Lecture 6: December ,28, 2001

Lecturer: Racheli Zakarin and Roded Sharan *Scribe: Ofer Molad and Yuval Altman*[1]

6.1 Bioinformatics Databases and Tools - Introduction

In recent years, biological databases have greatly developed, and became a part of the biologist's everyday toolbox (see, e.g., [4]). There are several reasons to search databases, for instance:

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

2. Some of the databases contain annotation which has already been added to a specific sequence. Finding annotation for the searched sequence or its *homologous sequences* can facilitate its research.

3. Find similar non-coding DNA stretches in the database: for instance repeat elements or regulatory sequences.

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

5. Search for *homologous proteins* - proteins similar in their sequence and therefore also in their presumed folding or structure or function.

Topics covered in this lecture:

1. *Primary sequence databases* - Protein databases and nucleotide databases. Characteristics and specific examples.

2. *Text based searching* - Motivation. Tools for textual search.

3. *Sequence based searching* - Query types. FastA, Blast, SW.

4. *Significance of scores* - Analysis of scoring models for sequence alignment.

[1]Based on scribe by Naomi Keren and Guy Kol, winter 2000, and on lecture slides by Dr. Racheli Kreisberg-Zakarin, fall 2001.

5. *Multiple sequence alignments* - Motivation. Techniques. Examining the ClustalW tool.

6. *secondary databases* - Databases of high level data representation. Examples.

6.2 Primary sequence databases

6.2.1 Introduction

In the early 1980's, several primary database projects evolved in different parts of the world (see table 6.1). There are two main classes of databases: DNA (nucleotide) databases and protein databases. The primary sequence databases have grown tremendously over the years.

DNA (nucleotide)		**Protein**	
EMBL	UK	PIR	US
GenBank	US	MIPS	Germany
DDBJ	Japan	Swiss-Prot	Swiss
Celera	Celera	TrEMBL	Swiss
		NRL_3D	US
		GenPept	US

Table 6.1: List of primary sequence databases and their locations.

Today they suffer from several problems, unpredicted in early years (when their sizes were much smaller):

- Databases are regulated by users rather than by a central body (except for Swiss-Prot).
- Only the owner of the data can change it.
- Sequences are not up to date.
- Large degree of redundancy in databases and between databases.
- Lack of standard for fields or annotation.

6.2.2 Protein Databases (Amino Acid Sequence)

PIR - International Protein Sequence Database)

PIR - The Protein Sequence Database [20] was developed in the early 1960's. It is located at the National Biomedical Research Foundation (NBRF). Since 1988 it has been maintained by PIR-International (see [21]).

PIR currently contains 250,417 entries (Release 70.0, September 30, 2001). It is split into four distinct sections, that differ in quality of the data and the level of annotation:

PIR1 - fully classified and annotated entries.

PIR2 - preliminary entries, not thoroughly reviewed.

PIR3 - unverified entries, not reviewed.

PIR4 - conceptual translations.

PIR home page: [20]. For a sample PIR entry, see [23].

Swiss-Prot

Swiss-Prot (home page: [35]) was established in 1986. It is maintained collaboratively by SIB (Swiss Institute of Bioinformatics) and EBI/EMBL. Provides high-level annotations, including description of protein function, structure of protein domains, post-translational modifications, variants, etc. It aims to be minimally redundant. Swiss-Prot is linked to many other resources, including other sequence databases. For a sample entry, see figures 6.1, 6.2, 6.3.

TrEMBL - Translated EMBL

Translated EMBL (home page: [36]) was created in 1996 as a computer annotated supplement to Swiss-Prot. It contains translations of all coding sequences in the EMBL nucleotide sequence database. SP-TrEMBL contains entries that will be incorporated into Swiss-Prot REM-TrEMBL contains entries that are not destined to be included in Swiss-Prot, (for example, T-cell receptors, patented sequences). The entries in REM-TrEMBL have no accession number.

GenPept

GenPept is a supplement to the GenBank nucleotide sequence database. Its entries are translation of coding regions in GenBank entries. They contain minimal annotation, primarily extracted from the corresponding GenBank entries. For the complete annotations, one must refer to the GenBank entry or entries referenced by the accession number(s) in the GenPept entry. For a sample GenPet entry, see [9].

NRL_3D

NRL_3D is produced and maintained by PIR. It contains sequences extracted from the Protein DataBank (PDB) (see [45]). The entries include secondary structure, active site, binding site and modified site annotations, details of experimental method, resolution, R-factor, etc. NRL_3D makes the sequence data in the PDB available for both text based and

General information	
Entry name	**POLG_WNV**
Accession number	**P06935**
Created	Rel. 06, 1-JAN-1988
Sequence update	Rel. 06, 1-JAN-1988
Annotation update	Rel. 40, 16-OCT-2001
Description and origin of the Protein	
Description	GENOME POLYPROTEIN [CONTAINS: CAPSID PROTEIN C (CORE PROTEIN); MAT (ENVELOPE PROTEIN M); MAJOR ENVELOPE PROTEIN E; NONSTRUCTURAL PRO NS2B, NS4A AND NS4B; PROTEASE/HELICASE (EC 3.4.21.98) (NS3); RNA-DIRECTI POLYMERASE (EC 2.7.7.48) (NS5)].
Organism source	West Nile virus (WN).
Taxonomy	Viruses; ssRNA positive-strand viruses, no DNA stage; Flaviviridae; Flavivirus.
NCBI TaxID	11082
References	
[1]	Castle,E., Leidner,U., Nowak,T., Wengler,G., **Primary structure of the West Nile flavivirus genome region coding for all nonstructural** (1986) *Virology* **149**:10 Position: SEQUENCE FROM N.A. Medline: 86124703 PubMed: 3753811
[2]	Castle,E., Nowak,T., Leidner,U., Wengler,G., **Sequence analysis of the viral core protein and the membrane- associated proteins V1 a flavivirus West Nile virus and of the genome sequence for these proteins.**

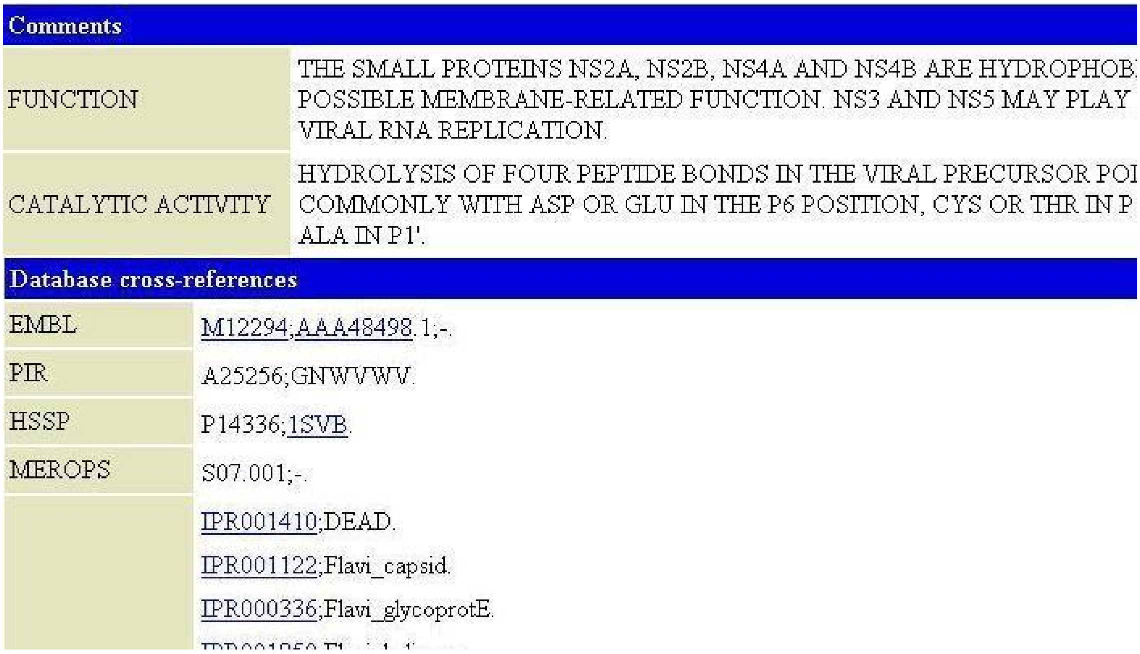
Figure 6.1: Sorce: [35]. A sample Swiss-Prot entry, part 1.

sequence-based searching. It also provides cross-reference information for use with the other PIR Protein Sequence Databases. For NRL_3D information, and sample entry, see [22].

Summary of protein sequence databases

- PIR(1-4) - comprehensive, poor quality of annotation (even in PIR1).
- Swiss-Prot - poor sequence coverage, highly structured, excellent annotation.
- GenPept most comprehensive, poor quality of annotation.
- NRL_3D - least comprehensive but is directly relating to structural information.

When searching for a protein sequence, it is recommended to search *all* databases.



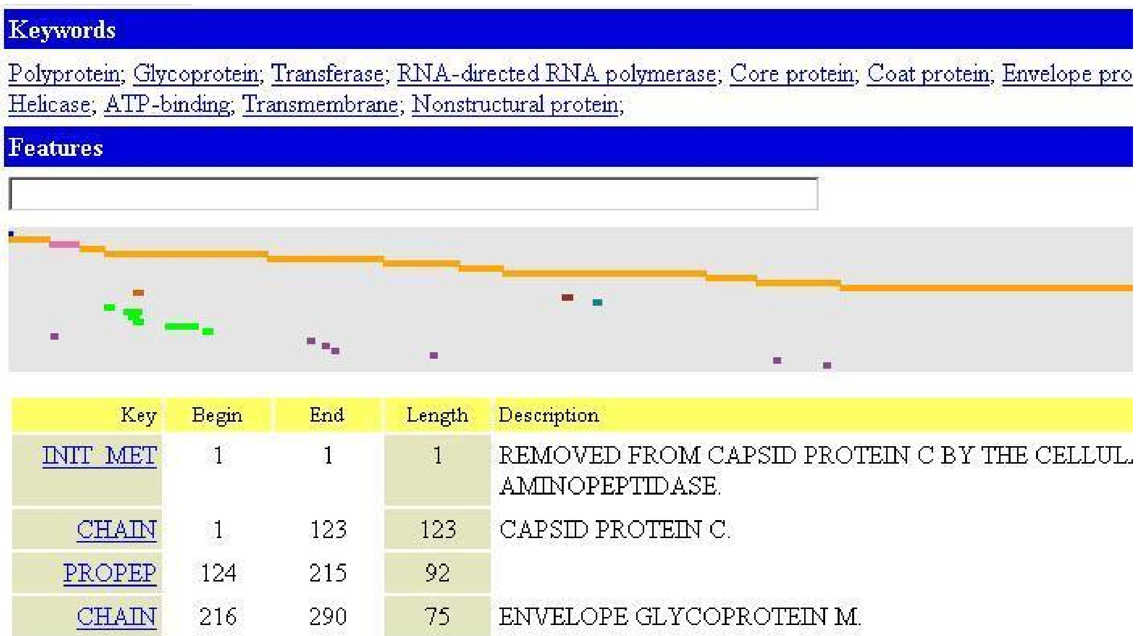
Comments

FUNCTION	THE SMALL PROTEINS NS2A, NS2B, NS4A AND NS4B ARE HYDROPHOB POSSIBLE MEMBRANE-RELATED FUNCTION. NS3 AND NS5 MAY PLAY VIRAL RNA REPLICATION.
CATALYTIC ACTIVITY	HYDROLYSIS OF FOUR PEPTIDE BONDS IN THE VIRAL PRECURSOR POI COMMONLY WITH ASP OR GLU IN THE P6 POSITION, CYS OR THR IN P ALA IN P1'.

Database cross-references

EMBL	M12294;AAA48498.1;-.
PIR	A25256;GNWVWV.
HSSP	P14336;1SVB.
MEROPS	S07.001;-.
	IPR001410;DEAD. IPR001122;Flavi_capsid. IPR000336;Flavi_glycoprotE.

Figure 6.2: Sorce: [35]. A sample Swiss-Prot entry, part 2.



Keywords

Polyprotein; Glycoprotein; Transferase; RNA-directed RNA polymerase; Core protein; Coat protein; Envelope pro Helicase; ATP-binding; Transmembrane; Nonstructural protein;

Features

Key	Begin	End	Length	Description
INIT_MET	1	1	1	REMOVED FROM CAPSID PROTEIN C BY THE CELLUL AMINOPEPTIDASE.
CHAIN	1	123	123	CAPSID PROTEIN C.
PROPEP	124	215	92	
CHAIN	216	290	75	ENVELOPE GLYCOPROTEIN M.

Figure 6.3: Sorce: [35]. A sample Swiss-Prot entry, part 3.

6.2.3 DNA Databases (Nucleotide Sequences)

The growth rate of DNA databases is much higher than that of the protein databases. This is because most of the DNA is not coding for proteins and because DNA sequencing is the most prominent source of database entries. Figure 6.4 illustrates the semi-exponential growth of DNA databases along the years.

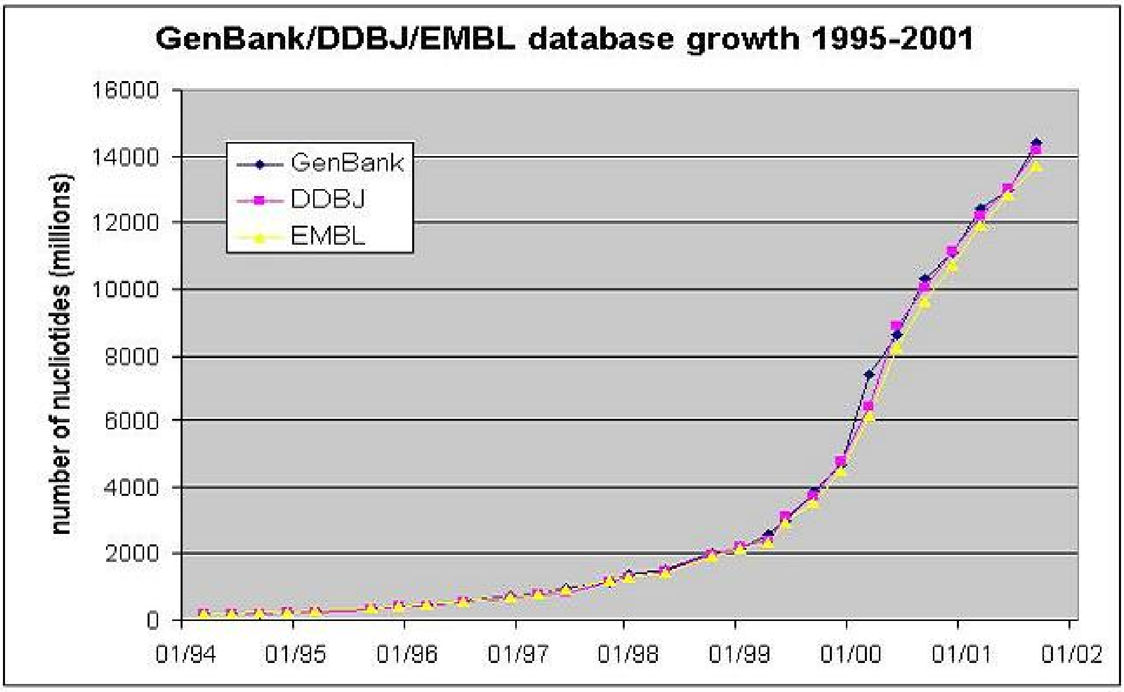


Figure 6.4: Sorce: [29]. The DNA database growth.

The large DNA databases are: Genbank (US), EMBL (Europe - UK), DDBJ (Japan). These databases are quite similar regarding their contents and are updating one another periodically. This was is a result of the International Nucleotide Sequence Database Collaboration.

EMBL

EMBL is a DNA sequence database from European Bioinformatics Institute (EBI). See EBI home page: [30]. EMBL includes sequences from direct submissions, from genome sequencing projects, scientific literature and patent applications. Its growth is exponential, on 3.12.01 it contained 15,386,184,380 bases in 14,370,773 records. EMBL supports several retrieval tools: SRS for text based retrieval and Blast and FastA for sequence based retrieval. See [31] for more information and for a sample EMBL entry. EMBL is divided into several divisions.

The division differ by the amount of sequences and by the quality of the data. See figure 6.5 for division statistics.

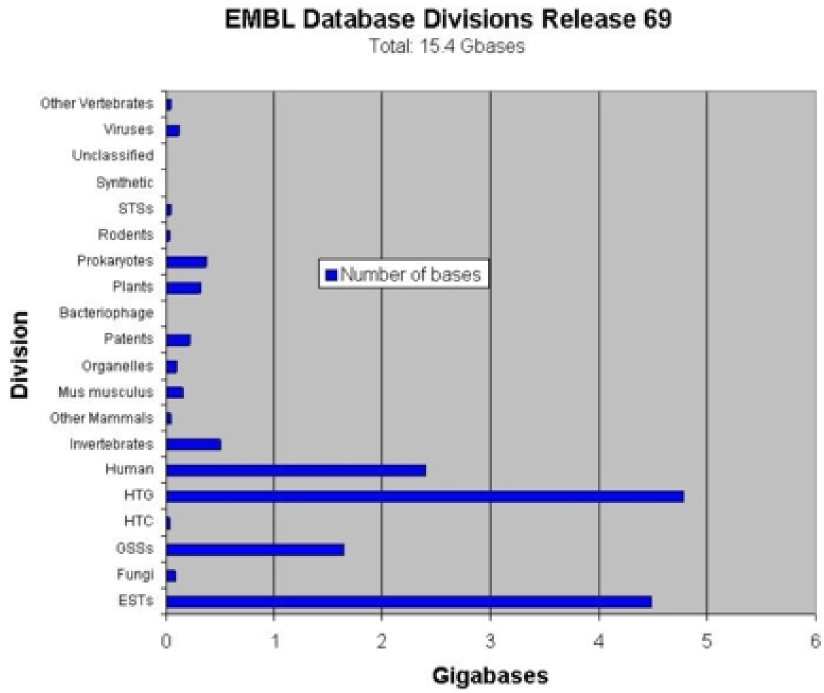


Figure 6.5: Sorce: [31]. EMBL divisions and number of bases in each division.

GenBank

GenBank is a DNA sequence database from National Center Biotechnology Information (NCBI). See NCBI home page: [38]). It incorporates sequences from publicly available sources (direct submission and large-scale sequencing). Like EMBL it is also split into smaller, discrete divisions (see table 6.2). This facilitates an efficient search. See [43] for more information and for a sample GenBank entry.

Genome databases of specific organisms

These are smaller databases that present an integrated view of a particular biological system. Here, sequence data is only the first level of abstraction; It contains other levels of biological

Division Code	**Description**
PRI	primate sequences
ROD	rodent sequences
MAM	other mammalian sequences
VRT	other vertebrate sequences
INV	invertebrate sequences
PLN	plant, fungal, and algal sequences
BCT	bacterial sequences
RNA	structural RNA sequences
VRL	viral sequences
PHG	bacteriophage sequences
SYN	synthetic sequences
UNA	unannotated sequences
EST	EST sequences (expressed sequence tags)
PAT	patent sequences
STS	STS sequences (sequence tagged sites)
GSS	GSS sequences (genome survey sequences)
HTG	HTGS sequences (high throughput genomic sequences)

Table 6.2: Source: [8]. GenBank divisions. The biggest division is the EST; Due to its rapid growth, it is divided into 23 pieces.

information. This leads to an overall understanding of the genome organization. An example is the Flybase, a comprehensive biological database of the Drosophila (see [18]).

Glossary

ESTs (Expressed Sequence Tags) - Short fragments of mRNA samples that are taken from a variety of tissues and organisms. These samples are amplified and sequenced. The sequencing is done in one read pass, therefore the ESTs are a non-accurate source of information. There are about 6 million sequenced ESTs (more than 1/3 cloned from human) .

STSs (Sequence-Tagged Sites) - Short genomic samples that serve as genomic markers.

HTGS (High Throughput Genomic Sequences) - Sequences obtained in the course of sequencing the whole genome. The records of this databases are classified according to their level of advancement towards sequence completion.
Phase 0 - Single or few pass reads of a single clone (not contigs).
Phase 1 - Unfinished, may be unordered, unoriented contigs, with gaps.

Phase 2 - Unfinished, ordered, oriented contigs, with or without gaps.
Phase 3 - Finished, no gaps (with or without annotation).

6.3 Text based searching

6.3.1 How to Perform Database-Searching?

As the amount of biological relevant data is increasing so rapidly, knowing how to access and search this information is essential. The two main ways of searching are:

Text based search - Searching the annotations. Examples: SRS, GCG's Lookup, Entrez.

Sequence based search - Searching the sequence itself. Examples: Blast, FastA, SW.

6.3.2 Text based retrieval tools

The listed retrieval systems allow text searching in a multitude of molecular biology database and provide links to relevant information for entries that match the search criteria. The systems differ in the databases they search and the links they have to other information.

SRS (Sequence Retrieval System)

SRS had been developed at the EBI. It provides a homogeneous interface to over 80 biological databases (see SRS help at [25]). 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. For each of the 80 available databases, there is a short description, including its last release. Before entering a query, one selects one or more of the databases to search. It is possible to send the query results as a batch query to a sequence search tool. The SRS is highly recommended for use. SRS entrance page: [24].

Entrez

Entrez is a molecular biology database and retrieval system, developed by the NCBI (see Entrez help at [42]). It is an entry point for exploring the NCBI's integrated databases. The Entrez is easy to use, but unlike SRS, the search is limited. It does not allow customization with an institutes preferred databases. Entrez entrance page: [41].

6.4 Sequence Based Searching

DNA search versus Protein search

The straight forward technique to search a DNA sequence is to search it against DNA databases. However, it is possible to translate a coding DNA sequence into a protein sequence, and then search it against protein databases. Let us compare the two techniques:

- A DNA sequence is a string of length n over an alphabet of size 4. Its protein translation is a string of length $n/3$ over an alphabet of size 20. Statistically, the expected number of random matches in some arbitrary database is larger for a DNA sequence.
- DNA databases are much larger than protein databases, and they grow faster. This also means more random hits.
- Translation of a DNA sequence to a protein sequence causes loss of information.
- Protein sequences are more biologically preserved than DNA sequences.

Bottom line: Translating DNA to a protein yields better search results. When possible (i.e. for a coding DNA sequence), it is the recommended technique.

Protein sequences are always searched against protein databases. Translating them to DNA is ambiguous and results in a large number of possible DNA sequences. The analysis in the previous paragraph also discourages translation to DNA.

Homology modeling

As stated, a primary goal of sequence search is to find sequences which are homologous to the query sequence. Such a homologous sequence shares sequence similarity with the query sequence. The similarity is derived from common ancestry and conservation throughout evolution. Homologous proteins are similar in their structure. This is the basis for *homology modeling* structure determination through the structure of similar proteins.

Evaluating search tools

The main goal in searching is finding the relevant information and avoiding non relevant information. We therefore define:

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".

When one chooses which algorithm to use, there is a trade off between these two figures of merit. It is quiet trivial to create an algorithm which will optimize one of these properties. The problem is to create an algorithm that will perform well with respect to both of them. A second criteria for evaluating algorithm is its time performance.

We will examine three main search tools: FastA (better for nucleotides than for proteins), BLAST (better for proteins than for nucleotides) and SW-search (more sensitive than FastA or BLAST, but much slower).

6.4.1 FastA

FastA is a sequence comparison software that uses the method of Pearson and Lipman [6]. The basic FastA algorithm assumes a query sequence and a database over the same alphabet. Practically, FastA is a family of programs, allowing also cross queries of DNA versus protein. The program variants are listed in table 6.3.

PROGRAM	*FUNCTION*
fasta3	scan a protein or DNA sequence library for similar sequences
fastx/y3	compare a DNA sequence to a protein sequence database, comparing the translated DNA sequence in forward and reverse frames.
tfastx/y3	compares a protein to a translated DNA data bank
fasts3	compares linked peptides to a protein databank
fastf3	compares mixed peptides to a protein databank

Table 6.3: Source: [33]. Variants of the FastA algorithm. Note: fastx3 uses a simpler, faster algorithm for alignments that allows frameshifts only between codons; fasty3 is slower but produces better alignments with poor quality sequences because frameshifts are allowed within codons (source: [32]).

Under different circumstances it is favorable to use different programs:

- To identify an unknown protein sequence use either FastA3 or tFastX3.
- To identify structural DNA sequence: (repeated DNA, structural RNA) use FastA3, first with $ktup = 6$ and then with $ktup = 3$.
- To identify an EST use FastX3 (check whether the EST codes for a protein homologous to a known protein).
- Use $ktup = 1$ for oligonucleotides (length < 20).

FastA3 (Fastx3, etc.) is the current version of FastA. FastA is available directly via the FastA3 server [28], or it can be accessed through one of the retrieval systems ,e.g., the GenWeb mirror site at the Weizmann Institute [16].

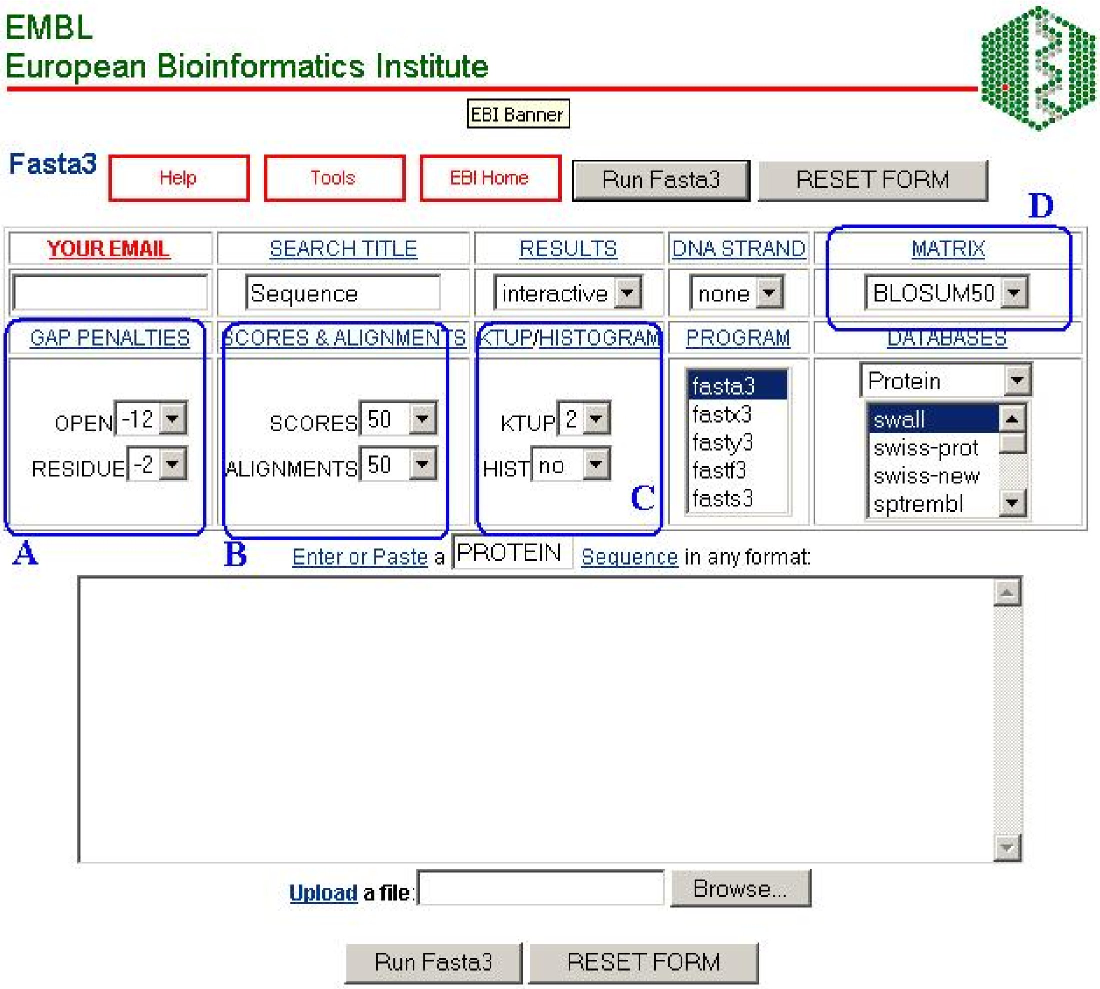


Figure 6.6: Sorce: [28]. FastA query screen. **A** - Default gap opening penalty: -12 for proteins, -16 for DNA. Default gap extension penalty: -2 for proteins, -4 for DNA. **B** - Max number of scores and alignments is 100. **C** - The larger the word-length the less sensitive, but faster the search will be. **D** - Default matrix: Blosum50. Lower PAM and higher blosum detect close sequences. Higher PAM and lower blosum detect distant sequences.

FastA - Steps

- Hashing: FastA locates regions of the query sequence and matching regions in the database sequences that have high densities of exact matches of k-tuple subsequences. The *ktup* parameter controls the length of the k-tuple.

- Scoring: The ten highest scoring regions are scored again using a scoring matrix. The score for such a pair of regions is saved as the $init_1$ score.

- Introduction of Gaps: FastA determines if any of the initial regions from different diagonals can 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.

- Alignment: 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.

- Random Sequence Simulation: In order to evaluate the significance of such alignment, FastA empirically estimates the score distribution from the alignment of many random pairs of sequences. More precisely, the characters of the query sequences are reshuffled (to maintain bias due to length and character composition) and searched against a random subset of the database. This empirical distribution is extrapolated, assuming it is an extreme value distribution. Each alignment to the real query is assigned a Z-score and an E-score. For a formal definition of Z-score and E-score, see Section 6.5.

FastA Output

The standard FastA output contains a list of the best alignment scores and a visual representation of the alignments. See figures 6.8, 6.7. When evaluating FastA E-scores, the following rule of thumb can be applied: Sequences with E-score less than 0.01 are almost always found to be homologous. Sequences with E-score between 1 and 10 frequently turn out to be related as well.

FastA uses a statistical model in order to determine a threshold E-score above which results are returned. However, sometimes the assumptions of this statistical model fail. The reliability of the sequence statistics for a given query can be quickly confirmed by looking at the histogram of observed and expected similarity scores (see [44]). The FastA histogram is an optional output. A sample histogram is shown in figure 6.9.

```
FASTA (3.39 May 2001) function [optimized, BL50 matrix (15:-5)] ktup: 2
 join: 36, opt: 24, gap-pen: -12/ -2, width:  16
 Scan time:  1.990
The best scores are:                                       opt bits E(103115)
SW:GRXB_BACAN Q9ZFB5 SPORE GERMINATION PROTEIN XB   ( 359) 1174  246 8.3e-65
SW:GRBB_BACSU P39570 SPORE GERMINATION PROTEIN B2   ( 368)  173   45 0.00024
SW:GRKB_BACSU P49940 SPORE GERMINATION PROTEIN KB   ( 373)  168   44 0.00048
SW:Y129_METJA Q57593 HYPOTHETICAL PROTEIN MJ0129.   ( 170)  119   34    0.25
SW:YK10_YEAST P36125 HYPOTHETICAL 32.0 KDA PROTEI   ( 273)  110   32     1.2
SW:C560_CHOCR P48934 SUCCINATE DEHYDROGENASE CYTO   ( 127)  102   30     2.1
SW:ABGT_ECOLI P46133 AMINOBENZOYL-GLUTAMATE TRANS   ( 510)  108   32     2.5
SW:Y346_MYCTU O06297 HYPOTHETICAL 52.2 KDA TRANSP   ( 487)  106   31     3.2
SW:MVIN_BORBU O51750 VIRULENCE FACTOR MVIN HOMOLO   ( 512)  106   31     3.4
SW:CIN2_RAT P04775 SODIUM CHANNEL PROTEIN, BRAIN    (2005)  113   33     3.4
SW:SL54_HUMAN Q9NY91 LOW AFFINITY SODIUM-GLUCOSE    ( 659)  107   32     3.5
SW:NTCP_MOUSE O08705 SODIUM/BILE ACID COTRANSPORT   ( 362)  102   30     4.5
SW:WECF_ECOLI P27835 PROBABLE 4-ALPHA-L-FUCOSYLTR   ( 450)  103   31     4.6
SW:CIN2_HUMAN Q99250 SODIUM CHANNEL PROTEIN, BRAI   (2005)  110   33     5.2
SW:NU4M_ASTPE P11992 NADH-UBIQUINONE OXIDOREDUCTA   ( 460)  102   31     5.4
SW:NU6M_ALBCO P48922 NADH-UBIQUINONE OXIDOREDUCTA   ( 155)   95   29     6.5
SW:COX2_MYCTU Q10375 PROBABLE CYTOCHROME C OXIDAS   ( 363)   99   30     6.9
SW:NFRB_ECOLI P31599 BACTERIOPHAGE N4 ADSORPTION    ( 745)  101   31     8.8
SW:YM16_RECAM O21266 HYPOTHETICAL PROTEIN YMF16.    ( 260)   95   29     9.4
```

Figure 6.7: Sorce: [28]. A sample FastA output: alignment scores. Column 1-3 detail the name and annotation of the record. Columns 4-7 are the FastA scores.

```
>>SW:GRBB_BACSU P39570 SPORE GERMINATION PROTEIN B2.      (368 aa)
 initn:  31 init1:  31 opt: 173  Z-score: 200.6  bits: 44.7 E(): 0.00024
Smith-Waterman score: 173;  25.143% identity (26.667% ungapped) in 175 aa

                 10        20        30        40        50
EMBOSS   AKQMVNFFQIALVLIGSTGIINHVIIIPMLLDHSGRDS-WISIIILSLVYIIWIPCVF
                :. .:..:.:: :   :. .:    ..:  : :. :.. ....:: .   .
SW:GRB MRKSEHKLTFMQTLIMISSTLIGAGVLTLPRSAAETGSPSGWLMILLQGVIFIIIVLLFL
               10        20        30        40        50        60

        60        70        80        90       100         110
EMBOSS IVHKYTREEHLFSWLMRNYGGFITYPLLSIIVLYLIILGTVTLKETLT--FFSFYLPETP
          . .  . ::.      : :: . :   : ::.:  : : ..  .     .:.: ..
SW:GRB PFLQKNSGKTLFKLNSIVAGKFIGFLLNLYICLYFI--GIVCFQARILGEVVGFFLLKNT
               70        80        90         100       110
```

Figure 6.8: Sorce: [28]. A sample FastA output: alignment of the query sequence against the result sequences.

```
gtt1_drome.aa: 209 aa
 >gi|121694|sp|P20432|GTT1_DROME GLUTATHIONE S-TRANSFERASE 1-1 (CLASS-THETA)
 vs  NBRF Annotated Protein Database (rel 56) library
searching /seqlib/lib/pir1.seq 5 library

       opt      E()
< 20    13      0:=
  22     0      0:          one = represents 22 library sequences
  24     0      0:
  26     0      0:
  28     1      3:*
  30    11     19:*
  32    46     75:===*
  34   242    204:=========*=
  36   493    419:===================*===
  38   788    692:===============================*====
  40  1055    965:===========================================*====
  42  1275   1180:=====================================================*====
  44  1299   1302:===========================================================*
  46  1251   1326:=========================================================  *
  48  1186   1269:======================================================   *
  50  1077   1158:=================================================   *
  52   907   1018:==========================================    *
  54   849    870:=======================================*
  56   714    727:=================================*
  58   570    596:========================== *
  60   456    483:=====================*
  62   393    387:=================*
  64   313    308:=============*=
  66   268    243:===========*=
  68   219    192:========*=
  70   191    150:======*==
  72   127    117:=====*
  74    93     91:====*
  76    91     71:===*=
  78    44     55:==*
  80    33     43:=*
  82    22     33:=*
  84    32     26:=*
  86    19     20:*
  88    19     16:*          inset = represents 1 library sequences
  90     8     12:*
  92     8      9:*         :========*
  94     5      7:*         :===== *
  96     2      6:*         :==   *
  98     3      4:*         :===*
 100     1      3:*         := *
 102     3      3:*         :==*
 104     0      2:*         : *
 106     1      2:*         :=*
 108     0      1:*         :*
 110     0      1:*         :*
 112     0      1:*         :*
 114     0      1:*         :*
 116     0      0:          *
 118     1      0:=         *=
>120     7      0:=         *=======
```

Figure 6.9: Source: [44]. Histogram of FASTA3 similarity scores - Results of search of a *Drosophila* class-theta glutathione transferase against the annotated PIR1 protein sequence database. The initial histogram output is shown. The shaded section indicates the region that is most likely to show discrepancies between observed and expected number of scores when the statistical model fails.

6.4.2 BLAST - Basic Local Alignment Search Tool

Blast programs use a heuristic search algorithm. The programs use the statistical methods of Karlin and Altschul [2]. 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. It is possible to use one's private database with BLAST. To this it is required to convert it to the BLAST format. Direct pointer: The BLAST at NCBI [39]. BLAST can also be run through one of the retrieval systems (recommended). For example: GeneWeb mirror site at the Weizmann Institute [16].

BLAST is a family of programs. Table 6.4 details the BLAST variants and their use.

Goal/Question	**Database**	**BLAST Program**
Is the query sequence represented in the database?	Choose a current nucleic acid database. Select from among organism-specific (e.g.: yeast), inclusive (e.g., nonredundant), or specialized set (e.g., dbEST, dbSTS, GSS, HTG) databases.	blastn.
Are there homologs or evolutionary relatives of the query sequence in the database? Are there proteins whose function is related to the query sequence?	Choose a protein database if the query is protein or DNA expected to encode a protein because amino acid searches are more sensitive.	blastp for amino acid queries; blastx for translated nucleic acid queries. Use **t**blastn or **t**blastx for comparisons of an amino acid or translated nucleic acid query versus a translated nucleic acid database.

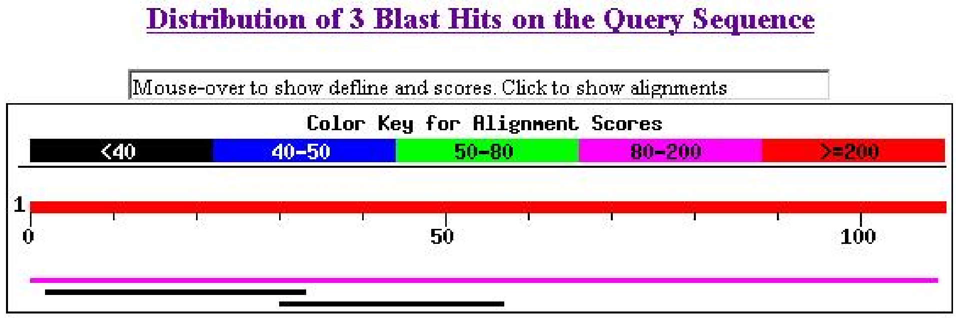
Table 6.4: Source: [40]. Variants of BLAST.

The BLAST 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. BLAST finds similar words between the query and each database sequence. It then extends such words to obtain *high-scoring sequence pairs (HSPs)* (BLAST parlance for local ungapped alignments). BLAST calculates statistics analytically, are calculated statistically like in FastA.

The BLAST graphical output is similar to FastA output. A sample output screen is shown in figure 6.10.

6.4.3 The Smith-Waterman Tool

Smith-Waterman (SW) searching method compares the query to each sequence in the database. SW uses the full Smith-Waterman algorithm for pairwise comparisons [7]. It also uses search




```
                                                                 Score    E
Sequences producing significant alignments:                      (bits)  Value

gi|17380160|sp|P13423|PAG_BACAN  Protective antigen precurso...   191   2e-49
gi|14285548|sp|Q9K1F5|LPXB_NEIMB  LIPID-A-DISACCHARIDE SYNTHASE    27   4.7
gi|11131224|sp|P57155|ARGE_BUCAI  ACETYLORNITHINE DEACETYLAS...    26   9.3

                                 Alignments

>gi|17380160|sp|P13423|PAG_BACAN Protective antigen precursor (PA) (PA-83) (PA83)
           translocating protein) [Contains: PA-20 (PA20); PA-63
           (PA63)]
          Length = 764

 Score =  191 bits (484), Expect = 2e-49
 Identities = 102/120 (85%), Positives = 104/120 (86%), Gaps = 10/120 (8%)

Query: 1   MKKRKVLIPLLALSTILVSTTGNLEVIQAEVKQENRLLNXXXXXXQGLLGYYFSDLNFQA 60
            MKKRKVLIPL+ALSTILVS+TGNLEVIQAEVKQENRLLN      QGLLGYYFSDLNFQA
Sbjct: 1   MKKRKVLIPLMALSTILVSSTGNLEVIQAEVKQENRLLNESESSSQGLLGYYFSDLNFQA 60
```

Figure 6.10: Sorce: [39]. A sample BLAST output screen. There are three sections: 1. A graphical representation of the alignments. 2. Scores: for each result a line containing name, annotation and BLAST scores. 3. Alignment of the query sequence against the results sequence.

results to generate statistics. Since SW searching is exhaustive, it is the slowest method. A special hardware + software (Bioccelerator) is used to accelerate the application. A Bioccelerator can be found in the TAU bio-informatics department. Direct pointer: [26]. It also can be run through the Weizmann Institute site [16].

6.4.4 Comparison of the Programs

- Concept:
  SW and BLAST produce local alignments, while FastA is a 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 homologous sequences on the average (but the opposite can also happen - if there are no identical residues conserved, but this is infrequent). It also gives better separation between true hits and random hits.
- 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.

6.4.5 Tips for DB Searches

- Use the latest database version.
- Run BLAST first, then depending on your results run a finer tool (FastA, Ssearch, SW, Blocks, etc.).
- Whenever possible, use protein or translated nucleotide sequences.
- $E < 0.05$ 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, since 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.

6.5 Significance of Scores

6.5.1 The Problem

An important question that software bioinformatics tools are trying to answer is how meaningful an alignment score is. A user may submit different queries into different databases, and it is important to find a means to estimate the how "significant" an alignment score is, regardless of the specific query or the specific database. This section will discuss the different *Statistical Enumerators* that the different tools that are used in order to estimate this significance level. Most of this section is based on an article by Pagni and Jongeneel [5].

A practical application of these statistical enumerators is setting the score threshold for the results that are displayed in sequence search engines. This threshold should include most positive results, while minimizing the number of *false positives* –alignments that are included in the list of results, but have no biological basis. The easy case is when the distribution of the scores for true alignments is very different from the distribution of the scores for alignments of random sequences (figure 1). A more complex case is when the true alignment score distribution and the random alignment score distribution share a common area along the score axis (figure 2). In this case it is hard to distinguish real alignments from random alignments. In this case, a means of determining the confidence level of the score is crucial.

In applications such as profile building or PSI-BLAST, the determination of accurate confidence scores is crucial. These applications make automated iterative use of results, in order to generate more results. This makes errors, such as false positives, disastrous for those algorithms.

6.5.2 Statistical Estimators

This section will define the different types of statistical estimators used in the analysis of the validity of an alignment score.

Z-score

The *Z-score* is an old, yet commonly used statistical estimator for the validity of statistical results, including alignment scores. It is defined by the number of standard deviations that separate an observed score from the average random score. In other words, it is the difference between the observed score and the average random score, normalized by the standard deviation of the distribution. A higher Z-score means that the score can be trusted with a higher confidence level.

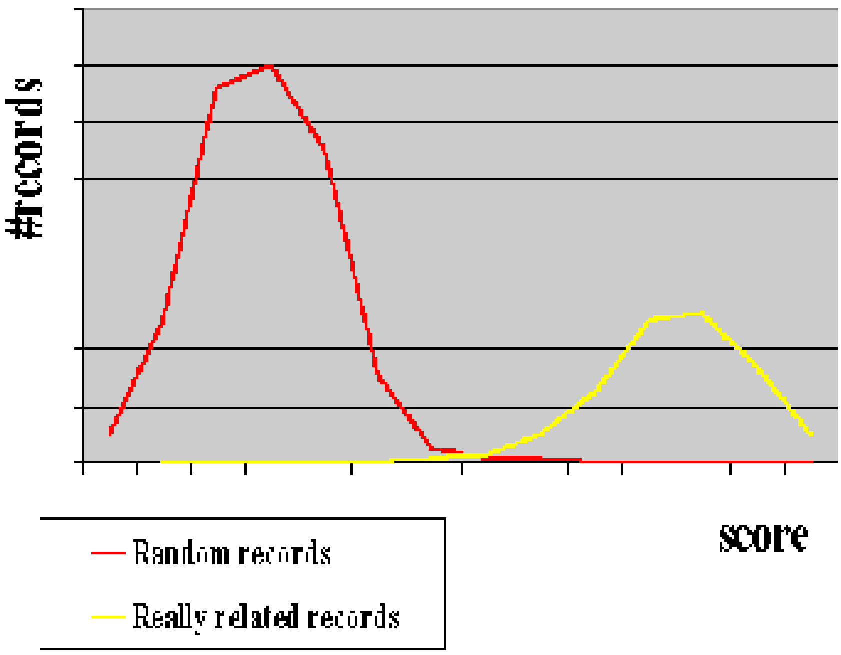


Figure 6.11: Easy Case: Illustration of an easy case of estimating significance. The score of really related records are distributed away from random records and thus can easily identified.

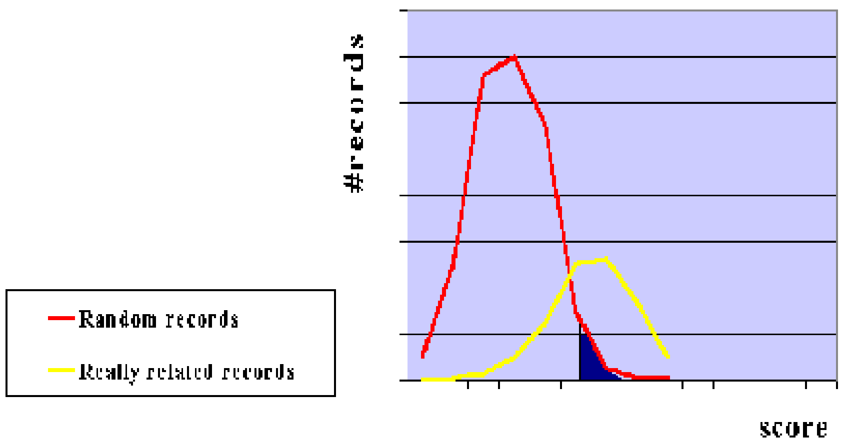


Figure 6.12: Complex Case: Illustration of a complex case of estimating significance. The dark area represents the number of random records (shuffeled query sequnce) that exceed the query score. In this case the common area betweeb the random plot and the real plot is large, which makes it hard to distinguish between the real and random ones.

E-value

The *E-value* is the most frequently used statistical estimator for the validity of alignment scores. It is defined as the expected number of false positives with a score higher than the observed score. This value is dependant, obviously, on the number of random alignments, determined by the size of the aligned sequences. A lower E-value indicates that the score has a higher confidence level.

P-value

Once we have calculated the E-value, E, for a certain score, we can go one step further. The *P-value* is the probability of the observed score – the probability that a certain score occurred by chance. To find a formula for the P-value, let us define a random variable Y_E as the number of random records achieving an E-value of E or better. This random variable has a Poisson distribution with the parameter λ=E. The probability that no random events have a lower score then our score, i.e. that $Y_E = 0$, decreases exponentially with our score - s. Therefore, that probability that at least one random record achieved a better score then our E-value can be computed using the following simple formula [1]:

$$P = 1 - e^{-E}$$

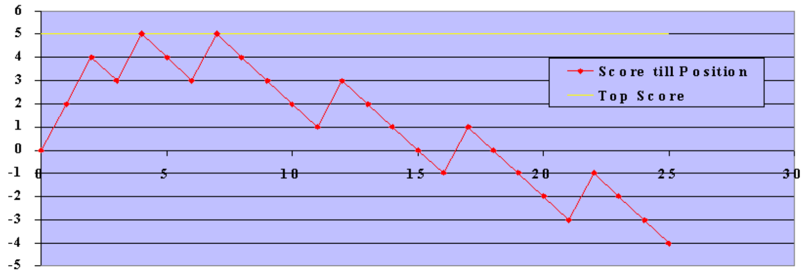
Like the E-value, this value is dependent on the size of the database. A lower P-value means that the score has a higher confidence level. This estimator is not widely used for determining the validity of sequence alignment scores.

6.5.3 A model for gap free alignments

This section will first discuss the distribution of a gap free alignment of two random sequences. Then it will introduce the extreme value distribution, an alternative model for the distribution of maximal alignment score of a query against a database. Then it will discuss the difference between this distribution and the normal distribution, and give a few notes on when each model is valid.

The gap free alignment problem

The gap free alignment process of a two short random sequences can be described as a random walk (figure 3). A positive score is given for each match, and a negative score is given for each mismatch. We assume in this model that the expectation of the score is negative, or else longer random alignments would receive better scores than shorter alignments. The probability that such a random walk will achieve a score higher than a threshold x, decreases exponentially with x. Thus the maximal gap free alignment problem for two random sequences produces a negative exponential distribution.




```
accggtagagtgctgatgatctatg
|| |  |    |    |    |
acggccatgcagtgcttatgatcga
```

Figure 6.13: Random walk: The score for a match is +2 and the punishment for a mismatch is -1, As shown,the expectancy for the whole walk is negative. The probability that the Top Score will be larger than X decreases exponentially with x.

The Extreme Value Distribution

We now attempt to predict the distribution of the local alignment scores of two long sequences. The following analysis is based on the work of Karlin and Altschul. [3] We can think of this score as a maximum of many local alignment scores of short sequences. i.e. we are looking for the distribution of the maximum of many decreasing exponential random variables.

Given a set of independent and identically distributed random variables x_1,x_2... with a distribution that decays fast for large values of the x_i's (such as the exponential distribution), the distribution of

$$X_n \equiv \max\{x_1, ..., x_n\}$$

was defined by Gambel [bib] as the *extreme value distribution*:

$$P(X_n > x) = 1 - \exp(-Ke^{-\lambda x})$$

where K and λ are parameters.

This distribution does not depend on the specifics of the distributions of the x_i's. The major feature of this density is that it is skewed. Unlike the normal distribution, which is symmetrical, this distribution has a "steeper" left tail, and a smoother "right" tail. This right tail can be estimated by a decreasing exponential. This feature is very useful for our purposes, because we are interested in large scores, and this approximation is much more easily computed in real application where response time is crucial.

6.5.4 The Gapped Alignment Problem

The common problem of gapped alignment is much more difficult than the problem of the gap free alignment. The extreme value distribution, which is an analytic model from the gap free alignment, can be used to some extent in some gapped alignments, but not in others. Unfortunately, there is no analytic model for gapped alignments, though some measurements had been conducted on random databases. This measurements lead to some interesting results. The results show that the Extreme Value distribution model can be applied to higher gap penalties, while the normal distribution is better for lower gap penalties. In fact, we witness two distinct phases in our models. Lower gap penalties correspond with the normal distribution, higher gap penalties with the Extreme Value distribution, and there are almost no gap penalties in between.

6.5.5 Models Used in Popular Tools

BLASTN

This tool simply ignores the gap penalties in its E-value estimation. This allows it to rely heavily on the analytic model of the extreme value distribution. The parameters of the distribution can be calculated analytically from the problem parameters. : Specifically, the length of the sequences and the similarity matrix. After finding the distribution, BLASTN calculates the bit score:

$$B = \frac{\lambda S - \ln K}{\ln 2}$$

This score, unlike the raw score, is measured in standard units, and is independent of the distribution, and thus it is more instructive. Clearly, it is linear to the raw score. Since the E-scores are a decreasing exponential to the raw scores, the E-score is derived from the following approximation, taking into consideration the length of the aligned sequences:

$$E = mn2^{-B}$$

where m is the database length, n is the query length.

FASTA

Unlike BLASTN, which uses a statistical model with an extensive theory behind it, FASTA attempts to give good estimates for e-values using values of tens of thousands of random sequences aligned during the course of the algorithm. The FASTA algorithm uses the following steps for the estimation:

- Random alignment scores are collected through the course of the algorithm. This is possible because FASTA has a heuristic, which produces alignment scores for pairs of sequences very quickly. FASTA assumes that the searched database is large enough, so a heterogeneous sample of scores is collected.

- Scores are assigned into bins of a histogram based on the length of the sequence matched. The best scores are removed from each bin, so that possible “positive” scores will not be taken into account, since we are interested in finding the number of expected false positives above our score.

- The expected value of a random alignment against the database is calculated. The expected value is the result of a linear regression of the data against the logarithm of the length of the sequence.

- For each score alignment score for which FASTA needs an E-value, FASTA first calculates the Z-value. This is done using the standard deviation of the random scores from the expected value of a random alignment with the same length of the analyzed alignment.

- The conversion of the Z-value into E-value follows the assumption that the distribution of the random scores is an Extreme Value Distribution. To get the E-value, FASTA multiplies the number of sequences in the database by the probability that such sequence will have a value higher then our score. This probability can be directly calculated from the number of standard deviations separating our score from the average score.

6.6 Multiple sequence alignments

6.6.1 Introduction

Sometimes it is necessary to align a number of sequences, in order to identify regions of homology between them. From the Biologist point of view, this alignment is an important tool in characterizing protein families, determining consensus sequences, finding secondary and tertiary structure of new sequences, and construction of phylogenetic trees according to the similarity level between within the aligned sequences.

There are two different approaches to multiple sequence alignment. The first – alignment of similar sequences of nucleotides or amino acids, taking into account physio-chemical properties and mutation data. The Second – alignment of sequences solely according to secondary and tertiary structure. The resulting alignment can, understandably, differ greatly between the two approaches.

6.6.2 Multiple Sequence Alignment Tools

ClustalW [27] is popular software for multiple sequence alignment. It can be used on either DNA or proteins. The output of ClustalW is a multiple alignment, shown graphically, and it can even construct phylogenetic trees according to the alignment. ClustalW creates alignments in a format called GCG. However, most alignment software and viewers uses another format, the Fasta format. The program tofasta [10] converts files from GCG format to fasta format.

The program JalView [15] is an excellent multiple alignment viewer. It uses colors to distinguish regions of homology from regions that are less similar.

6.7 Secondary Databases

There are various databases containing secondary structure information. Each has its advantages and disadvantages, so it is advisable to try more than one database when searching. This section will show some popular databases.

6.7.1 Prosite

The Prosite database [37] is based on SwissPort and thus is very well annotated, but small. Characterization of protein families is done by the single most conserved motif observed in a multiple sequence alignment of known homologous. These conserved motifs usually relate to biological functions such as active sites or binding sites. The search in Prosite does not require an exact match in structure. Prosite enables searches using complex patterns. It is possible to search textually using regular expressions for names of known proteins, etc. It is also possible to scan a protein sequence using prosite for structural pattern matches. The database is well cross-linked to SwissProt and TrEMBL.

6.7.2 FingerPrints

Unlike Prosite, FingerPrints has an improved diagnostic reliability which is achieved by using more than one conserved structural motif to characterize a protein family. With FingerPrints, many motifs are encoded using ungapped and unweighed local alignments.

The input to FingerPrints is a small multiple alignment, which has some conserved motifs. These motifs are searched for in the database, and only sequences that match all the motifs are considered for further analysis. With the new alignment, the database is searched for more sequences until no further complete fingerprint matches can be identified. These final aligned motifs constitute the refined fingerprint that is entered into the database.

6.7.3 Blocks

Blocks [11] uses multiply aligned ungapped segments corresponding to the most highly conserved regions of proteins. Block Searcher [14] ,Get Blocks [13] and Block Maker [12] are aids to 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.

6.7.4 Profiles

The Profiles databases [19] uses the notion of profiles to achieve a good detection of distant sequence relationships. A profile is a scoring table with multiple alignment information for the whole sequences, not just for conserved regions. Profiles are weighted to indicate:

- what types of residues are allowed at what positions.
- where insertions and deletions (INDELs) are allowed (not within core secondary structures).
- where the most conserved regions are located.

Profiles provide a sensitive means of detecting distant sequence relationships, where only a few residues are well conserved. The inherent complexity of profiles renders them to be highly potent discriminators.

The ISREC (Swiss Institute for Experimental Research) has created a compendium of profiles, allowing to find even distant homologous. Each of those profiles has separate data and family annotations.

6.7.5 Pfam

Pfam [46] uses a different method for its database. High quality seed alignments are used to create Hidden Markov Models to which sequences are aligned. Pfam has to classes of alignments, according to their credibility:

- Pfam-a – Non-edited seed alignments which are deemed to be accurate.
- Pfam-b – Alignments derived by automatic clustering of the SwissPort database. These alignments are, of course, less reliable.

6.7.6 eMotif

eMotif [17], also known as identify, uses data from Blocks and FingePrints to generate consensus expressions from the conserved regions of sequence alignments.
eMotif adopts a “fuzzy” algorithm which allows certain amino acid alternations. This allows eMotif to find homologous sequences that other programs can not find, but it results in a lot of noise. This trade-off shows why it is important to use multiple programs when searching for information.

6.7.7 InterrPro

InterPro [34] is an interface to several secondary databases: ProSite, prints, ProDom and Pfam. It has an intuitive interface both for text and sequence-based searches, and since it incorporates several databases, it is very recommended.

Bibliography

[1] A.Dembo and S.Karlin. Strong limit theorems of empirical functionals for large excedances of partial sums of i.i.d variables. *Annuals of Probability*, 19(4):1737–1755, 1991.

[2] 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.

[3] 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.

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

[5] Pagni M. and Jongeneel CV. Making sense of score statistics for sequence alignments. *Briefings in Bioinformatics*, 2(1):51–67, 2001.

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

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

[8] `ftp://genbank.sdsc.edu/pub/release.notes/gb107.release.notes`.

[9] `http://bioinfo.md.huji.ac.il/databases/genpept.html`.

[10] `http://bioinfo.tau.ac.il/GCG/html/unix/tofasta.html`.

[11] `http://blocks.fhcrc.org/`.

[12] `http://blocks.fhcrc.org/blockmkr/make_blocks.html`.

[13] `http://blocks.fhcrc.org/blocks-bin/getblock.sh`.

[14] http://blocks.fhcrc.org/blocks/blocks_search.html.

[15] http://circinus.ebi.ac.uk:6543/michele/jalview/help.html.

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

[17] http://dna.Stanford.EDU/identify.

[18] http://flybase.bio.indiana.edu/.

[19] http://isrec.isb-sib.ch/software/PFSCAN_for_m.html.

[20] http://pir.georgetown.edu/.

[21] http://pir.georgetown.edu/pirwww/aboutpir/collaborate.html.

[22] http://pir.georgetown.edu/pirwww/dbinfo/nrl3d.html.

[23] http://pir.georgetown.edu/pirwww/dbinfo/sample-hahu.html.

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

[25] http://srs.ebi.ac.uk/srs6/man/mi_srswww.html.

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

[27] http://www2.ebi.ac.uk/clustalw.

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

[29] http://www.dna.affrc.go.jp/htdocs/growth/index.html.

[30] http://www.ebi.ac.uk/.

[31] http://www.ebi.ac.uk/embl/.

[32] http://www.ebi.ac.uk/fasta33/fasta3x.txt.

[33] http://www.ebi.ac.uk/fasta3/help.html.

[34] http://www.ebi.ac.uk/interpro.

[35] http://www.ebi.ac.uk/swissprot/.

[36] http://www.ebi.ac.uk/swissprot/Information/information.html.

[37] http://www.expasy.ch/prosite.

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

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

[40] http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/guide.html.

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

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

[43] http://www.ncbi.nlm.nih.gov/Genbank/GenbankOverview.html.

[44] http://www.people.virginia.edu/~wrp/papers/mmol98f.pdf.

[45] http://www.rcsb.org/pdb/.

[46] http://www.sanger.ac.uk/Software/Pfam.