Bioinformatics

UNIT IV

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.

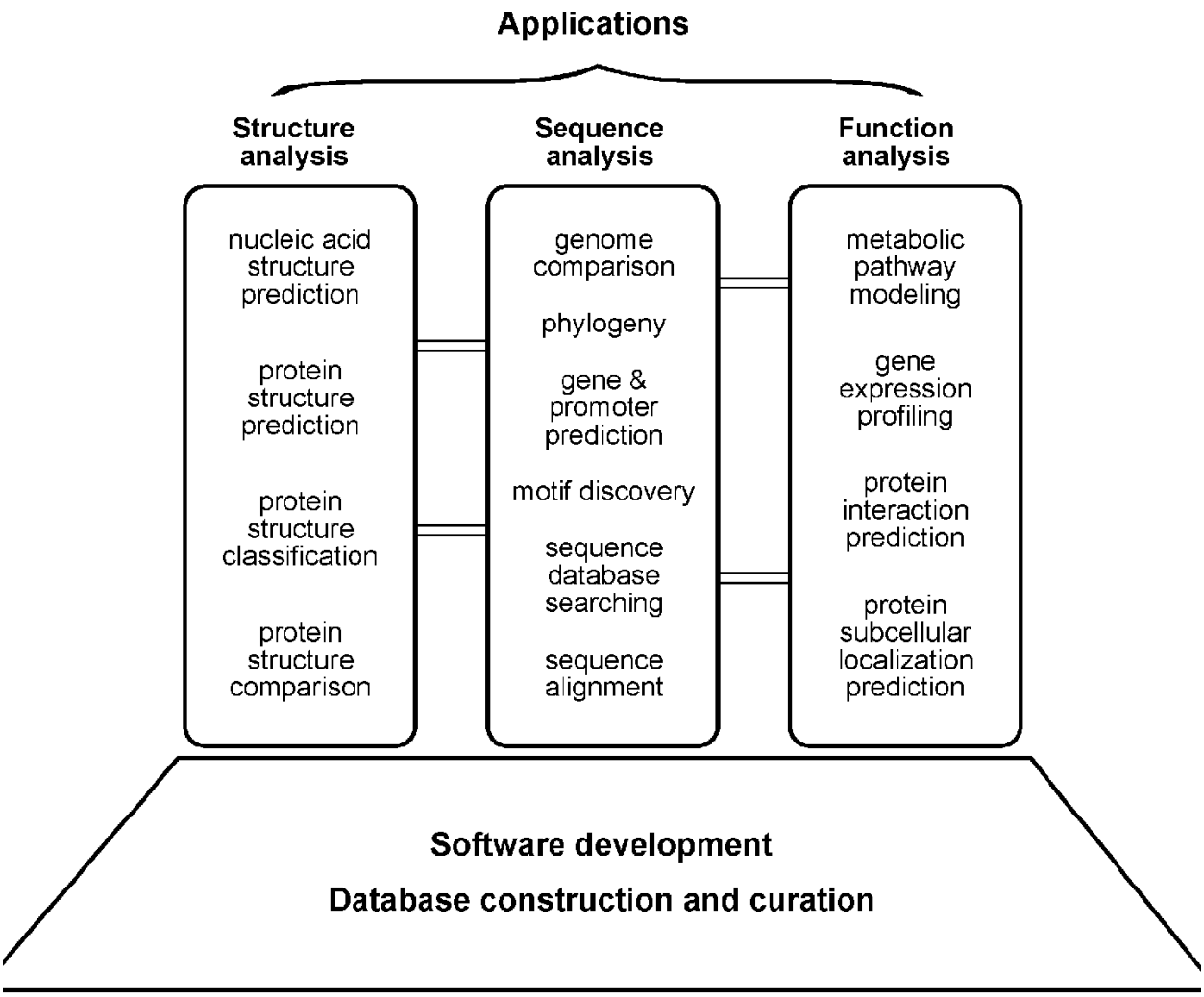
Bioinformatics differs from a related field known as computational biology. Bioinformatics is limited to sequence, structural, and functional analysis of genes and genomes and their corresponding products and is often considered computational molecular biology. However, computational biology encompasses 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

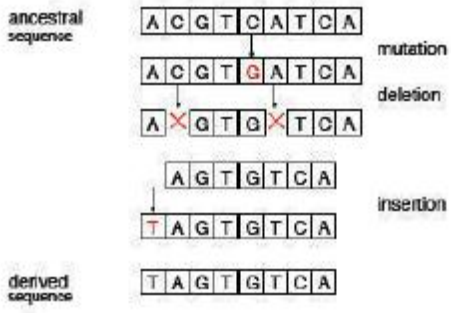
SEQUENCE ALIGNMENT ANALYSIS

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---

------ l l l l -----

-------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.

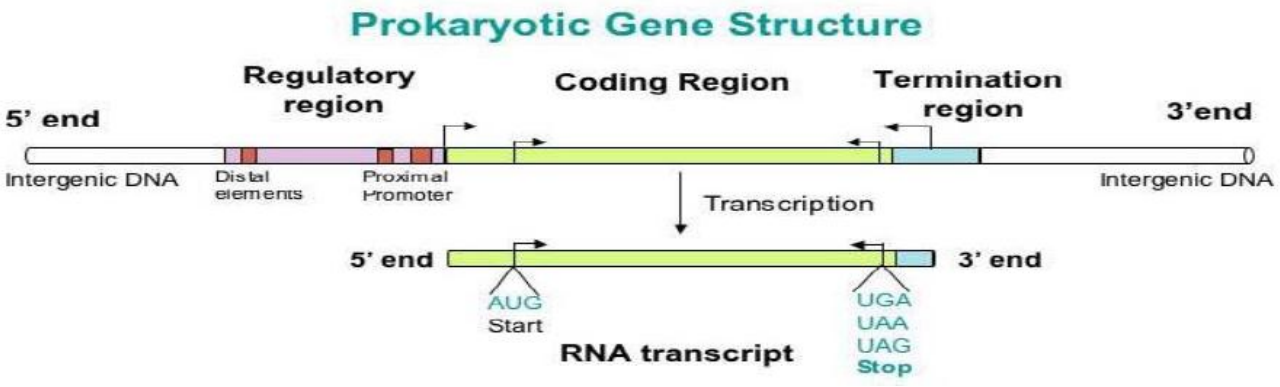
The E-value gives an indication of the statistical significance of a given pairwise alignment and reflects the size of the database and the scoring system used. The lower the E-value, the more significant the hit. A sequence alignment that has an E-value of 0.05 means that this similarity has a 5 in 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.

Bioinformatics Approaches to Gene Prediction

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.

ClustalW2

Introduction

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

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

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

Matrix (Protein Only)	Description	Abbreviation
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

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

Matrix (Protein Only)	Description	Abbreviation
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
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).

BLAST

BLAST, the Basic Local Alignment Search Tool, is perhaps the most widely used bioinformatics tool ever written. It is an alignment that determines "local alignments" between a query and a database. It uses an approximation of the Smith-Waterman algorithm.

BLAST consists of two components: a search algorithm and computation of the statistical significance of solutions.

- Basic Local Alignment Search Tool (BLAST) was developed as a new way to perform seq. similarity search. BLAST is faster than FASTA while being nearly as sensitive.
- The minimal "word" (*k*-tuple) length is slightly higher than in FASTA, 3 for proteins and 11 for DNA.
- The steps used by the BLAST algorithm:
- The seq is optionally filtered to remove low-complexity regions (AGAGAG…)
- A list of words of certain length is made
- Using substitution scores matrixes (like PAM or BLOSUM62) the query seq. words are evaluated for matches with any DB seq. and these scores (log) are added
- A cutoff score (*T*) is selected to reduce number of matches to the most significant ones
- The above procedure is repeated for each word in the query seq.
- The remaining high-scoring words are organised into efficient search tree and rapidly compared to the DB seq.
- If a good match is found then an alignment is extended from the match area in both directions as far as the score continue to grow. In the latest version of BLAST more time-efficient method is used
- The next step is to determine those high scoring pairs (HSP) of seq., which have score greater than a cutoff score (*S*). S is determined empirically by examining a range of scores found by comparing random seq. and by choosing a value that is significantly greater.
- Then BLAST determines statistical significance of each HSP score. The probability *p* of observing a score *S* equal to or greater than *x* is given by the equation: $p\ (S \geq x) = 1 - \exp(-e^{-\lambda(x-u)})$, where u = [log (Km'n')]/λ and K and λ are parameters that are calculated by BLAST for amino acid or nucleotide substitution scoring matrix, n' is effective length of the query seq. and m' is effective length of the database seq.
- On the next step a statistical assessments is made in the case if two or more HSP regions are found and certain matching pairs are put in descending order in the output file as far as their similarity/ score is concerned.

There are a number of different variants of the BLAST program:

• BLASTN: compares a DNA query sequence to a DNA sequence database;
• BLASTP: compares a protein query sequence to a protein sequence database;
• TBLASTN: compares a protein query sequence to a DNA sequence database
• BLASTX: compares a DNA query sequence (6 frames translation) to a protein sequence database
• TBLASTX: compares a DNA query sequence (6 frames translation) to a DNA sequence database (6 frames translation)