Year	Sem.	Subject Code	Title of the paper	Hours/ Week
2018 -2019 onwards	VI	18BBO63C	CORE PAPER XII BIOTECHNOLOGY AND BIOINFORMATICS	5

Unit – V

DNA sequence databases (EMBL, GenBank and DDBJ), Sequence file formats, Sequence alignment – local and global -Pair wise and multiple sequence analysis, BLAST implications, Gene identification and prediction. Proteomics: Protein sequence databases (SWISS PROT and PDB), Protein visualization tools (Rasmol) - Protein structure prediction - Homology modeling of protein (Swiss pdb Viewer).

Prepared by
Dr. M M Sudheer Mohammed
Associate Professor of otany
Mobile-9443274469

III BSc
Bioinformatics Notes: UNIT - V

Bioinformatics is an interdisciplinary science, emerged by the combination of various disciplines like biology, computer science, information technology, mathematics and statistics, to develop methods for storage, retrieval and analyses of biological data. Paulien Hogeweg, a Dutch system-biologist, was the first person who used the term "Bioinformatics" in 1970, referring to the use of information technology for studying biological systems.

Bioinformatics is an interdisciplinary research area at the interface between biological science and computer science. A variety of definitions exist in the literature and on the World Wide Web; some are more inclusive than others. Bioinformatics is a union of biology and informatics. Bioinformatics involves the technology that uses computers for storage, retrieval, manipulation, and distribution of information related to biological macromolecules such as DNA, RNA, and proteins.

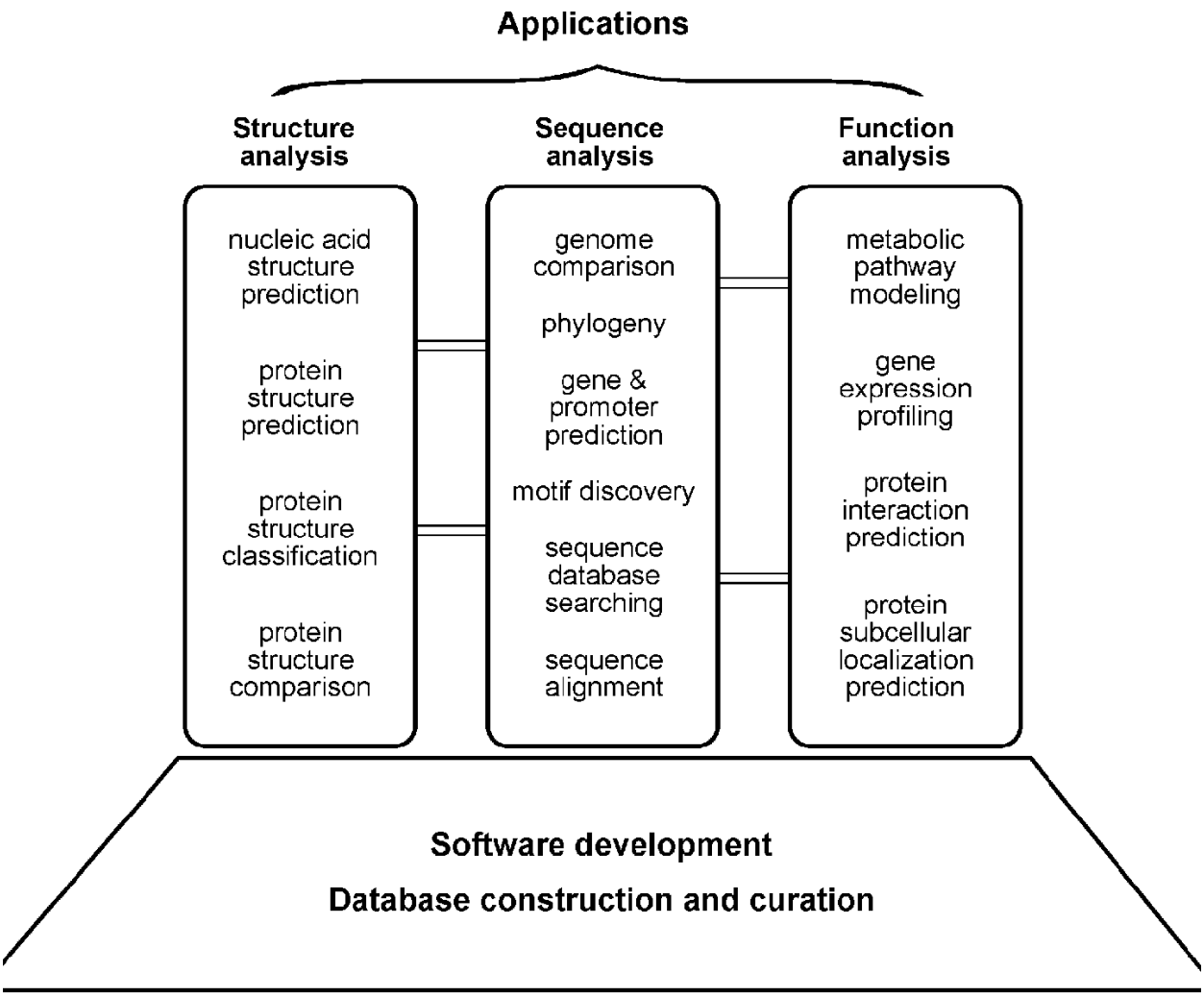
Bioinformatics differs from a related field known as computational biology. Bioinformatics is limited to sequence, structural, and functional analysis of genes and genomes and their corresponding products and is often considered computational molecular biology. However, computational biology encompasses all biological areas that involve computation. For example, mathematical modeling of ecosystems, population dynamics, application of the game theory in behavioral studies, and phylogenetic construction using fossil records all employ computational tools, but do not necessarily involve biological macromolecules.

Beside this distinction, it is worth noting that there are other views of how the two terms relate. For example, one version defines bioinformatics as the development and application of computational tools in managing all kinds of biological data, whereas computational biology is more confined to the theoretical development of algorithms used for bioinformatics.

Bioinformatics consists of two subfields: the development of computational tools and databases and the application of these tools and databases in generating biological knowledge to better understand living systems. These two subfields are complementary to each other. The tool development includes writing software for sequence, structural, and functional analysis, as well as the construction and curating of biological databases. These tools are used in three areas of genomic and molecular biological research: molecular sequence analysis, molecular structural analysis, and molecular functional analysis. The analyses of biological data often generate new problems and challenges that in turn spur the development of new and better computational tools.

The areas of sequence analysis include sequence alignment, sequence database searching, motif and pattern discovery, gene and promoter finding, reconstruction of evolutionary relationships, and genome assembly and comparison. Structural analyses include protein and nucleic acid structure analysis, comparison, classification, and prediction. The functional analyses include gene expression profiling, protein-protein interaction prediction, protein subcellular localization prediction, metabolic pathway reconstruction, and simulation.

The three aspects of bioinformatics analysis are not isolated but often interact to produce integrated results. For example, protein structure prediction depends on sequence alignment data; clustering of gene expression profiles requires the use of phylogenetic tree construction methods derived in sequence analysis. Sequence-based promoter prediction is related to functional analysis of co expressed genes. Gene annotation involves a number of activities, which include distinction between coding and noncoding sequences, identification of translated protein sequences, and determination of the gene's evolutionary relationship with other known genes; prediction of its cellular functions employs tools from all three groups of the analyses.



coexpressed genes. Gene annotation involves a number of activities, which include distinction between coding and noncoding sequences, identification of translated protein sequences, and determination of the gene's evolutionary relationship with other known genes; prediction of its cellular functions employs tools from all three groups of the analyses.

Computational tools are routinely used for characterization of genes, determining structural and physiochemical properties of proteins, phylogenetic analyses, and performing simulations to study how biomolecule interact in a living cell.

Biological databases

Biological databases can be broadly classified in to sequence and structure databases. Sequence databases is applicable to both nucleic acid sequences and protein sequences, whereas structure database is applicable to only Proteins

Sequence Databases

Biological sequence database refers to a vast collection of information about biological molecules such as nucleic acids, proteins and other biopolymers, each molecule to be identified by a unique key. The stored information is not only important for future use but also serves as a tool for primary sequence analyses. With the advancement of high throughput sequencing techniques, the sequencing has reached to a whole-genome scale, which is generating a massive amount of data every day. The submission and storage of this biological sequence information (DNA/RNA/PROTEIN) to become freely available to the scientific community has led to the development of various databases worldwide. Each database has become an autonomous representation of a molecular unit of life. Thus, an understanding of these databases will help to retrieve important information from these data collections relevant to one's project.

The primary DNA sequence databases are repositories (store house) for raw sequence data, and can be accessed freely over the World Wide Web. There are three such important databases; comprising the International Nucleotide Sequence Database Collaboration. These are **GenBank** maintained by the National Center for Biotechnology Information (NCBI), the **DNA Databank of Japan** (DDBJ) and the Nucleotide Sequence Database maintained by the European Molecular Biology Laboratory (EMBL), and new sequences can be deposited in any of the database since they exchange data on a daily basis.

The databases contain not only sequences but also extensive annotations. Annotation means obtaining useful information; that is, the structure and function of genes and other genetic elements, from raw sequence data to differences in gene structure and genome organization.

As an example, the **molecular file format of a GenBank file**, shows that much of the introductory part as self-explanatory, containing information such as the locus name, the accession number, the source species, literature references, and the date of submission. An important section of the file is the features table, which describes interesting features of the sequence.

```
LOCUS       HUMBTEB      4859 bp    mRNA            PRI       07-FEB-1999
DEFINITION  Human mRNA for GC box binding protein, complete cds.
ACCESSION   D31716
VERSION     D31716.1  GI:505081
KEYWORDS    GC box binding protein; zinc finger.
SOURCE      Homo sapiens germline cDNA to mRNA, clone_lib:placenta.
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
REFERENCE   1
            (.....)
REFERENCE   2  (bases 1 to 4859)
  AUTHORS   Ohe,N., Yamasaki,Y., Sogawa,K., Inazawa,J., Ariyama,T., Oshimura,M.
            and Fujii-Kuriyama,Y.
  TITLE     Chromosomal localization and cDNA sequence of human BTEB, a GC box
            binding protein
  JOURNAL   Somat. Cell Mol. Genet. 19 (5), 499-503 (1993)
  MEDLINE   94120483
COMMENT     Submitted (31-May-1994) to DDBJ by:
            Yoshiaki Fujii-Kuriyama
            (.....)
FEATURES             Location/Qualifiers
     source          1..4859
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
                     /clone_lib="placenta"
                     /germline
     gene            1265..1999
                     /gene="BTEB"
     CDS             1265..1999
                     /gene="BTEB"
                     /note="three-times repeated zinc finger motif"
                     /codon_start=1
                     /product="GC box binding protein"
                     /protein_id="BAA06524.1"
                     /db_xref="GI:1060891"
                      translation="MSAAAYMDFVAAQCLVSISNRAAVPEHGVAPDAERLRLPEREVT
                     KEHGDPGDTWKDYCTLVTIAKSLLDLNKYRPIQTPSVCSDSLESPDEDMGSDSDVTTE
                     SGSSPSHSPEERQDPGSAPSPLSLLHPGVAAKGKHASEKRHKCPYSGCGKVYGKSSHL
                     KAHYRVHTGERPFPCTWPDCLKKFSRSDELTRHYRTHTGEKQFRCPLCEKRFMRSDHL
                     TKHARRHTEFHPSMIKRSKKALANAL"
BASE COUNT     1285 a   1111 c   1193 g   1270 t
ORIGIN      Chromosome 9, q13.
        1 cacgttgggt gacataatgg ggttttttta attatagatt cacactgcat ttattcatca
```

Fig; 1 Molecular file (Flat file) format of a GenBank file

The main sequence databases have a number of subsidiaries for the storage of particular types of sequence data. For example, dbEST is a division of GenBank, which is used to store expressed sequence tags (ESTs). Other divisions of GenBank include dbGSS, which is used to store single-pass genomic sequences (genome survey sequences), dbSTS, which is used to store sequence tagged sites (unique genomic sequences that can be used as physical markers), and the HTG (high-throughput genomic) division, which is used to store unfinished genomic sequence data.

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

The DDBJ Center, a part of NIG, is funded as a supercomputing center. The web services, including submission systems, data retrieval systems, Web API, DDBJ Read Annotation Pipeline, and backend databases are performed on the NIG supercomputer system. The current commodity based cluster was implemented in 2012.

The sequences collected from the submitters are stored in the form of an entry in the database. Each entry consists of a nucleotide sequence, author information, reference, organism from which the sequence is determined, properties of the sequence etc.

```
LOCUS       AB003522                1192 bp    DNA     linear   PLN 14-FEB-2004
DEFINITION  Arabidopsis thaliana leucoplast genes for larger subunit of
            Rubisco, beta subunit of coupling factor one, partial cds.
ACCESSION   AB003522
VERSION     AB003522.1
KEYWORDS    .
SOURCE      leucoplast Arabidopsis thaliana (thale cress)
  ORGANISM  Arabidopsis thaliana
            Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
            Spermatophyta; Magnoliophyta; eudicotyledons; Gunneridae;
            Pentapetalae; rosids; malvids; Brassicales; Brassicaceae;
            Camelineae; Arabidopsis.
REFERENCE   1  (bases 1 to 1192)
  AUTHORS   Kobayashi,H.
  TITLE     Direct Submission
  JOURNAL   Submitted (06-MAY-1997) to the DDBJ/EMBL/GenBank databases.
            Contact:Hirokazu Kobayashi
            University of Shizuoka, Graduate School of Nutritional and
            Environmental Sciences; 52-1 Yada, Shizuoka, Shizuoka 422, Japan
REFERENCE   2
  AUTHORS   Isono,K., Niwa,Y., Satoh,K. and Kobayashi,H.
  TITLE     Evidence for transcriptional regulation of plastid photosynthesis
            genes in Arabidopsis thaliana roots
  JOURNAL   Plant Physiol. 114, 623-630 (1997)
REFERENCE   3
  AUTHORS   Isono,K. and Kobayashi,H.
  TITLE     Distinct control of expression of plastid genes with different
            promoter  structures in Arabidopsis thaliana
  JOURNAL   Unpublished (1997)
COMMENT     A region encoding promoters of rbcL for the large subunit of
            Rubisco and atpB/E operon for beta and epsilon subunits of
            coupling factor one cloned and sequenced.

            The rbcL ////////////// (Applied Biosystems) following the
            manufacture's instruction.

            The nucleotide sequence compiled here is that between primers No.1
            and No.2. The sequence data was completed on January 31 1992.
FEATURES             Location/Qualifiers
     source          1..1192
                     /db_xref="taxon:3702"
                     /ecotype="Columbia"
                     /mol_type="genomic DNA"
                     /organelle="plastid:leucoplast"
                     /organism="Arabidopsis thaliana"
     CDS             complement(<1..245)
```

```
                     /codon_start=1
                     /gene="atpB"
                     /product="beta subunit of coupling factor one"
                     /protein_id="BAA20945.1"
                     /transl_table=11
                     /translation="MRTNPTTSNPEVSIREKKNLGRIAQIIGPVLDVAFPPGKMPNIY
                     NALVVKGRDTLGQEINVTCEVQQLLGNNRVRAVAMSAT"
     misc_feature    complement(505)
                     /note="one of possible initiation site of atpB/E"
     regulatory      complement(510..515)
                     /note="one of possible -10 sequence"
                     /regulatory_class="minus_10_signal"
     regulatory      complement(537..542)
                     /note="one of possible -35_signal"
                     /regulatory_class="minus_35_signal"
     misc_feature    complement(707)
                     /note="one of possible initiation site of atpB/E"
     regulatory      complement(716..721)
                     /note="one of possible -10_sequence"
                     /regulatory_class="minus_10_signal"
     regulatory      complement(739..744)
                     /note="one of possible -35_signal"
                     /regulatory_class="minus_35_signal"
     regulatory      834..839
                     /regulatory_class="minus_35_signal"
     regulatory      858..863
                     /regulatory_class="minus_10_signal"
     misc_feature    868
                     /note="a putative transcription initiation site of rbcL"
     regulatory      1037..1041
                     /regulatory_class="ribosome_binding_site"
                     /standard_name="Shine-Dalgarno sequence"
     CDS             1047..>1192
                     /codon_start=1
                     /gene="rbcL"
                     /product="larger subunit of Rubisco"
                     /protein_id="BAA20946.1"
                     /transl_table=11
                     /translation="MSPQTETKASVGFKAGVKEYKLTYYTPEYETKDTDILAAFRVTP
                     QPGVP"
BASE COUNT          388 a          190 c          196 g          418 t
ORIGIN
        1 gtagcactca tagctacagc tctaactcga ttatttccta ataattgctg tacttcacaa
       61 gtcacattaa tttcttgacc aagagtatct cgacccttaa ccaccagagc attgtaaata
      121 ttaggcattt tgcccggggg gaaggctaca tccagtaccg gaccaatgat ttgggcgata
      181 cgtcccaggt ttttttttc acgtatcgaa acctctggat ttgaagtagt aggatttgtt
      241 ctcataataa aaaaaatatg ttaaattttg ttacgaattt tttcgaatac agaaaaaatc
      301 ttcgatagca aattaatcgg ttaattcaat aaaaagtggg agtaagcact cgatttcgtt
      361 ggtcccaccc aagcggatgt ggaattcaat tttttattca ttcaatgaag gaatagtcat
      421 tttcaagctc aactaactga aacctagttt taaaataaaa aatatatgaa taaaaaaatt
      481 ttttgcggaa agtcttttat ttttttatca taataggaat aggcaagcct ttgttttatc
      541 tagcgaattc gaaacggaac tttagttatg attcattatt tcgatctcat tagccttttt
      601 tttcgtattt tcatttttagc atatccggtt atgcgtccca tttattcatc ccttttagcaa
      661 ccccccccttg tttttcattt tcatggatga attccgcata ttgtcatatc taggatttac
      721 atatacaaca gatattactg tcaagagtga ttttattaat attttaattt taatattaaa
      781 tatttggatt tataaaaagt caaagattca aaacttgaaa aagaagtatt aggttgcgct
      841 atacatatga aagaatatac aataatgatg tatttggcga atcaaatatc atggtctaat
      901 aaagaataat tctgattagt tgataatttt gtgaaagatt cctgtgaaaa aggttaatta
      961 aatctattcc taatttatgt cgagtagacc ttgttgtttt gttttattgc aagaattcta
```

```
1021 aattcatgac ttgtagggag ggacttatgt caccacaaac agagactaaa gcaagtgttg
1081 ggttcaaagc tggtgttaaa gagtataaat tgacttacta tactcctgaa tatgaaacca
1141 aggatactga tatcttggca gcattccgag taactcctca acctggagtt cc
```

Fig; 2 Molecular file (Flat file) format of a DDBJ file

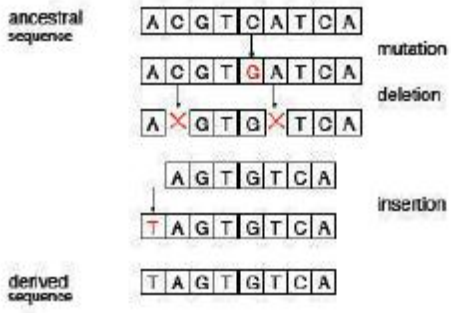
SEQUENCE ALIGNMENTS ANALYSIS

Sequence alignment is the process of lining up two or more sequences to achieve maximal levels of identity (and conservation, in the case of amino acid sequences) for the purpose of assessing the degree of similarity and the possibility of homology. Sequence similarity analy-sis is the single most powerful method for structural and functional inference available in databases. Sequence similarity analysis allows the inference of homology between proteins and homology can help one to infer whether the similarity in sequences would have similarity in function.

Genomes change over time, and the scarcity of ancient genomes makes it virtually impossible to compare the genomes of living species with those of their extinct ancestors. Thus, we are limited to comparing jus the genomes of living descendants. The goal of sequence alignment is to infer the edit operations that change a genome by looking only at these endpoints.

In practice, sequence evolution is mostly due to nucleotide mutations, deletions, and insertions.

1. A nucleotide mutation occurs when some **nucleotide in a sequence changes to some other nucleotide** during the course of evolution.
2. A nucleotide **deletion** occurs when some nucleotide is deleted from a sequence during the course of evolution.
3. A nucleotide **insertion** occurs when some nucleotide is added to a sequence during the course of evolution.



Note that these three events are all reversible. For example, if a nucleotide N mutates into some nucleotide M, it is also possible that nucleotide M can mutate into nucleotide N. Similarly, if nucleotide N is deleted, the event may be reversed if

nucleotide N is (re)inserted. Clearly, an insertion event is reversed by a corresponding deletion event.

Sequence similarity searches of databases enable us to extract sequences that are similar to a query sequence. Information about these extracted sequences can be used to predict the structure or function of the query sequence. Prediction using Similarity is a powerful and ubiquitous idea in bioinformatics. The underlying reason for this is molecular evolution. Any pair of DNA sequences will show some degree of similarity.

Sequence alignments are the first step in quantifying this in order to distinguish between chance similarity and real biological relationships. Alignments show the differences between sequences as changes (mutations), insertions or deletions (indels or gaps), and can be interpreted in evolutionary terms. Gap is a space introduced into an alignment to compensate for insertions and deletions in one sequence relative to another. In optimal alignment, non-identical characters and gaps are placed to bring as many identical or similar characters as possible into vertical register.

The sequence similarity analysis can be stated as—given two sequences how to find best alignment that can be obtained by sliding one sequence along the other. A major complication arises due to insertions or gaps in the alignment of sequences gaps in the alignment of sequences. To prevent the accumulation of too many gaps in an alignment, introduction of a gap causes the deduction of a fixed amount (the gap score) from the alignment score. Extension of the gap to encompass additional nucleotides or amino acid is also penalized in the scoring of an alignment. Usually, gap penalties (cost of inserting and extending gaps) are chosen to be length dependent. Typically, the cost of extending a gap (gap elongation) is 5-10 times lower than is the cost for introducing a gap (gap open). The process of alignment can be measured in terms of the number and length of gaps introduced, and the number of mis-matches remaining in the alignment.

A matrix relating such parameters represents the distance between two sequences. Various methodologies, mutation matrices (scoring matrices), dotplots, global and local sequence alignments and other algorithms are available to address the sequence alignment problem. Dynamic programming algorithms can calculate the best alignment of two sequences. Well-known variants are the Smith-Waterman algorithm (local alignments) and the Neeclleman-Wunsch algorithm (global alignments).

Local alignments are useful when sequences are not related over their full lengths, for example /proteins sharing only certain domains, or DNA sequences related only in exons. A simple alignment score measures the number or proportion of identically matching residues.

Gap penalties are subtracted from such scores to ensure that alignment algorithms produce biologically sensible alignments without too many gaps. Gap penalties may be constant

(independent of the length of the gap), proportional (proportional to the length of the gap), or affine (containing gap opening and gap extension contributions). Gap penalties can be varied according to the desired application. Sequence similarity can be quantified using the score from the alignment algorithm, percentage sequence identities, or more complex measures. Themost useful statistical measures are outlined below.

Similarity may exist between any sequences. Sequences are homologous only if they have evolved from a common ancestor. Homologous sequences often have similar biological functions (orthologues), but the mechanismof gene duplication allows homologous sequences to evolve different functions (paralogues). Protein sequences can be aligned to maximize amino acid identities; but this will not reveal distant evolutionary relationships. Protein coding sequences evolve slowly compared with most other parts of the genome, because of the need to maintain protein structure and function.

An exception to this is the fast evolution that might occur in the redundant copy of a recently duplicated gene.

Global Alignment

- Compares sequences and gives best overall alignment.
- Will return only the best matching segment for a given pair of sequences.
- May fail to find the best local region of similarity (e.g., a common motif) among the distantly related sequences.

Example: An alignment, given here, assumes that the two proteins are basically similar over the entire length of one another. The alignment attempts to match them to each other from end to end, even though parts of the alignment are not very convincing.

LGPSTKDFGKISESREFDN

I l l l l I

LNQLERSFGKINMRLEDA

In other words, the global sequence comparison algorithms seek to align every residue in one sequence with every residue in a second, in contrast to the more commonly used local sequence alignment algorithms, which seek only the strongest region of similarity between the two sequences. Global alignment algorithms are used for aligning families of sequences with similar lengths in preparation for phylogenetic analysis; global alignment scores can be transformed to the distance measures used for building evolutionary trees. Its similarity scores are rarely used to infer homology, however, as the distribution of global similarity scores is not well understood and thus it is difficult to assign a statistical significance to a global similarity score.

Local Alignment

- Finds regions of ungapped sequence with a high' degree of similarity.
- Better at finding motifs, especially for sequences that are different overall.
- Can return more than one matching segment for a given pair of sequences.

Example: An alignment searches for segments of the two sequences that match well. There is no attempt to force the entire sequences into an alignment, just those parts that appear to have good similarity, according to some criterion. Using the same sequences, given earlier as an example in the global alignment, one could get:

------FGKI---

------ ll ll -----

-------FGKI ------

It may seem that one should always use only the local alignments. However, it may be difficult to spot an overall similarity, as opposed to just a domain-to-domain similarity, if one uses only the local alignment. So the global alignment may be useful in some cases. The popular programs BLAST and FASTA for searching sequence databases produce local alignments.

Local alignment algorithms have two dramatic advantages over global alignment methods when searching sequence databases for statistically significant matches:

(1) the statistics of local similarity scores are well understood; and

(2) local alignments allow one to identify conserved domains in the proteins, which may not extend over the entire sequence.

Scoring Matrix

The correspondence between two aligned sequences can be expressed in terms of similarity/identity score. Scoring penalties are introduced to minimize the number of gaps. The total alignment score is then a function of the identity between aligned residues and the gap penalties incurred. A compilation of the similarity scores in pair-wise alignment into a matrix is called scoring matrix. Such matrices are constructed for:

- Evaluating match/mismatch between any two characters (residues).
- A score for insertion/deletion
- Optimization of total score.
- Evaluating the significance of the alignment.

Scoring matrices implicitly represent a particular theory of evolution. Elements of a matrix specify the weight to be assigned to a given comparison (i) by the measure of similarity for replacing one residue with another (similarity matrix), or (ii) by the cost for the replacement (distance matrix). Similarity matrices are used for database searching, while distance matrices are naturally used for phylogenetic tree construction.

The distance score (D) is usually calculated by summing up of mismatches in an alignment divided by the total number of matches and mismatches, which represents the number of changes required to change one sequence into the other, ignoring gaps.

PAM (Percent Accepted Mutation) matrix

Once the evolutionary relationship of two sequences is established, the residues that did exchange are similar (conservative mutations). This is the underlying principle behind the Dayhoff mutation data matrix compilation.

The Dayhoff mutation data matrix is based on the concept of the percentage-accepted mutation (PAM). Proteins are organized into families based on the degree of sequence similarity. From aligned sequences, a phylogenetic tree is derived showing graphically which sequences are mot related and therefore share a common branch on the tree. After the construction of the evolutionary trees, they are used with scoring matrices to evaluate the amino acid changes that occurred during evolution of the genes for the proteins in the organisms from which they originated. Subsequently, a set of tables (matrices), the percentage of amino acid mutations accepted by evolutionary selection, known as PAM tables are determined. PAM tables show which amino acids are most conserved and the corresponding positions in two sequences during evolution. Steps in the construction of mutation matrix are:

1. Align sequences that are at least 85% identical and determine pair exchange frequencies.

2. Compute frequencies of occurrence.

3. Compute relative mutabilities.

4. Compute a mutation probability matrix.

5. Compute evolutionary distance scale.

6. Calculate probability that two amino acid residues are aligned by evolutionary descendence to the probability that they are aligned by chance.

Limitatons of The PAM Model

The PAM model is built on the assumptions that are imperfect.

1. The replacement of any site (aminoacids) depends only on the amino acid at that site and the probability given by the table, is an imperfect representation of evolution. Replacement is not equally probable over entire sequence (e.g. local conserved sequences).

2. Each amino acid position is equally mutable is incorrect. Sites vary considerably in their degree of mutability.

3. Many sequences depart from average amino acid composition.

4. Errors in PAM1 are magnified in extrapolation to PAM250.

Blocks substi-tution matrix (BLOSUM)

In Blocks substi-tution matrix (BLOSUM) method, the starting data is conserved in blocks, and aligned in order to represent distant relationships more explicitly. In this method, the sequences of the indi-vidual proteins in each of the families are aligned in the regions defined by the blocks. Each column in the aligned sequences then provided a set of possible amino acid substitutions. The types of substitutions are then scored for all aligned patterns in the database and used to prepare a scoring matrix, the "BLOSUM" matrix, indicating the frequency of each type of substitution. More common (conservative) substitutions should represent a closer relation-ship between two amino acids in related proteins, and thus receive a more favorable score in sequence alignment. Conversely, radical substitutions should be less favored. Patterns of different identities are grouped in different groups—60% identical patterns are grouped under one substitution matrix blosum60, and those 80% alike under blosum80, and so on. BLOSUM matrix values are given as log-odds scores of the ratio of observed frequency of amino acid substitution dived by the frequency expected by chance. While PAM matrix is designed to track evolutionary origins of proteins, the BLOSUM model is designed to find their conserved domains. The better reliability of blocks method is due to:

1. Many sequences from aligned families are used to generate matrices.

2. Any potential bias introduced by counting multiple contributions from identical residue pairs is removed by clustering sequence segments on the basis of minimum percentage identity.

3. Clusters are treated as single sequences (Blosum60; Blosum80 etc.).

4. Log-odds matrix is calculated from the frequencies, Aij, of observing residue, i, in one cluster aligned against residue, j, in another cluster.

5. Derived from data representing highly conserved sequence segments from divergent proteins rather than data based on very similar sequences (as is the case with PAM matrices).

6. Detects distant similarities more reliably than Dayhoff matrices.

The BLAST Sequence Analysis Tool

Basic Local Alignment Search Tool (BLAST)

The comparison of nucleotide or protein sequences from the same or different organisms is a very powerful tool in molecular biology. By finding similarities between sequences, scientists can infer the function of newly sequenced genes, predict new members of gene families, and explore evolutionary relationships. Now that whole genomes are being sequenced, sequence similarity searching can be used to predict the location and function of protein-coding and transcription regulation regions in genomic DNA. Basic Local Alignment Search Tool (BLAST) is the tool most frequently used for calculating sequence similarity. BLAST comes in variations for use with different query sequences against different databases. All BLAST

applications, as well as information on which BLAST program to use and other help documentation, are listed on the BLAST homepage.

A sequence similarity search often provides the first information about a new DNA or protein sequence. A search allows scientists to infer the function of a sequence from similar sequences. There are many ways of performing a sequence similarity search, but probably the most popular method is the "Basic Local Alignment Search Tool" (BLAST). BLAST uses heuristics to produce results quickly. It also calculates an "expect value" that estimates how many matches would have occurred at a given score by chance, which can aid a user in judging how much confidence to have in an alignment. As the name implies, BLAST performs "local" alignments.

Most proteins are modular in nature, with one or more functional domains occurring within a protein. The same domains may also occur in proteins from different species. The BLAST algorithm is tuned to find these domains or shorter stretches of sequence similarity. The local alignment approach also means that an mRNA can be aligned with a piece of genomic DNA, as is frequently required in genome assembly and analysis. If instead BLAST started out by attempting to align two sequences over their entire lengths (known as a global alignment), fewer similarities would be detected, especially with respect to domains and motifs.

Basic Local Alignment Search Tool (BLAST) is from NCBI/GenBank (USA). It consists of a suite of algorithms, and they provide a fast, accurate and sensitive database searching. BLOSUM62 is the default-scoring matrix. BLAST works better on protein sequence databases. A general operational procedure is:

1. It takes each word (--short, fixed-length sequences based on the query) from the query sequence, optimally filtered to remove low-complexity regions and locates all similar words in the current test sequence. It initially throws away all database sequences that do not have a similar match.

2. If similar words are found (3 amino acids or 11 nucleotides), BLAST tries to expand the alignment to the adjacent words (gaps not allowed).

3. High-scoring segment pairs are generated. An HSP consists of two sequence fragments of arbitrary but equal length whose alignment is locally maximal and for which the alignment score is above the threshold score.

4. After all words are tested, a set of high-scoring segment pairs (HSPs) are chosen for that database sequence. Two sequences, a scoring system, and a threshold score define a set of HSPs.

5. Several non-overlapping HSPs may be combined in a statistical test to create a longer, more significant match.

A suite of BLAST programs is:

Un-gapped BLAST. The program may miss the similarity if two sequences do not have a single highly conserved region.

Gapped BLAST : Seeks only one from the un-gapped alignments that make up a significant match. Dynamic programming is used to extend a central pair of aligned residues in both directions to yield the final gapped alignment.

PSI-BLAST : Position-Specific Interactive BLAST is a generalized BLAST algorithm that incorporates both pair-wise and multiple sequence alignment methods. It is used for the identification of weak sequence similarities. It uses a position-specific score matrix in place of query sequence.

1. It takes as input a protein sequence and compares it to protein databanks, and constructs a multiple alignment from a Gapped BLAST search and generates a profile from any significant local alignment, called a "profile".

2. The profile is compared to the protein databases, again seeking best possible local alignments and PSI-BLAST estimates the statistical significance of the local alignments found, using "significant" hits to extend the profile search until convergence.

BLASTN : Compares the nucleotide query sequence against all nucleotide sequences in the non-redundant databases (DNA ® DNA). Suited for high-scoring matches; not suited for distant relationship matching.

BLASTP : Compares a protein query sequence against all protein sequences (gapped) in the non-redundant databases (Protein ® Protein). Suited for finding homologies.

BLASTX : The query nucleotide sequence will be translated in all six reading frames (each frame gapped) and the conceptual translation products are compared against all protein sequences in non-redundant databases (DNA translated ® protein). Suited for finding ESTs and new DNA searches for finding novel proteins.

TBLASTN : Compares a protein query sequence against nucleotide sequence databases, dynamically translated in all six reading frames (each frame gapped) (Protein ® DNA (translated). Suited for finding ESTs and novel proteins.

TBLASTX : Compares the six-frame translation of a nucleotide query sequence against the six-frame (ungapped) translation of nucleotide sequence databases (DNA (translated) ® DNA (translated). Suited for ESTs and gene structure annotations.

Once BLAST has found a similar sequence to the query in the database, it is helpful to have some idea of whether the alignment is "good" and whether it portrays a possible biological relationship, or whether the similarity observed is attributable to chance alone. BLAST uses statistical theory to produce a **bit score** and **expect value (E-value)** for each alignment pair (query to hit).

The bit score gives an indication of how good the alignment is; the higher the score, the better the alignment. In general terms, this score is calculated from a formula that takes into account the alignment of similar or identical residues, as well as any gaps introduced to align the sequences. A key element in this calculation is the "substitution matrix", which assigns a score for aligning any possible pair of residues. The BLOSUM62 matrix is the default for most BLAST programs, the exceptions being blastn and MegaBLAST (programs that perform nucleotide–nucleotide comparisons and hence do not use protein-specific matrices). Bit scores are normalized, which means that the bit scores from different alignments can be compared, even if different scoring matrices have been used.

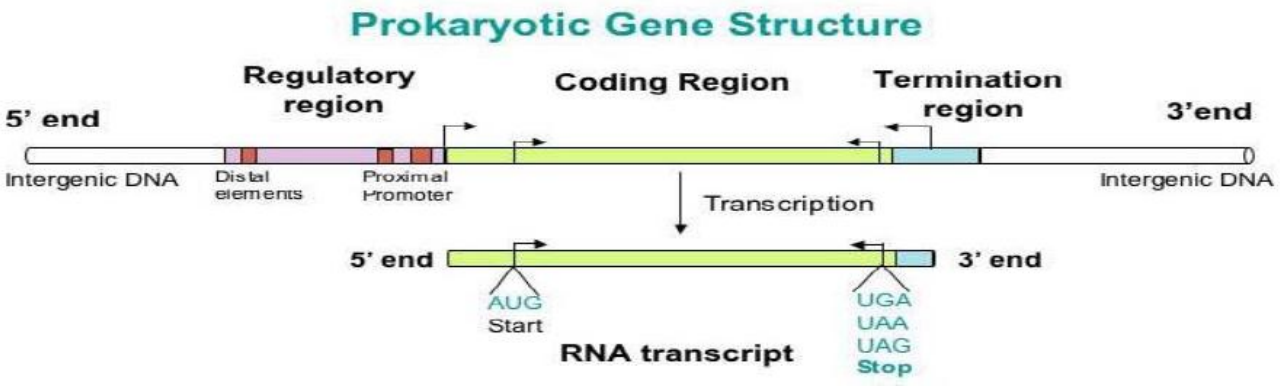
The E-value gives an indication of the statistical significance of a given pairwise alignment and reflects the size of the database and the scoring system used. The lower the E-value, the more significant the hit. A sequence alignment that has an E-value of 0.05 means that this similarity has a 5 in 100 (1 in 20) chance of occurring by chance alone. Although a statistician might consider this to be significant, it still may not represent a biologically meaningful result, and analysis of the alignments (see below) is required to determine "biological" significance.

Gene identification and prediction through Bioinformatics approaches

The process of identification of genomic DNA regions encoding proteins is defined as gene prediction or gene finding. Gene finding is one of the most significant process in understanding and analysis of an organism's genome after its sequencing. Bioinformatics approaches have great ability to predict the gene function based on its sequence alone. Further, gene finding process is able to predict structural genes which are fundamental basis for understanding biochemical process within the cells including transcription.

A DNA segment expressed for production of a functional product like a protein or RNA is called as a gene. Generally genes structure consist of following parts : *upstream* (intergenic region) , *promoter* (for example , TATA box with consensus sequence TATA(A/T)A(A/T), *first exon*(transcriptional start,5'-UTR), *intron(s)* (frequent stop codons), *exon(s)*(CDS/ORF and enhancer sites), *intron(s)* (frequent stop codons)*), last exon* Transcriptional stop, Poly A insertion sites , *downstream (intergenic region).*

Generally there are two types of genes based on organism: prokaryotic and eukaryotic genes which show following features: *prokaryotic genome:* small in size, high gene density, terminator important, no introns (or splicing), no RNA processing, similar promoters, and overlapping genes.



Eukaryotic genome: large in size, low gene density, terminator not important, presence of introns (or splicing), presence of RNA processing, heterogeneous promoters, polyadenylation. Knowledge of pattern recognition including gene feature and DNA characteristics are also important and prior to applying gene finding process, these are such as coding sequences (open reading frames (ORFs), GC-rich , CpG-content), PolyA-signals ,(consensus sequences), translational start and stop sites(start codons (ATG), stop one(TAA,TAG,TGA), splice sites, (consensus sequences) promoter regions(TATA, shine Dalgarno, Kozak consensus, CpG content, Prinbnow).

Eukaryotic gene structure

Totally gene finding methods can be divided into two types: laboratory based approaches and *computational* based approaches which itself consist of three types namely: *ab initio* methods, extrinsic methods (homology based) and comparative (statistical and HHM) based approaches.

1 *ab initio* (Intrinsic) methods:-

Predicts genes using only the genomic DNA sequence. It searches for signals and content (specific sequences, codon usage, GC content) of protein coding regions and statistical properties of the given DNA sequence. Example: GeneMarkS, Prodigal, Glimmer.

a) By identifying signal sensors in the genomic DNA.

Signals are short sequence segments of the DNA, that control translation or transcription. The various signals are

promoter: marks the begin of transcription

splice sites: 5' (donor) and 3' (acceptor) end of an intron

TIS: Contains the start codon (usually atg) and marks the begin of translation

stop codon: marks the end of translation (usually tga, taa or tag)

poly-A signal: triggers end of transcription.

These signals contain typical sequence motifs, but these motifs are not characteristic: The motifs occur also at positions where actually no signal is.

Example donor splice sites: (Almost) every intron begins with the dinucleotide gt, but that is not sufficiently specific and does not suffice for locating donor splice sites.

b) By identifying content sensors in the genomic DNA.

Coding sequences and non-coding sequences (introns, intergenic region) also typically have different base compositions. For example coding: bases g and c slightly more common non-coding: bases a and t slightly more common

Reading frame dependent hexamer frequencies is the most commonly used content sensor of current gene prediction programs.

2. Homology based (Extrinsic) methods:-

The given genome sequence is compared with an extrinsic genome (reference genome datasets) to find coding regions in the given genome. Example: BLAST

Gene structure is deduced using homologous sequences (EST, mRNA, protein). They are very accurate results when using homologous sequences with high similarity.

a) Alignment with cDNA

b) Alignment with ESTs

c) Protein Homology

Use local similarity between translated input DNA sequence and amino acid sequence from database to infer evidence about coding regions.

d) Cross-species DNA comparison

Consider the DNA sequences of two different species coding for the 'same' (or a similar) protein. Functional parts of the sequence, especially coding regions, tend to be more conserved.

With the huge amount of genomic data that are now available, a third way of predicting genes and other functional elements in genomic sequences is comparative sequence analysis. It is possible to identify functional regions in genomic DNA by comparing evolutionary related genomic sequences with each other. The rationale behind this approach is simple: during evolution, functional parts of sequences tend to be more highly conserved than non-functional parts, so local sequence conservation usually indicates biological functionality. Bafna and Huson (2000) utilized this fact and proposed gene-prediction methods that rely on comparing genomic sequences from related organisms.

The comparative gene-prediction approaches do not rely on statistical models derived from known genes of a given species, they can be applied to genome sequences from newly sequenced organisms where no training data are available - provided syntenic sequences are available from a second species at an appropriate evolutionary distance. With the increasing number of whole-genome sequencing projects, it will become easy to find syntenic sequence pairs from related organisms.

PAIR WISE AND MULTIPLE SEQUENCE ANALYSIS

Pairwise Sequence Alignment is used to identify regions of similarity that may indicate functional, structural and/or evolutionary relationships between two biological sequences (protein or nucleic acid).

Aim of sequence comparison through pairwise alignments

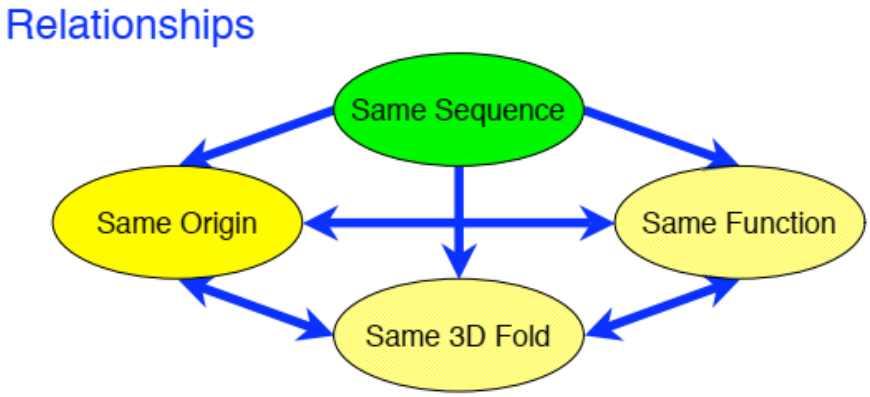
• Goal of pairwise comparison is to find conserved regions (if any) between two sequences

• Extrapolate information about our sequence using the known characteristics of the other sequence

Evolution of sequences

• Sequences evolve through mutation and selection! Selective pressure is different for each residue position in a protein (i.e. conservation of active site, structure, charge, etc.)

• Modular nature of proteins! Nature keeps re-using domains • Alignments try to tell the evolutionary story of the proteins.



Two similar regions of the Drosophila melanogaster Slit and Notch proteins

```
          970       980       990      1000      1010      1020
SLIT_DROME FSCQCAPGYTGARCETNIDDCLGEIKCQNNATCIDGVESYKCECQPGFSGEFCDTKIQFC
           ..:.:  :.  :.: ...:.: ..  : :.. : ::.. . :.: ::..:. :.  :. :
NOTC_DROME YKCECPRGFYDAHCLSDVDECASN-PCVNEGRCEDGINEFICHCPPGYTGKRCELDIDEC
            740       750       760       770       780       790
```

Concept of a sequence alignment

• Pairwise Alignment:

! Explicit mapping between the residues of 2 sequences

– Tolerant to errors (mismatches, insertion / deletions or indels)

– Evaluation of the alignment in a biological concept (significance)

- Tolerant to errors (mismatches, insertion / deletions or indels)
- Evaluation of the alignment in a biological concept (significance)

Number of alignments

- There are many ways to align two sequences
- Consider the sequence fragments below: a simple alignment shows some conserved portions

```
 CGATGCAGACGTCA
        ||||||||
CGATGCAAGACGTCA
```

but also:

```
CGATGCAGACGTCA
||||||||
CGATGCAAGACGTCA
```

- Number of possible alignments for 2 sequences of length 1000 residues:
  - ⇨ more than 10^{600} gapped alignments
    (Avogadro 10^{24}, estimated number of atoms in the universe 10^{80})

What is a good alignment ?

- We need a way to evaluate the biological meaning of a given alignment
- Intuitively we "know" that the following alignment:

```
CGAGGCACAACGTCA
|||  |||   ||||||
CGATGCAAGACGTCA
```

is better than:

```
ATTGGACAGCAATCAGG
|       ||   |      |
ACGATGCAAGACGTCAG
```

- We can express this notion more rigorously, by using a scoring system

Simple alignment scores

- A simple way (but not the best) to score an alignment is to count 1 for each match and 0 for each mismatch.

```
CGAGGCACAACGTCA
|||  |||   ||||||
CGATGCAAGACGTCA
```

⇨Score: 12

```
ATTGGACAGCAATCAGG
|      ||  |      |
ACGATGCAAGACGTCAG
```

⇨Score: 5

Importance of the scoring system

!discrimination of significant biological alignments

• Based on physico-chemical properties of amino-acids

! Hydrophobicity, acid / base, sterical properties, ...

! Scoring system scales are arbitrary

• Based on biological sequence information

! Substitutions observed in structural or evolutionary alignments of well studied protein families

! Scoring systems have a probabilistic foundation

Substitution matrices PAM/ BLOSUM

• In proteins some mismatches are more acceptable than others

• Substitution matrices give a score for each substitution of one aminoacid by another

• Positive score: the amino acids are similar, mutations from one into the other occur more often then expected by chance during evolution

• Negative score: the amino acids are dissimilar, the mutation from one into the other occurs less often than expected by chance during evolution

log-odd ratio

$$\log\left(\frac{observed}{expected\ by\ chance}\right)$$

• For a set of well known proteins:

• Align the sequences

• Count the mutations at each position

• For each substitution set the score to the log-odd ratio

Different kind of matrices

- PAM series (Dayhoff M., 1968, 1972, 1978)

Percent **A**ccepted **M**utation.
A unit introduced by Dayhoff *et al.* to quantify the amount of evolutionary change in a protein sequence. 1.0 PAM unit, is the amount of evolution which will change, on average, 1% of amino acids in a protein sequence. A PAM(x) substitution matrix is a look-up table in which scores for each amino acid substitution have been calculated based on the frequency of that substitution in closely related proteins that have experienced a certain amount (x) of evolutionary divergence.

⇨ Based on 1572 protein sequences from 71 families
⇨ Old standard matrix: PAM250

Different kind of matrices

- BLOSUM series (Henikoff S. & Henikoff JG., PNAS, 1992)

Blocks **Su**bstitution **M**atrix.
A substitution matrix in which scores for each position are derived from *observations* of the frequencies of substitutions in blocks of local alignments in related proteins. Each matrix is tailored to a particular evolutionary distance. In the BLOSUM62 matrix, for example, the alignment from which scores were derived was created using sequences sharing no more than 62% identity. Sequences more identical than 62% are represented by a single sequence in the alignment so as to avoid over-weighting closely related family members.

⇨ Based on alignments in the BLOCKS database
⇨ Standard matrix: BLOSUM62

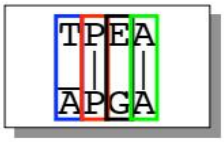
Limitations

• Substitution matrices do not take into account long range interactions between residues.
• They assume that identical residues are equal (whereas in reallife a residue at the active site has other evolutionary constraints than the same residue outside of the active site)
• They assume evolution rate to be constant.

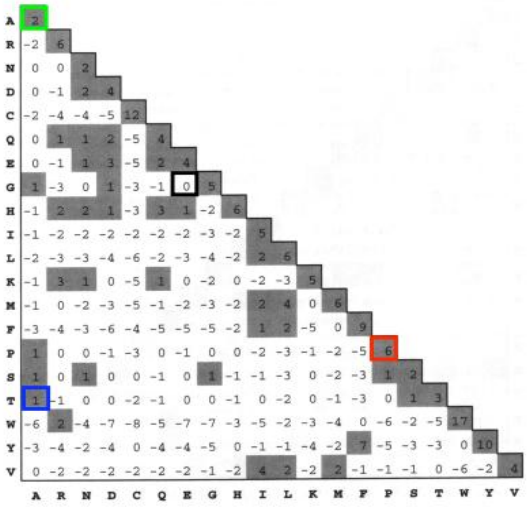
Amino acid substitution matrices

- Example: PAM250
- Most used: Blosum62

Raw score of an alignment



Score = 1 + 6 + 0 + 2 = 9



GAPS

Insertions or deletions

• Proteins often contain regions where residues have been inserted or deleted during evolution

• There are constraints on where these insertions and deletions can happen (between structural or functional elements like: alpha helices, active site, etc.)

Gaps in alignments

Gaps in alignments

```
GCATGCATGCAACTGCAT
|||||||||
GCATGCATGGGCAACTGCAT
```

can be improved by inserting a gap

```
GCATGCATG--CAACTGCAT
|||||||||  |||||||||
GCATGCATGGGCAACTGCAT
```

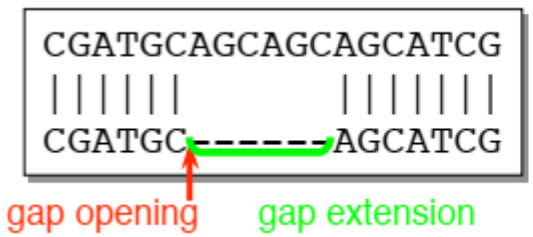
Gap opening and extension penalties

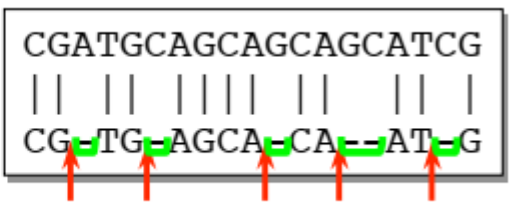
Costs of gaps in alignments

• We want to simulate as closely as possible the evolutionary mechanisms involved in gap occurence.

Example

• Two alignments with identical number of gaps but very different gap distribution. We may prefer one large gap to several small ones (e.g. poorly conserved loops between well-conserved helices)





Gap opening penalty

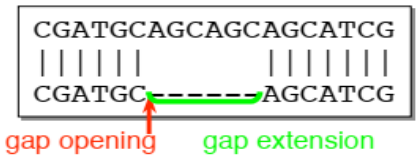
• Counted each time a gap is opened in an alignment (some programs include the first extension into this penalty)

Gap extension penalty

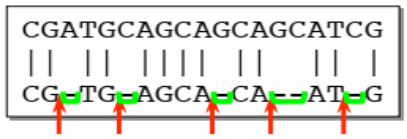
• Counted for each extension of a gap in an alignment

Example

- With a match score of 1 and a mismatch score of 0
- With an opening penalty of 10 and extension penalty of 1, we have the following score:



13 x 1 - 10 - 6 x 1 = -3



13 x 1 - 5 x 10 - 6 x 1 = -43

Alignments are evaluated according to their score
• Raw score
! It's the sum of the amino acid substitution scores and gap penalties (gap opening and gap extension)
! Depends on the scoring system (substitution matrix, etc.)
! Different alignments should not be compared based only on the raw score
• It is possible that a "bad" long alignment gets a better raw score than a very good short alignment.
! We need a normalised score to compare alignments !
! We need to evaluate the biological meaning of the score (p-value, e-value).
• Normalised score
! Is independent of the scoring system
! Allows the comparison of different alignments
! Units: expressed in bits

Statistics derived from the scores

- p-value
  - ⇨ Probability that an alignment with this score occurs by chance in a database of this size
  - ⇨ The closer the p-value is towards 0, the better the alignment

- e-value
  - ⇨ Number of matches with this score one can expect to find by chance in a database of this size
  - ⇨ The closer the e-value is towards 0, the better the alignment

- Relationship between e-value and p-value:
  - ⇨ In a database containing *N* sequences
    e = p x *N*

MULTIPLE SEQUENCE ALIGNMENT

A multiple sequence alignment is a collection of three or more protein (or nucleic acid) sequences that are partially or completely aligned. Homologous residues are aligned in columns across the length of the sequences. These aligned residues are homologous in an evolutionary sense: they are presumably derived from a common ancestor. The residues in each column are also presumed to be homologous in a structural sense: aligned residues tend to occupy corresponding positions in the three-dimensional structure of each aligned protein.

Aligned columns of amino acid residues characterize a multiple sequence alignment. This alignment may be determined because of features of the amino acids such as the following:

• There are highly conserved residues such as cysteine that are involved in forming disulfide bridges.

• There are conserved motifs such as a transmembrane domain or an immunoglobulin domain. We will encounter examples of protein domains and motifs (such as the PROSITE dictionary).

• There are conserved features of the secondary structure of the proteins, such as residues that contribute to ex helices, f3 sheets, or transitional domains.

• There are regions that show consistent patterns of insertions or deletions.

Typical Uses and Practical Strategies of Multiple Sequence Alignment When and why are multiple sequence alignments used?

• If a protein (or gene) you are studying is related to a larger group of proteins, this group membership can often provide insight into the likely function, structure, and evolution of that protein.

• Most protein families have distantly related members. Multiple sequence alignment is a far more sensitive method than pairwise alignment to detect homologs. Profiles (such as those described for PSIBLAST and hidden Markov models in Chapter 5) depend on accurate multiple sequence alignments.

• When one examines the output of any database search (such as a BLAST search), a multiple sequence alignment format can be extremely useful to reveal conserved residues or motifs in the output.

• If one is studying cDNA clones, it is common practice to sequence them. Multiple sequence alignment can show whether there are any variants or discrepancies in the sequences. Alignments of genomic DNA containing single nucleotide polymorphisms (SNP s are of interest, for example, in the identification of non-synonymous SNPs.

• Analysis of population data can provide insight into many biological questions involving evolution, structure, and function. The PopSet portion of Entrez (described below) contains nucleotide (and protein) population data sets that are viewed as multiple alignments.

• When the complete genome of any organism is sequenced, a major portion of the analysis consists of defining the protein families to which all the gene products belong. Database searches effectively perform multiple sequence alignments, comparing each novel protein (or gene) to the families of all other known genes.

There are many approaches to multiple sequence alignment; in the past decade many dozens of programs have been introduced. We may consider five algorithmic approaches:

(1) exact methods,

(2) progressive alignment (e.g., ClustalW),

(3) iterative approaches (e.g., PRALINE, IterAlign, MUSCLE),

(4) consistency-based methods (e.g., MAFFT, ProbCons), and

(5) structure-based methods that include information about one or more known three-dimensional protein structures to facilitate creation of a multiple sequence alignment (e.g., Expresso).

The programs we will describe in categories (3) to (5) are often overlapping; for example, all rely on progressive alignment and some combine iterative and structure-based approaches.

All the programs offer trade-offs in speed and accuracy. MUSCLE and MAFFTare fastest, and are thus most useful for aligning large numbers of sequences. ProbCons and T-Coffee, although slower, are more accurate in many applications.

Most programs produce reasonably consistent alignments, especially for relatively closely related protein or DNA sequences. Comparative studies of m ultiple sequence alignment algorithms have been performed based on tests against benchmark databases.

Some of the general conclusions include the following.

• Adding more homologs to a multiple sequence alignment improves its accuracy.

• As the group of sequences being multiply aligned begins to share less am ino acid identity, the accuracy of the alignments decreases. For groups of sequences that share less than 25 % identity, the problem becomes especially severe. Thompson et al. (1999) found that the best programs available at the time (P RRP, ClustalX, and SAGA) aligned about 60 % to 70 % of the amino acid residues for groups of proteins with < 25% identity, For multiple sequence alignments ofproteins sharing more identity (20% up to 40%), they found that on average 80 % of the residues were aligned properly.

• For highly divergent DNA sequences, programs that use local alignment (such as DiAlign and LAGAN) perform better than those using global alignment (such as ClustalW).

• Orphan sequences are proteins that are highly divergent members of a family. If we examined a multiple sequence alignment of retinol-binding protein (REP) from 10 species,

then added the distantly related odorant-binding protein (OBP) to that multiple sequence alignment, OBP would be considered an orphan. Orphans might be expected to disrupt the organization of a multiple sequence alignment, and yet they do not. Global alignmentalgorithms outperform local alignment methods for the introduction of orphans to an alignment.

• Separate multiple sequence alignments can be combined, such as a group of closely related myoglobins and a group of closely related neuroglobins. Iterative algorithms performed this task better than progressive alignment methods. However, many programs have difficulty in accurately producing a single alignment from a subset of alignments.

• Often, some proteins in a family contain large extensions at the amino- and/ or carboxy-terminals. Overall, local alignment programs dramatically outperformed global alignment programs at this task. For most multiple sequence alignment applications, global alignments are superior.

Progressive alignment Method with Clustal W2

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.

Aligning multiple sequences highlights areas of similarity which may be associated with specific features that have been more highly conserved than other regions. These regions in turn can help classify sequences or to inform experiment design.

Multiple sequence alignment is also an important step for phylogenetic analysis, which aims to model the substitutions that have occured over evolution and derive the evolutionary relationships between sequences.

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 be in GCG, FASTA, EMBL, PIR, NBRF or UniProtKB/Swiss-Prot format. Partially formatted sequences are not accepted. Adding a return to the end of the sequence may help certain applications understand the input. Note that directly using data from word processors may yield unpredictable results as hidden/control characters may be present. 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. Word processor files may yield unpredictable results as hidden/control characters may be present in the files. It is best to save files with the Unix format option to avoid hidden Windows characters. 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)	Description	Abbreviation
BLOSUM		blosum
PAM		pam
Gonnet		gonnet
ID		id

Default value is: Gonnet [gonnet]

DNA Weight Matrix (PW)

Slow pairwise alignment nucleotide sequence comparison matrix used to score alignment.

Matrix (Protein Only)	Description	Abbreviation
IUB		iub
ClustalW		clustalw

Default value is: IUB [iub]

Gap Open (PW)

Slow pairwise alignment score for the first residue in a gap.

Default value is: 10

Gap Extension (PW)

Slow pairwise alignment score for each additional residue in a gap.

Default value is: 0.1

KTUP

Fast pairwise alignment word size used to find matches between the sequences. Decrease for sensitivity; increase for speed.

Default value is: 1

Window Length

Fast pairwise alignment window size for joining word matches. Decrease for speed; increase for sensitivity.

Default value is: 5

Score Type

Fast pairwise alignment score type to output.

Order	Description	Abbreviation
percent		percent
absolute		absolute

Default value is: percent

Top Diags

Fast pairwise alignment number of match regions are used to create the pairwise alignment. Decrease for speed; increase for sensitivity.

Default value is: 5

Pair Gap

Fast pairwise alignment gap penalty for each gap created.

Default value is: 3

Step 3 - Multiple Sequence Alignment Options

Protein Weight Matrix

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

Matrix (Protein Only)	Description	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)	Description	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

No End Gaps

Multiple alignment disable the gap seperation penalty when scoring gaps the the ends of the alignment

Order	Description	Abbreviation
no		false
yes		true

Default value is: no [false]

Iteration

Multiple alignment improvement iteration type

Order	Description	Abbreviation
none	No iteration	none
tree	Iteration at each step of alignment process	tree
alignment	Iteration only on final alignment	alignment

Default value is: none

Num Iter

Maximum number of iterations to perform

Default value is: 1

Clustering

Clustering type.

Order	Description	Abbreviation
NJ	Neighbour-joining (Saitou and Nei 1987)	NJ
UPGMA	UPGMA clustering	UPGMA

Default value is: NJ

Output

Format for generated multiple sequence alignment.

Order	Description	Abbreviation
Aln w/numbers	ClustalW alignment format with base/residue numbering	aln1
Aln wo/numbers	ClustalW alignment format without base/residue numbering	aln2
GCG MSF	GCG Multiple Sequence File (MSF) alignment format	gcg

Order	Description	Abbreviation
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: Aln w/numbers [aln1]

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 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 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).

PROTEOMICS

PROTEIN SEQUENCE DATABASES (SWISS PROT AND PDB)

Proteomics is the large-scale study of proteomes. A proteome is a set of proteins produced in an organism, system, or biological context. We may refer to, for instance, the proteome of a species (for example, Homo sapiens) or an organ (for example, the liver). The proteome is not constant; it differs from cell to cell and changes over time. To some degree, the proteome reflects the underlying transcriptome. However, protein activity (often assessed by the reaction rate of the processes in which the protein is involved) is also modulated by many factors in addition to the expression level of the relevant gene.. Proteomics has enabled the identification of ever increasing numbers of protein.

Proteomics is used to investigate:

i. when and where proteins are expressed
ii. rates of protein production, degradation, and steady-state abundance
iii. how proteins are modified (for example, post-translational modifications (PTMs) such as phosphorylation)
iv. the movement of proteins between subcellular compartments
v. the involvement of proteins in metabolic pathways
vi. how proteins interact with one another
vii. Proteomics can provide significant biological information for many biological problems, such as which proteins interact with a particular protein of interest (for example, the tumour suppressor protein p53)? (Human example)
viii. which proteins are localised to a subcellular compartment (for example, the mitochondrion)? (Human example)

ix. which proteins are involved in a biological process (for example, circadian rhythm)? (Human example)

SWISS-PROT

SWISS-PROT is a curated protein sequence database. It is created at the Department of Medical Biochemistry of the University of Geneva and has been a collaborative effort of the Department and the European Molecular Biology Laboratory (EMBL), since 1987. SWISS-PROT is now an equal partnership between the EMBL and the Swiss Institute of Bioinformatics (SIB). The EMBL activities are carried out by its Hinxton Outstation, the European Bioinformatics Institute (EBI). It strives to provide a high level of annotation (such as the description of the function of a protein, its domains structure, post-translational modifications, variants, etc.), a minimal level of redundancy and high level of integration with other databases.

SWISS-PROT is a collection of confirmed protein sequences with annotations relating to structure, function, and protein family assignments. The related database TrEMBL is a translation of all coding sequences in the primary nucleic acid databases. The entries in TrEMBl are annotated less extensively than those in SWlSS-PROT, but are moved to SWISS-

PROT when reliable annotations become available. Both TrEMBl and Swiss-Prot have been incorporated into the UniProt (t;piversai Protein Resource), which also incorporates the PIR database

Recent developments of the database include format and content enhancements, cross-references to additional databases, new documentation files and improvements to TrEMBL, a computer-annotated supplement to SWISS-PROT. TrEMBL consists of entries in SWISS-PROT-like format derived from the translation of all coding sequences (CDSs) in the EMBL Nucleotide Sequence Database, except the CDSs already included in SWISS-PROT.

Its mission is to provide the scientific community with a comprehensive, high-quality and freely accessible resource of protein sequence and knowledge. The database sums up all the available information on a given protein in a given species, such as its function, cell location, pathological role, interactions and differences with other species. All this information has been curated beforehand, i.e. sorted, structured, reviewed and validated by expert people (known as biocurators). UniProtKB/Swiss-Prot is one of the most widely used databases and is recognized today as a gold standard reference and quality gatekeeper by the international scientific community. It is used by scientists from all fields of the life sciences as well as by clinicians. It also serves as a reference for patent organizations in the field of proteins and biotherapeutics, and is used by computer scientists to benchmark their prediction programs. The database also serves as a foundation for other databases intended for more specialized audiences; an example would be the SIB NeXtProt database recently developed for clinicians.

PROTEIN DATA BANK (PDB)

Through an internet information portal and downloadable data archive, the PDB provides access to 3D structure data for large biological molecules (proteins, DNA, and RNA). These are the molecules of life, found in all organisms on the planet. Knowing the 3D structure of a biological macromolecule is essential for understanding its role in human and animal health and disease, its function in plants and food and energy production, and its importance to other topics related to global prosperity and sustainability. RCSB PDB (Research Collaboratory for Structural Bioinformatics PDB) operates the US data center for the global PDB archive, and makes PDB data available at no charge to all data consumers without limitations on usage (Policies). The Vision of the RCSB PDB is to enable open access to the accumulating knowledge of 3D structure, function, and evolution of biological macromolecules, expanding the frontiers of fundamental biology, biomedicine, and biotechnology.

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 was established in 1971 at Brookhaven National Laboratory under the leadership of

Walter Hamilton and originally contained structures. However, as of now a total of 149174 structures, of which 138369 proteins, 3309 nucleic acids, 7464 NA- Protein complexes and 32 other are available.

The PDB Archive

- Grows at the rate of nearly 10% per year
- Used to download >2 million structure data files per day
- Managed by International collaboration US-Asia-Europe
- Manages “Big Data” as global Public Good

PDB Data

- Enable research in subject areas from Agriculture to Zoology (Analysis)
- Contributed data to nearly >1 million published research papers
- Used by >400 biological data resources

PDB Data Impact

- Basic and applied research
- Patent applications
- Discovery of lifesaving drugs
- Innovations that can lead to new product development and company formation
- STEAM education: PDB-101 provides curricula and online tools for teachers and students

PROTEIN STRUCTURE VISUALIZATION USING RASMOL

RasMol is an important scientific tool for visualisation of molecules created by Roger Sayle in 1992. RasMol is used by hundreds of thousands of users world-wide to view macromolecules and to prepare publication-quality images.

RasMol Features

RasMol is a molecular graphics program intended for the visualisation of proteins, nucleic acids and small molecules. The program is aimed at display, teaching and generation of publication quality images. RasMol runs on wide range of architectures and operating systems including Microsoft Windows, Apple Macintosh, UNIX and VMS systems.

The program reads in a molecule coordinate file and interactively displays the molecule on the screen in a variety of colour schemes and molecule representations. Currently available representations include depth-cued wireframes, 'Dreiding' sticks, spacefilling (CPK) spheres, ball and stick, solid and strand biomolecular ribbons, atom labels and dot surfaces.

The program reads in molecular coordinate files and interactively displays the molecule on the screen in a variety of representations and colour schemes. Supported input file formats include Protein Data Bank (PDB), Tripos Associates' Alchemy and Sybyl Mol2 formats, Molecular Design Limited's (MDL) Mol file format, Minnesota Supercomputer Center's (MSC) XYZ (XMol) format, CHARMm format, CIF format and mmCIF format files. If connectivity information is not contained in the file this is calculated automatically.

The loaded molecule can be shown as wireframe bonds, cylinder 'Dreiding' stick bonds, alpha-carbon trace, space-filling (CPK) spheres, macromolecular ribbons (either

smooth shaded solid ribbons or parallel strands), hydrogen bonding and dot surface representations. Atoms may also be labelled with arbitrary text strings. Alternate conformers and multiple NMR models may be specially coloured and identified in atom labels.

Different parts of the molecule may be represented and coloured independently of the rest of the molecule or displayed in several representations simultaneously. The displayed molecule may be rotated, translated, zoomed and z-clipped (slabbed) interactively using either the mouse, the scroll bars, the command line or an attached dial box. RasMol can read a prepared list of commands from a 'script' file (or via inter-process communication) to allow a given image or viewpoint to be restored quickly. RasMol can also create a script file containing the commands required to regenerate the current image. Finally, the rendered image may be written out in a variety of formats including either raster or vector PostScript, GIF, PPM, BMP, PICT, Sun rasterfile or as a MolScript input script or Kinemage.

The RasMol help facility can be accessed by typing "help <topic>" or "help <topic> <subtopic>" from the command line. A complete list of RasMol commands may be displayed by typing "help commands". A single question mark may also be used to abbreviate the keyword "help". Please type "help notices" for important notices.

Running RasMol Under Microsoft Windows

To start RasMol under Microsoft Windows, double click on the RasMol icon in the program manager. When RasMol first starts, the program displays a single main window (the display window) with a black background on the screen and provides the command line window minimized as a small icon at the bottom of the screen. The command line or terminal window may be opened by double clicking on this RasMol icon.

It is possible to specify either a coordinate filename or a script filename or both on the windows command line. A script file may be specified by adding the option '-script <filename>' to the command line. A molecule coordinate file may be specified by placing its name on the command line, optionally preceded by a file format option. If no format option is given, the specified coordinate file is assumed to be in PDB, CIF or mmCIF format. If both a coordinate file and a script file are specified on the command line, the molecule is loaded first, then the script commands are applied to it. If either file is not found, the program displays the error message 'Error: File not found!' and the user is presented the RasMol prompt.

It is also possible to specify the initial graphics window size or position or both the size and the position with the options '-height nnnn', '-width nnnn', '-xpos nnnn' and '-ypos nnnn'. The numeric values are in pixels. The position is specified in terms of the top left corner of the rendering area.

RasMol's Window

On all platforms RasMol displays two windows, the main **graphics or canvas window** with a black background and **a command line or terminal window**. At the top of the graphics window (or at the top of the screen for the Macintosh) is the RasMol menu bar. The contents of the menu bar change from platform to platform to support the local user interface guidelines; however, all platforms support the 'File', 'Display', 'Colours', 'Export', 'Options' and

'Settings' pull-down menus. The Main graphics window also has two scroll bars, one on the right and one at the bottom, that may be used to rotate the molecule interactively.

While the mouse pointer is located within the graphics area of the main display window, the mouse pointer is drawn as a cross-hair cursor, to enable the 'picking' of objects being displayed; otherwise the mouse pointer is drawn as an arrowhead. Any characters that are typed at the keyboard while the display window is in 'focus' (meaning active or foreground) are redirected to the command line in the terminal window. Hence you do not need continually to switch focus between the command line and graphics windows.

The display window may be resized at any point during the session. This has the effect of simply rescaling the image displayed on the canvas. RasMol imposes limits on the size of the display window such that the window must be large enough to display the menu and scroll bars and yet small enough to fit on a single screen. Attempts to enlarge the screen may fail owing to insufficient memory on the host machine, in which case RasMol reports the error message 'Renderer Error: Unable to allocate frame buffer!' or some similar error.

Mouse Controls

Here is a summary of RasMol's mouse click-and-drag controls. The 'set mouse' command mode defaults to 'set mouse rasmol', which gives the controls summarized below.

Action	Windows
Rotate X, Y	Left
Translate X, Y	Right
Rotate Z	Shift-Right
Zoom	Shift-Left
Slab Plane	Ctrl-Left

Command Line Interface

RasMol allows the execution of interactive commands typed at the RasMol prompt in the terminal window. Characters typed into either the terminal or the display window are processed on the command line. Each command must be given on a separate line terminated by a newline or carriage return character.. If a command is not recognised by RasMol, the program will generate an 'Unrecognised command!' error and redisplay the main prompt. If surplus information is given at the end of a command line, RasMol will execute the recognised command, but issue the warning message 'Warning: Ignoring rest of command!'. Some commands may prompt the user for more information. These commands display a different prompt and are discussed in the command reference.

PDB: Protein Structure Visualisation Using RasMol

Program

RasMol is a molecular graphics program intended for the visualization of proteins, nucleic acids and small molecules. The program is aimed at display, teaching and generation of publication quality images.
Rasmol is installed on your own computer if you have one.

Objective

After completing this laboratory, you should know how to:
Search the protein databank for protein structures
Download the primary structure (text format)
Manipulate proteins using molecular visualization software
Identify amino acids in an active site

Protein Visualization

The first step is to find the US hosted version of the PDB
Opening a PDB file and identifying the primary structure of a protein

1. One of the proteins we will be investigating is cytochrome c.
Using advanced search select keyword and search for "oxidized C2 cytochrome" in the databank (don't include quote marks).
2. Find PDB identifier 2C2C
Download the file for cytochrome c (oxidized form)
There are 4 icons under the PDB identifier

Click on the icon that looks like text
This is a preview of the data file

The information should look like this: (click here) It is of value to do this to make sure it is the file that you want to down load.

3. Now download the entire PDB file for viewing.
Click on the Icon with an arrow. The file can Either be saved to disk or opened directly
*You may also choose to look at the molecule using a PDB viewer (such as CHIME) by selecting [PDB viewer] or RasMol by selecting [Motifs-RasMol].
You will open a saved PDB file manually in RasMol.

Viewing the Molecule in RasMol

5. Open RasMol by clicking on the icon Raswin.exe (or RasMol).
If you prefer to launch RasMol from the Molecules R US website select: **Output requested** [*Motifs-RasMol]*
For the first time use of RasMol on a computer, Netscape (or Explorer) must know where to find the application. Select browse and find the folder containing *Raswin.exe* and click on it. The molecule should now be opened in RasMol. Future use will not require these steps.
RasMol contains two parts: a viewing window and a command line.
Viewing window: displays the molecule
Command line: enables the user to change the view by typing commands

6. Open the molecule by clicking **File: Open** and selecting the PDB file saved in your network account.
A wireframe molecule will appear in the viewing window.

You will need to view the molecule and command line for this exercise.
Expand the viewing window so that it fills the opper ¾ of the screen.
Expand the command window so that it fills the remaining ¼ of the screen.

Sometimes when Rasmol opens the command line is minimized on the lower task bar. If you can't find the command line window look along the bottom of your screen and click on the Rasmol type there.

7. Manipulate the molecule using the mouse:
Translation - Right mouse button
Rotation - Left mouse button
Rotation in a plane - Shift - Right mouse button
Zoom Shift - left mouse button

8. Locate residues (amino acids) and particular atoms by using the mouse to pick or select atoms (i.e., click on an atom and the result is shown in the command window). The command window tells you which atoms are being selected and to what amino acid they belong. If you can't see the amino acid selected zoom in using the mouse. If you don't have the primary sequence in front of you type **show sequence** in the command line to see the sequence in the command line.

Note: many of the following tasks can be accomplished by typing the correct command into the command line or accessing the task descriptor from the pull down menus. You may do either to accomplish the task. Many commands, however, are only available through the command line. The command line allows greater flexibility and detail control.

9. Change the viewing type by selecting **Display** from the viewing window menu bar.
If only certain residues are changing, then type **select all (or use pull down menu).**
Practice changing display types for the protein this can be accomplished using the pull down menu.

10. Display particular amino acids:
Type **select all** and choose **Display** - *wireframe*
Type **select ala** to look at all of the alanines in the protein (75 atoms should be selected)
Type **color red** (all alalines should be red)
To view alaline as ball and stick select **Display** - *Ball and Stick.*
You can select any amino acid (again, type show sequence if you need to know what amino acids are present in the molecule). Practice looking for the loctions of various amino acids by selecting them one at a time and changing their color and Display style.
Typical colors: Red, orange, yellow, green, blue, violet, purple, brown, gold, cyan, black, white, grey, greentint, greenblue,hotpink, magenta, pink, pinktint, redorange, skyblue, bluetint, yellowtint.
Other colors are generated by specifying the **R**ed**G**reen**B**lue triplet values as in **color[255,0,0]** for red; **color [0,255,0]** for green; and **color [0,0,255]** for blue and all values in between.
Change the molecule's color to purple by typing **select all** and then **color [255,0,255].** When you are finished, type **select all,** and reset the **display** to *wireframe* and color cpk**.**

11. Display particular atoms:
Type **select all** and then **wireframe** (this is an alternative way Of changing the display)
Cytochrome c contains an iron metal center - Type select iron to identify the iron in the protein.
Type **color orange** (iron should be orange).
View it as a ball and stick model.
The iron is in a porphyrin ring. Identify the porphyrin ring by typing **select ligand.**
Change its color to red and display it as a ball and stick .
Type **select iron** and change its color to *yellow.*
Type **select backbone** to select the protein backbone.

Change the display to ribbon by typing **ribbon.**
Change the backbone color to grey.
Select the sidechains by typing **select sidechain.**
Change the color to *greentint.*
Change the background color to white by typing **background white.**
Practice changing styles and colors and rotate and move the

Copy and paste your final picture into Word or Wordperfect to be included in your report. If possible print in color; if no indicate what colors were present on the screen in pencil on your printout.

12. You can select individual amino acids by typing **select #** (i.e., **select 12** will select amino acid number 12; **select 1-12** will select residues 1-12).
This is useful for identifying domains (structurally independent units in a protein) or to highlight regions of interest in a protein sequence (i.e., beta sheets, helices, turns).

PROTEIN STRUCTURE PREDICTION

HOMOLOGY MODELING OF PROTEIN USING SWISS-MODEL(Swiss pdb Viewer)

Homology modeling aims to build three-dimensional protein structure models using experimentally determined structures of related family members as templates. SWISS-MODEL workspace is an integrated Web-based modeling expert system. For a given target protein, a library of experimental protein structures is searched to identify suitable templates. On the basis of a sequence alignment between the target protein and the template structure, a three-dimensional model for the target protein is generated. Model quality assessment tools are used to estimate the reliability of the resulting models. Homology modeling is currently the most accurate computational method to generate reliable structural models and is routinely used in many biological applications. Typically, the computational effort for a modeling project is less than 2 h. However, this does not include the time required for visualization and interpretation of the model, which may vary depending on personal experience working with protein structures.

Homology modeling with swiss-model

The SWISS-MODEL is a simple and popular homology-modelling program and one of only few which available on the Internet. It uses the "building by fragments" method to construct the model on the template structures.

1. Open Swiss-Model at http://swissmodel.expasy.org/.
2. Link to **First Approach mode** (the upper link on the left frame).
3. The first data that we should supply (apart of the personal details) is the primary sequence of the protein we wish to model. In this exercise we will model the structure of a cyclic AMP dependent kinase (PKA). In order to get the primary sequence of this kinase you can enter to the ***Swissprot*** site (http://www.expasy.ch/sprot/), to type the accession number (P05132) and retrieve the entry. Save the sequence into local file in Fasta format. This new file should be later opened as a Fasta format file, meaning that it should begin with description line starting

with the character ">". Call this file as **pka.tfa**. Copy and paste this sequence also to the relevant window at the SWISS-MODEL form (without the description line). This is actually all you need to do in order to run SWISS-MODEL in simple first approach mode. However, we **will not run** the program in this mode, instead we will run in **First approach mode with a specific template**. We will supply a specific kinase structures that will serve as templates during the building. These will be the structures of two tyrosine protein kinases (PDB ID 1iep, chain A and 1k2p, chain A). Under "**Use a specific template**" insert: 1iepA. Send the request.

4**.** We will now learn how to run the program in optimized mode. **Open Swiss-PDB-viewer.**

5. Choose "**Load Raw Sequence to Model**" item of the "Swiss Model" menu to load the file **pka.tfa** that you previously created.

6. Choose the "Swiss-Model" item of the "Preferences" menu. Enter your name and e-mail address. Make sure that the address of the modeling server is: **http://swissmodel.expasy.org/cgi-bin/sm-submit-request.cgi** and that the address of the template server is: **http://swissmodel.expasy.org/cgi-bin/blastexpdb.cgi**

7. **Now we will choose and supply the template structure**. Get the PDB file **1iep** and save it locally on your computer. The program has also an option to choose the template files for you. We will not use this option now. Open the file by Swiss PDB viewer.

8. Choose "**Alignment**" from the "Window" menu.

9. Click on the **pka** name to make this layer active. Choose the "**Magic Fit**" option of the "**Fit**" menu. This will perform the sequence alignment. Choosing "**Improved Fit**" from the same menu will optimize the alignment.

10. Make sure all residues of the 2 proteins are selected. From the "Color" menu choose "color by alignment diversity", so you will be able to identify the conserved regions. 11. Choose the "**Update threading now**" item of the "SwissModel" menu (this item is not accessible if the "Update Threading Display automatically" item is enabled; which is the case by default).

12. After the initial automatic alignment we have now the freedom to change it. This is done with the mouse and the arrow keys. We can also make use of the mean force potential to help threading correctly a protein, although this tool should be used with caution. Make sure the current layer is pka, and click on the little arrow located at the right of the question mark of the Alignment Window. The window expands, and displays a curve depicting how each residue likes it's surrounding. If a residue is "happy", its energy is below zero, whereas unhappy residues will have energy above the zero axis. This is the mean force potential energy. Click on the "smooth" text, and set a smoothing factor of 1. It means that the energy of each residue will be the average of itself plus the energy of 1 flanking residue on each side. You can enable the "Auto Color by Threading Energy" item of the "SwissModel" menu to better see the potential on the structure. Click on the "E= XX" text, this will re-compute the energy for the current layer. **Note:** this tool provides hints and should be used in conjunction with other type of analyses! It works better for displacement of large fragments.

13. You can also evaluate how good your threading is by using the "aa making clashes" items of the "Select" menu. This will allow you to quickly focus on potentially problematic regions. You can then choose the "Fix Selected Side-chains" ("crude") item of the "Tools" menu, which will browse the rotamer library to choose the best rotamer, exactly as during a mutation process. By repeating the "Select aa making clashes" process, you should see that fewer

amino-acids are making problems. If not, this is probably a good clue that your threading is incorrect. **Important Note:** Actually fixing the side-chains is just for you, to evaluate prior to submitting the request, it will have absolutely no incidence onto the model building, as side-chains are reconstructed anyway.

14. When everything seems OK, you can submit a modeling request to Swiss-Model simply by choosing the "Submit modeling request" of the "SwissModel" menu. You will be asked to give a project name. By the default, you will get to your email a Swiss-PDB-Viewer project file, with your model aligned onto the templates you used, and ready for comparison.

15. While the server is working, we will compare the results of the first approach mode with the real structure for that protein. **Open the structural alignment program CE** (combinatorial extension) at the URL:

http://cl.sdsc.edu/ce/ce_align.html. For the first chain upload the model you obtained from Swiss-Model. Don't forget to mark the "User File" option instead of the "PDB" option. For the second chain, enter 1APM:E which are the PDB code and chain identifier for the real structure exists for PKA. Submit.

16. At the results page, look at the alignment. Notice that this sequence alignment was produced according to the structural information, without considering the sequence. It allows us to judge the structural similarity. Find the regions which were not properly built. What is the overall RMSD of the structural alignment? Is this significant?

17. Save the PDB format of the structural alignment in your computer and open it with **RasMol**. Find the regions not properly aligned by visual inspection and compare to your answer from the previous question.

18. Finally we will take a look at the at the **evaluation** report for this model. Look at the evaluation graphs obtained using Anolea, Gromos and Verify3d**.** Try to find regions which are suspected to be incorrect.

19. To conclude: structural model is basically easy to obtain, but we always be aware of how it was produced, check it with available tools and refine it if necessary.