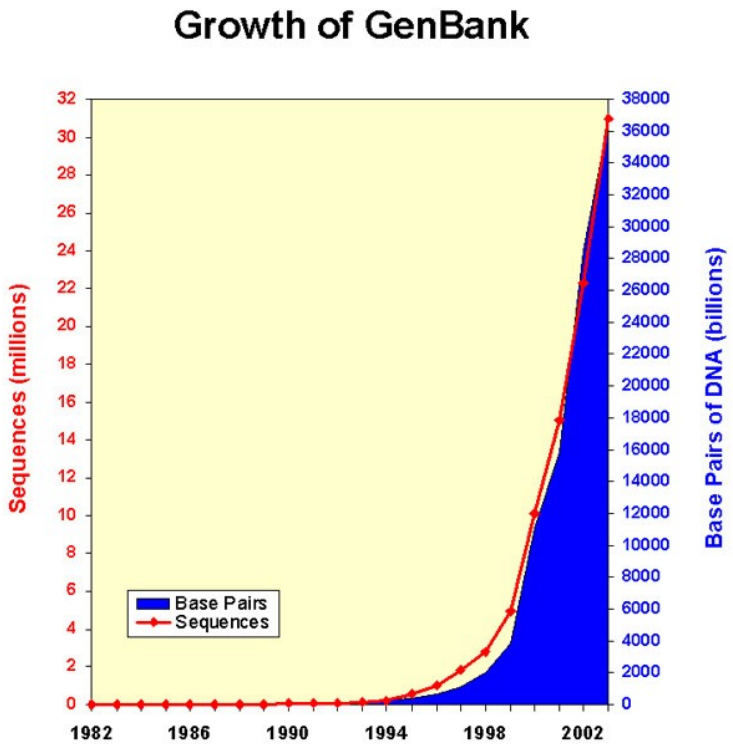
Sequence Databases

Sequence databases are experiencing explosive growth both in size and number. There are some genomes that have been completely sequenced, like Human, mice(mus musculus), e.coli, Arabidopsis Thaliana etc. There are also other sequencing projects which target to sequence complete genomes in the near future. Apart from these, there is a lot of sequencing data available from other species. In fact any DNA/protein sequence that is sequenced any where in the world will be entered into a database. The following are a few of the major sequence databases.

Genbank

Genbank is 'an annotated collection of all publicly available DNA sequences' [NIH]. Genbank is maintained by the NIH(National Institute of Health). Genbank is part of bigger effort called the 'International Nucleotide sequence Database Collaboration', which comprises of DDBJ(DNA Database of Japan), and the EMBL(European Molecular Biology Laboratory) apart from Genbank.

As of April 2004, Genbank consists of approximately 38 billion bases in 32 million sequences [NIH]. The following graph, provided by the NCBI, shows the growth of Genbank over the years.



Genbank is publicly available for searching using a variety of tools that will be discussed in the next section.

Databases that are more specific than Genbank will be necessary if somebody wants to narrow their search to a specific type of DNA sequences. Consequently, there are numerous specialized databases available. Some of the most important ones include:

dbEST: This is a database of the all ESTs(Expressed Sequence Tags). ESTs are short(300 – 500 bp) cDNA sequences that are derived the from the ends of mRNA sequences. ESTs provide valuable information about the coding sequences of a genome, and help in locating and identifying genes.

dbSTS: This is a database of all Sequence Tagged Sites. These are marker sequences in the genome that are helpful in sequencing and sequence assembly.

dbGSS: This database consists mainly of Genomic Survey Sequences, which are obtained by directly sequencing, or 'reading from' the genome. These may also contain sequences derived from mRNA, like the dbEST databases.

dbSNP: A database of all SNPs(Single Nucleotide Polymorphisms) in the human genome. A site in the human genome is termed as a SNP if at least 10% of the population have a mutated base at that site that is different from the rest of the population. For example, if 80% of the population have an 'A' at a particular site, and 20% have 'C', that site called a SNP, and 'A' and 'C' are called the two alleles for that SNP. Though most of the SNPs are bi-allelic, there can be some multi-allelic SNPs. SNPs are very common in the human genome: it is estimated that on average there is a SNP for every 600 nucleotides. SNPs are of great interest to biologists, as they are believed to be the causes for many diseases. Some SNPs have already been linked to diseases (for eg: breast cancer).

UniGene : 'UniGene is an experimental system for automatically partitioning GenBank sequences into a non-redundant set of gene-oriented clusters. Each UniGene cluster contains sequences that represent a unique gene, as well as related information such as the tissue types in which the gene has been expressed and map location.' - NCBI

Protein (amino acid) sequence databases:

PDB, Swissprot, PIR and PRF are some of the major databases out of the many protein databases available.

PDB (Protein Data Bank): Available at http://www.rcsb.org/pdb/
According to the website, PDB is the "single worldwide repository for the processing and distribution of 3-D structure data of large molecules of proteins and nucleic acids".

However, the most useful protein database, as far as sequence comparison and alignment are concerned, is the nr database, distributed by NCBI. This is the Database of all non-redundant amino acid sequences.

Structure databases, conserved domain databases

Entrez:
Entrez is the text based search and retrieval system provided by NCBI. Entrez is a central search engine that can be used to search all databases that are maintained by NCBI, which include journal articles, nucleotide and protein sequences, complete genomes, structure databases, Taxonomy, and others. Entrez supports all kinds of practical searches on these databases. Using entrez, one can retrieve all the sequences that belong to a particular organism, or retrieve a sequence through its accession number or global identifier, or even retrieve all the sequences that were submitted by a particular lab or person.

Entrez is available at http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi

PubMed:

PubMed is a database of over 14 million citations for biomedical articles. Pubmed contains citations to almost all the articles published in life sciences journals. Pubmed is part of the Entrez retrieval system

Sequence Alignment Tools for Biologists

BLAST

BLAST (Basic Local Alignment Search Tool) is probably the most popular sequence alignment tool used by biologists. The popularity of BLAST is mostly due to its speed. BLAST is a collection of programs, each targeted for a different kind of problem.

BLAST is a heuristic approach that concentrates on finding regions of high local similarity in alignments. i.e, it tries to find sequences in a given database that are highly similar to substrings of the input query. Blast uses an alphabet-scoring matrix (which generally is an identity matrix incase of DNA sequences) for computing the scores for alignments. The following terms will be necessary in explaining how BLAST works:

Segment Pair: Given two strings S_1 and S_2, a *segment pair* is a pair of equal-length substrings of S_1 and S_2, aligned without spaces.

Locally maximal segment pair: A segment pair whose alignment score decreases when the alignment is extended either to the right or to the left.

Maximal segment pair (MSP): A segment pair with the maximum score over all the segment pairs in S_1, S_2.

Given a fixed query sequence *P*, BLAST attempts to find all the database sequences that have an MSP with score greater than a certain cutoff with *P*.

BLAST functions based on the 'hot-spot' strategy. It finds positions of sequences in the database that are highly similar (exact-matches, in case of DNA sequences) to substrings in the input query sequence. These positions are called the 'hot-spots'. It then extends these hot-spots into locally maximal segment pairs. It reports all the locally maximal segment pairs that have an alignment score greater than a given cut-off value.

BLAST starts by enumerating all the *w*-length strings, or *w*-grams of the input query sequence P. For each such *w*-gram, it enumerates a list of all possible *w*-grams that result in an alignment score greater than a certain threshold *t*. It then finds occurrences of this list of *w*-grams in the database. The speed of BLAST is due to the fact that the database can be indexed based on these *w*-grams, and hence the hot-spots can be easily found. Therefore, once a value of *w* that is optimal for the given database is determined (from empirical results), the search will be quickly narrowed down to a relatively few hot-spots, or 'hits'.

The different BLAST programs:

blastn: blastn is for comparing DNA sequences against a database of DNA sequences.

blastp: blastp is for comparing protein sequences against a database of protein sequences.

blastx: blastx is for comparing DNA sequences against a database of protein sequences. blastx translates the given DNA sequence into protein sequences in all six reading frames(three in each direction) using the genetic code. It then basically runs a blastp on all the six amino acid sequences.

tblastn: tblastn does exactly the opposite of blastx – it takes an amino acid sequence as the query and searches a database of DNA sequences. It creates the translated versions of the DNA databases (again, in all six reading frames), and essentially calls blastp on these translated databases.

tblastx: searches for a translated query in a translated database.

Blast Output

The output of the BLAST programs typically has four fields: The first two fields give information about the matching sequence. The first gives the global identifier for the matching sequence, followed by the accession number of the sequence in the specific database from which the sequence has been obtained. The second filed gives a brief text description/name of the DNA/protein sequence.

The next two fields give information about the quality of the match itself. The third field gives the bit score of the match. The bit score is a measure of the statistical significance of the alignment. The bit score is independent of the size/nature of the database. Higher bit scores indicate better matches.

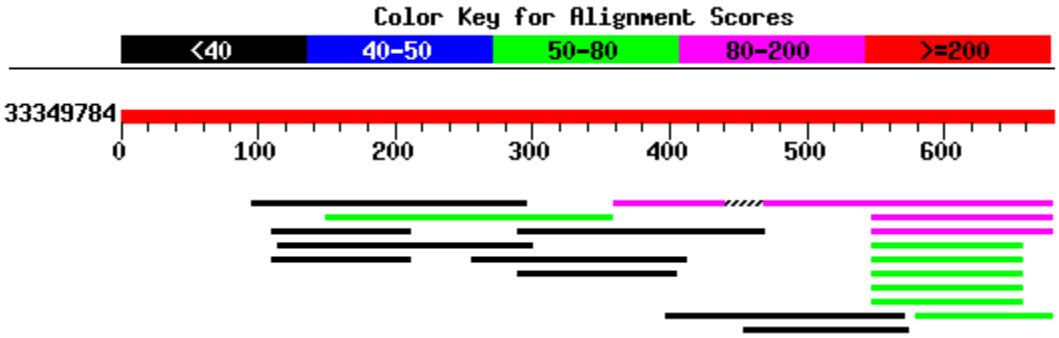
The e-value (expectation value) gives an estimation of the number of times you can expect a match with this score in the given database, just by chance. E-value takes into account the size and other details about the database. The smaller the e-value, the better is the match. In general, e-values larger than 10^{-4} are not considered to be significant.

Apart from listing the matches, BLAST also provides a graphical display of where the hits are located on the query sequence. The hits are color-coded according to their alignment scores. The following shows a sample output of blast:

```
RID: 1088704329-28478-190589094837.BLASTQ4

Query= gi|33349784|gb|CC882883.1|CC882883 02F6160-12B1-H12
UniformMu MuTAIL Library Zea mays genomic clone 02F0616-12B1-H12.
         (679 letters)


Database: All non-redundant GenBank CDS
translations+PDB+SwissProt+PIR+PRF excluding environmental samples
           1,881,758 sequences; 624,507,535 total letters


If you have any problems or questions with the results of this search
please refer to the BLAST FAQs

Taxonomy reports
```

Distribution of 24 Blast Hits on the Query Sequence

Mouse-over to show defline and scores. Click to show alignments



Sequences producing significant alignments:		Score (bits)	E Value
gi|13518388|ref|NP_084747.1|	hypothetical protein [Oenother...	119	3e-34
gi|11497575|ref|NP_054982.1|	hypothetical protein SpolCp078...	65	9e-20
gi|11467247|ref|NP_043079.1|	hypothetical protein [Zea mays...	94	2e-18
gi|48478723|ref|YP_024330.1|	hypothetical protein 133 [Sacc...	94	2e-18
gi|37533280|ref|NP_920942.1|	hypothetical protein ORF133 fr...	78	1e-13
gi|40253564|dbj|BAD05510.1|	ORF133; ORF within trnI intron ...	78	1e-13
gi|11466839|ref|NP_039436.1|	ORF133; ORF within trnI intron...	78	1e-13
gi|38346668|emb|CAE54562.1|	OSJNBa0079C19.21 [Oryza sativa ...	78	1e-13
gi|46805100|dbj|BAD17339.1|	rice chloroplast ORF133; ORF wi...	77	4e-13
gi|76358|pir||QIZMI	hypothetical protein C-123 - maize chlo...	65	2e-11
gi|7524727|ref|NP_042481.1|	ORF40e [Pinus thunbergii] >gi|7...	32	0.013
gi|29565684|ref|NP_817266.1|	ORF68b [Pinus koraiensis] >gi|...	40	0.053
gi|38347122|emb|CAE05099.2|	OSJNBa0009K15.19 [Oryza sativa ...	39	0.070
gi|15219789|ref|NP_176868.1|	lupeol synthase, putative / 2,...	34	2.2
gi|47196787|emb|CAF88687.1|	unnamed protein product [Tetrao...	33	3.8
gi|29833909|ref|NP_828543.1|	putative cytochrome C3-like hy...	33	5.0
gi|38637730|ref|NP_942704.1|	putative [NiFe] hydrogenase la...	33	6.5
gi|12658323|gb|AAK01090.1|	putative BSC1 sodium channel pro...	32	8.5
gi|46316081|ref|ZP_00216661.1|	COG3662: Uncharacterized pro...	32	8.5

PSI-BLAST: Position Specific Iterated BLAST

PSI-BLAST is a program designed to detect families of proteins, or to find a family to which a particular protein belongs to. The first iteration of PSI-BLAST starts as a normal blastp search using the BLOSUM62 alignment matrix. However, in the results page, there will be a check box against each matching amino acid sequence. The user can select the sequences (based annotation) that he thinks are interesting matches. PSI BLAST then computes a new position-specific alignment matrix based on the alignment between the selected sequences. It then searches the whole database again, based on the new position-specific alignment matrix. This process is repeated again and again, until there are no new matches or until the user is satisfied with the results.

FASTA

FASTA is another heuristic-based local alignment search tool that is in general slower but more accurate when compared to BLAST. The following is brief description of how FASTA works, summarized from [Gusfield, pp. 376-378].

FASTA requires the user select a value for a parameter called the '*ktup*', which determines the length of the 'hot-spots' to be found in the first step. FASTA finds pairs (*i*,*j*) such that *ktup*-length substring starting at position *i* in the query exactly matches *ktup*-length substring at position *j* in the database. Such pairs are called 'hot-spots'. The standard recommended values of *ktup* are six for DNA sequences and two for protein sequences. The hot-spots can be found efficiently by hashing *ktup*-tuples of the query and/or the database to a hash-table.

Each such hot-spot(*i*,*j*) can be considered as a *ktup*-length interval in diagonal (*i*-*j*) of the full dynamic programming table. A *diagonal run* a set of consecutive hot-spots on the same diagonal. The *score* for a diagonal-run is determined by giving a positive score for a hotspot and a negative score for a gap, the negative score being proportional to the length of the gap. The total score of the diagonal-run is the sum of the hot-spot score and the interspot scores. In the first step, FASTA finds the ten highest scoring diagonal runs using this scheme.

In step two, FASTA builds a dynamic programming table for each of the top 10 diagonal runs, using an alignment matrix like PAM or BLOSUM. Using the DP-tables, it finds and reports the single best sub-alignment in the top-ten diagonal runs. This single-best sub-alignment is called *init1*. Note that *init1* does not allow any spaces(gaps).

In step three, FASTA finds high-scoring alignments by combining sub-alignments that have a score greater than a given cut-off, allowing some spaces between sub-alignments this time. The bets such alignment is called *initn*.

In step four, FASTA returns to *init1*, forms a band of 16 or 32 diagonals around the diagonal containing *init1*, and computes the optimal local alignment in the subtable restricted to those 16 or 32 diagonals. The output of this step is reported as *opt*.

Finally, FASTA ranks sequences in the database by their *init1*, *initn* and *opt* scores, and reports the results.

Alignment Matrices:

PAM (Point Accepted Mutaion or Percent Accepted Mutation) is a unit of measure of the evolutionary divergence between two amino acid sequences.

"Two sequences S_1 and S_2 are defined as being one PAM unit diverged if a series of accepted point mutations has converted S_1 to S_2 with an average of one accepted point-mutation event per one-hundred amino acids" [Gusfield, p.381] . i.e, if there is one beneficial or non-lethal mutation for every hundred amino acids, then two amino acid sequences are one defined to be one PAM unit diverged.

A single position can undergo more than single mutation. Therfore 'one PAM unit' of divergence does not mean one percent sequence difference between the two sequences. To give an example, to say that two sequences are 100 PAM units diverged does not mean that they differ in *every* position. In general, amino acids that have diverged by 200 PAM units are expected to be identical in 25% of their positions, and sequences that are 250 PAM units diverged are expected to distinguishable from a pair of random sequences [Gusfield, p.382].

PAM matrices are to be used to compare two sequences that are a specific number of PAM units diverged. They encode the expected evolutionary change at the amino acid level. The (*i*,*j*) entry in the PAM *n* matrix reflects the likelihood of amino acid A_i being replaced by A_i in two sequences that are *n* PAM units diverged.

Theoretically, PAM matrices can be constructed by taking distinct pairs of homologous sequences that are known to be *n* PAM units apart. Each pair is aligned, and for each amino acid pair (A_i, A_j), we can calculate the frequency *f*(*i*,*j*), which is the number of times the amino acid A_i aligns opposite the amino acid A_j. The frequencies of the amino acids A_i and A_j, denoted by f_i and f_j, can also be enumerated. Then, the (*i*,*j*) entry for the ideal PAM *n* matrix will be:

$$\log \frac{f(i, j)}{f(i) f(j)}$$

Practically, however, the PAM matrices can not be obtained as described above. Due insertions and deletions, proper correspondence between the positions cannot be determined, especially incase of sequences that are many PAM units diverged. Besides, it is not easy to predict the exact number of PAM units by which the two sequences are diverged.

According to [Gusfield, p.383], Pam matrices are generally built by taking highly similar sequences that are expected to be only a few PAM units apart. Insertions/deletions are very few in case of such sequences, and even when they do occur, they can be detected very easily. The higher PAM matrices are then built by multiplying the lower PAM matrices with each other, or by themselves. If $M_n(i,j)$ is a PAM *n* matrix and $M_m(i,j)$ is a PAM *m* matrix, a PAM *mn* matrix is obtained by:

$$\log \frac{f(i) M_n(i, j) M_m(i, j)}{f(i) f(j)}$$

PROSITE, BLOCKS and BLOSSUM

PROSITE (http://au.expasy.org/prosite/) is a database of biologically significant patterns in proteins. According to the PROSITE website, PROSITE currently contains patterns and profiles specific to more than one thousand protein families. There are two kinds of entries in PROSITE: domain *signature*s, and domain *profile*s. The domain signatures in

PROSITE are written as regular expressions. A sample signature entry in PROSITE will look like:

W-x(9,11)-[VFY]-[FYW]-x(6,7)-[GSTNE]-[GSTQCR]-[FYW]-x(2)-P

The above entry means that any occurrence of the pattern should start with a W, anywhere between 9 and 11 amino acids of any type, followed by any one amino acid in [VFY], followed by any one amino acid in [FYW], etc.

Domain profiles are derived from a multiple alignment of family members of a protein family. Profiles are used when effective domain signatures can not be derived for a family.

PROSITE also provides tools to scan a given amino acid sequence in order to check if it matches any of the domains in the database.

BLOCKS (http://www.blocks.fhcrc.org) is a protein motif-database derived some what indirectly from the PROSITE database. The motifs in the BLOCKS database are called *blocks*. A block is a short contiguous interval in a multiple sequence alignment of amino acid sequences[Gusfield, p.386]. Therefore, each block in the BLOCKS matrix consists of multiple rows.

The blocks in the BLOCKS database are derived automatically, by looking for the most highly conserved regions in groups of proteins documented in the PROSITE database. The PROSITE patterns themselves are not directly used to make the BLOCKS database. The PROSITE pattern of a group may or may not be part of the blocks belonging to that group in the BLOCKS database The patterns in the PROSITE database always have biological significance. They are always derived based on the known functions or known structures of the proteins. However, a block in the BLOCKS database need not necessarily have a biological meaning. All the blocks are calibrated against the SWISSPROT database. It is these calibrated motifs that make up the BLOCKS database.

BLIMPS (Blocks IMProved Searcher) is a program to that can be used to search a protein sequence against the BLOCKS database.

BLOSUM Matrices

The BLOSUM (BLOcks SUbstitution Matrix) amino acid substitution matrices are derived from the BLOCKS database. The calculations used to compute the BLOSUM matrices are explained in [Gusfield, p.356-387]. To avoid situations in which two highly similar rows in a block have a strong impact of the scores of the amino acid substitution matrix, one of the rows in a pair of highly similar rows are removed from a block. Therefore, a BLOSUM x matrix is matrix derived from blacks in which no two pairs are more than x percent similar. BLOSUM 62 matrix is considered to be one of the most effective matrices.