Lecture 4: January 1, 1999

Lecturer: Irit Or *Scribe: Irit Gat and Tal Kohen*

4.1 Biological Databases and Retrieval Systems

In recent years, biological databases have greatly developed a lot, and became a part of the biologist's everyday toolbox [see eg. [7]]. There are several reasons to search databases :

1. When obtaining a new DNA sequence, one needs to know whether it has already been deposited in the databanks, or whether they contain any *homologous sequences* (sequences which are derived from a common ancestry) exist there.

2. Given a putative coding ORF, we can search for *Homologous proteins* - proteins similar in their folding or structure of function).

3. To find similar non-coding DNA stretches in the database : Repeat elements or regulatory sequences for instance.

4. There are other uses for specific purpose, like locating false priming sites for a set of PCR oligonucleotides.

4.1.1 Available Databases

Database Searching of DNA (Nucleotide Sequences)

- The large databases: Genbank (US), Embl (Europe), DDBJ (Japan). These databases are quite similar regarding their contents and are updating one each other.

- Genomic databases: Human (GDB), mouse (MGB), yeast (SGB), etc...

- Special databases: ESTs (expressed sequence tags), STSs (sequence-tagged sites), EPD (eukaryotic promotor database), REPBASE (repetitive sequence database) and many others.

Database Searching of Proteins (Amino Acid Sequence)

- The large databases are: Swiss-Prot (high level of annotation), PIR (protein identification resource).

- Translated databases, like SPTREMBL (translated EMBL), GenPept (translation of coding regions in GenBank) .
- Special databases, like PDB (sequences derived from the 3D structure Brookhaven PDB).

4.1.2 How to Perform Database-Searching

s the amount of biological relevant data is increasing so rapidly, knowing how to access and search this information is essential. There are three data retrieval systems of particular relevance to molecular biologist: Sequence Retrieval System (SRS), Entrez, DBGET.
These systems allow text searching of multiple molecular biology database and provide links to relevant information for entries that match the search criteria. The three systems differ in the databases they search and the links they have to other information.

Sequence Retrieval System (SRS)

SRS [17] is a homogeneous interface to over 80 biological databases that had been developed at the European Bioinformatics Institute (EBI) at Hinxton, UK (see also SRS help [18]. It includes databases of sequences, metabolic pathways, transcription factors, application results (like BLAST, SSEARCH, FASTA), protein 3-D structures, genomes, mappings, mutations, and locus specific mutations.
The web page listing all the databases contains a link to a description page about the database including the date on which it was last updated. You select one or more of the databases to search before entering your query.
After getting results you choose an alignment algorithm (like CLUSTALW, PHYLIP) enter parameters, and run it.
The SRS is highly recommended for use.

Entrez

Entrez [28] is a molecular biology database and retrieval system. Developed by the National Center for Biotechnology information (NCBI) (see Entrez help [29]). It is entry point for exploring distinct but integrated databases. Of the three text-based database systems, Entrez is the easiest to use, but also offers more limited information to search.

DBGET

DBGET [23] is an integrated database retrieval system, developed at the university of Tokyo (see DBGET help [24]). Provided access to 20 databases, one at a time. Having more limited options, the DBGET is less recommended than the two others.

4.1.3 DNA vs. Protein Searches

If we have a coding nucleotide sequence, we can translate it into protein sequence. (The other direction is, of-course, ambiguous, because the genetic code is degenerated). So, if we have a nucleotide sequence, should we search the DNA databases only? Or should we translate it to protein and search protein databases?
Usually, we should use proteins for database similarity searches when possible.
The reasons for this conclusion are:

- There are very different DNA sequences that code for similar protein sequences. We certainly do not want to miss those.
- When comparing DNA sequences, we get significantly more random matches than we get with proteins. There are several reasons for that:
  - DNA is composed of 4 characters: A,G,C,T. Hence, two unrelated DNA sequences are expected to have 25% similarity.
  - In contrast, protein sequence is composed of 20 characters (AA). The sensitivity of the comparison is improved. It is accepted that convergence of proteins is rare, meaning that high similarity between two proteins always means homology.
  - The DNA databases are much larger, and grow faster than protein databases. Bigger database means more random hits!
- For DNA we usually use identity matrices, while for protein more sensitive matrices like PAM and BLOSUM are used. This allows better search results.
- Proteins are rarely mutated during evolution. Due to their conservation, searching them reveals remote evolutionary relationships.

4.1.4 Specificity and Sensitivity of the Search Tools

- *Sensitivity*: the ability to detect "true positive" matches. The most sensitive search finds all true matches, but might have lots of false positives
- *Specificity* : the ability to reject "false positive" matches. The most specific search will return only true matches, but might have lots of false negatives (see figure 4.1)

When one chooses which algorithm to use, there is a trade off between these two characters. It quite trivial to create an algorithms which will do best one of these characters, the problem is to create algorithm which will consume both of them.

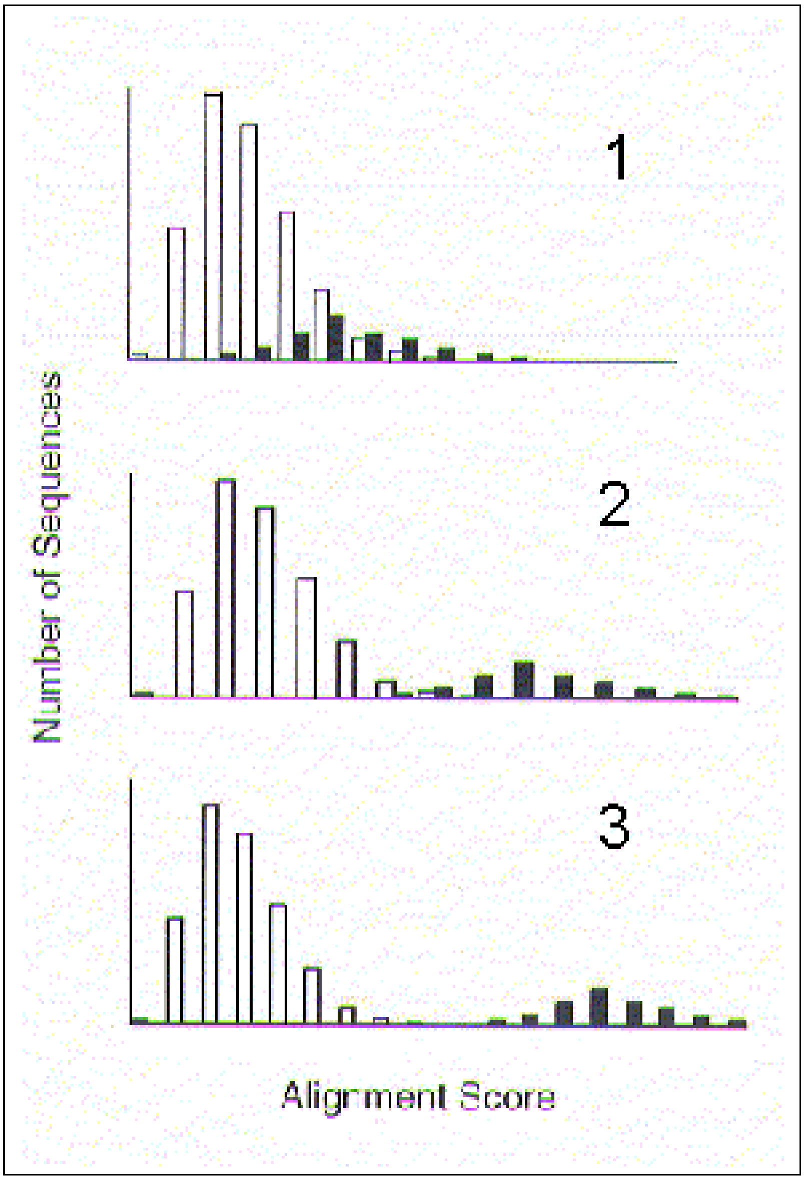

Figure 4.1: Specificity and sensitivity - The possibilities are: 1) Substantial overlap - Too many true positives are hidden by the background. All cutoffs are bad. A better model is required. 2) Small overlap - A few true positives have lower score than the highest random matches. An inclusive cutoff and visual inspection usually suffice. 3) Complete separation - All true positives are above the background. A simple cutoff suffices.

4.2 Main Algorithms for Database Searching

There are three main search tools: FastA, BLAST and SW-search.

4.2.1 The FastA Software Package

FastA is a sequence somparison software that uses the method of Pearson and Lipman [9]. The program compares a DNA sequence to a DNA database or a protein sequence to a protein database. Practically, FastA is a family of programs, which include: FastA, TFastA, Ssearch, etc.
Direct pointer: The fasta3 server at EBI: [20]
It also can be run through one of the retrieval systems (recommended). For example, GeneWeb mirror site at Weizmann Institute: [16]

Variants of FastA

- **FASTA** - Compares a DNA query sequence to a DNA database, or a protein query to a protein database, detecting the sequence type automatically.
  Versions 2 and 3 are in common use, version 3 having a highly improved score normalization method. It significantly reduces the overlap between the score distributions.

- **FASTX** - Compares a DNA query to a protein database. It may introduce gaps only between codons.

- **FASTY** - Compares a DNA query to a protein database, optimizing gap location, even within codons.

- **TFASTA** - Compares a protein query to a DNA database.

Sketch of the FastA Algorithm (see also lecture No. 3)

FastA locates regions of the query sequence and matching regions in the database sequences that have high densities of exact word matches. The score for such a pair of regiongs is saved as the $init_1$. Then it determines if any of the initial regions from different diagonals may be joined together to form an approximate alignment with gaps. Only non-overlapping regions may be joined. The score for the joined regions is the sum of the scores of the initial regions minus a joining penalty for each gap. The score of the highest scoring region, at the end of this step, is saved as the $init_n$ *score*.
After computing the initial scores, FastA determines the best segment of similarity between the query sequence and the search set sequence, using a variation of the Smith-Waterman algorithm. The score for this alignment is the *opt score*.

One of the few ways to evaluate the significance of such a score is to generate an empirical score distribution from the alignment of many random pairs of sequences having the same lengths as the two compared sequences. From this distribution, the Z_value (the number of standard deviation from the mean) for the alignment score of interest can then be estimated. Importantly, it should not be assumed that the score distribution is normal. Under reasonable assumptions the random score distribution for optimal ungaped local alignments can be proved to follow *extreme value distribution* (which proved to be significantly different from the normal distribution) [5]. In the current versions of FASTA and BLAST search programs, the evaluation of statistical significance best upon the extreme value distribution. These evaluations take the form of *E values.*

The E-value for a given alignment depends upon it's score as well as the lengths of both the query sequence and the database searched. It is the expectency of the number of distinct alignments with equivalent or superior score when using a random sequence. Thus, an E-value of five is not statistically significant whereas an E-value of 0.01 is. Scores of near $\sim 10^{-50}$ are now seen frequently and they suggest, with extremely high confidence, that the query protein is evolutionary related to the target matched in the database.

When the program finds similarity between your query sequence and a database sequence it is not always clear how significant this similarity really is. To evaluate if this similarity is statistically significance, you can run from the FASTA the package programs prss or prdf [20].

Programs in the FastA3 Package

- **FastA3** - Compare a protein sequence to a protein database, or a DNA sequence to a DNA database, using the FastA algorithm. Search speed and selectivity are controlled with the *ktup* (word-size) parameter. Searches with $ktup = 1$ are slower, but more sensitive, while $ktup = 2$ is faster but less effective.
  For proteins, the default is $ktup = 2$. For DNA, the default is $ktup = 6$. Use $ktup = 3$ or $ktup = 4$ more sensitivity. Use $ktup = 1$ for oligonucleotides (length < 20).

- **Ssearch3** - Compare a protein sequence to a protein database, or a DNA database, using the Smith-Waterman algorithm. It is very slow but much more sensitive for full-length proteins comparison.

- **Fastx3** - Compare a DNA sequence to a protein database, by comparing the translated DNA sequence in three frames and allowing gaps and frameshifts.

Which Program When?

- To identify an unknown protein sequence use either : FastA3, Ssearch3, tFastX3

- To identify structural DNA sequence : (repeated DNA, structural RNA) FastA3, Use first with $ktup = 6$ and than with $ktup = 3$.
- To identify an EST use FastX3 (check first whether the EST codes for a protein homologous to a known protein).

FastA Output

The output of FASTA contains information about all the hits (the score, the statistics of the score, the alignment itself, general information about the hit, Smith-Waterman score about the query and the hit).
At [20] and at [16] the output also contains a graph of frequency of expected and observed scores. (For calculating the z_score).
Running FASTA through SRS, enable to choose the output format. This is very recommended because it is exactly the details which biologists interested in.

Tips for FastA Results

- When $init_1 = init_0 = opt$: 100% homology over the matched stretch.
- When $init_n > init_1$: more than one matching region was found in the database sequence with poorly matching separating regions.
- When $opt > init_n$: the matching regions are greatly improved by adding gaps in one or both of the sequences.

4.2.2 BLAST - Basic Local Alignment Search Tool

Blast programs use a heuristic search algorithm. The programs use the statistical methods of Karlin and Altschul [1]. Blast programs were designed for fast database searching, with minimal sacrifice of sensitivity for distantly related sequences. The programs search databases in a special compressed format. To use your own private database with Blast, you need to convert it to the blast format.
Direct pointer: The BLAST at nbci: [27]
It also can be run through one of the retrieval systems (recommended). For example: GeneWeb mirror site at Weizmann Institute : [16]

Variants of BLAST

- **BLASTN** - Compares a DNA query to a DNA database. Searches both strands automatically. It is optimized for speed, rather than sensitivity.

- **BLASTP** - Compares a protein query to a protein database.

- **BLASTX** - Compares a DNA query to a protein database , by translating the query sequence in the 6 possible frames , and comparing each against the database (3 reading frames from each strand of the DNA) searching.

- **TBLASTN** - Compares a protein query to a DNA database, in the 6 possible frames of the database.

- **TBLASTX** - Compares the protein encoded in a DNA query to the protein encoded in a DNA database, in the $6 * 6$ possible frames of both query and database sequences.

- **BLAST2** - Also called *advanced BLAST*. It can perform gapped alignments.

- **PSI-BLAST** - (Position Specific Iterated) Performs iterative database searches

General View of How the BLAST Program Works (see also lecture No. 3):

- **BLAST** - The program compares the query to each sequence in database using heuristic rules to speed up the pairwise comparison. It first creates *sequence abstraction* by listing exact and similar words This is done in advance for each sequence in the database on the run for a certain query.
  BLAST finds similar words between the query and each database sequence, It then extends such words to obtain *high-scoring sequence pairs (HSPs)*. It also calculates statistics analytically like FastA does.

  BLAST 2.0 is a new version with new capabilities such as Gapped-Blast and PSI-Blast.

- **GAPPED BLAST** - This algorithm allows gaps to be introduced into the alignments. That means that similar regions are not broken into several segments (as in the older versions).
  This method reflects biological relationships much better than ordinary Blast.

- **PSI - BLAST** - (Position Specific Iterated) Blast provides a new automatic "profile like" search. The program first performs a gapped blast search of the database. The information of the significant alignments are then used by the program to construct a *position specific (PS)* score matrix. This matrix replaces the query sequence in the next round of database searching. The program may be iterated until no new significant alignments are found.

BLAST Output

This output is very similar to the FASTA output: It concains information for each hit (Including hit name, description, length,), score of hit (how many identical residues, how many residues contributing positively to the score, and the statistics of the score), and the local alignment itself.

4.2.3 The Smith-Waterman Tool

Smith-Waterman (SW) searching method compare query to each sequence in database using the full Smith-Waterman algorithm for pairwise comparisons [12]. It also uses search results to generate statistics.
Since SW searching is exhaustive, it is the slowest method. We use a special hardware + software (Bioccelerator) to execute the algorithm.
Direct pointer: [19]
It also can be run through the Weizmann Institute site: [16]

4.2.4 Comparison of the Programs

- Concept:
  SW and BLAST produce local alignments, while FASTA is global alignment tool. BLAST can report more than one HSP per database entry, while FASTA reports only one segment (match).

- Speed:
  BLAST $>$ FASTA $\gg$ SW
  BLAST (package) is a highly efficient search tool.

- Sensitivity:
  SW $>$ FASTA $>$ BLAST (old version!)
  FASTA is more sensitive, missing less homologues, (the opposite can also happen - if there are no identical residues conserved, but this is infrequent). It also gives better separation between true homologues and random hits. Usually when FASTA returns an unexpected hit.

- Statistics:
  BLAST calculates probabilities, and it sometimes fails entirely if some of the assumptions used are invalid. FastA calculates significance 'on the fly' from the given dataset which is more relevant but can be problematic if the dataset is small .

4.2.5 Tips for DB Searches

- Use latest database version
- Run Blast first, then depending on your results run a finer tool (fasta, ssearch, SW, blocks, etc..)
- Whenever possible, use the translated sequence.
- $E() < 0.05$ is statistically significant, usually biologically interesting. Check also $0.05 < E() < 10$ because you might find interesting hits.
- Pay attention to abnormal composition of the query sequence, it usually causes biased scoring.
- Split large query sequences (> 1000 for DNA, > 200 for protein).
- If the query has repeated segments, remove them and repeat the search.

4.3 Multiple Sequence Alignment

4.3.1 Why to Performing Multiple Alignments?

Multiple nucleotide or amino acid sequence alignment techniques are usually motivated by one of the following goals :

- Characterization of protein families, or identification of shared regions of homology in a multiple sequence (this may happen when a sequence search revealed homologies to several sequences)
- Determination of the consensus sequence of several aligned sequences.
- Help predicting the secondary and tertiary structures of new sequences.
- Preliminary step in molecular evolution analysis using Phylogenetic methods for constructing phylogenetic trees.

4.3.2 General View of How the Programs Work:

The most practical and widely used method for multiple sequence alignment uses hierarchical extensions of pairwise alignments. The algorithm is as follows:

1. Perform all pairwise comparisons between the sequences.

2. Perform cluster analysis on the pairwise data to generate a cluster hierarchy. This hierarchy may built in the form of a binary tree or a simple ordering.

3. Construct the multiple alignment by first aligning the most similar pair of sequences, then the next most similar pair and so on. Once an alignment of two sequences has been made, it remains fixed. Thus for a set of sequences A, B, C, D having aligned A with C and B with D the alignment of A, B, C, D is obtained by comparing the alignment of A and C with that of B and D using averaged scores at each aligned position.

4.3.3 Choosing Sequences for Alignment

General considerations :

- The more sequences to align the better.
- Don't include similar (> 80%) sequences, because it may cause a drift of the result to one direction.
- Sub-groups should be pre-aligned separately, and one member of each subgroup should be included in the final multiple alignment.

4.3.4 Pileup - Multiple Alignment in GCG

GCG (Genetic Computer Group) is a package of sequence analysis programs which can be run trough [22]. Pileup creates multiple sequence alignment from a group of related sequences using progressive, pairwise alignment method of Feng and Doolittle [2]. It can also plot a tree showing the clustering relationships used to create the alignment.
The input file for Pileup is a list of sequence file names or sequence codes in the database.
Pileup follows the general scheme outlined in section 4.3.2. The clustering strategy called UPGMA that stands for Unweighted Pair-Group Method using Arithmetic average [13].

The clustering algorithm it uses starts each clusters one sequence each, and iteratively constructs larger clusters. In each iteration, it merges the two clusters whose pairwise alignment distance is the smallest. Cluster pairwise alignment is a simple extension of sequence alignment. For a pairwise alignment of clusters of sequences, the comparison score between any two positions in those clusters is simply the arithmetic average of the scores for all possible symbol comparisons at those positions. When gaps are inserted into a cluster to produce an alignment, they are inserted at the same position in all of the sequences of the cluster [8]. The full multiple alignment is obtained once all the sequences have been clustered into one cluster. This hierarchical clustering is naturally described by a dendrogram, which Pileup

can plot (see figure 4.2).
As a general rule, Pileup can align up to 500 sequences, with any single sequence in the final alignment restricted to a maximum length of 7000 characters (including gap characters inserted into the sequence by Pileup to create the alignment). However, the longer are the sequences in the alignment, the number of sequences Pileup can handle decreases.

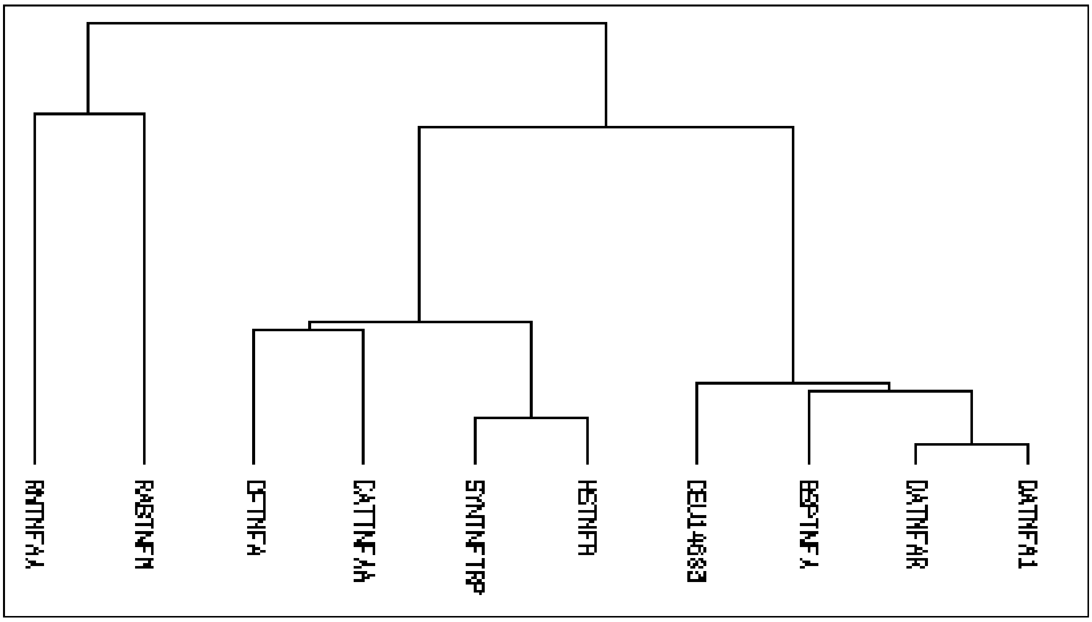

Figure 4.2: Dendrogram from Pileup// Distances along the vertical axis is proportional to the difference between sequences.Distance along the horizontal axis has no significance at al

4.3.5 Displaying a Multiple Alignment in GCG

There are several programs to display the multiple alignment nicely:

1. The **Pretty** program prints sequences with their columns aligned and can display a consensus for the alignment, allowing the user to look at relationships among the sequences.

2. The **PrettyBox** program displays the alignment graphically with the conserved regions of the alignment as shaded boxes. The output is in Postscript format.

ShadyBox

ShadyBox is a multiple alignment editor program which enables you to box and shade residues or segments of the aligned sequences. The program works on a msf or Pretty

HSTNFR GGGAAGAG—TTCCCCAGGGACCTCTCTCTAATCAGCCCTCTGGCCCAG——GCAG SYNTNFTRP GGGAAGAG—TTCCCCAGGGACCTCTCTCTAATCAGCCCTCTGGCCCAG——GCAG CFTNFA ——————————————————-TGTCCAG——ACAG CATTNFAA GGGAAGAG—CTCCCACATGGCCTGCAACTAATCAACCCTCTGCCCCAG——ACAC RABTNFM AGGAGGAAGAGTCCCCAAACAACCTCCATCTAGTCAACCCTGTGGCCCAGATGGTCACCC RNTNFAA AGGAGGAGAAGTTCCCAAATGGGCTCCCTCTCATCAGTTCCATGGCCCAGACCCTCACAC OATNFA1 GGGAAGAGCAGTCCCCAGCTGGCCCCTCCTTCAACAGGCCTCTGGTTCAG——ACAC OATNFAR GGGAAGAGCAGTCCCCAGCTGGCCCCTCCTTCAACAGGCCTCTGGTTCAG——ACAC BSPTNFA GGGAAGAGCAGTCCCCAGGTGGCCCCTCCATCAACAGCCCTCTGGTTCAA——ACAC CEU14683 GGGAAGAGCAATCCCCAACTGGCCTCTCCATCAACAGCCCTCTGGTTCAG——ACCC ** *

Figure 4.3: CLUSTAL W (1.7) multiple sequence alignment

output file, and produces a postscript output file. The original input file is not changed.

4.3.6 ClustalW- for Multiple Alignment

ClustaW is a general purpose multiple alignment program for DNA or proteins. [4].
ClustalW improves the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice.
It can create multiple alignments, manipulate existing alignments, perform profile analysis and create phylogentic trees. Alignment can be constructed either slow but accurate or by fast but more approximated.
The input file for clustalW is a file containing all sequences in one of the following formats: NBRF/PIR, EMBL/SwissProt, Pearson (Fasta), GDE, Clustal, GCG/MSF, RSF.

One of the variant of ClustalW is ClastalX [14]. This variant provides a new window-based user interface to the ClustalW program. NCBI site: [26]
(see the next figure for an example)

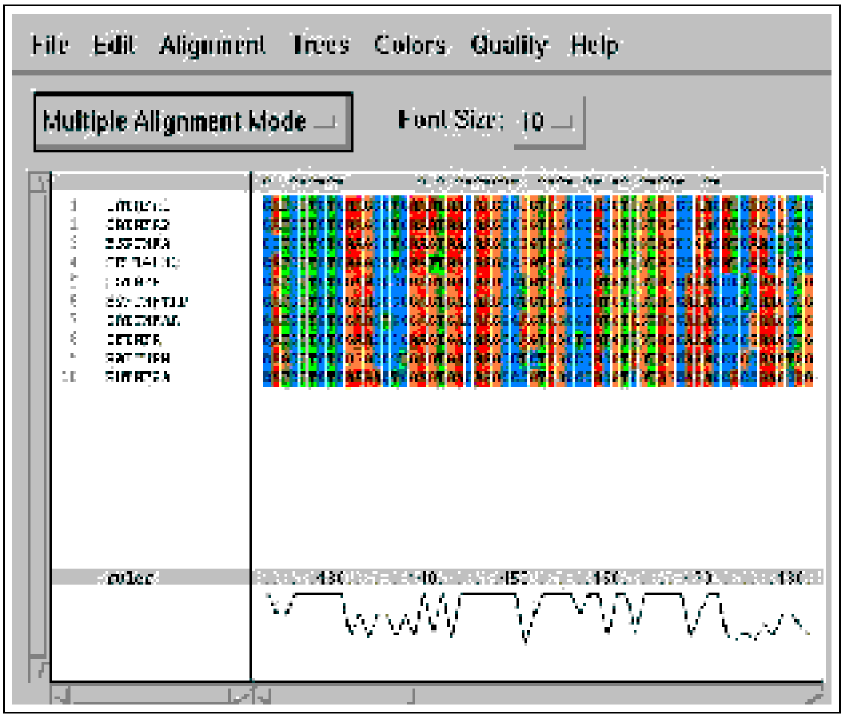

Figure 4.4: An example of alignment from ClustalX

4.4 Blocks Database and Tools

Blocks are multiply aligned ungapped segments corresponding to the most highly conserved regions of proteins. Block Searcher, Get Blocks and Block Maker are aids for detection and verification of protein sequence homology. They compare a protein or DNA sequence to a database of protein blocks , retrieve blocks, and create new blocks, respectively.
The BLOCKS web server at URL: [15]

4.4.1 General View of How the Program Works

See [10], Many known proteins can be grouped into families according to functional and sequence similarities. The similarity of the proteins across the sequences in each family is far from uniform. While some regions are clearly conserved, others display little sequence similarity. Often the conserved regions are crucial to the protein's function, including, for example, enzymatic catalytic sites. Such conserved regions can be used to probe an uncharacterized sequence to indicate its function .

The description of a protein family by its conserved regions focuses on the family's characteristic and distinctive sequence features, thus reducing noise compared to alignments which handle all positions uniformly. Databases of conserved features of protein families can be utilized to classify sequences from proteins, cDNAs and genomic DNA. The database was constructed from sequences of protein families using a fully automated procedure. Searching the Blocks database with a sequence query allows detection of one or more blocks representing a family. A best set of blocks representing each protein group is found automatically by the two-step PROTOMAT system [3]. The first step incorporates a motif finder. Currently there is use of MOTIF algorithm [11]. MOTIF exhaustively evaluates spaced triplets of amino acids that are common to multiple sequences. There is also implemented a Gibbs sampling motif finder that iteratively optimizes random 'seeds' for blocks [6]. The MOTIF and Gibbs algorithms generate similar block sets for the sequences used in the Blocks Database. The second step of the PROTOMAT system combines and refines the original blocks and assembles a optimal set of blocks that is consistently found in most of the sequences in the group.

4.4.2 Other Uses of the Blocks Database

The automated construction and extensive data in the Blocks database make it suitable for uses other than protein classification. The local alignments of sequence segments provided data for the BLOSUM series of amino acid substitution matrices. These matrices performed very well in sequence database searches. The Blocks database was also used to test and

compare different methods for weighting sequences to reduce redundancy.
Many blocks are made up of sequence segments with known functions such as ligand binding regions, catalytic domains or trans-membranal domains. This can be a resource for research on specific domains. For example, when studying protein-nucleotide binding sites one can search for block families annotated as having such sites or for blocks containing the known signature of the sites. The blocks found can help refine the signature and even reveal unannotated sites.

4.4.3 The Blocks Searcher Tool

For searching a database of blocks, the first position of the sequence is aligned with the first position of the first block, and a score for that amino acid is obtained from the profile column corresponding to that position. Scores are summed over the width of the alignment, and then the block is aligned with the next position.
This procedure is carried out exhaustively for all positions of the sequence for all blocks in the database, and the best alignments between a sequence and entries in the Blocks database are reported. The score of the block is designed to indicate how well does the query sequence represent the block group.

Typically, a group of proteins has more than one region in common and their relationship is represented as a series of blocks separated by unaligned regions. Naturally, more high scoring blocks strong then the supposed relation between the wuery and the group.

4.4.4 The Block Maker Tool

Block Maker finds conserved blocks in a group of two or more unaligned protein sequences, which are assumed to be related. Input file must contain at least two sequences. The sequences must be in FastA format. Results are returned by e-mail.

4.5 Recommended Sites on the Web

A collection of bioinformatic links: [25]
Biocatalog of Programs: [21]

Bibliography

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol*, 215:403–410, 1990.

[2] D. F. Feng and R. F. Doolitle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25:351–360, 1987.

[3] S. Henikoff and J. G. Henikoff. automated assembly of protein blocks for database searching. *Nucleic Acid Res*, 19(23):6565–6572, 1991.

[4] j. D. Thompson, D. G. Higgins, and T. J. Gibson. Clastal w : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap panelties, and weight matrix choice. *Nucleic Acid Res*, 22(22):4673–4680, 1994.

[5] S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA*, 87:2264–2268, 1990.

[6] C. E. Lawrence, S. F. Altschul, M. S. Bogusky, J. S. Liu, A. F. Neuwald, and j. C. Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–214, 1993.

[7] F. Lewitter. Text-based database searching. *Trends Guide to Bioinformatics*, pages 3–5, 1998.

[8] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[9] R. W. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*, 85:2444–2448, 1988.

[10] S. Pietrokovski, J. G. Henikoff, and S. Henikoff. The blocks database- a system for protein classification. *Nucleic acid research*, 24(1):197–200, 1996.

[11] H. O. Smith and T. M. Chandrasegaran. Finding sequence motifs in groups of functionally related proteins. *Proc. Natl. Acad. Sci. USA*, 87(2):826–830, 1990.

[12] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

[13] P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy.* W.H Freeman and company, San Francisco, California,USA, 1973.

[14] J. D. Thompson, T. J. Gibson, F. Plewniac, F. Jeanmougin, and D. G. Higgins. The clustalx windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. *Nucleic Acid Res.*, 25:4876–4882, 1997.

[15] http://blocks.fhcrc.org/.

[16] http://dapsas1.weizmann.ac.il\bcd\bcd_parent\bcd_bioccel\bioccel.html..

[17] http://srs/ebi/ac/uk/.

[18] http://srs.ebi.ac.uk/srs5/man/mi_srswww.html.

[19] http://www2.ebi.ac.uk/bic_sw/.

[20] http://www2.ebi.ac.uk/fasta3/.

[21] http://www.ebi.ac.uk/biocat/biocat_form.html.

[22] http://www.gcg.com.

[23] http://www.genome.ad.jp/dbget/dbget2.html .

[24] http://www.genome.ad.jp/dbget/dbget_manual.html .

[25] http://www.ii.uib.no/~inge/list.html.

[26] http://www.ncbi.nlm.nih.gov/.

[27] http://www.ncbi.nlm.nih.gov/BLAST/.

[28] http://www/ncbi.nlm.nih.gov/Entrez/.

[29] http://www/ncbi.nlm.nih.gov/Entrez/entrezhelp.html.