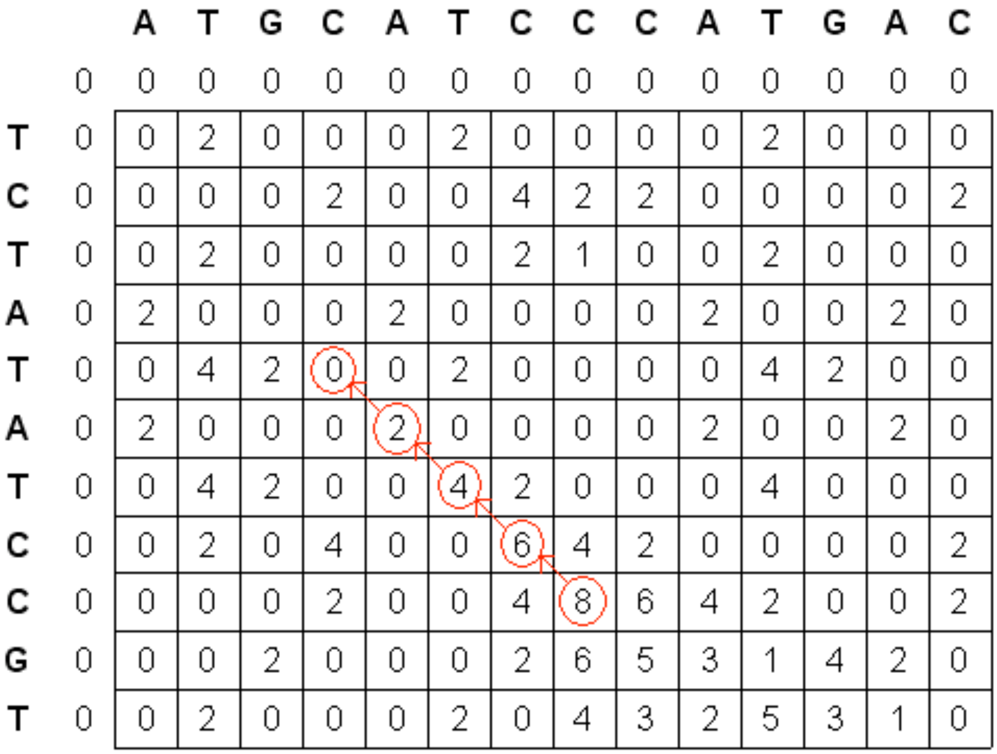
Find the best local alignment between these two sequences:

```
ATGCATCCCATGAC
TCTATATCCGT
```

Using -2 as a gap penalty, -3 as a mismatch penalty, and 2 as the score for a match.

Solution:

Traceback begins at the highest value (which is also the alignment score).



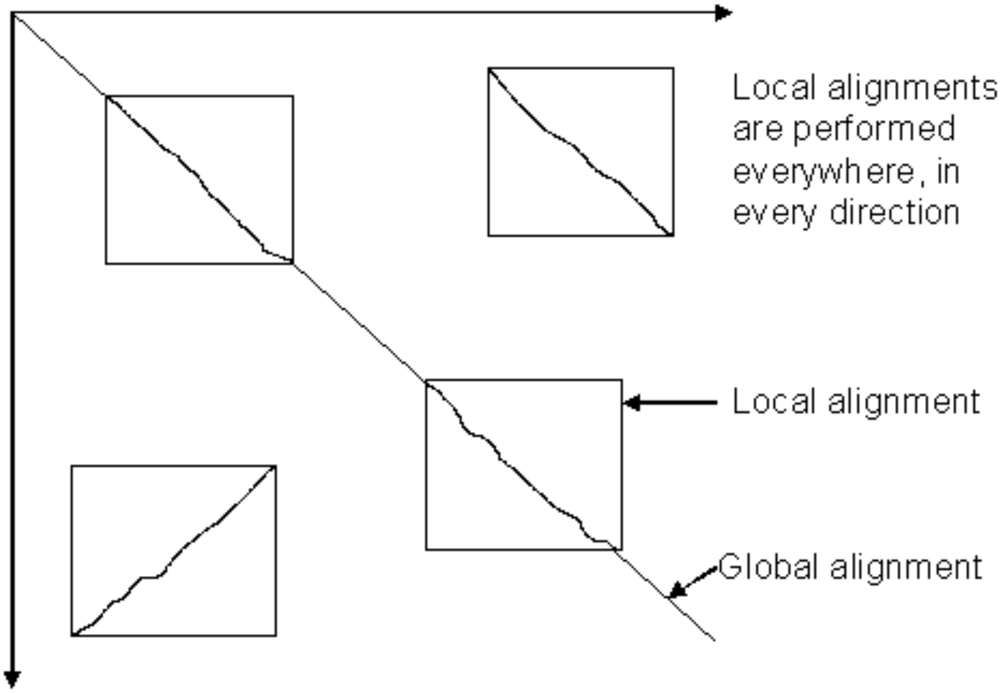
		A	T	G	C	A	T	C	C	C	A	T	G	A	C
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	2	0	0	0	2	0	0	0	0	2	0	0	0
C	0	0	0	0	2	0	0	4	2	2	0	0	0	0	2
T	0	0	2	0	0	0	0	2	1	0	0	2	0	0	0
A	0	2	0	0	0	2	0	0	0	0	2	0	0	2	0
T	0	0	4	2	0	0	2	0	0	0	0	4	2	0	0
A	0	2	0	0	0	2	0	0	0	0	2	0	0	2	0
T	0	0	4	2	0	0	4	2	0	0	0	4	0	0	0
C	0	0	2	0	4	0	0	6	4	2	0	0	0	0	2
C	0	0	0	0	2	0	0	4	8	6	4	2	0	0	2
G	0	0	0	2	0	0	0	2	6	5	3	1	4	2	0
T	0	0	2	0	0	0	2	0	4	3	2	5	3	1	0

Which yields the alignment:

```
ATCC
||||
ATCC
```

With an alignment score of 8.

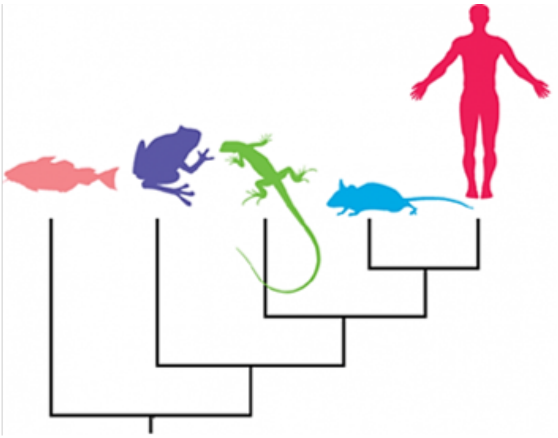
Local alignments are performed everywhere possible along two sequences.



When trying to find the best local alignments corresponding to a global alignment, a sub-matrix is created with the highest positive score for all alignments above a given threshold. Therefore, the same thing that the MSS was doing on a linear matrix, the Smith-Waterman alignment does on a rectangular matrix.

UNIT -V

Phylogenetic Tree



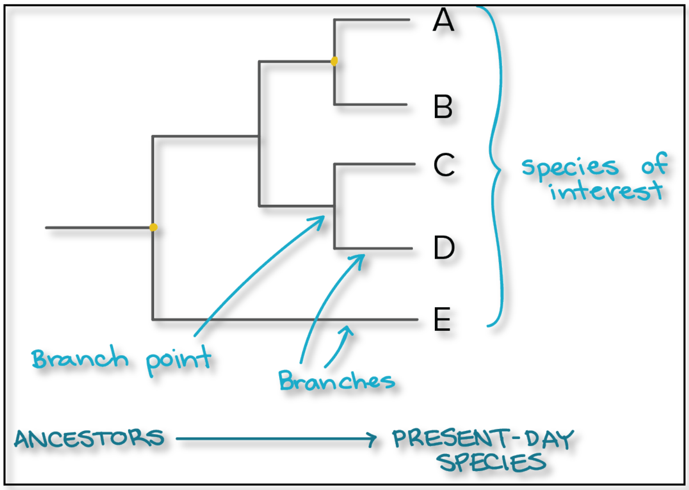
A phylogenetic tree, also known as a phylogeny, is a diagram that depicts the lines of evolutionary descent of different species, organisms, or genes from a common ancestor. Phylogenies are useful for organizing knowledge of biological diversity, for structuring classifications, and for providing insight into events that occurred during evolution.

Anatomy of a phylogenetic tree

When we draw a phylogenetic tree, we are representing our best hypothesis about how a set of species (or other groups) evolved from a common ancestor

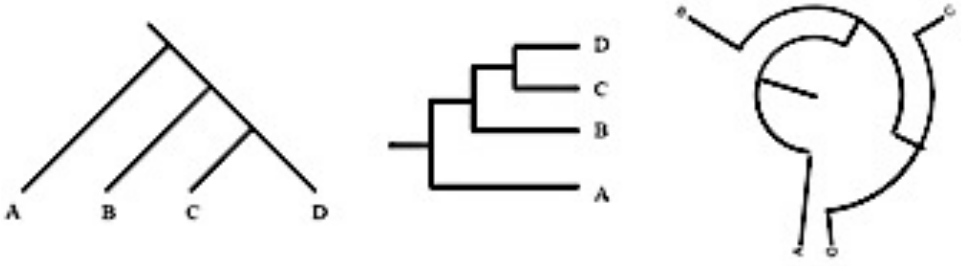
In a phylogenetic tree, the species or groups of interest are found at the tips of lines referred to as the tree's **branches**. For example, the phylogenetic tree below represents relationships between five species, A, B, C, D, and E, which are positioned at the ends of the branches:

The pattern in which the branches connect represents our understanding of how the species in the tree evolved from a series of common ancestors. Each branch point (also called an **internal node**) represents a **divergence** event, or splitting apart of a single group into two descendant groups.



At each branch point lies the **most recent common ancestor** of all the groups descended from that branch point. For instance, at the branch point giving rise to species A and B, we would find the most recent common ancestor of those two species. At the branch point right above the **root** of the tree, we would find the most recent common ancestor of all the species in the tree (A, B, C, D, E).

Similarly, tree diagrams can depict the same information yet be oriented in different ways. The three trees in Figure 6, for example, have the same topology and thus the same evolutionary implications. In each case, the first divergence event separated the lineage that gave rise to tip A from the lineage that gave rise to tips B, C, and D. The latter lineage then split into two lineages, one of which developed into tip B, and the other which gave rise to tips C and D. What this means is that C and D share a more recent common ancestor with each other than either shares with A or B. Tips C and D are therefore more closely related to each other than either is to tip A or tip B. The diagram also shows that tips B, C, and D all share a more recent common ancestor with each other than they do with tip A. Because tip B is an equal distance (in terms of branch arrangement) from both C and D, we could say that B is equally related to C and D. Likewise, B, C, and D are all equally related to A.



Neighbour Joining Method

The Neighbour Joining method is a method for re-constructing phylogenetic trees, and computing the lengths of the branches of this tree. In each stage, the two nearest nodes of the tree (the term "nearest nodes" will be defined in the following paragraphs) are chosen and defined as neighbours in our tree. This is done recursively until all of the nodes are paired together.

The algorithm was originally written by Saitou and Nei, 1987. In 1988 a correction for the paper was sent by Studier & Keppler. The correction was first of all for the proof of the algorithm, and second of all made a slight change to the algorithm which brought the efficiency down to O(n3) (as is explained in the following paragraphs). We will first of all describe the original algorithm, and then elaborate on the changes made by Studier & Kepler.

What are neighbours?

Neighbours are defined as a pair of OTU's (OTU=operational taxonomic units, or in other words – leaves of the tree), who have one node connecting them. For instance, in the tree in figure 1, nodes A and B are neighbours (connected by only one internal node), and nodes C and D are neighbours, whereas nodes A and C (for example) are not neighbours.

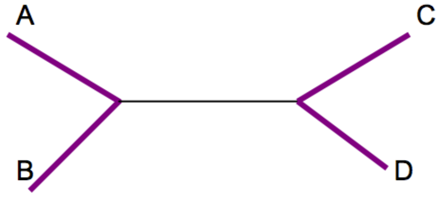


Figure 1

How do we find neighbours, and how de we construct our tree?

1. We start off with a star tree:

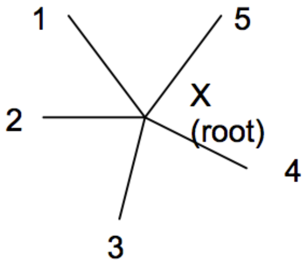


Figure 2(a)

2. We define some kind of distance parameter between our nodes (1 through 5), and enter this parameter into a distance matrix. (see following paragraphs). The columns and rows of the matrix represent nodes, and the value i,j of the matrix represent the distance between node i and node j. Note that the matrix is symmetric, and that the diagonal is irrelevant, therefore only the top half (or lower half) are enough.
3. We pick the two nodes with the lowest value in the matrix defined in step 2. These are defined as neighbours. For example, assuming nodes 1 and 2 are the nearest, we define them as neighbours.
4. The new node we have added is defined as node X.
5. We now define the distance between node X and the rest of the nodes, and enter these distances into our distance matrix. We remove nodes 1 and 2 from our distance matrix.
6. We compute the branch lengths for the branches that have been joined (for figure 2(b), these are branches 1-X and 2-X).
7. We repeat the process from stage 2 – once again we look for the 2 nearest nodes, and so on.

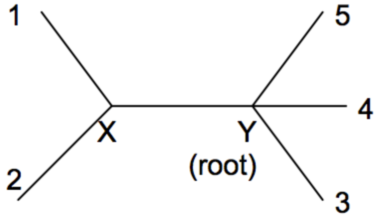


Figure 2(b)

Advantages of NJ

- Fast (suited for large datasets)
- Does not require ultrametric data: suited for datasets comprising lineages with largely varying rates of evolution
- Permits correction for multiple substitutions

Disadvantages of NJ

- Information is reduced (distance matrix based)
- Gives only one tree (out of several possible trees)
- The resulting tree depends on the model of evolution used

UPGMA

The unweighted pair-group method with arithmetic mean (UPGMA) is a popular distance analysis method.

UPGMA characteristics

- UPGMA is the simplest method for constructing trees.
- The great disadvantage of UPGMA is that it assumes the same evolutionary speed on all lineages, i.e. the rate of mutations is constant over time and for all lineages in the tree. This is called a 'molecular clock hypothesis'.
- This would mean that all leaves (terminal nodes) have the same distance from the root. In reality the individual branches are very unlikely to have the same mutation rate. Therefore, *UPGMA frequently generates wrong tree toplogies*!
- Generates rooted trees (re-rooting is not allowed!)
- Generates ultrametric trees

The UPGMA algorithm

- UPGMA starts with a matrix of pairwise distances D[1..n, 1..m].
- In the following text each sample (taxon, operational taxonomic unit=OTU) is denoted as a 'cluster'.
- starts by assigning all clusters (samples) to a star-like tree

1. Find that pair (cluster i and j) with the smallest distance value in the distance matrix: D[i,j].
2. Define a new cluster comprising cluster i and j:
   a. Cluster i is connected by a branch to the common ancestor node. The same applies for cluster j.
   b. Therefore, the distance D[i,j] is split onto the two branches. So, each of the two branches obtains a length of D[i,j]/2.
2. If i and j were the last 2 clusters, the tree is finished. If not the algorithm finds a new cluster called u.
3. Define the distance from u to each other cluster (k, with k <> i or j) to be an average of the distances dki and dkj.
   a. For 'Weighted PGMA (WPGM)': dku = dki+dkj/2).
   b. For 'Complete linkage': dku = max(dki, dkj).
   c. For 'Single linkage': dku = min(dki, dkj).

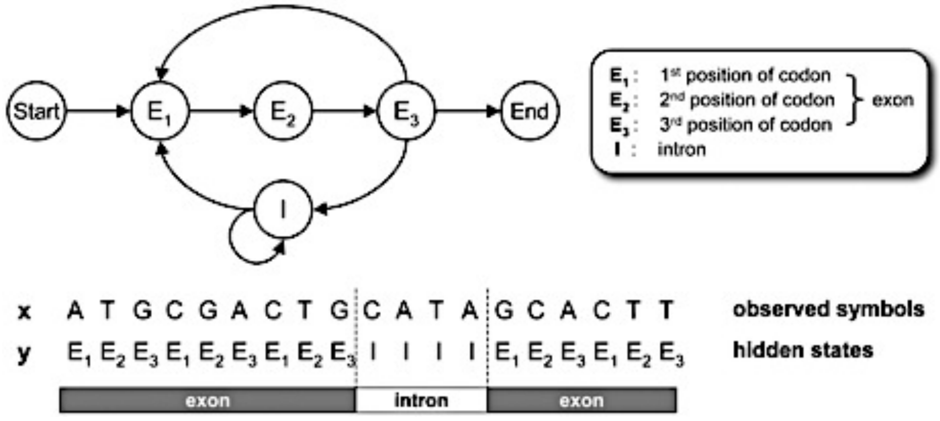
4. Go back to step 1 with one less cluster. Clusters i and j are eliminated, and cluster u is added to the tree.

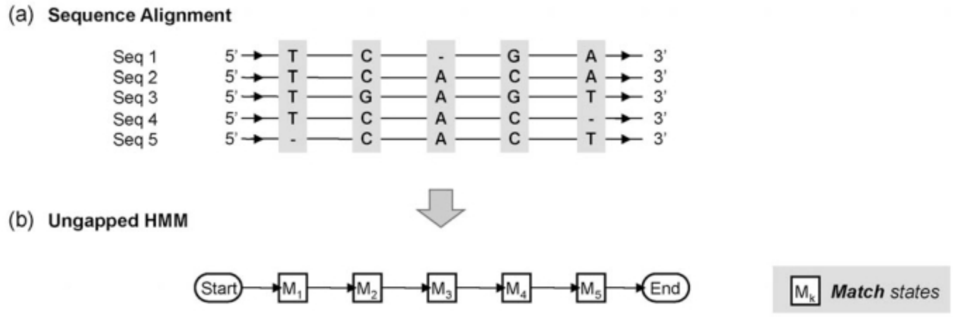
A *hidden Markov model (HMM)* is a statistical model that can be used to describe the evolution of observable events that depend on internal factors, which are not directly observable. We call the observed event a `symbol' and the invisible factor underlying the observation a `state'. An HMM consists of two stochastic processes, namely, an invisible process of hidden states and a visible process of observable symbols. The hidden states form a *Markov chain*, and the probability distribution of the observed symbol depends on the underlying state. For this reason, an HMM is also called a doubly-embedded stochastic process.

Markov Processes Diagram 1 depicts an example of a Markov process. The model presented describes a simple model for a stock market index. The model has three states, Bull, Bear and Even, and three index observations up, down, unchanged. The model is a finite state automaton, with probabilistic transitions between states. Given a sequence of observations, example: up-down-down we can easily verify that the state sequence that produced those observations was: Bull-Bear-Bear, and the probability of the sequence is simply the product of the transitions, in this case 0.2 × 0.3 × 0.3.

HMMs for detecting domains in protein sequences

There are numerous methods for detecting homology between protein sequences, but they fall into two basic categories: pair-wise methods (such as Blast[and FASTA) and profile methods. As mentioned earlier, HMMs are a profile method and although pair-wise methods are much faster, profile methods are able to detect much more distant relationships. Probably the most commonly used profile method is Position Specific Iterative Blast (PSI-Blast) which is comparable in performance (see later), but HMMs are also widely used for remote homology detection.


Start
E_1
E_2
E_3
End
I
E_1 : 1st position of codon
E_2 : 2nd position of codon
E_3 : 3rd position of codon
} exon
I : intron
x A T G C G A C T G C A T A G C A C T T
observed symbols
y E_1 E_2 E_3 E_1 E_2 E_3 E_1 E_2 E_3 I I I I E_1 E_2 E_3 E_1 E_2 E_3
hidden states
exon
intron
exon



Sequence homology

Well over 80% of the biological information concerning protein sequences reported in the published literature has been inferred by homology".

Proteins are the fundamental building blocks of life, and they are completely specified by a simple linear DNA sequence code. Since the invention of dideoxy sequencing it has been very easy to determine these sequences, and millions have already been determined. The rate at which new sequences are being determined is rising faster than that dictated by Moore's law, and more importantly the complete sequences of whole genomes, including the human, are now available.

The sequence alone does not tell us anything about the protein which it codes for. Further information can be obtained by experimentation, but it is very expensive and time consuming. The number of proteins for which there is experimental information is very small relative to the number of proteins for which the sequence is now known. However almost all sequences are homologous to some other sequences, and this homology can be detected using computer algorithms which are very cheap and fast. Therefore information about a large number of sequences can be inferred from knowledge about a small number.

Sequence homology provides reliable information about the large number of sequences being generated by the sequencing projects. The value of this information makes homology methods the most important computational tool in genome biology.

Sequence analysis

In bioinformatics, **sequence analysis** is the process of subjecting a DNA, RNA or peptide sequence to any of a wide range of analytical methods to understand its features, function, structure, or evolution. Methodologies used include sequence alignment, searches against biological databases, and others. Since the development of methods of high-throughput production of gene and protein sequences, the rate of addition of new sequences to the databases increased exponentially. Such a collection of sequences does not, by itself, increase the scientist's understanding of the biology of organisms. However, comparing these new sequences to those with known functions is a key way of understanding the biology of an organism from which the new sequence comes. Thus, sequence analysis can be used to assign function to genes and proteins by the study of the similarities between the compared sequences. Nowadays, there are many tools and techniques that provide the sequence comparisons (sequence alignment) and analyze the alignment product to understand its biology.

Sequence analysis in molecular biology includes a very wide range of relevant topics:

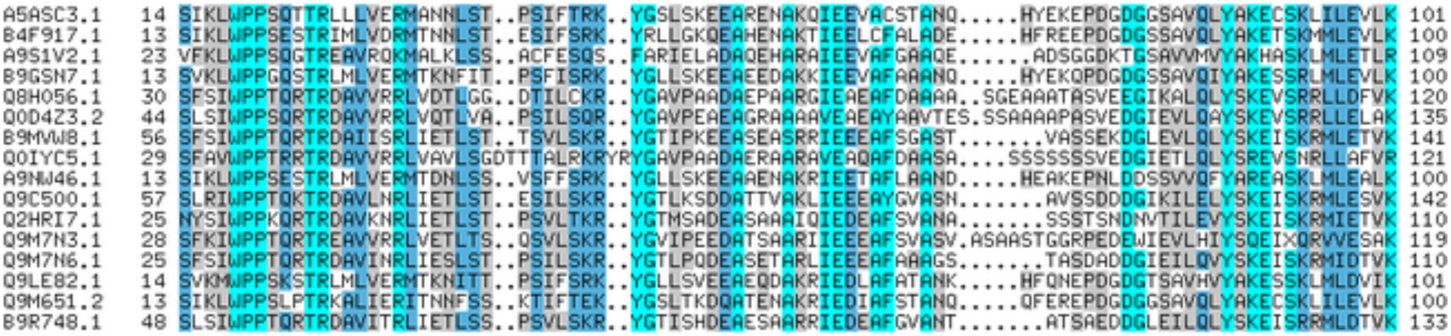
- The comparison of sequences in order to find similarity, often to infer if they are related (homologous)
- Identification of intrinsic features of the sequence such as active sites, post translational modification sites, gene-structures, reading frames, distributions of introns and exons and regulatory elements
- Identification of sequence differences and variations such as point mutations and single nucleotide polymorphism (SNP) in order to get the genetic marker.
- Revealing the evolution and genetic diversity of sequences and organisms
- Identification of molecular structure from sequence alone

In chemistry, sequence analysis comprises techniques used to determine the sequence of a polymer formed of several monomers. In molecular biology and genetics, the same process is called simply "sequencing".

In marketing, sequence analysis is often used in analytical customer relationship management applications, such as NPTB models (Next Product to Buy).

In sociology, sequence methods are increasingly used to study life-course and career trajectories, patterns of organizational and national development, conversation and interaction structure, and the problem of work/family synchrony. This body of research has given rise to the emerging subfield of social sequence analysis.

Sequence Alignment



```
A5ASC3.1    14 SIKLWPPSQTTRLLLVERMANNLST..PSIFTRK..YGSLSKEEARENAKQIEEVACSTANQ.....HYEKEPDGDGGSAVQLYAKECSKLILEVLK 101
B4F917.1    13 SIKLWPPSESTRIMLVDRMTNNLST..ESIFSRK..YRLLGKQEAHENAKTIEELCFALADE.....HFREEPDGDGSSAVQLYAKETSKMMLEVLK 100
A9S1V2.1    23 VFKLWPPSQGTREAVRQKMALKLSS..ACFESQS..FARIELADAQEHARAIEEVAFGAAQE......ADSGGDKTGSAVVMVYAKHASKLMLETLR 109
B9GSN7.1    13 SVKLWPPGQSTRLMLVERMTKNFIT..PSFISRK..YGLLSKEEAEEDAKKIEEVAFAAANQ.....HYEKQPDGDGSSAVQIYAKESSRLMLEVLK 100
Q8H056.1    30 SFSIWPPTQRTRDAVVRRLVDTLGG..DTILCKR..YGAVPAADAEPAARGIEAEAFDAAAA..SGEAAATASVEEGIKALQLYSKEVSRRLLDFVK 120
Q0D4Z3.2    44 SLSIWPPSQRTRDAVVRRLVQTLVA..PSILSQR..YGAVPEAEAGRAAAAVEAEAYAAVTES.SSAAAAPASVEDGIEVLQAYSKEVSRRLLELAK 135
B9MVW8.1    56 SFSIWPPTQRTRDAIISRLIETLST..TSVLSKR..YGTIPKEEASEASRRIEEEAFSGAST.......VASSEKDGLEVLQLYSKEISKRMLETVK 141
Q0IYC5.1    29 SFAVWPPTRRTRDAVVRRLVAVLSGDTTTALRKRYRYGAVPAADAERAARAVEAQAFDAASA....SSSSSSSVEDGIETLQLYSREVSNRLLAFVR 121
A9NW46.1    13 SIKLWPPSESTRLMLVERMTDNLSS..VSFFSRK..YGLLSKEEAAENAKRIEETAFLAAND.....HEAKEPNLDDSSVVQFYAREASKLMLEALK 100
Q9C500.1    57 SLRIWPPTQKTRDAVLNRLIETLST..ESILSKR..YGTLKSDDATTVAKLIEEEAYGVASN.......AVSSDDDGIKILELYSKEISKRMLESVK 142
Q2HRI7.1    25 NYSIWPPKQRTRDAVKNRLIETLST..PSVLTKR..YGTMSADEASAAAIQIEDEAFSVANA.......SSSTSNDNVTILEVYSKEISKRMIETVK 110
Q9M7N3.1    28 SFKIWPPTQRTREAVVRRLVETLTS..QSVLSKR..YGVIPEEDATSAARIIEEEAFSVASV.ASAASTGGRPEDEWIEVLHIYSQEIXQRVVESAK 119
Q9M7N6.1    25 SFSIWPPTQRTRDAVINRLIESLST..PSILSKR..YGTLPQDEASETARLIEEEAFAAAGS.......TASDADDGIEILQVYSKEISKRMIDTVK 110
Q9LE82.1    14 SVKMWPPSKSTRLMLVERMTKNITT..PSIFSRK..YGLLSVEEAEQDAKRIEDLAFATANK.....HFQNEPDGDGTSAVHVYAKESSKLMLDVIK 101
Q9M651.2    13 SIKLWPPSLPTRKALIERITNNFSS..KTIFTEK..YGSLTKDQATENAKRIEDIAFSTANQ.....QFEREPDGDGGSAVQLYAKECSKLILEVLK 100
B9R748.1    48 SLSIWPPTQRTRDAVITRLIETLSS..PSVLSKR..YGTISHDEAESAARRIEDEAFGVANT.......ATSAEDDGLEILQLYSKEISRRMLDTVK 133
```

Example multiple sequence alignment

There are millions of protein and nucleotide sequences known. These sequences fall into many groups of related sequences known as protein families or gene families. Relationships between these sequences are usually discovered by aligning them together and assigning this alignment a score. There are two main types of sequence alignment. Pair-wise sequence alignment only compares two sequences at a time and multiple sequence alignment compares many sequences in one go. Two important algorithms for aligning pairs of sequences are the Needleman-Wunsch algorithm and the Smith-Waterman algorithm. Popular tools for sequence alignment include:

- Pair-wise alignment - BLAST
- Multiple alignment - ClustalW, PROBCONS, MUSCLE, MAFFT, and T-Coffee.

A common use for pairwise sequence alignment is to take a sequence of interest and compare it to all known sequences in a database to identify homologous sequences. In general the matches in the database are ordered to show the most closely related sequences first followed by sequences with diminishing similarity. These matches are usually reported with a measure of statistical significance such as an Expectation value.

Profile comparison

In 1987, Michael Gribskov, Andrew McLachlan, and David Eisenberg introduced the method of profile comparison for identifying distant similarities between proteins. Rather than using a single sequence, profile methods use a multiple sequence alignment to encode a profile which contains information about the conservation level of each residue. These profiles can then be used to search collections of sequences to find sequences that are related. Profiles are also known as Position Specific Scoring Matrices (PSSMs). In 1993, a probabilistic interpretation of profiles was introduced by David Haussler and colleagues using hidden Markov models. These models have become known as profile-HMMs.

In recent years, methods have been developed that allow the comparison of profiles directly to each other. These are known as profile-profile comparison methods.

Sequence assembly

Sequence assembly refers to the reconstruction of a DNA sequence by aligning and merging small DNA fragments. It is an integral part of modern DNA sequencing. Since presently-available DNA sequencing technologies are ill-suited for reading long sequences, large pieces of DNA (such as genomes) are often sequenced by (1) cutting the DNA into small pieces, (2) reading the small fragments, and (3) reconstituting the original DNA by merging the information on various fragment.

Recently sequencing multiple species at one time is one of the top research target. Metagenomics is studying microbial communities directly obtained from environment. Different from cultured microorganism from lab, the wild sample usually contains dozens, sometimes even thousands types of microorganisms from their original habitats. Recovering the original genomes is a real challenging work. Most recently Projects:

Global Ocean survey (GOS)
Data Download
Human Microbiome Project (HMP)
Data Download
Earth Microbiome Project (EMP)

Gene prediction

Gene prediction or gene finding refers to the process of identifying the regions of genomic DNA that encode genes. This includes protein-coding genes as well as RNA genes, but may also include prediction of other functional elements such as regulatory regions. Gene finding is one of the first and most important steps in understanding the genome of a species once it has been sequenced. In general the prediction of bacterial genes is significantly simpler and more accurate than the prediction of genes in eukaryotic species that usually have complex intron/exon patterns.Identifying genes in long sequences remains a problem, especially when the number of genes is unknown. Hidden markov model can be part of the solutions. Machine learning has played a significant role in predicting the sequence of transcription factors. Traditional sequencing analyzing used focused on the statistical parameters of nucleotide sequence itself Another way is identifying homologous sequence based on other known gene sequence. Those two methods are both focusing on sequence. However, nowadays the shape feature of these molecules such as DNA and protein have also been studied and proposed to have an equivalent influence on the behaviors of these molecular as the sequence, if not higher.

Protein Structure Prediction

Target protein structure, with Calpha backbones of 354 predicted models for it submitted in the CASP8 structure-prediction experiment.

The 3D structures of molecules are of great importance to their functions in nature. Since structural prediction of large molecules at an atomic level is largely intractable problem, some biologists introduced ways to predict 3D structure at a primary sequence level. This includes biochemical or statistical analysis of amino acid residues in local regions and structural inference from homologs (or other potentially related proteins) with known 3D structures.

There have been a large number of diverse approaches to solve the structure prediction problem. In order to determine which methods were most effective a structure prediction competition was founded called CASP (Critical Assessment of Structure Prediction).

Benefits of DNA sequencing:-

DNA sequencing enables the scientists to determine genome sequence. Human genome project is the biggest example of DNA sequencing. When the human genome was sequenced back in 2001, many issue rose but now after many year, we can see its impacts on medical and pharmaceutical research. Scientists are now able to identify the genes which are responsible for causing genetic diseases like Alzheimer's disease, Cystic fibrosis, myotonic dystrophy and many other diseases caused by the disability of genes to function properly. Many types of acquired diseases like cancers can also be detected by observing certain genes.

Applications of DNA sequencing:-

Forensics:-

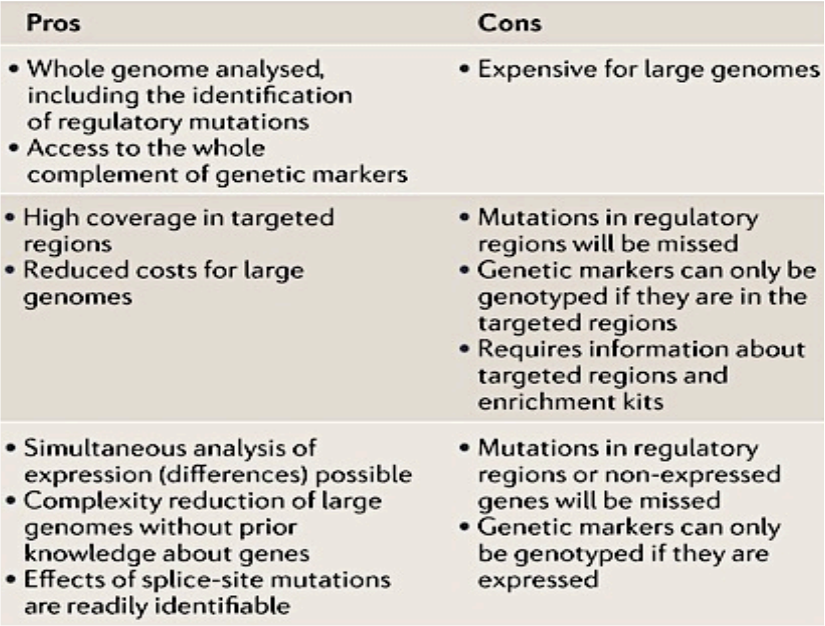
DNA sequencing has been applied in forensics science to identify particular individual because every individual has unique sequence of his/her DNA. It is particularly used to identify the criminals by finding some proof from the crime scene in the form of hair, nail, skin or blood samples. DNA sequencing is also used to determine the paternity of the child. Similarly, it also identifies the endangered and protected species.

Medicine:-

In medical research, DNA sequencing can be used to detect the genes which are associated with some heredity or acquired diseases. Scientists use different techniques of genetic engineering like gene therapy to identify the defected genes and replace them with the healthy ones.

Agriculture:-

DNA sequencing has played vital role in the field of agriculture. The mapping and sequencing of the whole genome of microorganisms has allowed the agriculturists to make them useful for the crops and food plants. For example, specific genes of bacteria have been used in some food plants to increase their resistance against insects and pests and as a result the productivity and nutritional value of the plants also increases. These plants can also fulfill the need of food in poor countries. Similarly, it has been useful in the production of livestock with improved quality of meat and milk.



Pros	Cons
• Whole genome analysed, including the identification of regulatory mutations • Access to the whole complement of genetic markers	• Expensive for large genomes
• High coverage in targeted regions • Reduced costs for large genomes	• Mutations in regulatory regions will be missed • Genetic markers can only be genotyped if they are in the targeted regions • Requires information about targeted regions and enrichment kits
• Simultaneous analysis of expression (differences) possible • Complexity reduction of large genomes without prior knowledge about genes • Effects of splice-site mutations are readily identifiable	• Mutations in regulatory regions or non-expressed genes will be missed • Genetic markers can only be genotyped if they are expressed