CHAPTER 7

Database Searching for Similar Sequences

INTRODUCTION

DATABASE SIMILARITY SEARCHES have become a mainstay of bioinformatics. Large sequencing projects in which all the genomic DNA sequence of an organism is obtained have become quite commonplace. The genomes of a number of model organisms have been sequenced, including the budding yeast *Saccharomyces cerevisiae*, the bacterium *Escherichia coli*, the worm *Caenorhabditis elegans*, the fruit fly *Drosophila melanogaster*, and the human species *Homo sapiens*. These species have also been subjected to intense biological analysis to discover the functions of the genes and encoded proteins. Thus, there is a good deal of information available as to the biological function of particular sequences in model organisms that may be exploited to predict the function of similar genes in other organisms. In addition to genomic DNA sequences, complete cDNA copies of messenger RNAs that carry all the sequence information for the protein products have also been obtained for some of the expressed genes of various organisms. Translation of these cDNA copies provides a close-to-correct prediction of the sequence of the encoded proteins. Because obtaining intact cDNA sequences is laborious and time-consuming, a common practice is to make a library of partial cDNA sequences from the expressed genes, and then to perform high-throughput, low-accuracy sequencing of a large number of these partial sequences, known as expressed sequence tags (ESTs). The objective of an EST project is to find enough sequence of each cDNA and to have enough accuracy in the sequence that the amino acid sequence of a significant length of the encoded protein can be predicted. Overlapping ESTs can then be combined, and interesting ones can be found by database similarity searches. The full cDNA sequence of these genes of interest may then be obtained. Once all the sequence information is collected and placed in the sequence databases, the big task at hand is to search through the databases to locate similar sequences that are predicted to have a similar biological function through a close evolutionary relationship.

Sequence database searches can also be remarkably useful for finding the function of genes whose sequences have been determined in the laboratory. The sequence of the gene of interest is compared to every sequence in a sequence database, and the similar ones are identified. Alignments with the best-matching sequences are shown and scored. If a query sequence can be readily aligned to a database sequence of known function, structure, or biochemical activity, the query sequence is predicted to have the same function, structure, or biochemical activity. The strength of these predictions depends on the quality of the alignment between the sequences. As a rough rule, if more than one-half of the amino acid sequence of query and database proteins is identical in the sequence alignments, the prediction is very strong. As the degree of similarity decreases, confidence in the prediction also decreases. The programs used for these database searches provide statistical evaluations that serve as a guide for evaluation of the alignment scores.

Previous chapters have described methods for aligning sequences or for finding common patterns within sequences. The purpose of making alignments is to discover whether or not sequences are homologous or derived from a common ancestor gene. If a homology relationship can be established, the sequences are likely to have maintained the same function as they diverged from each other during evolution. If an alignment can be found that would rarely be observed between random sequences, the sequences are predicted to be related with a high degree of confidence. The presence of one or more conserved patterns in a group of sequence is also useful for establishing evolutionary and structure–function relationships among sequences.

The above methods of establishing sequence relationships have been utilized in database searches that are summarized in Table 7.1. In addition to standard searches of a sequence database with a query sequence (Table 7.1A), a matrix representation of a family of related protein sequences may be used to search a sequence database for additional proteins that are in the same family (Table 7.1B,C,D,), or a query protein sequence may be searched for the presence of sequence patterns that represent a protein family to determine whether the sequence belongs to that particular family (Table 7.1E). Genomic DNA sequences may also be searched for consensus regulatory patterns such as those representing transcription factor-binding sites, promoter recognition signals, or mRNA splicing sites; these types of searches are discussed in Chapter 8.

SEQUENCE SIMILARITY SEARCH WITH A SINGLE QUERY SEQUENCE

Searching a sequence database for sequences that are similar to a query sequence is the most common type of database similarity search. The search provides a list of database sequences with which the query sequence can be aligned. Once a list is available, additional searches may be performed using one of the initially found sequences as a query sequence. In this manner, the search may be expanded to find more distant relatives of the initial query sequence. Once a family of related sequences is found, the entire sequence may be aligned in a multiple sequence alignment, or the sequences may be analyzed for the occurrence of short regions of similarity, as described later in the chapter. Chapter 10 describes the use of those repetitive searches to identify families of paralogous proteins. Web sites and computational resources that support this type of database similarity searching are described in Table 7.2.

A common reason for performing a database search with a query sequence is to find a related gene in another organism. For a query sequence of unknown function, a matched gene may provide a clue as to function. Alternatively, a query sequence of known function (e.g., a yeast gene) may be used to search through sequences of a particular organism (e.g, a plant) to identify a gene that may have the same function. Sequences of an organism that are collected for such purposes include genomic sequences (sequences of BAC clones or the assembled sequence of an entire chromosome), EST sequences, and cDNA/protein sequences for particular genes. Database similarity searches may use one type of sequence (e.g., an EST sequence) to find matching EST sequences, genomic DNA sequences, or cDNA/protein sequences in the same organism. The Institute for Genomic Research (TIGR) has indexed a large number of EST sequences of model organisms in this manner (Table 7.2). These indexed databases may also be searched with a query sequence to identify related sequences.

ALLOWING FAST SEARCHES

When database searches were first attempted, machine size and speed were limiting factors that prevented use of a full alignment program, such as the dynamic programming algorithm, for each search. Although these considerations no longer apply due to the availability of more powerful machines, the sheer number of such searches that are presently performed on whole genomes creates a need for faster procedures. Hence, two methods that are at least 50 times faster than dynamic programming were developed. These methods follow a heuristic (tried-and-true) method that almost always works to find related sequences in a database search but does not have the underlying guarantee of an optimal solution like the dynamic programming algorithm. The first rapid search method was FASTA, which found short common patterns in the query and database sequences and

Table 7.1. *Types of database searches for proteins*

Type of search	Target database	Method	Type of query data	Examples of programs used, location (also see Tables 7.2, 7.4, 7.7, and 7.8)	Results of database search
A. Sequence similarity search with query sequence	protein sequence database (or genomic sequences[a])	search for database sequence that can be aligned with query sequence	single sequence, e.g., DAHQSNGA	FASTA (TFASTA[a]), SSEARCH http://fasta.bioch.virginia.edu/fasta/ BLASTP (TBLASTN[a]) http://www.ncbi.nlm.nih.gov/BLAST/ WU-BLAST http://blast.wustl.edu/	list of database sequences having the most significant similarity scores

B. Alignment search with profile (scoring matrix[b,d] with gap penalties)	protein sequence database	prepare profile from a multiple sequence alignment (Profilemake) and align profile with database sequence	profile representing gapped multiple sequence alignment, e.g.,
			`D-HQSNGA`
			`ESHQ-YTM`
			`EAHQSN-L`
			`EGVQSYSL`
				PROFILESEARCH
				ftp.sdsc.edu/pub/sdsc/biology
					list of database sequences that can be aligned with the profile

C. Search with position-specific scoring matrix[c,d] (PSSM) representing ungapped sequence alignment (BLOCK)	protein sequence database	prepare PSSM from ungapped region of multiple sequence alignment or search for patterns of same length in unaligned sequences,[c] then use for database search	PSSM representing ungapped alignment, e.g.,
			`DAHQSN`
			`ESHQSY`
			`EAHQSN`
			`EGVQSY`
				MAST
				http://meme.sdsc.edu/meme/website/mast.html
					list of database sequences with one or more patterns represented by PSSM but not necessarily in the same order

D. Iterative alignment search for similar sequences that starts with a query sequence, builds a gapped multiple alignment, and then uses the alignment to augment the search[d]	protein sequence database	uses initial matches to query sequence to build a type of scoring matrix and searches for additional matches to the matrix by an iterative search method[d]	builds matches to query sequence, e.g.,
			`DAHQSNGA`
			iteration 1
			`  H-SNGA`
			`EAHQSN-L`
			↓
			further iterations
				PSI-BLAST
				http://www.ncbi.nlm.nih.gov/BLAST/
					PSI-BLAST finds a set of sequences related to each other by the presence of common patterns (not every sequence may have same patterns).

E. Search query sequence for patterns representative of protein families[e]	database of patterns found in protein families	search for patterns represented by scoring matrix or hidden Markov model (profile HMM)[e]	single sequence, e.g., DAHQSNGA	PROSITE http://www.expasy.ch/prosite INTERPRO http://www.ebi.ac.uk/interpro PFAM http://www.sanger.ac.uk/Pfam CDD/IMPALA http://www.ncbi.nlm.nih.gov/Structure/cdd/cdd.shtml (also see Table 9.5)	list of sequence patterns found in query sequence

[a] Searches of this type include the use of programs that search nucleic acid databases for matches to a query protein sequence by automatically translating the nucleic acid sequences in all six possible reading frames (TFASTA, TBLASTN). These searches may be useful when only genomic sequences or partial cDNA sequences (expressed sequence tag or EST sequences) of an organism are available. Genomic sequences that encode proteins may also have been found by gene prediction programs (Chapter 8). The predicted protein is then usually entered in the protein sequence databases. Matches to these predicted proteins may be found by searches of the protein sequence databases. These gene predictions are error-prone (see Chapter 8).

[b] A multiple sequence alignment that includes gaps may be represented by a profile, a type of scoring matrix discussed in Chapter 4, page 161. The consecutive rows of the matrix represent columns of the multiple sequence alignment, and the column values represent the distribution of amino acids in each column of the alignment. The profile includes extra columns with gap opening and extension penalties. The profile is aligned to a sequence by sliding the profile along the sequence and finding the position with the best alignment score by means of a dynamic programming method. The alignment may include gaps in the database sequence. The best scoring alignments are with database sequences that have a pattern similar to that represented by the profile.

[c] The position-specific scoring matrix (PSSM), or weight matrix as it is sometimes called, is a representation of a multiple sequence alignment that has no gaps (a BLOCK). The matrix may be made from a multiple sequence alignment or by searching for patterns of the same length in a set of sequences using pattern-finding or statistical methods, e.g., expectation maximization, Gibbs sampling, ASSET, and by aligning these patterns, as discussed in Chapter 4. The consecutive columns of the matrix represent columns of the aligned patterns and the rows represent the distribution of amino acids in each column of the alignment. The PSSM columns include log odds scores for evaluating matches with a target sequence. The matrix is used to search a sequence for comparable patterns by sliding the matrix along the sequence and, at each position in the sequence, evaluating the match at each column position using the matrix values for that column. The log odds scores for each column are added to obtain a log odds score for the alignment to that sequence position. High log odds scores represent a significant match.

[d] Using a scoring matrix instead of a single query sequence can enhance a database search because the matrix represents the greater amount of sequence variation found in a multiple sequence alignment. Amino acid representation in each column of the alignment is also reflected in the matrix scores for that column; the more common an amino acid, the higher the score for a match to that amino acid. Note also that the matrix does not store any information about correlations between sequence positions. Thus, if two amino acids are commonly found together in the sequences at two positions of the alignment, these will each be independently scored by the matrix, but there will be no information as to their co-occurrence (or covariation) in the sequences. Since this type of information is missing, the matrix can give high scores to patterns that include new combinations of amino acids not found in the original set of sequences. Scoring covariation in sequence positions is discussed further in Chapters 5, 8, and 9.

[e] Pattern databases are described in Chapter 9.

Table 7.2. *Web resources for performing database searches with a simple query sequence*

Server/program	Web address or FTP site	Reference
BLAST—Basic Local Alignment Search Tool[a]	http://www.ncbi.nlm.nih.gov/BLAST FTP to ncbi.nlm.nih.gov/blast/executables	Altschul et al. (1990, 1997); Altschul and Gish (1996)
WU-BLAST[b]	sites that run WU-BLAST 2.0 are listed at http://blast.wustl.edu programs obtainable at http://blast.wustl.edu/blast/executables with licensing agreement	Altschul et al. (1990, 1997); Altschul and Gish (1996)
FASTA[c]	http://fasta.bioch.virginia.edu/fasta FTP to ftp.virginia.edu/pub/fasta	Pearson (1995, 1996, 1998, 2000)
BCM Search Launcher (Baylor College of Medicine)	http://dot.imgen.bcm.tmc.edu:9331/	see Web site
TIGR gene indices search	http://www.tigr.org	see Web site

Additional resources for performing a database sequence search using a dynamic programming method are described in Table 7.7. There are also many other BLAST and FASTA servers on the Web, including ones for searches in specific organisms (see Chapter 10). The TIGR site is given as an example of such a site.

[a] A stand-alone BLAST server may also be established on a local machine running Windows, UNIX, or MacOS.

[b] Executable programs for UNIX platforms are available from the FTP site. Note the advice given to increase search speed in protein searches by an order of magnitude (http://blast.wustl.edu/blast/TO-FLY.html

[c] Executable programs that run on PC, Macintosh, or UNIX platforms are available from the FTP site. The FASTA package also includes programs for performing pair-wise sequence alignments and for a statistical analysis of alignment scores (see Chapter 3). A number of Web sites offer FASTA database search, including the FASTA server and the BCM Search Launcher.

joined these into an alignment. BLAST, the next method, was similar to FASTA but gained a further increase in speed by searching only for rarer, more significant patterns in nucleic acid and protein sequences. BLAST is very popular due to availability of the program on the World Wide Web through a large server at the National Center for Biotechnology Information (NCBI) (http://ncbi.nlm.nih.gov) and at many other sites. The NCBI BLAST server site receives tens of thousands of requests a day. Both FASTA and BLAST have undergone evolution to recent versions that provide very powerful search tools for the molecular biologist and are freely available to run on many computer platforms. They are discussed further below.

With the more recent increased speed and size of computers and algorithmic improvements in the Smith-Waterman dynamic programming algorithm (described in Chapter 3), database similarity searches may also be performed by a search based on a full sequence alignment. The searches are 50-fold or more slower than FASTA and BLAST, but control experiments have revealed that more distantly related sequences will usually be found in a database search, provided that the appropriate statistical methods are used. A popular version of the Smith-Waterman program is SSEARCH (FTP to ftp.virginia.edu/pub/fasta), which is also available on Web sites but usually should be established on a local computer due to the length of time required for a search. Another recently introduced method for sequence alignment that has been used in database searches is the Bayes block aligner, described in Chapter 3 (p. 126). This program found more remotely similar sequences in protein families based on three-dimensional structure than did SSEARCH but is a much slower method (Zhu et al. 1998).

DNA VERSUS PROTEIN SEARCHES

One very important principle for database searches is to translate DNA sequences that encode proteins into protein sequences before performing a database search. DNA sequences comprise only four nucleotides, whereas protein sequences comprise 20 amino acids. Due to the fivefold larger variety of sequence characters in proteins, it is much easier to detect patterns of sequence similarity between protein sequences than between DNA sequences. Pearson (1995, 1996, 2000) has proven that searches with a DNA sequence encoding a protein against a DNA sequence database yield far fewer significant matches than searches using the corresponding protein sequence. To assist with an analysis based on translation of DNA sequences, both BLAST and FASTA provide programs that translate the query DNA sequence, the database DNA sequence, or both sequences in all six reading frames before making comparisons. An example of an exception to this rule would be a comparison of nucleic acid sequences in the same organism to locate other database entries of the same sequence. In such cases, a nucleic acid search would be needed.

When comparing methods of searching protein sequence databases, the sensitivity and selectivity of the methods should be considered. Sensitivity refers to the ability of the method to find most of the members of the protein family represented by the query sequence. Selectivity refers to the ability of the method not to find known members of other families as false positives. Ideally, both sensitivity and selectivity should be as high in quality as possible. A suitable method for describing both features is to describe the degree of coverage of families at a given level of false positives. Although similarity among many family members based on sequence similarity is readily identifiable, for some family members the similarity is weak and difficult to identify.

Identification of protein families is easier when the families are based on sequence similarity rather than on structural similarity, as discussed in detail in Chapter 9. Proteins that have the same structural features may have little, if any, sequence similarity. To facilitate a match of the query protein to a protein of known three-dimensional structure, protein sequences are grouped into families based on sequence similarity. All members of this family have sequence similarity with at least one of the remaining members, but not necessarily with all of the members, as illustrated in Figure 7.1. Families that include a protein of known three-dimensional structure are then identified. If a similarity search identifies a match of a query sequence with a member of such a protein family, the query sequence may be predicted to have a similar structure.

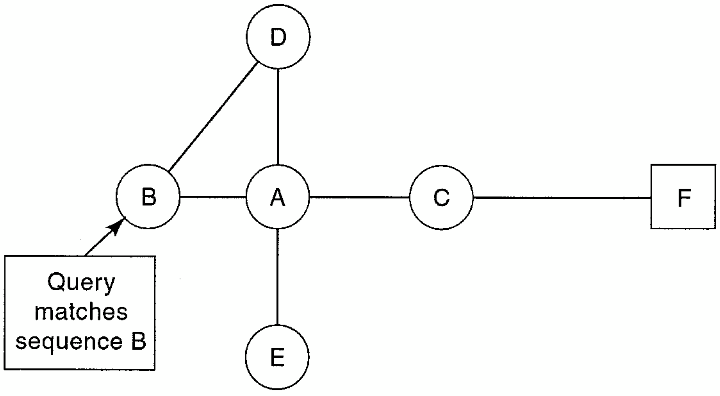


Figure 7.1. Structural prediction in database similarity searches. Sequences A–F refer to six members of a protein family defined by sequence similarity between some or all of the members. The sequences are represented as nodes on a graph, and similarity between sequences is represented by joining the nodes with a line (or edge). Note that not all nodes are joined. Thus, sequence A has detectable similarity to sequence B and to sequence C, but the relationship between sequence B and sequence C is not easily detectable. Suppose that sequence C can be aligned with sequence F, a protein of known three-dimensional structure. Hence, all members of this family may be predicted to have the same structure provided that the pair-wise alignments are significant and convincing. To gain further support for this prediction, improved alignments and identification of more family members to help bridge the similarity gaps are needed.

For protein sequence searches, two recent developments have greatly assisted with the finding of more distantly related sequences. First, combinations of amino acid substitution matrices and gap penalty scores that are most suitable for searches have been identified. Second, improved methods for establishing the statistical significance of a sequence alignment have been developed. Thus, whether a weak alignment between a query sequence and a database sequence is significant can be quite readily and confidently assessed. These topics are extensively discussed in Chapter 3 and on the book Web site and are reviewed below. Use of these new tools has also greatly improved the ability to balance sensitivity of a database search with selectivity.

SCORING MATRICES FOR SIMILARITY SEARCHES

There are a number of choices of amino acid substitution matrices for use in similarity searches of protein sequence databases (Henikoff and Henikoff 2000). The best performing matrices are now widely used, and they often are the default choice of the database search program. The most important consideration to be made is that the scoring matrix be in the log odds form so that statistical significance of the search results can be properly evaluated. In the log odds matrix, each matrix entry is the observed frequency of substitution of amino acids A and B for each other in proteins known to be related divided by the expected frequency of a chance substitution based on the frequency of A and B in proteins; the resulting ratio is then converted to a logarithm. The score is simply the logarithm of the odds that a pair of aligned amino acids is found because the sequences are related to a chance alignment of the pair in an alignment between unrelated sequences. The log odds form is useful because the probabilities that successive pairs in an alignment are related is the product of the odds of each pair. When log odds values are used, the probabilities may be found by addition in a much simpler calculation. Choice of the best scoring matrix for sequence alignments is discussed in detail in Chapter 3 and on the book Web site and is reviewed below.

PAM250 Scoring Matrix

For a long time, the Dayhoff PAM250 matrix was used for database searches. This scoring matrix is based on an evolutionary model that predicts the types of amino acid changes over long periods of time. The matrix is based on tallying the observed amino acid changes in a closely related group of proteins that were 85% identical. The proteins were organized into an evolutionary tree, and the predicted amino acid changes in the tree were used to estimate the frequency of substitution of each amino acid for another. These frequencies were then normalized to those expected if 1% of the sequence were to change, giving the PAM1 matrix. This level of change roughly corresponds to those amino acid changes expected over a period of 50 million years of evolutionary history. The substitution frequencies in the PAM1 matrix were then extrapolated to predict the changes occurring over longer periods of evolutionary time. For example, if D substitutes for E in the first PAM period, then in the second period, there is an additional chance that D might substitute for E. However, it is also possible that in a second PAM period the initial D substitution might revert to E or change to any other amino acid. As more time passes, the type and frequency of each substitution between the beginning and end of the time period will change. PAM250 represents a period of time at which only 20% of the amino acids will remain unchanged, but the expected frequencies are extrapolated many times from those observed in proteins that are 85% similar. Additional information concerning more recent substitution matrices that are based on an evolutionary model is discussed in Chapter 3 and on the book Web site. For many types of database searches, the PAM250 scoring matrix has been replaced by the BLOSUM matrices.

BLOSUM62 Scoring Matrix

The amino acid substitution matrix used by the BLAST programs is the BLOSUM62 scoring matrix. This matrix represents frequencies of amino acid substitutions observed in a large number of related proteins, including some quite similar and some quite different protein sequences. The observed substitutions are all lumped together to provide average frequencies of substitutions without regard to the degree of divergence between sequences. This approach appears to be more suitable for similarity searches in databases than using the Dayhoff PAM250 matrix, probably because sequences separated by any evolutionary distance may be more readily recognized. The Dayhoff matrices are also based on a much smaller data set than the BLOSUM62 matrix. The BLOSUM scoring matrices were generated by S. Henikoff and J.G. Henikoff (1992), who searched for common sequence patterns (blocks) of the same length among all of the related proteins in the Prosite catalog (see p. 428). They then added some additional related sequences in the current databases at the time and scored the columns in a multiple sequence alignment of these patterns for amino acid substitutions. In scoring the columns, some amino acid substitutions were much more common than others because many of the sequences had the same amino acid. The resulting BLOSUM matrices have a number to designate how much these repeated occurrences were weighted. The BLOSUM62 matrix uses only 62% of the repeats in one column and thereby reduces the relative weight given to those substitutions in the matrix. Another scoring matrix, BLOSUM50, which weights the repeated substitutions somewhat less, has been found to be more suitable for database searches by the FASTA and SSEARCH programs, which use different algorithms from BLAST. BLOSUM matrices give the best results when the appropriate gap opening and gap extension are used, as discussed in Chapter 3.

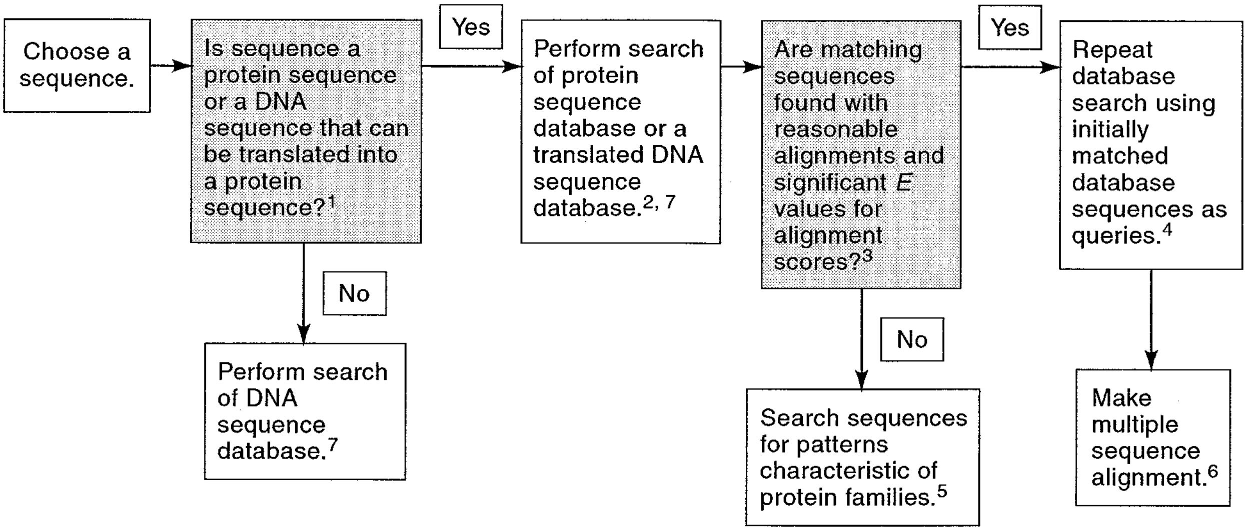
Other Scoring Matrices

In addition to the BLOSUM amino acid substitution matrices, a number of other scoring matrices have been devised. The usefulness of various combinations of search programs and substitution matrices for identifying that largest possible number of related sequences in a database search, including remotely related sequences, has been studied in considerable detail. These studies are extensively reviewed and referenced in Chapter 3 and on the book Web site.

LIMITING OUTPUT

Database similarity search programs tend to produce large volumes of output. It can become difficult to screen this volume of material and to assess whether or not the more remotely related sequences are really related to the query sequence. Thus, it is important to limit the sequence output; there are some relatively simple procedures that may be followed for each program, as described below. For searches of protein databases, avoid repetitive alignments with the same sequence by limiting searches to the protein sequence databases that are well curated, such as SwissProt and PIR, as opposed to translated GenBank sequences (the Genpept database).

METHODS


Choose a sequence.
Is sequence a protein sequence or a DNA sequence that can be translated into a protein sequence?1
Yes
No
Perform search of DNA sequence database.7
Perform search of protein sequence database or a translated DNA sequence database.2, 7
Are matching sequences found with reasonable alignments and significant E values for alignment scores?3
Yes
No
Search sequences for patterns characteristic of protein families.5
Repeat database search using initially matched database sequences as queries.4
Make multiple sequence alignment.6

1. Translation of protein-encoding DNA sequences into protein sequences before performing sequence comparisons has been shown to be a more effective way to identify related genes than direct comparisons of untranslated DNA sequences. This method also corrects for different codon usage, base composition, and other DNA sequence variations by different organisms. However, to search for a matching DNA sequence in the same organism (e.g., a section of genomic DNA that is thought to encode a protein is used as a query against an EST database for the organism), a nucleic acid search is more appropriate. If the entire sequence does not encode a protein (e.g., the sequence is a genomic sequence that includes introns), the sequence can be translated in all six reading frames to locate open reading frames that may specify the amino acid sequence of a protein. The predicted translation product may then be compared to a protein sequence database or a DNA sequence database that is translated in all six reading frames. Alternatively, a gene annotation of the genomic DNA sequence—a predicted amino acid sequence for the protein encoded by the gene that has been entered into the protein sequence database—may be used, as described in Chapter 8. Masking low-complexity regions and sequence repeats in the query sequence is also necessary in many cases because such regions tend to give high-scoring alignments.
2. The carefully annotated protein sequence database (e.g., PIR, SwissProt) will provide a more manageable output list of matched sequences. However, investigators may also wish to expand the search to include predicted genes from gene annotations of genomic sequences (see note 1 and Chapter 8) that are frequently entered into the DNA sequence translation databases (e.g., DNA sequences in the GenBank DNA sequence databases automatically translated into protein sequences and placed in the Genpept protein sequence database). To compare a protein or predicted protein sequence to EST sequences of an organism, the ESTs should be translated into all six possible reading frames (Pearson 2000).
3. A matched database sequence that is listed should have a small *E* score and a reasonable alignment with the query sequence (or translations of protein-encoding DNA sequences should have these same features). The *E* (expect value) of the alignment score between the sequences gives the statistical chance that an unrelated sequence in the database or a random sequence could have achieved such a score with the query sequence, given as many sequences as there are in the database. The smaller the *E* score, the more significant the alignment. A cutoff value in the range of 0.01–0.05 is used (Pearson 1996). However, the alignment should also be examined for absence of repeats of the same residue or residue pattern because these patterns tend to give false high alignment scores. Filtering of low-complexity regions from the query sequence in a database search helps to reduce the number of false positives. The alignment should also be examined for reasonable amino acid substitutions and for the appearance of a believable alignment (see Chapter 3 flowchart for a summary). One of the sequences

may be shuffled many times, and each random sequence may be realigned with the other sequence to obtain a score distribution for a set of unrelated sequences. This distribution may then be used to evaluate the significance of the true alignment score (Chapter 3).

4. Including these extra steps may find additional members of a protein family that has too low a sequence similarity to the original query sequence to be detected in the first search.
5. These types of searches are discussed later in the chapter.
6. Methods and considerations that need to be made for producing a multiple sequence alignment are discussed in Chapter 4. Additional relationships among the matched sequences may be found by performing a phylogenetic analysis based on the multiple sequence alignments as described in Chapter 6. Such a phylogenetic analysis can reveal which sequence of several found in an organism is most closely related to a query sequence and therefore is the most likely of the group to have the same function as the query sequence.
7. For performing a large number of searches, there is a definite advantage to setting up the search programs on a local machine, especially since versions of the programs that run on most computer platforms are available. One can then set up batch commands or scripts (shell or Perl scripts) for processing the sequences and managing the returned data. The NCBI staff provides assistance in the form of SEALS (a system for analysis of lots of sequences) at http://ncbi.nlm.nih.gov/Walker/SEALS/index.html (Walker and Koonin 1997).

FASTA SEQUENCE DATABASE SIMILARITY SEARCH

FASTA is a program for rapid alignment of pairs of protein and DNA sequences. Rather than comparing individual residues in the two sequences, FASTA instead searches for matching sequence patterns or words, called *k*-tuples (Wilbur and Lipman 1983; Lipman and Pearson 1985; Pearson and Lipman 1988). These patterns comprise *k* consecutive matches in both sequences. The program then attempts to build a local alignment based on these word matches. Due to the ability of the algorithm to find matching sequences in a sequence database with high speed, FASTA is useful for routine searches of this type. Comparable methods are the BLAST algorithm, which is faster, and of comparable sensitivity for protein queries, and a Smith-Waterman dynamic programming algorithm, which is much slower but more sensitive when full-length protein sequences are used as queries. Detailed performance studies of these methods have been made, one showing that the Smith-Waterman dynamic programming algorithm and FASTA outperformed BLAST (Pearson 1995). The FASTA programs have all undergone recent enhancements that have improved detection of more remotely related sequences. For sequence fragments, FASTA is as good as Smith-Waterman methods. For DNA searches, FASTA is theoretically better able than BLAST to find matches because a *k*-tuple smaller than the minimum obligatory one of 7 (default size 11) for BLASTN (3 for TBLASTN, BLASTX, TBLASTX) may be used. For reviews on using FASTA, see Pearson (1995, 1996, 1998). The following information is largely based on these reviews and on information provided in the FASTA distribution package.

FASTA3

FASTA has gone through a series of updates and enhancements leading to version 3, denoted FASTA3. FASTA3 has improved methods of aligning sequences and of calculating the statistical significance of alignments. These changes result in a greatly increased ability of FASTA3 to detect distantly related sequences. The FASTA package is available by anonymous FTP from ftp.virginia.edu/pub/.

FASTA compares an input DNA or protein sequence to all of the sequences in a target sequence database and then reports the best-matched sequences and local alignments of these matched sequences with the input sequence. The input sequence is usually in the standard FASTA format, but it is also very easy to change sequence formats, as described in Chapter 2. FASTA finds sequence similarities between the query sequence and each database sequence in four steps illustrated in Figure 7.2.

In the initial stage of a search for regions of similarity, FASTA uses an algorithmic method known as hashing, illustrated in Table 7.3. In this method, a lookup table showing the positions of each word of length *k*, or *k*-tuple, is constructed for each sequence. The relative positions of each word in the two sequences are then calculated by subtracting the position in the first sequence from that in the second. Words that have the same offset position are in phase and reveal a region of alignment between the two sequences. Using hashing, the number of comparisons increases linearly in proportion to average sequence length. In contrast, the number of comparisons in dot matrix and dynamic programming methods increases between the square and the cube of the average sequence length. In FASTA, the *k*-tuple length is user-defined and is usually 1 or 2 for protein sequences (i.e., either the positions of each of the individual 20 amino acids or the positions of each of the 400 possible dipeptides are located). For nucleic acid sequences, the *k*-tuple is 4–6, and is much longer than for protein sequences because short *k*-tuples are much more common due to the four-letter alphabet of nucleic acids. The larger the *k*-tuple chosen, the more rapid, but less thorough, a database search.

Significance of FASTA Matches

The methods used by FASTA to report the significance of a database search were revised in later versions, and use of the latest version FASTA3 is strongly recommended. Similar methods are used by the database search program SSEARCH, which is based on a slower Smith-Waterman type of alignment. The statistical scores provide a reliable indication as to whether or not the alignment scores for sequences found in a database search are significant. This analysis provides the probability that scores between unrelated sequences could reach as high a value as those found for the higher-scoring alignments (Pearson 1998). The statistical distribution of scores found in a database search follows the extreme value distribution, described in detail in Chapter 3 (p. 96).

Recall that the parameters of the extreme value distribution, u and λ, vary with the length and composition of the sequences being compared, and also with the particular scoring system. In database searches, the expected score between the query sequence and an unrelated database sequence increases in proportion to the logarithm of the length of the database sequence. The parameters change when a different scoring system, e.g., a different scoring matrix or gap penalty, is used. FASTA calculates these parameters from the scores found with unrelated sequences during the database search. Some of the sequence scores in the database search arise from matches with related sequences and must be removed before the statistical calculations are performed. FASTA performs these tasks in the following manner:

1. The average score for database sequences in the same length range is determined.
2. The average score is plotted against the logarithm of average sequence length in each length range.
3. The points are then fitted to a straight line by linear regression.
4. A z score, the number of standard deviations from the fitted line, is calculated for each score.

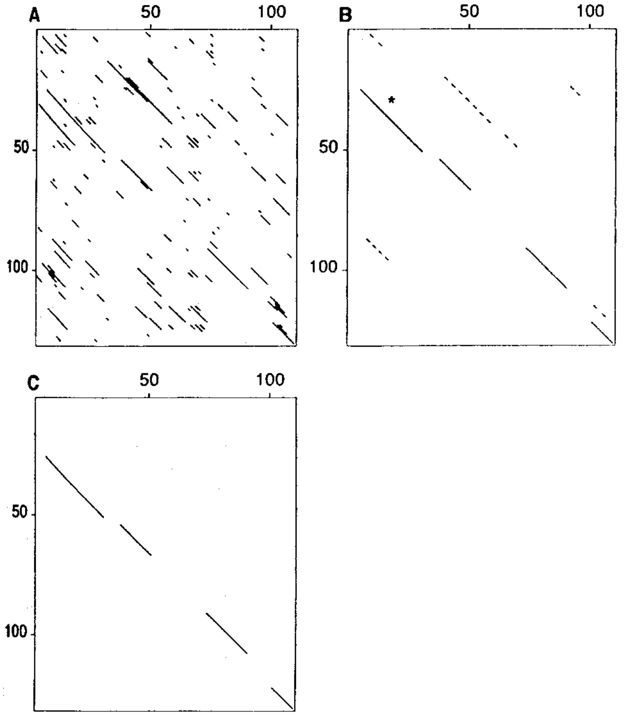


Figure 7.2. Methods used by FASTA to locate sequence similarities. (*A*) The 10 best-matching regions in each sequence pair are located by a rapid screen. First, all sets of *k* consecutive matches are found by a rapid method described below. For DNA sequences, *k* is usually 4–6 and for protein sequences, 1–2. Second, those matches within a certain distance of each other (for proteins, 32 for *k*=1 and 16 for *k*=2) are joined along with the region between them into a longer matching region without gaps. The regions with the highest density of matches are identified. The calculation is very much like a dot matrix analysis described in Chapter 3, but is calculated in fewer steps. The diagonals shown in *A* represent the locations of these common patterns initially found in the two sequences. (*B*) The highest-density regions of protein sequences identified in *A* are evaluated using an amino acid substitution matrix such as a PAM or BLOSUM scoring matrix. A corresponding matrix may also be used for DNA sequences. The highest-scoring regions, called the best initial regions (INIT1), are identified and used to rank the matches for further analysis. The best-scoring INIT1 region is shown marked by an asterisk. (*C*) Longer regions of identity of score INITN are generated by joining initial regions with scores greater than a certain threshold. The INITN score is the sum of the scores of the aligned individual regions less a constant gap penalty score for each gap introduced between the regions. Later versions of FASTA include an optimization step. When the INITN score reaches a certain threshold value, the score of the region is recalculated to produce an OPT score by performing a full local alignment of the region using the Smith-Waterman dynamic programming algorithm. By improving the score, this step increases the sensitivity but decreases the selectivity of a search (Pearson 1990). INITN and OPT scores are used to rank database matches. Finally, not shown, an optimal local alignment between the input query sequence and the best-scoring database sequences is performed based on the Smith-Waterman dynamic programming algorithm. (Reprinted, with permission, from Pearson and Lipman 1988.)

5. High-scoring, presumably related sequences, and also very low scoring alignments that do not fit the straight line are removed from consideration.
6. Steps 1–5 are repeated one or more times.
7. The known statistical distribution of alignment scores is used to calculate the probability that a *Z* score between unrelated or ran-

Table 7.3. *Lookup method for finding an alignment*

```
position    1  2  3  4  5  6  7  8  9  10  11
sequence  1 n  c  s  p  t  a · · · · ·

position    1  2  3  4  5  6  7  8  9  10  11
sequence  2                a  c  s  p   r   k
```

amino acid	position in protein A	position in protein B	offset pos A - pos B
a	6	6	0
c	2	7	-5
k	-	11	
n	1	-	
p	4	9	-5
r	-	10	
s	3	8	-5
t	5	-	

```
Note the common offset for the 3 amino acids c, s, and p.
A possible alignment is thus quickly found

protein 1 n c s p t a
            | | |
protein 2 a c s p r k
```

Shown are fragments of two sequences that share a pattern c-s-p. All of the positions at which a given character is found are listed in a table. The positions of a given character in one of the sequences are then subtracted from the positions of the same character in the second sequence, giving an offset in location. When the offsets for more than one character are the same, a common word is present that includes those characters. Common words, or *k*-tuples, in two sequences are found by this method in a number of steps proportional to the sequence lengths.

dom sequences of the same lengths as the query and database sequence could be greater than z,

$$P\,(Z > z) = 1 - \exp(-e^{-1.2825z\,-\,0.5772})\mathrm{ss} \qquad (1)$$

The derivation of this equation is given in Chapter 3, page 108.

The expectation E of observing, in a database of D sequences, no alignments with scores higher than z is given by e^{-DP} and that of observing at least one score z is $E \simeq 1 - e^{-DP}$. For $P<0.1$, this relationship is approximated by $E \simeq DP$ as indicated below.

$$E\,(Z > z) = D \times P\,(Z > z) \qquad (2)$$

8. Normalized similarity scores are calculated for each score by the formula $z' = 50 + 10z$. Thus, an alignment score with a standard deviation of 5 has a normalized score of 100. These normalized scores are reported in the program output.
9. The significance of an alignment score between a given sequence and a database sequence may be further analyzed by aligning a sequence with a shuffled library or a shuffled sequence with an unshuffled library (Pearson 1996) as described in Chapter 3, page 116.

An example of a database search with FASTA, vers. 3 is shown in Figure 7.3.

Versions of FASTA

There are several implementations of the FASTA algorithm (W. Pearson, release notes for FASTA vers. 3 and earlier releases; Pearson et al. 1997; Pearson 1998) using newly developed algorithms (Zhang et al. 1997):

1. FASTA compares a query protein sequence to a protein sequence library to find similar sequences. FASTA also compares a DNA sequence to a DNA sequence library.
2. TFASTA compares a query protein sequence to a DNA sequence library, after translating the DNA sequence library in all six reading frames.
3. FASTF/TFASTF and FASTS/TFASTS compare a set of short peptide fragments, as obtained from analysis of a protein, against a protein sequence database (FASTF/FASTS) or a DNA sequence database translated in all six reading frames (TFASTF/TFASTS). The FASTF programs analyze a set of fragments following cleavage and sequencing of protein bands resolved by electrophoreseis and the FASTS programs data from a mass spectrometry analysis of a protein. Note that a different sequence format is required to specify the separate peptides (see http://fasta.bioch.virginia.edu/fasta/).

Additional programs have been developed that are designed to align a DNA sequence with a protein sequence, allowing gaps and frameshifts. If a DNA sequence has a high possibility of errors, such as EST sequences, the translated sequence may be inaccurate due to amino acid changes or frameshifts. These programs are designed to go around such errors by allowing gaps and frameshifts in the alignments. FASTX and TFASTX allow only frameshifts between codons, whereas FASTY and TFASTY allow substitutions and frameshifts within a codon. These programs have been shown to be very useful for gene panning, the search for related sequences in EST databases (Retief et al. 1999).

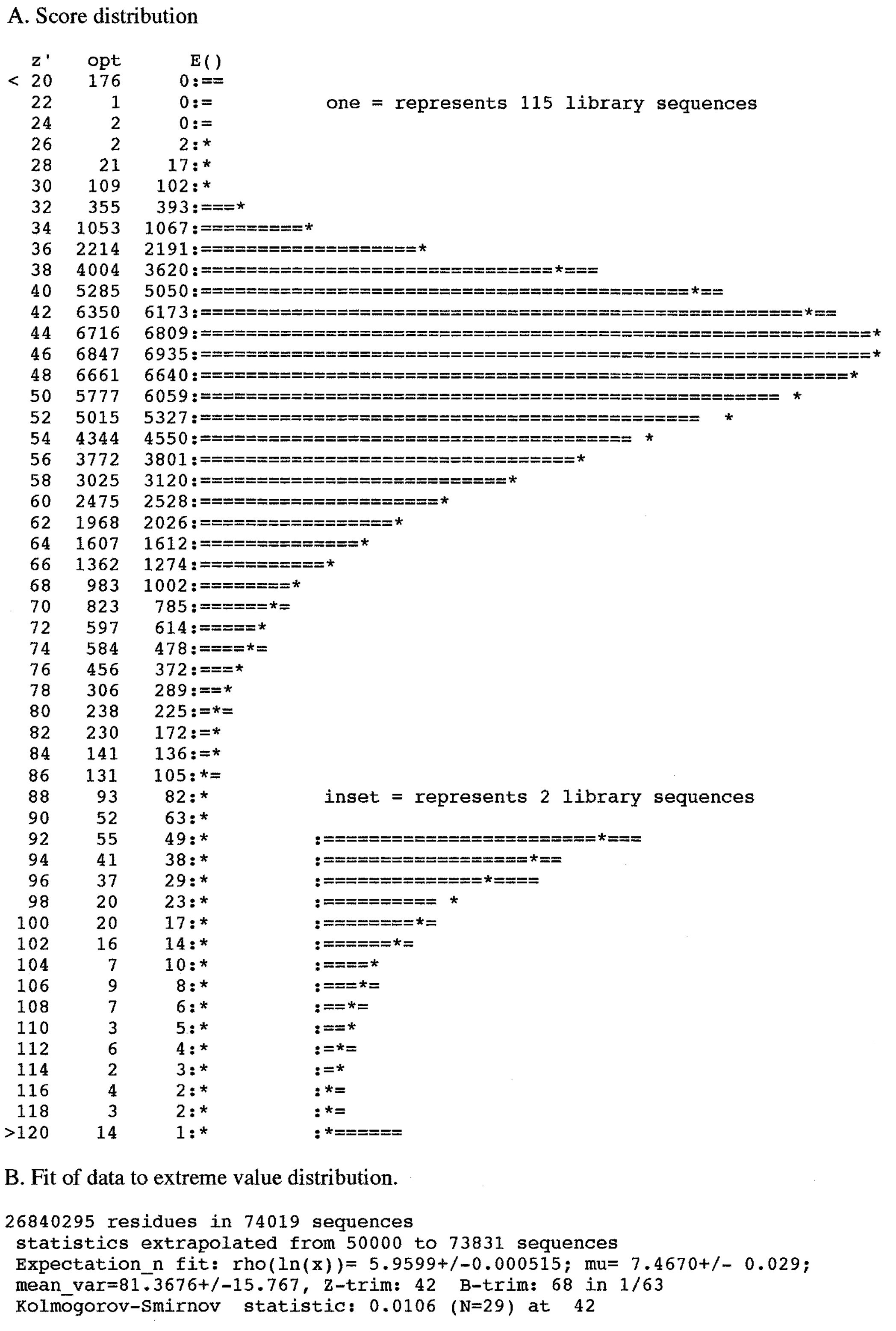
1. FASTX and FASTY translate a query DNA sequence in all three reading forward frames and compare all three frames to a protein sequence database.
2. TFASTX and TFASTY compare a query protein sequence to a DNA sequence database, translating each DNA sequence in all six possible reading frames.

The above FASTA suite of programs is available as executable binary files for most computer systems including Windows, Macintosh, and UNIX platforms (ftp.virginia.edu/pub/FASTA).

The FASTA algorithm has also been adapted for searching through a pattern database instead of a sequence database (Ladunga et al. 1996). FASTA-pat and FASTA-swap are accessible at the Baylor College of Medicine Web site (http://dot.imgen.bcm.tmc.edu:9331/seq-search/Options/fastapat.html). Instead of comparing a query sequence to a sequence database, these programs compare the query sequence to a pattern database that contains patterns representative of specific protein families (see below, p. 326). A match between the query sequence and specific database patterns is an indication of a familial relationship between that sequence and the sequences from which those database patterns were generated.

Matching Regions of Low Sequence Complexity

FASTA and SSEARCH (described below) do not provide a method for avoiding low-complexity sequences or sequence repeats (Pearson 1998). Such regions can lead to higher scores between the query and database sequences than for other sequence pairs, thus giving the appearance that the sequences are related when they are actually not related. An



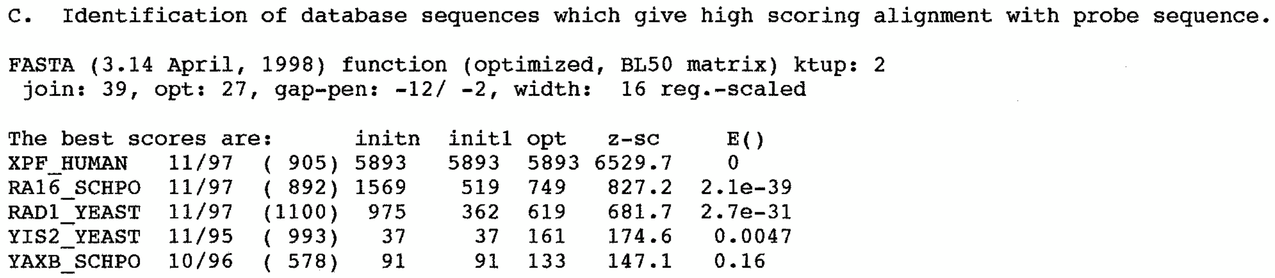
A. Score distribution

```
  z'   opt     E()
< 20   176      0:==
  22     1      0:=         one = represents 115 library sequences
  24     2      0:=
  26     2      2:*
  28    21     17:*
  30   109    102:*
  32   355    393:===*
  34  1053   1067:=========*
  36  2214   2191:===================*
  38  4004   3620:===============================*===
  40  5285   5050:===========================================*==
  42  6350   6173:=====================================================*==
  44  6716   6809:===========================================================*
  46  6847   6935:===========================================================*
  48  6661   6640:=========================================================*
  50  5777   6059:=================================================== *
  52  5015   5327:============================================  *
  54  4344   4550:====================================== *
  56  3772   3801:=================================*
  58  3025   3120:===========================*
  60  2475   2528:=====================*
  62  1968   2026:=================*
  64  1607   1612:==============*
  66  1362   1274:===========*
  68   983   1002:========*
  70   823    785:======*=
  72   597    614:=====*
  74   584    478:====*=
  76   456    372:===*
  78   306    289:==*
  80   238    225:=*=
  82   230    172:=*
  84   141    136:=*
  86   131    105:*=
  88    93     82:*         inset = represents 2 library sequences
  90    52     63:*
  92    55     49:*        :========================*===
  94    41     38:*        :==================*==
  96    37     29:*        :==============*====
  98    20     23:*        :========== *
 100    20     17:*        :========*=
 102    16     14:*        :======*=
 104     7     10:*        :====*
 106     9      8:*        :===*=
 108     7      6:*        :==*=
 110     3      5:*        :==*
 112     6      4:*        :=*=
 114     2      3:*        :=*
 116     4      2:*        :*=
 118     3      2:*        :*=
>120    14      1:*        :*======
```

B. Fit of data to extreme value distribution.

```
26840295 residues in 74019 sequences
 statistics extrapolated from 50000 to 73831 sequences
 Expectation_n fit: rho(ln(x))= 5.9599+/-0.000515; mu= 7.4670+/- 0.029;
 mean_var=81.3676+/-15.767, Z-trim: 42  B-trim: 68 in 1/63
 Kolmogorov-Smirnov  statistic: 0.0106 (N=29) at  42
```

Figure 7.3. *Figure continues on next page.*

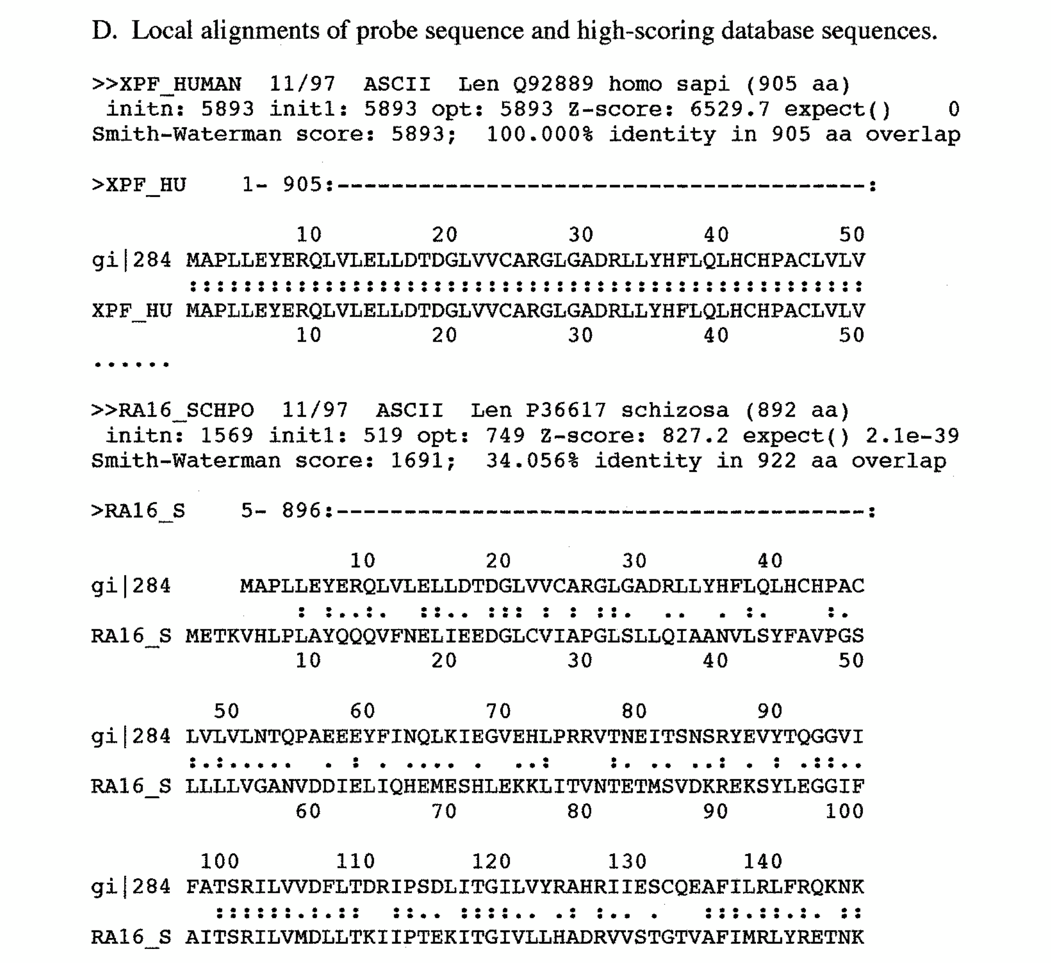


```
C.  Identification of database sequences which give high scoring alignment with probe sequence.

FASTA (3.14 April, 1998) function (optimized, BL50 matrix) ktup: 2
 join: 39, opt: 27, gap-pen: -12/ -2, width:  16 reg.-scaled

The best scores are:      initn  init1 opt   z-sc     E()
XPF_HUMAN   11/97   ( 905) 5893   5893  5893 6529.7    0
RA16_SCHPO  11/97   ( 892) 1569    519  749   827.2  2.1e-39
RAD1_YEAST  11/97   (1100)  975    362  619   681.7  2.7e-31
YIS2_YEAST  11/95   ( 993)   37     37  161   174.6   0.0047
YAXB_SCHPO  10/96   ( 578)   91     91  133   147.1   0.16
```

Figure 7.3. *Continued.* Example of a FASTA, Vers. 3 search. The SwissProt protein database was searched with the human XPF DNA repair protein on a local UNIX server with a locally written Web page interface. The recommended (default) BLOSUM50 amino acid scoring matrix and gap penalties of $-12/-2$ were used. Actual z scores are normalized to a mean of 50 and a standard deviation of 10 (normalized scores are indicated in this version of the program output in A as z', in B as z, and in C as Z). These values may be converted back to actual z scores for statistical calculations by subtracting 50 and dividing by 10. (*A*) Histogram of the normalized similarity scores and the expected score distribution. The first column gives the lower score in each range of scores, the second labeled "opt" is the number of optimized scores in that range, and the third labeled "E()" is the number of alignment scores expected to be in that range for unrelated sequences based on the extreme value distribution and the calculated values of u and λ. The "=" signs outline an approximate curve for the actual score distribution and the "*" gives the same information for the expected score distribution. Note the excellent agreement between the observed and expected numbers until a normalized score >120 is reached, at which point some high-scoring alignments are revealed. (*B*) An evaluation of the fit of the data to the expected curve is given by the Kolmogorov-Smirnov statistic, which compares the maximum deviation between the observed and expected values. In his FASTA distribution notes, W. Pearson indicates that statistic values <0.10 (for $N=30$) reveal excellent agreement. If this statistic is >0.2, he suggests repeating the analysis with higher gap penalties, e.g., -16, -4 rather than -12, -2. (*C*) Database sequences that have high normalized alignment scores are listed along with the raw init1, initn, opt, z' score, and E() for a z' score of that value. E() gives the probability that alignment of the query sequence with D database sequences unrelated to the query sequence could generate at least one such z' score. Note that the first row of scores is that for aligning the query sequence with a database copy of itself, followed by very high-scoring alignments to two yeast DNA repair genes on the next rows. (*D*) Smith-Waterman local alignments are shown along with additional information about the percent identity. A ":" in the alignment is an identity and "." is a conservative substitution. Included is a sketch indicating the extent to which the sequences can be locally aligned.

Figure continues on next page.

update of this feature can be anticipated in the near future. The BLAST2 programs described below filter regions of low complexity in both DNA and protein query sequences. Programs and Web sites for this purpose are described below in the description of BLAST. The program PRSS in the FASTA distribution package provides a straightforward way of establishing whether or not low complexity plays a role in the alignment score between two sequences. These programs shuffle the matching library sequences many times and realign each of the shuffled sequences with the query sequence. Two levels of shuffling are possible, one at the level of individual amino acids and a second at the level of sequence segments of a chosen length. The first method explores the possibility that restricted amino acid composition plays a role in the alignment, and the second that particular regions in the query sequence, such as sequence repeats, influence the score. If low complexity at either level is a problem, high scores will be produced when shuffled sequences are aligned with the query sequence. The distribution of scores from alignment between shuffled and query sequences is used to compute the statistical significance of the actual alignment score between the sequences. An example of using PRSS is presented in Chapter 3 (p. 116).



D. Local alignments of probe sequence and high-scoring database sequences.

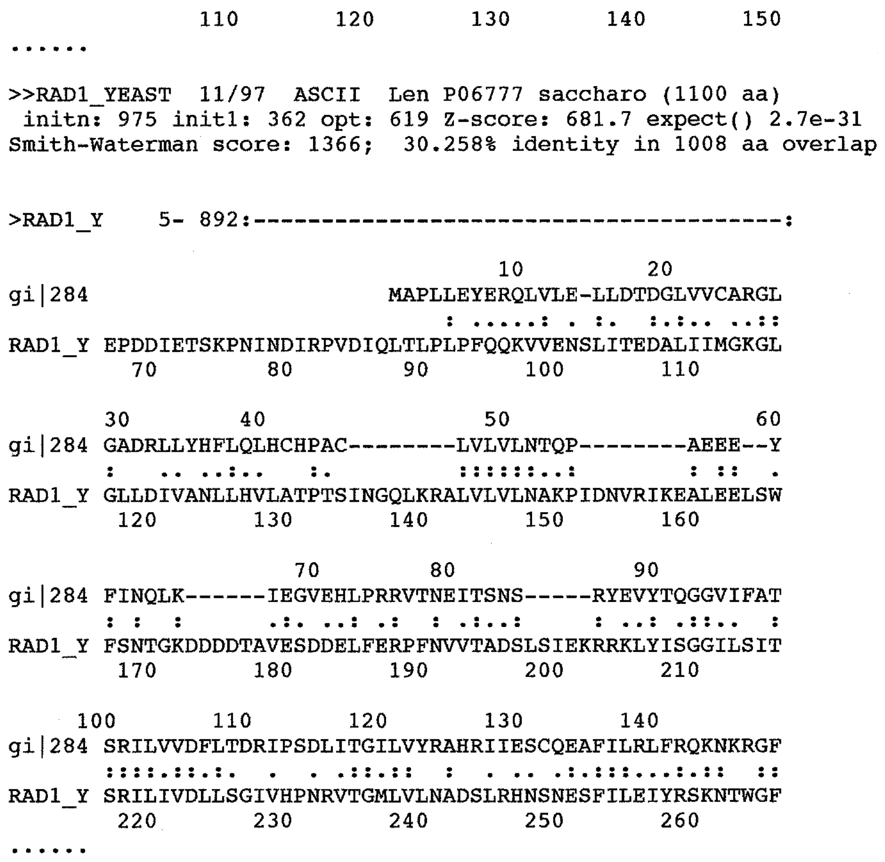
```
>>XPF_HUMAN  11/97  ASCII  Len Q92889 homo sapi (905 aa)
 initn: 5893 init1: 5893 opt: 5893 Z-score: 6529.7 expect()    0
Smith-Waterman score: 5893;  100.000% identity in 905 aa overlap

>XPF_HU    1- 905:---------------------------------------:

               10        20        30        40        50
gi|284 MAPLLEYERQLVLELLDTDGLVVCARGLGADRLLYHFLQLHCHPACLVLV
       ::::::::::::::::::::::::::::::::::::::::::::::::::
XPF_HU MAPLLEYERQLVLELLDTDGLVVCARGLGADRLLYHFLQLHCHPACLVLV
               10        20        30        40        50
......

>>RA16_SCHPO  11/97  ASCII  Len P36617 schizosa (892 aa)
 initn: 1569 init1: 519 opt: 749 Z-score: 827.2 expect() 2.1e-39
Smith-Waterman score: 1691;  34.056% identity in 922 aa overlap

>RA16_S    5- 896:---------------------------------------:

                   10        20        30        40
gi|284     MAPLLEYERQLVLELLDTDGLVVCARGLGADRLLYHFLQLHCHPAC
               : :..:.  ::.. ::: : : ::.  ..  . :.    :.
RA16_S METKVHLPLAYQQQVFNELIEEDGLCVIAPGLSLLQIAANVLSYFAVPGS
               10        20        30        40        50

         50        60        70        80        90
gi|284 LVLVLNTQPAEEEYFINQLKIEGVEHLPRRVTNEITSNSRYEVYTQGGVI
       :.:.....  . : . .... .  ..:    :. .. ..: . : .::..
RA16_S LLLLVGANVDDIELIQHEMESHLEKKLITVNTETMSVDKREKSYLEGGIF
               60        70        80        90       100

        100       110       120       130       140
gi|284 FATSRILVVDFLTDRIPSDLITGILVYRAHRIIESCQEAFILRLFRQKNK
         ::::::.:.::  ::.. ::::.. .: :.. .   :::.::.:. ::
RA16_S AITSRILVMDLLTKIIPTEKITGIVLLHADRVVSTGTVAFIMRLYRETNK
```

Figure 7.3. *Continued.*

Recommended Steps for a FASTA Search

The following strategy is recommended for searches with FASTA for finding the most homologous sequences in a database search while avoiding false-negative matches (Pearson 1996, 2000):

1. Look for agreement between the real and theoretical distribution of scores. If the query sequence has a low-complexity, repeated domain or if the gap penalties are set too low, there may be an excess of unrelated sequences with *E* less than 0.1. If there is an excess of three- to fivefold more sequences than expected in the score range of 80–110, repeat the search after removing the low-complexity regions from the query sequence (see page 308 for a description of this method) or else increase the gap penalties from −12/−2 to −14/−4. Another test to apply is to examine the number of high-scoring unrelated sequences with *E* smaller than 1.0. If there are more than 5–10 such sequences, the analysis is suspect.



```
              110       120       130       140       150
......

>>RAD1_YEAST  11/97  ASCII  Len P06777 saccharo (1100 aa)
 initn: 975 init1: 362 opt: 619 Z-score: 681.7 expect() 2.7e-31
Smith-Waterman score: 1366;  30.258% identity in 1008 aa overlap


>RAD1_Y    5- 892:---------------------------------------:

                                    10         20
gi|284                      MAPLLEYERQLVLE-LLDTDGLVVCARGL
                                : .....: . :.  :.:.. ..::
RAD1_Y EPDDIETSKPNINDIRPVDIQLTLPLPFQQKVVENSLITEDALIIMGKGL
         70        80        90       100       110

       30        40                50                  60
gi|284 GADRLLYHFLQLHCHPAC--------LVLVLNTQP--------AEEE--Y
       :   .. ..:..   :.         ::::::..:        : ::  .
RAD1_Y GLLDIVANLLHVLATPTSINGQLKRALVLVLNAKPIDNVRIKEALEELSW
        120       130       140       150       160

                     70        80             90
gi|284 FINQLK------IEGVEHLPRRVTNEITSNS-----RYEVYTQGGVIFAT
       : :  :      .:. ..: .:  : .:..:     : ..: .::..  :
RAD1_Y FSNTGKDDDDTAVESDDELFERPFNVVTADSLSIEKRRKLYISGGILSIT
        170       180       190       200       210

     100       110       120       130       140
gi|284 SRILVVDFLTDRIPSDLITGILVYRAHRIIESCQEAFILRLFRQKNKRGF
       ::::.::.:.  .  . .::.::  :  . .. .:.:::...:.::  ::
RAD1_Y SRILIVDLLSGIVHPNRVTGMLVLNADSLRHNSNESFILEIYRSKNTWGF
        220       230       240       250       260
......
```

Figure 7.3. *Continued.*

2. Recall that the expect score *E* of a database match is the number of times that an unrelated database sequence would obtain a score higher than *z* just by chance. For a match to be significant, *E* should be < 0.01–0.05. If the search has correctly identified homologous sequences, the corresponding *E* values should be much less than 0.01, whereas scores between unrelated sequences should be much greater than this value; e.g, at least 0.5. If there are no *E* scores less than 0.1, the search has not found any sequences with significant similarity to the query sequence.
3. If there are no matches with *E* less than 0.1, repeat the search with FASTA with *k*-tuple = 1, or else use the Smith-Waterman dynamic programming method with a program such as SSEARCH. If the program now finds matches with *E* less than 0.02, the sequences may be homologous, if there is not a low-complexity region in the query sequence. Computer experiments with FASTA have revealed that sequences with scores of 0.2–10 may also be homologous but have marginal sequence similarity. For further study of this possibility, select some of these marginal sequences and use them as query sequences for additional database searches with FASTA. Additional family members with significant similarity may then be found.
4. Confirm homology of marginal matches by shuffling the query or database sequence many times to calculate the significance of the real alignment. The program PRSS described in Chapter 3, page 116, performs this task.

5. Protein sequence alignments with 50% identity in a short 20- to 40-amino-acid region are common in unrelated proteins. To be truly significant, the alignment should extend over a longer region.

BASIC LOCAL ALIGNMENT SEARCH TOOL (BLAST)

The BLAST algorithm was developed as a new way to perform a sequence similarity search by an algorithm that is faster than FASTA while being as sensitive. A powerful computer system dedicated to running BLAST has been established at NCBI, National Library of Medicine. Access to this BLAST system is possible through the Internet (http://www.ncbi.nlm.nih.gov/) as a Web site and through a BLAST E-mail server. There are also numerous other Web sites that provide a BLAST database search. In addition to the BLAST programs developed at the NCBI, an independent set of BLAST programs has been developed at Washington University (see Table 7.2). These programs perform similarity searches using the same methods as NCBI-BLAST and produce gapped local alignments. The statistical methods used to evaluate sequence similarity scores are different, and thus WU-BLAST and NCBI-BLAST can produce different results (see box below, point 11).

The BLAST Web server at http://www.ncbi.nlm.nih.gov/ is the most widely used one for sequence database searches and is backed up by a powerful computer system so that there is usually very little wait. Like FASTA, the BLAST algorithm increases the speed of sequence alignment by searching first for common words or *k*-tuples in the query sequence and each database sequence. Whereas FASTA searches for all possible words of the same length, BLAST confines the search to the words that are the most significant. For proteins, significance is determined by evaluating these word matches using log odds scores in the BLOSUM62 amino acid substitution matrix. For the BLAST algorithm, the word length is fixed at 3 (formerly 4) for proteins and 11 for nucleic acids (3 if the sequences are translated in all six reading frames). This length is the minimum needed to achieve a word score that is high enough to be significant but not so long as to miss short but significant patterns. FASTA theoretically provides a more sensitive search of DNA sequence databases because a shorter word length may be used. Like FASTA, the BLAST algorithm has gone through several developmental stages. The most recent gapped BLAST, or BLAST2, is recommended, as older versions of BLAST are reported to overestimate the significance of database matches (Brenner et al. 1998). The most important recent change is that BLAST reports the significance of a gapped alignment of the query and database sequences. Former versions reported several ungapped alignments, and it was more difficult to evaluate their overall significance. The statistical analysis of sequence alignments that made this change possible is discussed in detail in Chapter 3, page 97.

Steps Used by the BLAST Algorithm

Steps for searching a protein sequence database by a query protein sequence include the following (Altschul et al. 1990, 1994, 1997; BLAST Web server help pages):

1. The sequence is optionally filtered to remove low-complexity regions that are not useful for producing meaningful sequence alignments (see below).

2. A list of words of length 3 in the query protein sequence is made starting with positions 1, 2, and 3; then 2, 3, and 4, etc.; until the last 3 available positions in the sequence are reached (word length 11 for DNA sequences, 3 for programs that translate DNA sequences).
3. Using the BLOSUM62 substitution scores, the query sequence words in step 1 are evaluated for an exact match with a word in any database sequence. The words are also evaluated for matches with any other combination of three amino acids, the object being to find the scores for aligning the query word with any other three-letter word found in a database sequence. There are a total of $20 \times 20 \times 20 = 8000$ possible match scores for this one sequence position. For example, suppose that the three-letter word PQG occurs in the query sequence. The likelihood of a match to itself is found in the BLOSUM62 matrix as the log odds score of a P-P match, plus that for a Q-Q match, plus that for a G-G match $= 7 + 5 + 6 = 18$. These scores are added because the BLOSUM62 matrix is made up of logarithms of odds of finding a match in sequences. To find three consecutive matches, the likelihoods of each pair are multiplied because we are asking that all characters match at the same time—the first pair and the second and the third. Adding logarithms of scores is the equivalent of multiplying the raw odds scores. Similarly, matches of PQG to PEG would score 15, to PRG 14, to PSG 13, and to PQA 12. For DNA words, a match score of +5 and a mismatch score of −4 are used, corresponding to the changes expected in sequences separated by a PAM distance of 40 (see p. 90).
4. A cutoff score called neighborhood word score threshold (*T*) is selected to reduce the number of possible matches to PQG to the most significant ones. For example, if this cutoff score *T* is 13, only the words that score above 13 are kept. In the above example, the list of possible matches to PQG will include PEG (15) but not PQA (12). The list of possible matching words is thereby shortened from 8000 of all possible to the highest scoring number of approximately 50.
5. The above procedure is repeated for each three-letter word in the query sequence. For a sequence of length 250 amino acids, the total number of words to search for is approximately $50 \times 250 = 12{,}500$.
6. The remaining high-scoring words that comprise possible matches to each three-letter position in the query sequence are organized into an efficient search tree for comparing them rapidly to the database sequences.
7. Each database sequence is scanned for an exact match to one of the 50 words corresponding to the first query sequence position, for the words to the second position, and so on. If a match is found, this match is used to seed a possible ungapped alignment between the query and database sequences.
8. (a) In the original BLAST method, an attempt was made to extend an alignment from the matching words in each direction along the sequences, continuing for as long as the score continued to increase, as illustrated below. The extension process in each direction was stopped when the accumulated score stopped increasing and had just begun to fall a small amount below the best score found for shorter extensions. At this point, a larger stretch of sequence (called the HSP or high-scoring segment pair), which has a larger score than the original word, may have been found.

```
        L  P    P  Q  G    L  L  QUERY SEQUENCE
        M  P    P  E  G    L  L  DATABASE SEQUENCE
                <WORD>           THREE LETTER WORD FOUND
                                 INITIALLY
                 7 2 6           BLOSUM62 scores, word
                                 score = 15
          <------    ------>
EXTENSION TO LEFT    EXTENSION TO RIGHT
          2  7    7  2  6    4  4
          <         HSP     >   HSP SCORE = 9 + 15 + 8 = 32
```

(b) In the later version of BLAST, called BLAST2 or gapped BLAST, a different and much more time-efficient method is used. The method starts by making a list of high-scoring matching words, as in steps 1–4 above, with the exception that a lower value of *T*, the word cutoff score, such as 11 in our example, is used. This change results in a longer word list and matches to lower-scoring words in the database sequences. Matches between the query sequence and one database sequence are illustrated below. The x's mark positions of the words with scores at least as high as the new value of *T*. The object is to use these short matched regions lying on the same diagonal and within distance A of each other as the starting points for a longer ungapped alignment between the words. Once found, these joined regions are then extended using the method in part (a). Usually only a few such regions are extended. Because the new matches depend on finding two contiguous words, it is necessary to use a lower value of *T* to maintain the same level of sensitivity for detecting sequence similarity. The newly found diagonals are then scored by summing the scores of the individually matched sequence pairs (Fig. 7.4).

9. The next step is to determine whether each HSP score found by one of the above methods is greater in value than a cutoff score *S*. A suitable value for *S* is determined empirically by examining the range of scores found by comparing random sequences, and by choosing a value that is significantly greater. The high scoring pairs (HSPs) matched in the entire database are identified and listed.
10. BLAST next determines the statistical significance of each HSP score. A probability that two random sequences, one the length of the query sequence and the other the entire length of the database (which is approximately equal to the sum of the lengths of all of the database sequences), could achieve the HSP score is calculated. The topic of sequence statistics is discussed in detail in Chapter 3 and therefore the procedure is only reviewed briefly here. The main problem encountered is that scores between random sequences can reach extremely high values and become higher, the longer the random sequences. The probability p of observing a score S equal to or greater than x is given by the equation,

$$p\,(S \geq x) = 1 - \exp(-e^{-\lambda(x-u)}) \tag{3}$$

where $u = [\log\,(Km'n')]/\lambda$ and where K and λ are parameters that are calculated by BLAST for the amino acid substitution scoring matrix, n' is the effective length of the query sequence, and m' is the effective length of the database sequence. Methods for calculating the parameters K and λ are described in

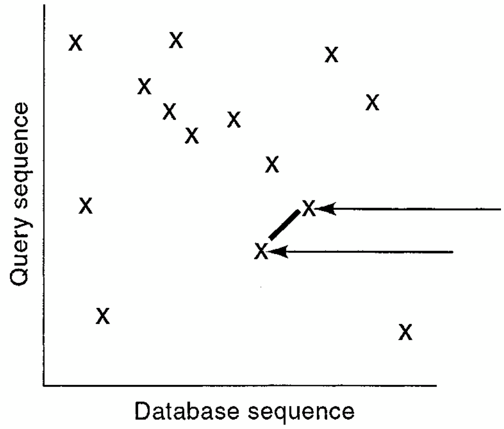


Figure 7.4. Scoring diagonals in BLAST2.

Chapter 3 and in greater detail on the book Web site.

The effective sequence lengths are the actual lengths of the query and database sequences less the average length of an alignment between two random sequences of the same length. m' and n' are calculated from the following relationship:

$$m' \approx m - (\ln Kmn)/H \tag{4}$$

$$n' \approx n - (\ln Kmn)/H \tag{5}$$

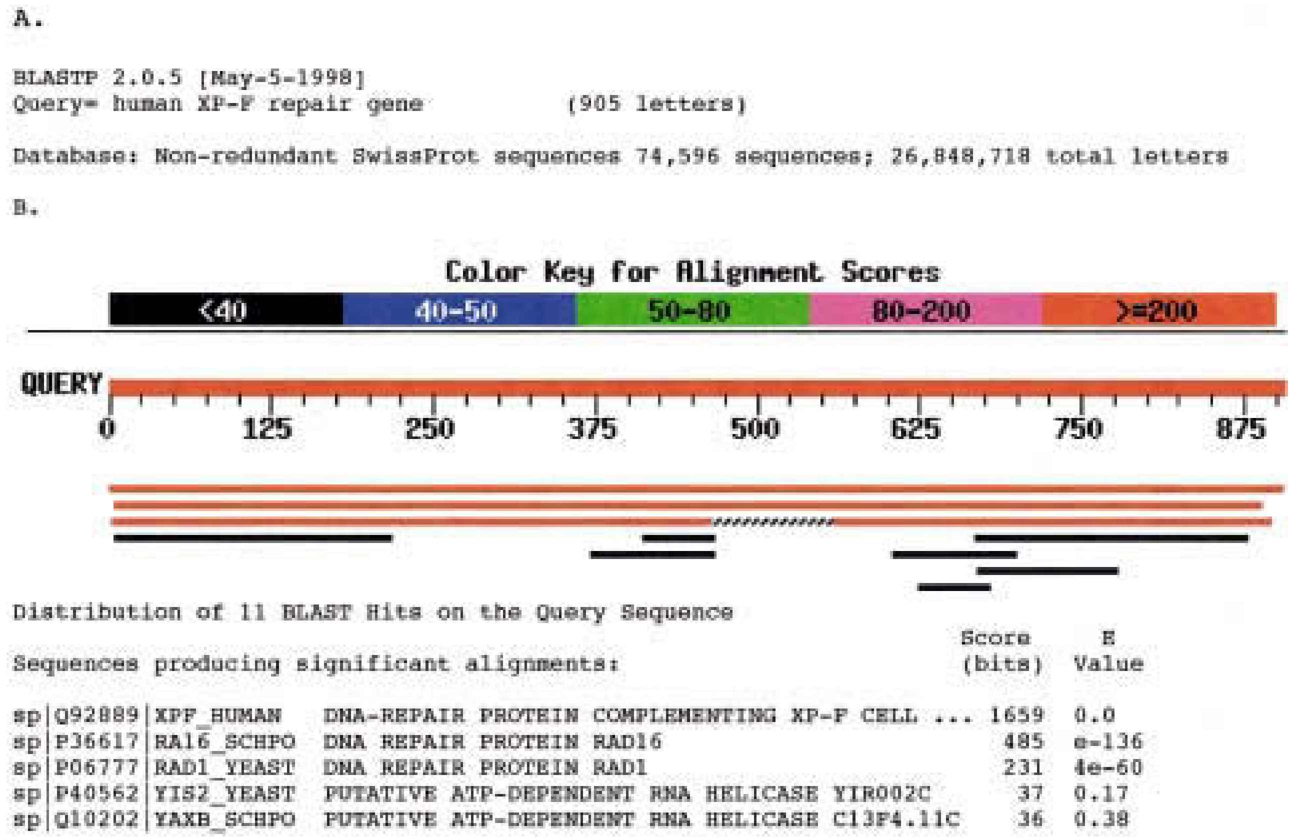
where H is the average expected score per aligned pair of residues in an alignment of two random sequences (Altschul and Gish 1996). H is calculated from the relationship $H = (\ln Kmn)/l$, where l is the average length of the alignment that can be achieved between random sequences of lengths m and n using the same scoring system as used in the database search. l is measured from actual alignments of random sequences. H is similar to the relative entropy of a scoring matrix described in Chapters 3 and 4, except that in this case, H is calculated from alignments of random sequences for a given scoring matrix, usually BLOSUM62. The basis for using these reduced lengths in statistical calculations is that an alignment starting near the end of one of the sequences is likely not to have enough sequence to build an optimal alignment. Using this correction also provides an improved match to statistical theory (Altschul and Gish 1996).

Note that the higher the value of H for a scoring matrix–gap penalty combination, the smaller the correction to the sequence length in Equations 4 and 5. Hence, to obtain alignments with shorter sequences, a scoring system with a higher H value is the most suitable combination. For example, for protein queries in the length range 50–85, the BLAST help pages recommend using BLOSUM80 with gap penalties $(-10,-1)$ instead of BLOSUM62 with gap penalties $(-11,-1)$ because the value of H is higher. To see these recommendations, click on the matrix link on the main BLAST page. For the BLOSUM62 scoring matrix and ungapped alignments, these values are $K = 0.14$ and $\lambda = 0.318$. The probability of the HSP score given by the above equation is adjusted to account for the multiple comparisons performed in the database search. The expectation E of observing a score $S \geq x$ in a database of D sequences is approximately given by the Poisson distribution,

$$E \approx 1 - e^{-p(s > x)\,D} \qquad (6)$$

and for $p < 0.1$, E is approximately pD. The expectation is the chance that a score as high as the one observed between two sequences will be found by chance in a search of a database of size D. Thus, $E = 1$ means that there is a chance that 1 unrelated sequence will be found in the database search. A similar expectation E is calculated by FASTA and SSEARCH.

11. Sometimes, two or more HSP regions that can be made into a longer alignment will be found, thereby providing additional evidence that the query and database sequences are related. In such cases, a combined assessment of the significance will be made. Suppose that two HSP scores are found; one set is 65 and 40, and the second 52 and 45. Which combination of scores is more significant, the one with the highest score (65 versus 52) or the one with the higher of the lower score of each set (45 versus 40)? Two methods have been used by BLAST for calculating this probability (Altschul and Gish 1996). One, the Poisson method, assumes that the probability of the multiple scores is higher when the lower score of each set is higher (45 is better than 40). The other, the sum-of-scores method, calculates the probability of the sum of the scores. In this example, 65 + 40 = 105 is more significant than 52 + 45 = 97. Earlier versions of NCBI-BLAST use the Poisson method; WU-BLAST (Washington University BLAST) and gapped BLAST use the sum-of-scores method. The most recent versions of NCBI-BLAST2 perform a local gapped alignment of the sequences and calculate the expect value of the alignment score. Such calculations became possible when it was realized that a statistical score could be calculated for gapped alignments (see Chapter 3, p. 112; Altschul and Gish 1996). To calculate the significance of the gapped alignment score, values of K and λ are determined on the basis of the alignment scores of random sequences using a combination of scoring matrix and gap penalties, and Equations 3 and 4 are then used.
12. Smith-Waterman local alignments are shown for the query sequence with each of the matched sequences in the database. Earlier versions of BLAST produced only ungapped alignments that included the initially found HSP. If two HSPs were found, two separate alignments were produced because the two regions could not be aligned without gaps. BLAST2 versions produce a single alignment with gaps that can include all of the initially found HSP regions. From the discussion of improvements in the dynamic programming alignment in Chapter 3 and on the book Web site, recall that the procedure of aligning of sequences may be divided into subalignments of the sequences, one starting at some point in sequence 1 and going to the beginning of the sequences, and another starting at the distal ends of the sequences and ending at the same position in sequence 1. A similar method is used to produce an alignment starting with the alignment between the central pair in the highest-scoring region of the HSP pattern as a seed for producing a gapped alignment of the sequences. The score of the alignment is obtained and the expect value for that score is calculated.
13. When the expect score for a given database sequence satisfies the user-selectable threshold parameter E, the match is reported. An example of a BLASTP v2 output file is shown in Figure 7.5.

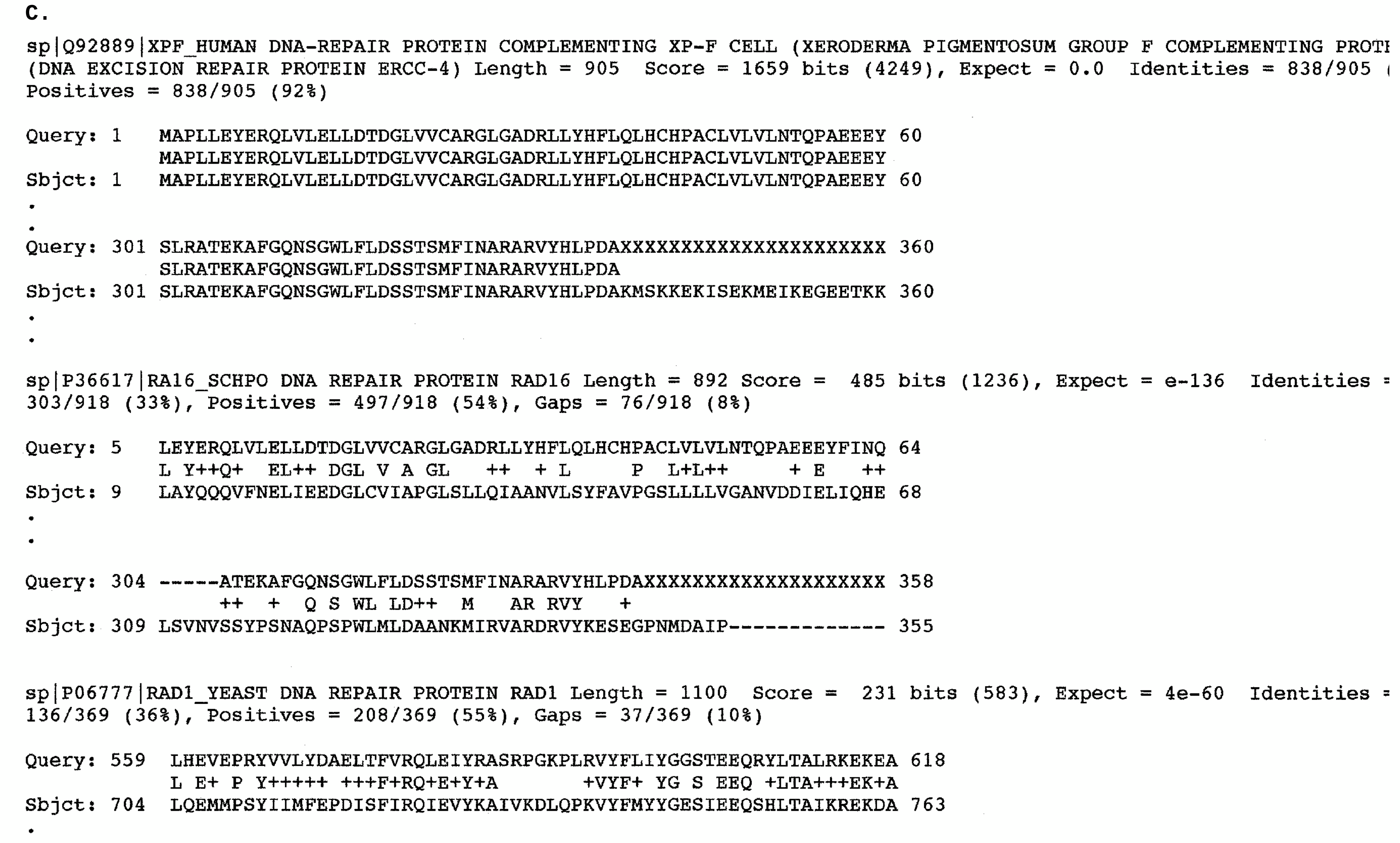


```
A.

BLASTP 2.0.5 [May-5-1998]
Query= human XP-F repair gene         (905 letters)

Database: Non-redundant SwissProt sequences 74,596 sequences; 26,848,718 total letters

B.
```

```
Distribution of 11 BLAST Hits on the Query Sequence
                                                                 Score    E
Sequences producing significant alignments:                     (bits)  Value

sp|Q92889|XPF_HUMAN   DNA-REPAIR PROTEIN COMPLEMENTING XP-F CELL ... 1659   0.0
sp|P36617|RA16_SCHPO  DNA REPAIR PROTEIN RAD16                        485   e-136
sp|P06777|RAD1_YEAST  DNA REPAIR PROTEIN RAD1                         231   4e-60
sp|P40562|YIS2_YEAST  PUTATIVE ATP-DEPENDENT RNA HELICASE YIR002C      37   0.17
sp|Q10202|YAXB_SCHPO  PUTATIVE ATP-DEPENDENT RNA HELICASE C13F4.11C    36   0.38
```

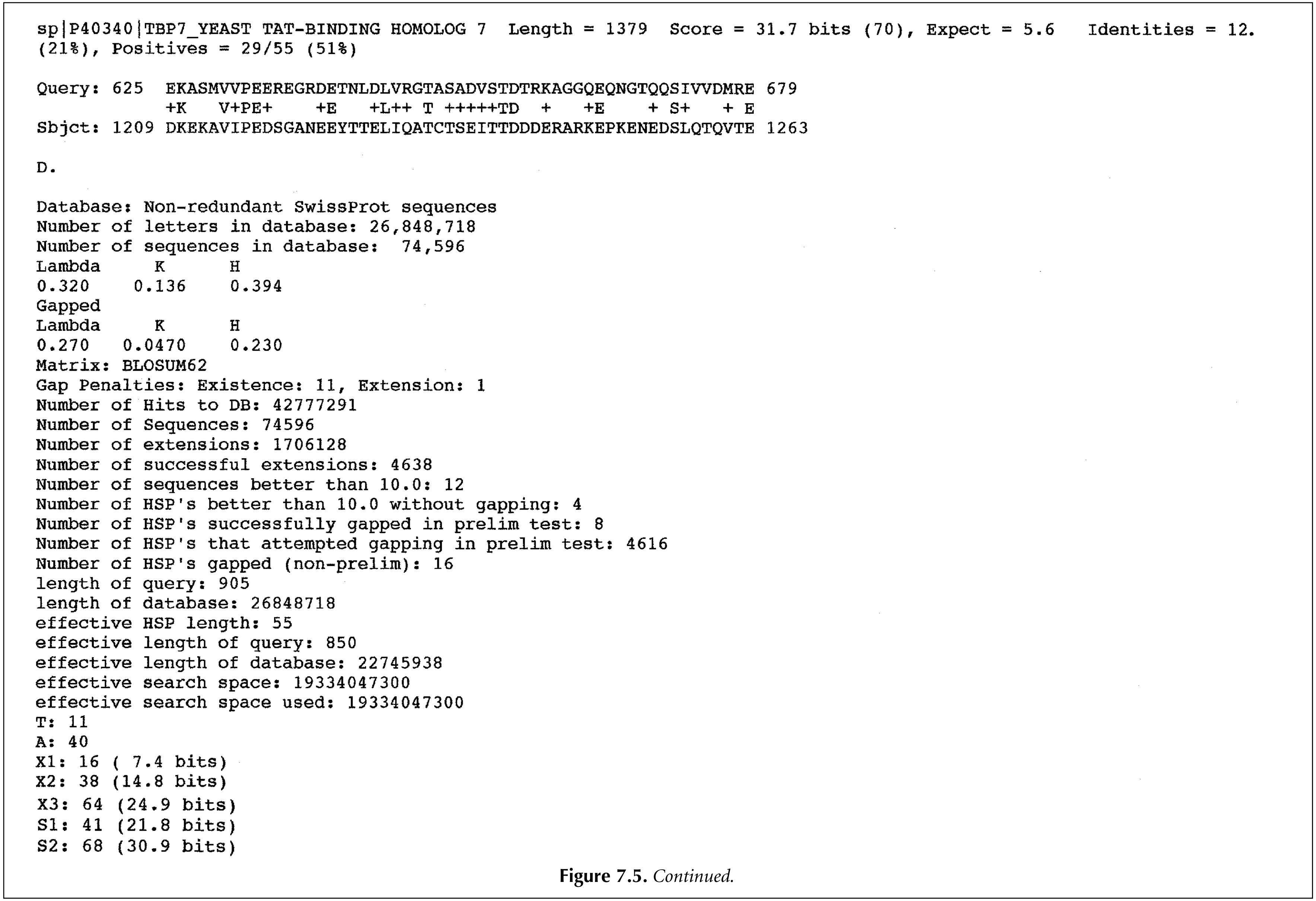
Figure continues on next page.

Figure 7.5. Example of BLASTP output. The BLAST server at http://www.ncbi.nlm.nih.gov/BLAST/, advanced version BLAST2 was given the human XP-F DNA repair sequence in FASTA format (providing the sequence accession number is another option). Program option BLASTP, database option SwissProt, and default program settings (gapped alignment, expectation value = 10, and low-complexity filtering), and 10 descriptions and alignments were chosen. Expectation value is the number of matches expected by chance between the query sequence and random or unrelated database sequences from a database of the size used. If the program is allowed to report all of the matches that it finds, the number found will include at least this many matches with unrelated sequences in the database. The BLAST Web site has excellent help pages that should be consulted, especially when new updates of BLAST have revised Web pages. For example, one revised page did not provide an option for changing the amino acid substitution matrix from the default BLOSUM62 scoring matrix. The comments given below summarize the results of the above BLAST2 search. (*A*) BLAST version number, query sequence, and sequence database are identified. (*B*) First, a graphical representation of the extent to which database sequences match the query sequence is shown. Note that three database sequences can be aligned with the entire length of the query sequence and are therefore likely to be highly significant alignments. Other alignments found are only with portions of the query sequence. The mouse may be used to go directly to the alignments represented in the graph. The scores of the requested 10 highest-scoring database sequences and the 1 identical database sequence are reported, one in each row. Each row includes the database sequence identifier where "sp" indicates a SwissProt match followed by SwissProt accession number and locus name, the score of the alignment in bits (see *C*), and the expectation value (*E*) of the alignment. *E* values of 0.0 and e^{-136} (which is 10^{-136}) in the first and second rows indicate that the match is highly significant. The first match is to the query sequence itself and the next two matches are closely related to the query sequence as indicated by their low *E* scores. If older versions of BLAST that give ungapped alignments or the ungapped option is used, or if the results are from BLASTX and TBLASTN searches, each row may have an additional column displaying *n*, the number of HSPs found and the probability of the sum of these HSP scores, as indicated in step 11 above. (*C*) Gapped alignments between the query sequence and the matched database sequences are shown. The query sequence is named as such and the database sequence is called the subject sequence. Note the filtering of a low-complexity region in the query sequence indicated by the replacement of sequence by X. Gaps are indicated by a dash. Shown in each alignment are the sequence ID and length, and the score of the alignment in bits ("score" is the sum of log odds scores of each matching amino acid pair in the alignment less gap penalties; the raw score in bits is the log odds score in units of logarithms to the base 2). The score shown in the program output is in units of normalized bits = ($\lambda \times$ raw score $- \ln K$) / ln 2. This number is independent of the scoring matrix used, but the raw score in bits is also shown in parentheses. The expectation value *E* of chance matches of unrelated sequences from a database of this size, percent identities in the alignment, percent positives in the alignment (identities plus positive scoring matches in the BLOSUM62 matrix), and percent of the alignment that is gaps are also shown. (*D*) Statistical information about the search is provided, including the numbers found in the steps outlined above. The statistical parameters *K* and λ, which are different for gapped and ungapped alignments (Chapter 3), and the gap penalty scores are also shown. This information will be useful as a basis for adjustment of the basic input parameters.



```
C.
sp|Q92889|XPF_HUMAN DNA-REPAIR PROTEIN COMPLEMENTING XP-F CELL (XERODERMA PIGMENTOSUM GROUP F COMPLEMENTING PROTI
(DNA EXCISION REPAIR PROTEIN ERCC-4) Length = 905  Score = 1659 bits (4249), Expect = 0.0  Identities = 838/905 (
Positives = 838/905 (92%)

Query: 1   MAPLLEYERQLVLELLDTDGLVVCARGLGADRLLYHFLQLHCHPACLVLVLNTQPAEEEY 60
           MAPLLEYERQLVLELLDTDGLVVCARGLGADRLLYHFLQLHCHPACLVLVLNTQPAEEEY
Sbjct: 1   MAPLLEYERQLVLELLDTDGLVVCARGLGADRLLYHFLQLHCHPACLVLVLNTQPAEEEY 60
.
.
Query: 301 SLRATEKAFGQNSGWLFLDSSTSMFINARARVYHLPDAXXXXXXXXXXXXXXXXXXXXXX 360
           SLRATEKAFGQNSGWLFLDSSTSMFINARARVYHLPDA
Sbjct: 301 SLRATEKAFGQNSGWLFLDSSTSMFINARARVYHLPDAKMSKKEKISEKMEIKEGEETKK 360
.
.

sp|P36617|RA16_SCHPO DNA REPAIR PROTEIN RAD16 Length = 892 Score =  485 bits (1236), Expect = e-136  Identities =
303/918 (33%), Positives = 497/918 (54%), Gaps = 76/918 (8%)

Query: 5   LEYERQLVLELLDTDGLVVCARGLGADRLLYHFLQLHCHPACLVLVLNTQPAEEEYFINQ 64
           L Y++Q+  EL++ DGL V A GL   ++  + L     P  L+L++     + E   ++
Sbjct: 9   LAYQQQVFNELIEEDGLCVIAPGLSLLQIAANVLSYFAVPGSLLLLVGANVDDIELIQHE 68
.
.

Query: 304 -----ATEKAFGQNSGWLFLDSSTSMFINARARVYHLPDAXXXXXXXXXXXXXXXXXXXX 358
                ++  +  Q S WL LD++  M   AR RVY   +
Sbjct: 309 LSVNVSSYPSNAQPSPWLMLDAANKMIRVARDRVYKESEGPNMDAIP------------- 355


sp|P06777|RAD1_YEAST DNA REPAIR PROTEIN RAD1 Length = 1100  Score =  231 bits (583), Expect = 4e-60  Identities =
136/369 (36%), Positives = 208/369 (55%), Gaps = 37/369 (10%)

Query: 559  LHEVEPRYVVLYDAELTFVRQLEIYRASRPGKPLRVYFLIYGGSTEEQRYLTALRKEKEA 618
            L E+ P Y+++++ +++F+RQ+E+Y+A       +VYF+ YG S EEQ +LTA+++EK+A
Sbjct: 704  LQEMMPSYIIMFEPDISFIRQIEVYKAIVKDLQPKVYFMYYGESIEEQSHLTAIKREKDA 763
.
```

Figure 7.5. *Continued.*



```
sp|P40340|TBP7_YEAST TAT-BINDING HOMOLOG 7  Length = 1379  Score = 31.7 bits (70), Expect = 5.6   Identities = 12.
(21%), Positives = 29/55 (51%)

Query: 625  EKASMVVPEEREGRDETNLDLVRGTASADVSTDTRKAGGQEQNGTQQSIVVDMRE 679
            +K   V+PE+    +E   +L++ T +++++TD  +   +E    + S+   + E
Sbjct: 1209 DKEKAVIPEDSGANEEYTTELIQATCTSEITTDDDERARKEPKENEDSLQTQVTE 1263

D.

Database: Non-redundant SwissProt sequences
Number of letters in database: 26,848,718
Number of sequences in database:  74,596
Lambda     K      H
0.320    0.136    0.394
Gapped
Lambda     K      H
0.270   0.0470    0.230
Matrix: BLOSUM62
Gap Penalties: Existence: 11, Extension: 1
Number of Hits to DB: 42777291
Number of Sequences: 74596
Number of extensions: 1706128
Number of successful extensions: 4638
Number of sequences better than 10.0: 12
Number of HSP's better than 10.0 without gapping: 4
Number of HSP's successfully gapped in prelim test: 8
Number of HSP's that attempted gapping in prelim test: 4616
Number of HSP's gapped (non-prelim): 16
length of query: 905
length of database: 26848718
effective HSP length: 55
effective length of query: 850
effective length of database: 22745938
effective search space: 19334047300
effective search space used: 19334047300
T: 11
A: 40
X1: 16 ( 7.4 bits)
X2: 38 (14.8 bits)
X3: 64 (24.9 bits)
S1: 41 (21.8 bits)
S2: 68 (30.9 bits)
```

Figure 7.5. *Continued.*

Sequence Filtering

The BLAST programs include a feature for filtering the query sequence through programs that search for low-complexity regions or for sequence repeats. Note that filtering is only applied to the query sequence and not to the database sequence. These regions are marked with an X (protein sequences) or N (nucleic acid sequences) and are then ignored by the BLAST program. Such regions tend to give high scores that do not reflect sequence similarity but rather the occurrence of low-complexity or repetitive sequences. Removing these types of sequences increases emphasis on the more significant database hits. The NCBI programs SEG and PSEG are used for amino acid sequences, and NSEG for nucleic acid sequences (Wootten and Federhen 1993, 1996). The SEG programs are available by anonymous FTP from ncbi.nlm.nih.gov/pub/seg, including documentation. The program DUST is also used for DNA sequences (see http://www.ncbi.nlm.nih.gov/BLAST/filtered.html).

Regions of low-complexity or repetitive sequences may be readily visualized in a dot matrix analysis of a sequence against itself (see Chapter 3, p. 63). Low-complexity regions with a repeat occurrence of the same residue can appear on the matrix as horizontal and vertical rows of dots representing repeated matches of one residue position in one copy of the sequence against a series of the same residue in the second copy. Repeats of a sequence pattern appear in the same matrix as short diagonals of identity that are offset from the main diagonal (see Fig. 3.6). Sequence complexity may also be analyzed by examining the fraction of all possible residues that are represented in a sequence window.

The compositional complexity in a window of sequence of length L is given by (Wootten and Federhen 1996)

$$K = 1/L \log_N (L!/\prod_{\text{all } i} n_i!) \tag{7}$$

where N is 4 for nucleic acid sequences and 20 for protein sequences, and n_i are the numbers of each residue in the window. K will vary from 0 for very low complexity to 1 for high complexity. Thus, complexity is given by:

For the sequence GGGG,

$$L! = 4 \times 3 \times 2 \times 1 = 24$$

$$n_G = 4\ n_C = 0\ n_T = 0\ n_A = 0$$

$$\prod_{\text{All } i} n_i = 4 \times 3 \times 2 \times 1 \times 0! \times 0! \times 0! = 24 \times 1 \times 1 \times 1 = 24$$

$$K = 1/4 \log_4 (24/24) = 0$$

For the sequence CTGA,

$$L! = 24$$

$$n_G = n_C = n_T = n_A = 1$$

$$\prod n_i = 1$$

All $K = 1/4 \log_4 (24/1) = 0.573$

Compositional complexities are sometimes calculated to logarithms to the base 2 to produce scores in bit units. A sliding window (usually 12 residues) is moved along the sequence, and the complexity is calculated at each position. Regions of low complexity are identified, neighboring regions are then combined, and the resulting region is then reduced to a single optimal segment by a minimization procedure. SEG is used for analysis of either proteins or nucleic acids by the above methods. PSEG and NSEG are similar to SEG but are set up for analysis of protein and nucleic acid sequences, respectively. These versatile programs may also be used for locating specific sequence patterns that are characteristic of exons (Chapter 8) or protein structural domains (Chapter 9). In database searches involving comparisons of genomic DNA sequences with EST sequence libraries, use of repeat masking is important for filtering the output to the most significant matches (Claverie 1996).

In addition to low-complexity regions, BLAST will also filter out repeat elements (such as human SINE and LINE retroposons; see Chapter 10). Another filtering program for repeats of periodicity less than 10 residues (XNU; Claverie and States 1993) is used by the BLAST stand-alone programs, but is not available on the NCBI server.

Another Web server, RepeatMasker (http://ftp.genome.washington.edu/cgi-bin/) screens sequence for interdispersed repeats known to be present in mammalian genomes and also can filter out low-complexity regions (A.F.A. Smeet and P. Green, see Web site above). A dynamic programming search program (cross-match, P. Green, see Web site) performs a search of a repeat database with the query sequence (Claverie 1996). A database of repetitive elements (Repbase) maintained at http://www.girinst.org/!server/repbase.html) by the Genetics Information Research Institute (Jurka 1998) can also be used for this purpose.

Other BLAST Programs and Options

There are a number of different versions of the BLAST program for comparing either nucleic acid or protein query sequences with nucleic acid or protein sequence databases. If necessary, the programs translate nucleic acid sequences in all six possible reading frames to compare them to protein sequences. These BLAST programs are shown in Table 7.4 along with the types of alignment, gapped or ungapped, that they produce. Table 7.5 lists the databases that are available, and Table 7.6 lists the options and parameter settings that are available on the BLAST server. These various options may be chosen and are also described on the main BLAST Web page at http://www.ncbi.nlm.nih.gov/. The results produced by a sample BLASTP version 2 output are shown and described in Figure 7.5.

1. BLAST CLIENT (BLASTcl3) is a network-client BLAST that may be established on a local machine and used to access the BLAST2 server (FTP at ncbi.nlm.nih.gov/blast/network/netblast) rather than using a Web browser.
2. Stand-alone BLAST. Executable versions of all of the BLAST programs for Windows, Macintosh, and UNIX platforms are available (FTP at ncbi.nlm.nih.gov/blast).
3. BLAST E-mail server. When the BLAST server is busy so that the interactive Web page is slow and unresponsive, an alternative is to send the job by E-mail and to have the results returned by E-mail. A standard format is required in the E-mail message, as shown in Figure 7.6. The format changes periodically, therefore it is a good idea to send for the current format by sending the message help to the BLAST E-mail server, BLAST@ncbi.nlm.nih.gov. Note that there are obligatory and optional lines in the E-mail message.

Other BLAST-related Programs

1. BLAST-enhanced alignment utility (BEAUTY). BEAUTY adds additional information to BLAST search results, including figures summarizing the information on the locations of HSPs and any already known domains and sites present in the matching

Table 7.4. *BLAST programs provided by the National Center for Biotechnology Information*

Program	Query sequence	Database	Type of alignment[a]
BLASTP	protein	protein	gapped
BLASTN	nucleic acid	nucleic acid	gapped
BLASTX	translated nucleic acid[b]	protein	each frame gapped
TBLASTN	protein	translated nucleic acid[b]	each frame gapped
TBLASTX[c]	translated nucleic acid[b]	translated nucleic acid[b]	ungapped

[a] Type of alignment available between query and database sequences in BLAST2. A gapped alignment is usually preferred, if available. BLASTX and TBLASTN generate gapped alignment for each reading frame found and may use sum statistics. TBLASTX provides only ungapped alignments and sum statistics. Ungapped alignments available as option for BLASTP and BLASTN.

[b] Nucleic acid sequence is translated in all six possible reading frames and then compared to the protein sequence.

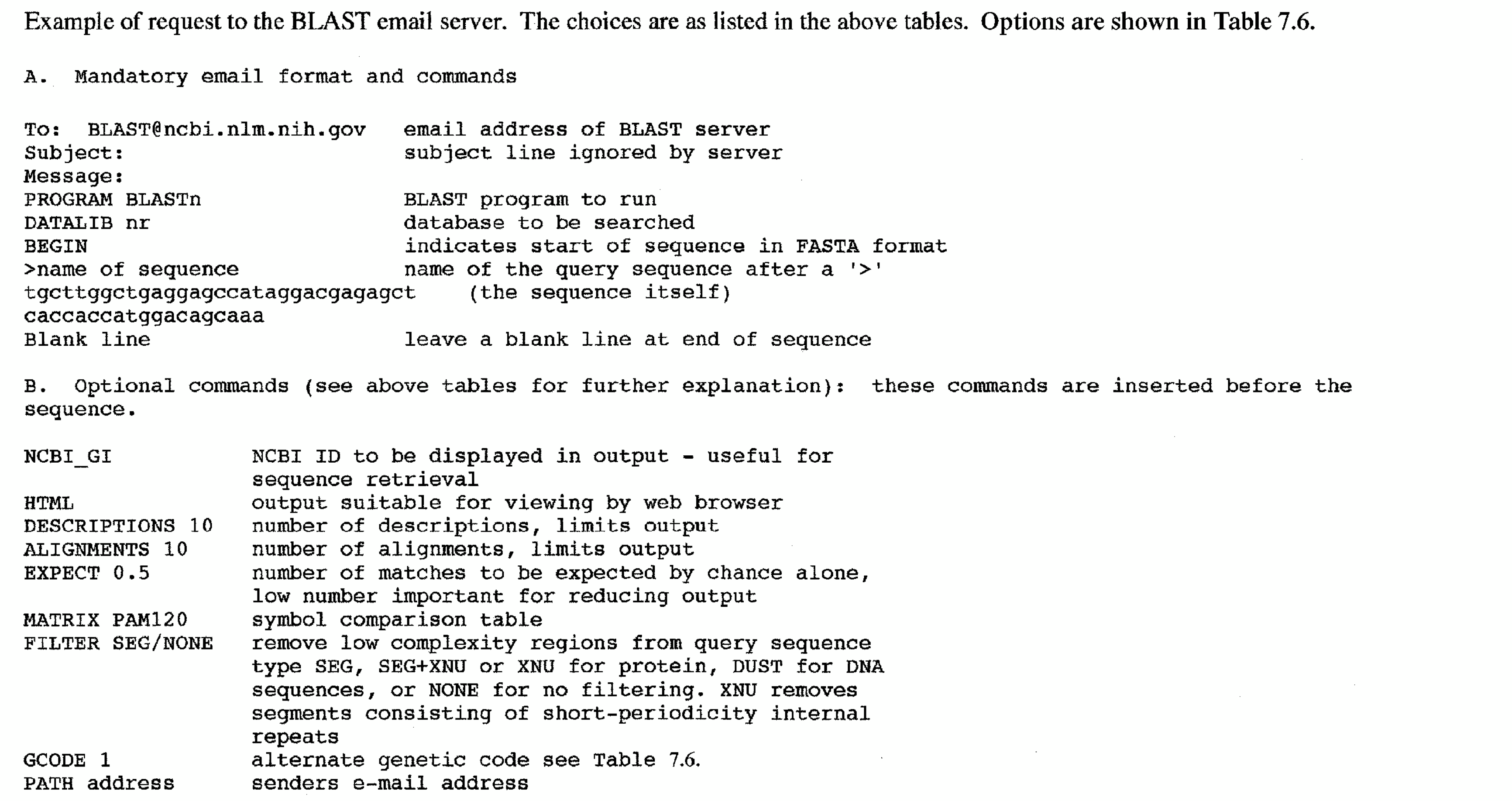
[c] TBLASTX is a heavy user of computer resources and therefore cannot be used with the nr nucleic acid database on the BLAST Web page.

Table 7.5. *Databases available on BLAST Web server*

Database	Description
	A. Peptide sequence databases
nr	translations of GenBank DNA sequences with redundancies removed, PDB, SwissProt, PIR, and PRF
month	new or revised entries or updates to nr in the previous 30 days
swissprot	latest release of the SwissProt protein sequence database[a]
Drosophila genome	provided by Celera and Berkeley Drosophila genome project
yeast	yeast (*Saccharomyces cerevisiae*) genomic sequences
E. coli	*E. coli* genomic sequences
pdb	sequences of proteins of known three-dimensional structure from the Brookhaven Protein Data Bank
yeast	yeast (*S. cerevisiae*) protein sequences
E. coli	*E. coli* genomic coding sequence translations
pdb	sequences of proteins of known three-dimensional structure from the Brookhaven Protein Data Bank
kabat [kabatpro]	Kabat's database of sequences of immunological interest
alu	translations of select *Alu* repeats from REPBASE, a database of sequence repeats
	B. Nucleotide sequence databases
nr	GenBank, EMBL, DDBJ, and PDB sequences with redundancies removed (EST, STS, GSS, and HTGS sequences excluded)
month	new or revised entries or updates to nr in the previous 30 days
dbest[b]	EST sequences from GenBank, EMBL, and DDBJ with redundancies removed
dbsts[b]	STS sequences from GenBank, EMBL, and DDBJ with redundancies removed
htgs[b]	high-throughput genomic sequences
kabat [kabatnuc]	Kabat's database of sequences of immunological interest
vector	vector subset of GenBank
mito	database of mitochondrial sequences
alu	select *Alu* repeats from REPBASE, a database of sequence repeats; suitable for masking *Alu* repeats from query sequences
epd	eukaryotic promoter database
gss[b]	genome survey sequences, includes single-pass genomic data, exon-trapped sequences, and *Alu* PCR sequences

[a] The SwissProt database is carefully curated but not always up to date because updates are released after longer intervals. SwissProt and PIR are the preferred protein databases for searches because the nr protein database is a composite of several databases and has duplicates of many sequences. Unfortunately, PIR is not provided as a separate choice on the database menu.

[b] Databases containing sequences that may have been less accurately determined.

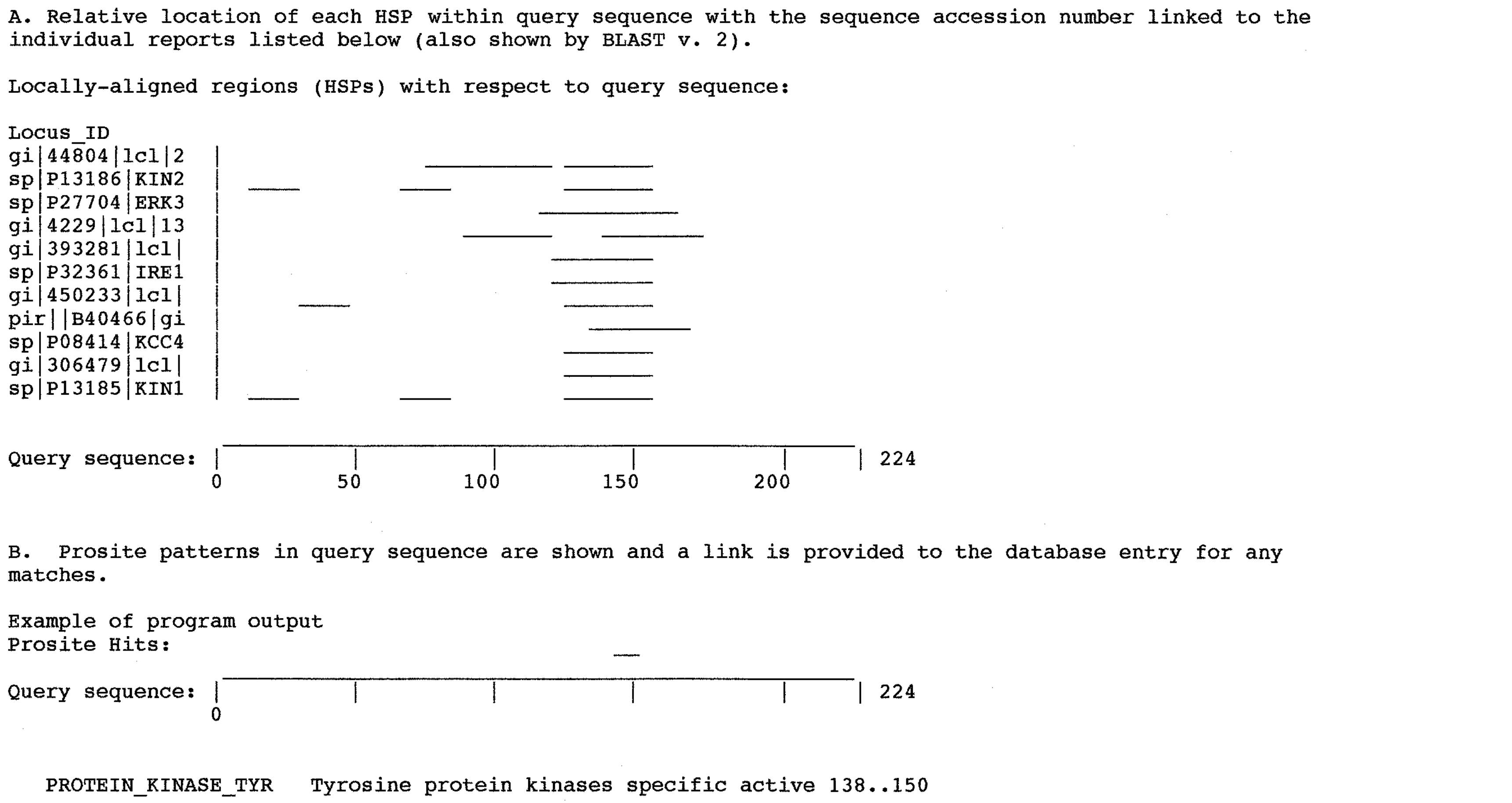


Example of request to the BLAST email server. The choices are as listed in the above tables. Options are shown in Table 7.6.

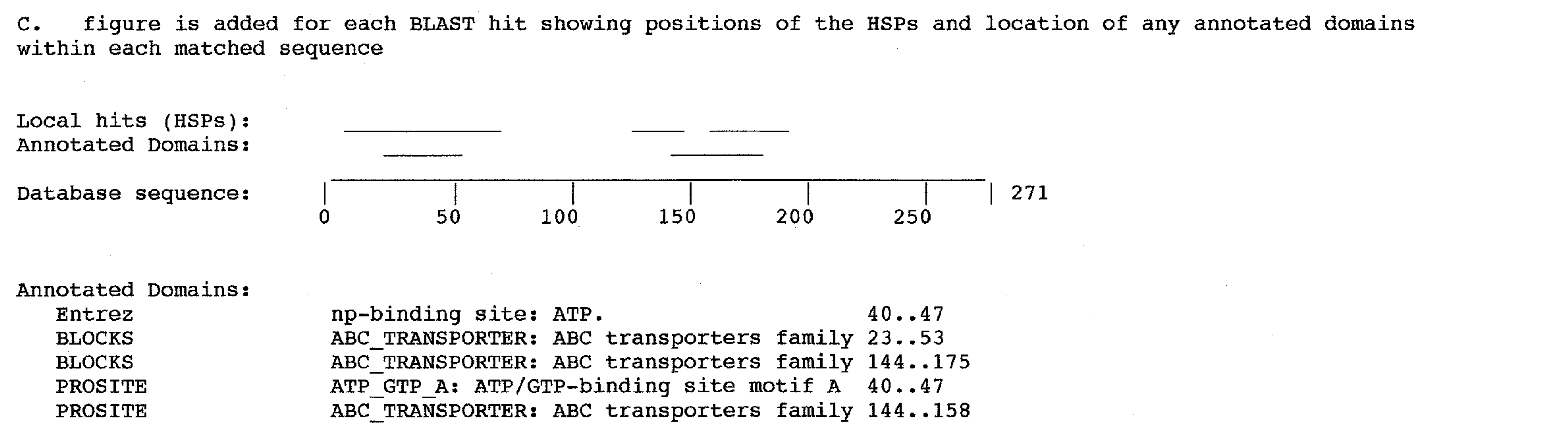
```
A.  Mandatory email format and commands

To:  BLAST@ncbi.nlm.nih.gov   email address of BLAST server
Subject:                      subject line ignored by server
Message:
PROGRAM BLASTn                BLAST program to run
DATALIB nr                    database to be searched
BEGIN                         indicates start of sequence in FASTA format
>name of sequence             name of the query sequence after a '>'
tgcttggctgaggagccataggacgagagct    (the sequence itself)
caccaccatggacagcaaa
Blank line                    leave a blank line at end of sequence

B.  Optional commands (see above tables for further explanation):  these commands are inserted before the
sequence.

NCBI_GI             NCBI ID to be displayed in output - useful for
                    sequence retrieval
HTML                output suitable for viewing by web browser
DESCRIPTIONS 10     number of descriptions, limits output
ALIGNMENTS 10       number of alignments, limits output
EXPECT 0.5          number of matches to be expected by chance alone,
                    low number important for reducing output
MATRIX PAM120       symbol comparison table
FILTER SEG/NONE     remove low complexity regions from query sequence
                    type SEG, SEG+XNU or XNU for protein, DUST for DNA
                    sequences, or NONE for no filtering. XNU removes
                    segments consisting of short-periodicity internal
                    repeats
GCODE 1             alternate genetic code see Table 7.6.
PATH address        senders e-mail address
```

Figure 7.6. Example of request to the BLAST E-mail server.

Example: BEAUTY program output



```
A. Relative location of each HSP within query sequence with the sequence accession number linked to the
individual reports listed below (also shown by BLAST v. 2).

Locally-aligned regions (HSPs) with respect to query sequence:

Locus_ID
gi|44804|lcl|2  |                __________ _______
sp|P13186|KIN2  |  ____        ____         _______
sp|P27704|ERK3  |                         ___________
gi|4229|lcl|13  |                   _______    ________
gi|393281|lcl|  |                          ________
sp|P32361|IRE1  |                          ________
gi|450233|lcl|  |      ____                 _______
pir||B40466|gi  |                             ________
sp|P08414|KCC4  |                           _______
gi|306479|lcl|  |                           _______
sp|P13185|KIN1  |  ____        ____         _______

                 __________________________________________________
Query sequence: |          |          |          |           |     | 224
                0         50        100        150         200


B.  Prosite patterns in query sequence are shown and a link is provided to the database entry for any
matches.

Example of program output
Prosite Hits:                                   __
                 __________________________________________________
Query sequence: |          |          |          |           |     | 224
                0


   PROTEIN_KINASE_TYR   Tyrosine protein kinases specific active 138..150
```



```
C.   figure is added for each BLAST hit showing positions of the HSPs and location of any annotated domains
within each matched sequence


Local hits (HSPs):       ____________          ____  ______
Annotated Domains:          ______                ________
                        __________________________________________________
Database sequence:      |         |        |        |        |        |    | 271
                        0       50      100      150      200      250


Annotated Domains:
   Entrez               np-binding site: ATP.                    40..47
   BLOCKS               ABC_TRANSPORTER: ABC transporters family 23..53
   BLOCKS               ABC_TRANSPORTER: ABC transporters family 144..175
   PROSITE              ATP_GTP_A: ATP/GTP-binding site motif A  40..47
   PROSITE              ABC_TRANSPORTER: ABC transporters family 144..158
```

Figure 7.7. Example of BEAUTY output.

Table 7.6. *Options and parameter settings available on the BLAST server*

Parameter	Range of choices or values	Function
Descriptions	0–500	number of matching sequences to report
Alignments	0–500	number of alignments to show
Expect	0.001–1000	number of matches from unrelated sequences expected by chance from the selected database smaller values decrease chance of reporting of such matches
Filter	yes or no	removes regions of low sequence complexity from the query sequence because they can give misleading high scoring matches
NCBI-gi	yes or no	gi identifier shown in output
Genetic code	various codon use tables[a]	for translation of nucleic acid sequences
Graphical overview	yes or no	useful display of matches to the query sequence mouse may be used to show alignment
Advanced options	—	type into space provided[b]

[a] Codon tables include standard, vertebrate mitochondrial, yeast mitochondrial, mold mitochondrial, invertebrate mitochondrial ciliate nuclear, echinoderm mitochondrial, euplotid nuclear, bacterial, alternative yeast nuclear, ascidian mitochondrial, flatworm mitochondrial, and blepharisma macronuclear. These are numbered 1–15, respectively, for E-mail access.

[b] Options include (where *n* is an integer 0,1,2, . . .): −*G n*, penalty or cost to open a gap; −*E n*, penalty to extend a gap; −*q n* penalty for a mismatch in BLASTN; −*r n*, match score in BLASTN; −*W n*, initial word size; −v n, number of descriptions; *b n*, number of alignments to show; and –*E r*, expect value where r is a real number such as 10.0. For example, to set the gap opening penalty to 10 and the gap extension penalty to 2, click the mouse on the advanced options form and then type −G 10 −E 2. For more advanced searches of the entire proteome of an organism using stand-alone BLAST on a local machine, additional options must be used, for example, effective database size, to obtain reliable statistical results.

database sequences (Worley et al. 1995). To make this enhanced type of analysis possible, a database of domains and sites was created for use with the BEAUTY program. A new database of sequence domains and sites was made showing for each sequence in ENTREZ the possible location of patterns in the Prosite catalog, the BLOCKS database, and the PRINTS protein fingerprint database. This information is displayed in the following example of the program output (Fig. 7.7). The BEAUTY program is accessible on the BCM Search Launcher (http://dot.imgen.bcm.tmc.edu:9331/seq-search/protein-search.html).

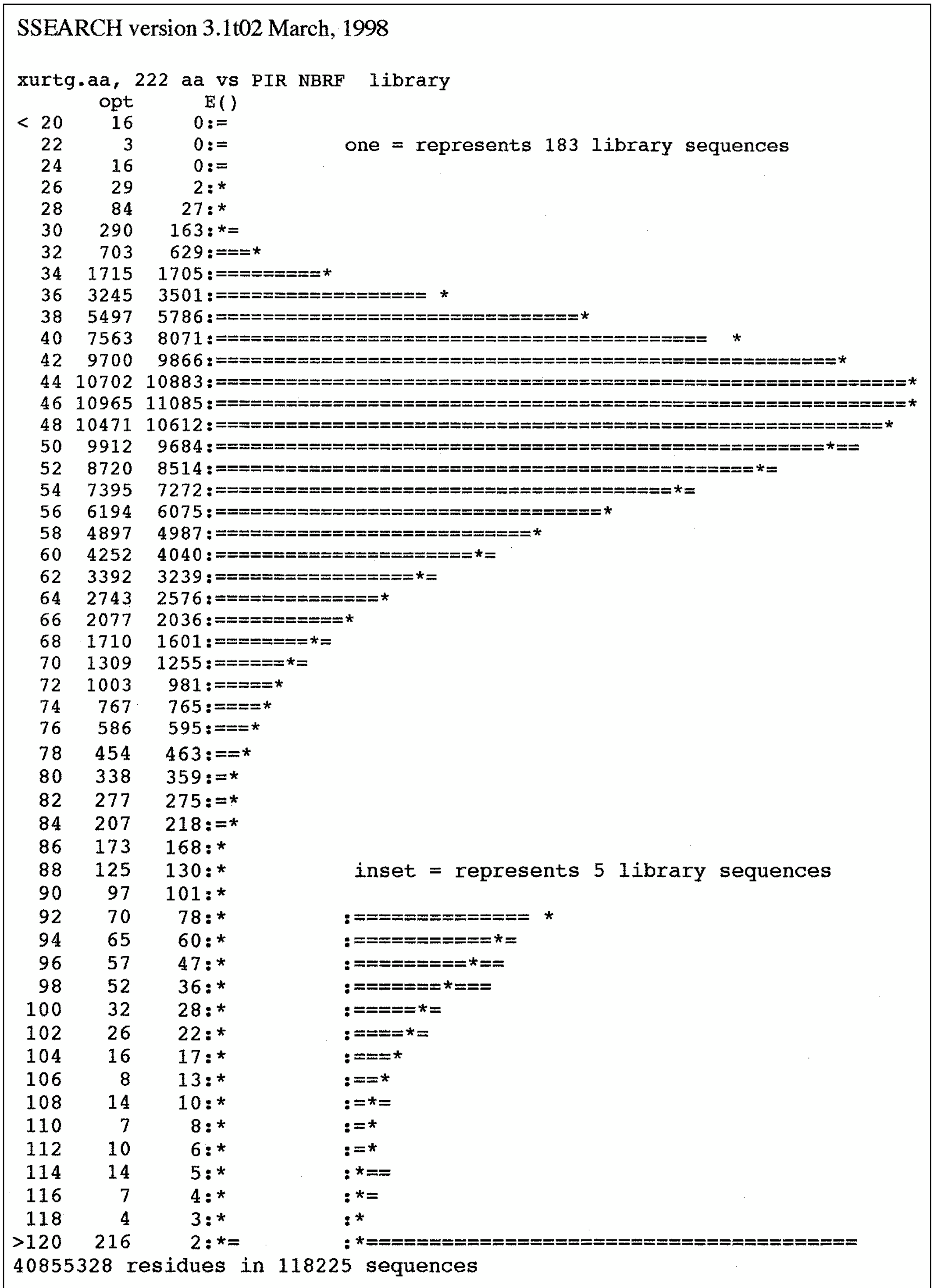
2. BLAST searching with a Cobbler sequence. The BLOCKS server (http://www.blocks.fhcrc.org) offers a variety of BLAST searches that use as a query sequence a consensus sequence derived from multiple sequence alignment of a set of related proteins. This consensus sequence, called a Cobbler sequence (Henikoff and Henikoff 1997), is used to focus the search on residues that are in the majority in each column of the multiple sequence alignment, rather than on any one particular sequence. Hence, the search may detect additional database sequences with variation unlike that found in the original sequences, yet still representing the same protein family. An example of a Cobbler sequence is shown in the BLOCKS search example on page 325**.**
3. BLAST2. This program uses the BLASTP or BLASTN algorithms for aligning two sequences and may be reached at http://www.ncbi.nlm.nih.gov/gorf/bl2.html. This site should be useful for aligning very long sequences, but sequences >150 kb are not recommended.

DATABASE SEARCHES WITH THE SMITH-WATERMAN DYNAMIC PROGRAMMING METHOD

The objective of similarity searching in a sequence database is to discover as many sequences as possible that are similar to the query sequence. For proteins, the resulting collection of sequences may represent a sequence family. Because there may be $< 20\%$ amino acid identity (an alignment has this many identical residues) between some family members, finding such distant relatives is a difficult task. The aforementioned programs, FASTA and BLAST, are designed to find database sequences related to a query sequence rapidly and with high reliability. They achieve their speed by searching first for short identical patterns in the query sequence and each database sequence and then by aligning the sequences starting at these patterns. Because patterns are very often found in related sequences, the methods work most of the time. FASTA and BLAST are not based on an algorithm that guarantees the best or optimal alignment, but instead on a heuristic method that works most of the time in practice; thus, they may fail to detect some distant sequence relationships.

The Smith-Waterman dynamic programming algorithm discussed in Chapter 3 is mathematically designed to provide the best or optimal local alignment between two sequences and is therefore expected to be the most reliable method for finding family members in a database search. Several studies discussed below have shown that such is the case. The disadvantage of using dynamic programming is that it is 50–100 times slower than FASTA and BLAST, and until recently a search could take up to several hours on a typical medium-sized machine. With the advent of faster and more powerful computers and improvements in the dynamic programming algorithm discussed in Chapter 3 (and on the book Web site), it is now possible to perform database searches in an hour or less. Some institutions have gone so far as to establish a powerful system of several computers linked together in a parallel architecture that allows a search to be performed within minutes. Several of these sites listed below offer public access through the Web (Table 7.5). It is important to examine the site for use of up-to-date databases and use of an appropriate statistical analysis. Detection of distant sequence relationships depends on use of the statistical methods that have been developed for BLAST and FASTA. For routine use of dynamic programming methods for database searches, establishing the program SSEARCH (FTP to ftp.virginia.edu/pub/fasta; Pearson 1991; Pearson and Miller 1992) and the appropriate sequence databases on a local UNIX server is recommended.

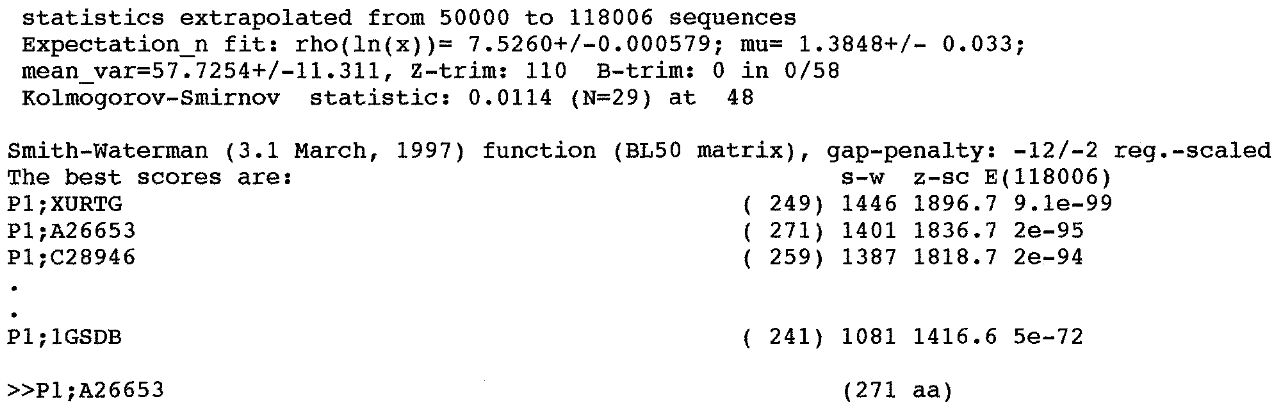
In several studies (Pearson 1995, 1996, 1998; Agarwal and States 1998; Brenner et al. 1998), it has been shown that using SSEARCH, which is based on the Smith-Waterman dynamic programming algorithm, is more suitable for identifying related proteins of limited sequence similarity than FASTA and BLAST in a database search. In several of these studies, known members of protein families are used as a query sequence searching for the remaining members in a protein sequence database. In another study, the performance of the sequence analysis methods was determined using protein sequences of known structural relationships (Brenner et al. 1998). The results are presented in terms of the sensitivity and selectivity of the algorithm, or the ability to identify correct family members, including some that are only weakly similar, without incorrectly identifying other unrelated proteins as members (Pearson 1995, 1998). The ability to discriminate true from false matches depends on the use of appropriate amino acid substitution matrices, gap opening and extension penalties that provide local alignments, and a careful statistical analysis of the search results using the extreme value distribution to predict scores from unrelated sequences (Brenner et al. 1998; Pearson 1998). The program SSEARCH has the necessary



SSEARCH version 3.1t02 March, 1998

```
xurtg.aa, 222 aa vs PIR NBRF  library
       opt      E()
< 20    16     0:=
  22     3     0:=          one = represents 183 library sequences
  24    16     0:=
  26    29     2:*
  28    84    27:*
  30   290   163:*=
  32   703   629:===*
  34  1715  1705:=========*
  36  3245  3501:================== *
  38  5497  5786:===============================*
  40  7563  8071:==========================================  *
  42  9700  9866:=====================================================*
  44 10702 10883:===========================================================*
  46 10965 11085:===========================================================*
  48 10471 10612:=========================================================*
  50  9912  9684:====================================================*==
  52  8720  8514:==============================================*=
  54  7395  7272:=======================================*=
  56  6194  6075:=================================*
  58  4897  4987:===========================*
  60  4252  4040:======================*=
  62  3392  3239:=================*=
  64  2743  2576:==============*
  66  2077  2036:===========*
  68  1710  1601:========*=
  70  1309  1255:======*=
  72  1003   981:=====*
  74   767   765:====*
  76   586   595:===*
  78   454   463:==*
  80   338   359:=*
  82   277   275:=*
  84   207   218:=*
  86   173   168:*
  88   125   130:*         inset = represents 5 library sequences
  90    97   101:*
  92    70    78:*        :============== *
  94    65    60:*        :============*=
  96    57    47:*        :=========*==
  98    52    36:*        :=======*===
 100    32    28:*        :=====*=
 102    26    22:*        :====*=
 104    16    17:*        :===*
 106     8    13:*        :==*
 108    14    10:*        :=*=
 110     7     8:*        :=*
 112    10     6:*        :=*
 114    14     5:*        :*==
 116     7     4:*        :*=
 118     4     3:*        :*
>120   216     2:*=       :*=======================================
40855328 residues in 118225 sequences
```

Figure 7.8. *Figure continues on next page.*



```
 statistics extrapolated from 50000 to 118006 sequences
 Expectation_n fit: rho(ln(x))= 7.5260+/-0.000579; mu= 1.3848+/- 0.033;
 mean_var=57.7254+/-11.311, Z-trim: 110  B-trim: 0 in 0/58
 Kolmogorov-Smirnov  statistic: 0.0114 (N=29) at  48

Smith-Waterman (3.1 March, 1997) function (BL50 matrix), gap-penalty: -12/-2 reg.-scaled
The best scores are:                                      s-w  z-sc E(118006)
P1;XURTG                                           ( 249) 1446 1896.7 9.1e-99
P1;A26653                                          ( 271) 1401 1836.7 2e-95
P1;C28946                                          ( 259) 1387 1818.7 2e-94
.
.
P1;1GSDB                                           ( 241) 1081 1416.6 5e-72

>>P1;A26653                                               (271 aa)
```

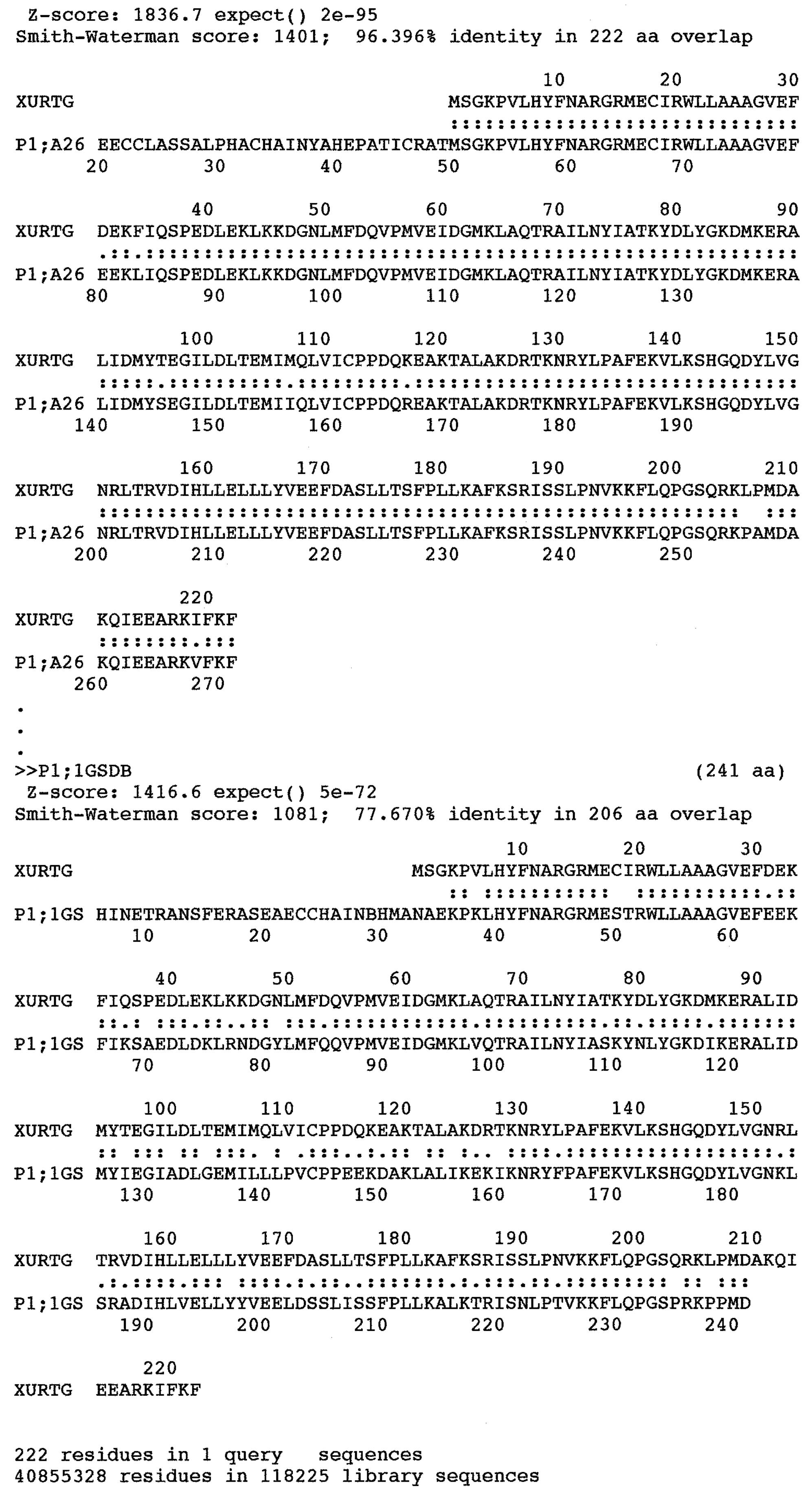
Figure 7.8. Example of SSEARCH. The PIR database was searched with the rat glutathione transferase sequence (EC 2.5.1.18). PIR was used to avoid multiple reports of the same sequence that is obtained with combined databases such as Genpept. SSEARCH was obtained from ftp.virginia.edu/FASTA and compiled on a local UNIX server. The PIR database was accessed by the program from the Genetics Computer Group sequence libraries, which are locally available. SSEARCH was run in the UNIX command line mode since a Web page interface was not available. The program output is very similar to that of FASTA which is described in detail in Fig. 7.3. Note that, if not specified otherwise, SSEARCH uses the BLOSUM50 scoring matrix with gap penalties −12/−2. Like FASTA, the program calculates the statistical parameters λ and *K* from the alignment scores calculated for 50,000 unrelated sequences, and then uses these parameters to calculate the *E*-value scores of the alignment scores with related sequences. The *z* values are calculated from a linear regression of the scores against the logarithm of the sequence length, and deviations from this line are converted to standard *z* scores, as described for FASTA. The glutathione transferases are a large and diverse group of sequences, some of which share very little sequence similarity with the others (Pearson 1996). The large number of normalized scores >120 indicates that a large number of related sequences were found in PIR. Only a few of the alignments are shown, and the alignment of the query sequence with itself is omitted. *Figure continues on next page.*

features and is available for database searches. The reliability of the statistical scores reported by FASTA, BLAST2, and SSEARCH has been determined using sequences of known structural relatedness as a guide. The *E*-value scores reported by FASTA and SSEARCH are reliable, with the number of false positives agreeing with the scores. BLAST2 *E*-value scores also appear to be reliable (see Brenner et al. 1998).

An example of an SSEARCH vers. 3 database search is given in Figure 7.8. Several guest Web sites for performing a database search with the Smith-Waterman dynamic programming algorithm are listed in Table 7.7.

DATABASE SEARCHES WITH THE BAYES BLOCK ALIGNER

From the discussion so far, it is apparent that the fastest and most convenient way to perform sequence database searches is with the FASTA and BLAST2 programs. The much slower Smith-Waterman dynamic programming programs, such as SSEARCH, may find more distantly related sequences. The significance of the alignment scores can be accurately evaluated by these programs. A even better method for detection of distant sequence relationships has been described; this is the Bayes block aligner (Zhu et al. 1998), which was previously discussed in Chapter 3 (p. 126). This program requires several series of computational steps roughly proportional to the product of the sequence lengths and is therefore considerably slower than SSEARCH. As an indication of length of time required, alignment of two standard-sized proteins scoring 7 blocks with all available BLOSUM or PAM matrices on the author's 500 MHz laptop with 500 megabytes of memory running the Linux operating system took less than 10 seconds.



```
 Z-score: 1836.7 expect() 2e-95
Smith-Waterman score: 1401;  96.396% identity in 222 aa overlap

                                             10        20        30
XURTG                                MSGKPVLHYFNARGRMECIRWLLAAAGVEF
                                     ::::::::::::::::::::::::::::::
P1;A26 EECCLASSALPHACHAINYAHEPATICRATMSGKPVLHYFNARGRMECIRWLLAAAGVEF
      20        30        40        50        60        70

               40        50        60        70        80        90
XURTG  DEKFIQSPEDLEKLKKDGNLMFDQVPMVEIDGMKLAQTRAILNYIATKYDLYGKDMKERA
       .::.::::::::::::::::::::::::::::::::::::::::::::::::::::::::
P1;A26 EEKLIQSPEDLEKLKKDGNLMFDQVPMVEIDGMKLAQTRAILNYIATKYDLYGKDMKERA
      80        90       100       110       120       130

              100       110       120       130       140       150
XURTG  LIDMYTEGILDLTEMIMQLVICPPDQKEAKTALAKDRTKNRYLPAFEKVLKSHGQDYLVG
       :::::.::::::::::.:::::::::.:::::::::::::::::::::::::::::::::
P1;A26 LIDMYSEGILDLTEMIIQLVICPPDQREAKTALAKDRTKNRYLPAFEKVLKSHGQDYLVG
     140       150       160       170       180       190

              160       170       180       190       200       210
XURTG  NRLTRVDIHLLELLLYVEEFDASLLTSFPLLKAFKSRISSLPNVKKFLQPGSQRKLPMDA
       :::::::::::::::::::::::::::::::::::::::::::::::::::::::  :::
P1;A26 NRLTRVDIHLLELLLYVEEFDASLLTSFPLLKAFKSRISSLPNVKKFLQPGSQRKPAMDA
     200       210       220       230       240       250

              220
XURTG  KQIEEARKIFKF
       ::::::::.:::
P1;A26 KQIEEARKVFKF
     260       270
.
.
.
>>P1;1GSDB                                                (241 aa)
 Z-score: 1416.6 expect() 5e-72
Smith-Waterman score: 1081;  77.670% identity in 206 aa overlap

                                          10        20        30
XURTG                             MSGKPVLHYFNARGRMECIRWLLAAAGVEFDEK
                                     :: :::::::::::  :::::::::::.::
P1;1GS HINETRANSFERASEAECCHAINBHMANAEKPKLHYFNARGRMESTRWLLAAAGVEFEEK
          10        20        30        40        50        60

            40        50        60        70        80        90
XURTG  FIQSPEDLEKLKKDGNLMFDQVPMVEIDGMKLAQTRAILNYIATKYDLYGKDMKERALID
       ::.: :::.::..:: :::.::::::::::::.::::::::::.::.:::::.:::::::
P1;1GS FIKSAEDLDKLRNDGYLMFQQVPMVEIDGMKLVQTRAILNYIASKYNLYGKDIKERALID
          70        80        90       100       110       120

           100       110       120       130       140       150
XURTG  MYTEGILDLTEMIMQLVICPPDQKEAKTALAKDRTKNRYLPAFEKVLKSHGQDYLVGNRL
       :: ::: :: :::. : .:::..:.:: :: :.. ::::.::::::::::::::::::.:
P1;1GS MYIEGIADLGEMILLLPVCPPEEKDAKLALIKEKIKNRYFPAFEKVLKSHGQDYLVGNKL
         130       140       150       160       170       180

           160       170       180       190       200       210
XURTG  TRVDIHLLELLLYVEEFDASLLTSFPLLKAFKSRISSLPNVKKFLQPGSQRKLPMDAKQI
       .:.::::.::: ::::.:.::..:::::::.:.:::.::.::::::::: :: :::
P1;1GS SRADIHLVELLYYVEELDSSLISSFPLLKALKTRISNLPTVKKFLQPGSPRKPPMD
         190       200       210       220       230       240

           220
XURTG  EEARKIFKF


222 residues in 1 query   sequences
40855328 residues in 118225 library sequences
```

Figure 7.8. *Continued.*

Table 7.7. *Examples of guest Web sites for performing a database search based on the Smith-Waterman dynamic programming algorithm*

Server/program	Reference	Web address
BCM Search Launcher (with programming links to several servers)	Baylor College of Medicine	http://dot.imgen.bcm.tmc.edu:9331/ seq-search/protein-search.html
bic-sw[a]	Bic server European Bioinformatics Institute	http://www.ebi.ac.uk/bic_sw/
Mpsearch[b]	National Institute of Agrobiological Resources, Tsukuba, Japan	http://www.dna.affrc.go.jp/htbin/mp_PP.pl
Scanps	G.Barton, European Bioinformatics Institute	http://barton.ebi.ac.uk; http://www.ebi.ac.uk/scanps
SSEARCH E-mail server	DNA Databank of Japan	http://www.ddbj.nig.ac.jp/E-mail/homology.html
Swat[c]	Phil Green, University of Washington	http://www.genome.washington.edu/UWGC/ analysistools/swat.htm

A comprehensive list of servers for these types of analyses may be found at http://www.sdsc.edu/ResTools/biotools/biotools1.html.

[a] Bic-sw provides a combination of amino acid scoring matrix and gap penalties and also length-normalized *z* scores (similar to FASTA and BLAST) which are most appropriate for resolving more distantly related sequences.

[b] MPSearch is an extremely fast implementation of the Smith-Waterman dynamic programming algorithm by J.F. Collins and S. Sturrock, Biocomputing Resource Unit, the University of Edinburgh, distribution rights by Oxford Molecular Ltd. An E-mail server is at http://www.gen-info.osaka-u.ac.jp/. Some versions of the Mpsearch algorithm at this site use the same penalty for all gaps, others use gap opening and extension penalties. The former is designed to find similar sequences in which gaps are less important in the alignment, the latter the more distant sequence alignments. An on-line manual is available at http://www.dna.affrc.go.jp/htdocs/MPsrch/MPsrchMain.html. Current versions of these programs rank the sequences found by two kinds of scoring systems. A statistical analysis is performed but the scores do not appear to be length-normalized. Hence, the sensitivity of the program may not exceed that shown by FASTA (Pearson 1996).

[c] Includes Smith-Waterman and Needleman-Wunsch search algorithms. Calculates statistical significance using extreme value statistics (like FASTA and BLAST).

Evaluation of programs for finding related proteins is usually based on searches in databases for families using sequence similarity (Pearson 1998). A more difficult type of evaluation is based on the searches of structural databases (Brenner et al. 1998). In these databases, discussed in Chapter 9, the sequences have been organized into families having similar three-dimensional structures. Three of these databases representing groups of proteins that have less than 25%, 35%, or 45% identities (Hobohm et al. 1992) were searched using representatives of structural families in each. In each case, the block aligner slightly but significantly outperformed SSEARCH in finding structural relatives. For example, at the 1% false-positive level, the Bayes block aligner found an average of 14.4% of the proteins in the less-than-25% identity group, whereas SSEARCH with usual scoring matrix, gap penalties, and statistical score options found 12.9%, a difference of 1.5%. In addition, the Bayes block aligner can align sequences that have very little similarity but provide alignments that closely match those found by a careful structural analysis described in Chapter 9 using the VAST program (Madej et al. 1995). A similar study (Brenner et al. 1998) compared the ability of BLAST2, FASTA, and SSEARCH to identify proteins in the families of the SCOP structural database (Murzin et al. 1995, and see Chapter 9).

A Web page describing Bayesian bioinformatics and the source of the Bayes block aligner software is located at http://www.wadsworth.org/resnres/bioinfo/.

The Bayes block aligner uses a new method for producing sequence alignments. The method, discussed in detail in Chapter 3, starts by finding all possible blocks, which are patterns without gaps, that are located in two sequences. A large number of possible alignments between two sequences are generated by aligning combinations of blocks. Gaps will be present between the blocks, as illustrated in Figure 7.9. The sequence alignments are

```
Sequence 1 xxxxxx---o--xxxxxxxxxxxooo-oooxxxxxx
Sequence 2 xxxxxxooooooxxxxxxxxxxxo-ooo-oxxxxxx
           block1      block2            block3
```

Figure 7.9. Alignment found by the Bayes block aligner. The alignment between two sequences includes ungapped blocks (marked by x where aligned x's may be identical or substitutions; there will be at east one identity in each block used to identify the block) and intervening unaligned regions with gaps (marked by o for unaligned residue and − for a gap). These two regions are designed to represent conserved structural alignments in the protein core and variable surface loops, respectively. A large number of alignments of this type involving different combinations of blocks are found. These alignments are then evaluated by a set of scoring matrices. The best alignment is then derived by a Bayesian statistical analysis, described in Chapter 3.

scored only where the sequences are aligned in the blocks: There is no gap penalty as in the dynamic programming method of alignment.

Alignments are also scored differently by the Bayes block aligner than by the dynamic programming method. In the Bayes block aligner, a set of amino acid substitution matrices is used. Each scoring matrix models a different degree of substitution between the sequences, and the matrices that best represent this degree should give the highest alignment scores. When PAM-type matrices are used, the evolutionary distance between parts of sequences can be estimated knowing the best-scoring matrix. When the analysis has been completed, there are a large number of possibilities to sort out, including choices of block number, alignments, and scoring matrices.

By using a Bayesian statistical analysis of the results, it is possible to derive block alignments in which amino acids in each sequence are most often associated, regardless of the many possible choices. These alignments are represented as the posterior probability of aligning those amino acids given the initial preferences of block number and scoring matrices. It is these block alignments that are statistically the best representation of the alignment between two sequences. From the Bayesian analysis, the probability that the sequences are related may be calculated from the posterior probabilities of block number by examining the analysis for evidence that the block number is greater than zero. If this calculation, described in Chapter 3 (p. 130), yields a probability greater than 0.5, the Bayesian analysis supports a relationship between the sequences (Zhu et al. 1998)

DATABASE SEARCHES WITH A SCORING MATRIX OR PROFILE

The methods for database searching discussed so far in this chapter are based on using a single query sequence to search a sequence database. Another method of database searching is to use the variation found in a multiple sequence alignment of a set of related sequences to search for matching database sequences. This enhanced type of search will locate database sequences that match new combinations of sequence characters in the multiple sequence alignment. For example, if column 1 of a multiple sequence alignment includes the amino acids P and Q and column 2 the amino acids D and E, then database sequences that match all four combinations of these two amino acids can be found, whereas only the combinations found in the original sequences would be matched if single query sequences were to be used.

Multiple sequence alignments represent the occurrence of one or more patterns common to a set of sequences. These patterns may be relatively short or may include long conserved stretches of sequence. In Chapter 4, two methods for identifying a common set of patterns in sequences were described. The first extracts a set of patterns from a multiple sequence alignment, which can be produced by methods such as dynamic programming, genetic algorithms, or hidden Markov models. The second uses pattern finding and statistical methods, including expectation maximization and Gibbs sampling methods, to locate patterns in unaligned sequences. Hidden Markov models are also useful for representing a set of conserved patterns that includes gaps, in a protein family. The resulting PROFILE HMM (Durbin et al. 1998) may then be used to search a query sequence for matches to the set of patterns. Chapter 4 should be consulted for a discussion of these methods; the relevant programs and Web sites are described below.

To search a sequence database for matches to a set of patterns, the sequence information is stored as a matrix of 20 rows, one row for each amino acid, and *n* columns, one column for each column in the multiply aligned sequence patterns. In addition, there may be extra rows for ambiguous or unidentified symbols and, in the sequence profile matrix, there are rows for gap opening and extension penalties. Examples are shown in Figures 4.11 and 4.12 in Chapter 4.

The simplest scoring matrix, the position-specific scoring matrix (PSSM), represents an alignment of sequence patterns of the same length (no gaps). The production of a PSSM is also discussed in Chapter 4 (p. 192). To summarize, the sequence patterns are first aligned as a multiple sequence alignment so that corresponding residues are in the same column. Raw amino acid counts are first found by summing the numbers in each column of the alignment, and these numbers are placed in the corresponding columns of the scoring matrix, one for each amino acid in the designated row. These counts are then adjusted by a weighting method designed to prevent overrepresentation of the amino acids in the more closely related sequences. Otherwise, the matrix would be more tuned to those sequences than to the less-alike ones in the group. To these raw scores, additional counts are added based on previously observed general types of amino acid variations in alignments of related proteins. The idea behind this strategy is that the small number of sequences usually present in these alignments does not represent the full range of expected amino acid variations. Therefore, additional pseudocounts are added based on substitution patterns found in an amino acid substitution matrix or representative blocks in the BLOCKS database (Dirichlet mixtures). The statistical basis for adding counts is that including prior information in the form of pseudocounts should increase the sensitivity of the scoring matrix. The sum of the raw and additional counts in each column is then divided by the expected frequency of the amino acid from the data or from other sources. The resulting ratio represents the odds for finding a match of another related sequence to the column divided by the chance of a random match with an unrelated sequence. For ease in multiplying probabilities by adding their logarithms, each odds score is converted to a log odds score, usually to logarithms to the base 2. The log odds score for each column in the alignment is placed in the corresponding column of the matrix, and there is one row of scores for each amino acid that is the same width as the pattern window. The resulting PSSM is easy to align with a sequence, as discussed below.

SEARCHING SEQUENCE DATABASES WITH A POSITION-SPECIFIC SCORING MATRIX OR SEQUENCE PROFILE

Aligning a PSSM with a protein sequence is illustrated in Figure 7.10. Every possible sequence position is scored by sliding the matrix along the sequence one position at a time. The amino acid substitution scores in each column of the PSSM are used to evaluate each sequence position. Positions with the highest scores are the best matches of the corresponding set of sequence patterns with the sequence. In searches of a sequence database, those sequences with a region that is a close match to the pattern will produce the highest scores and may be readily identified.

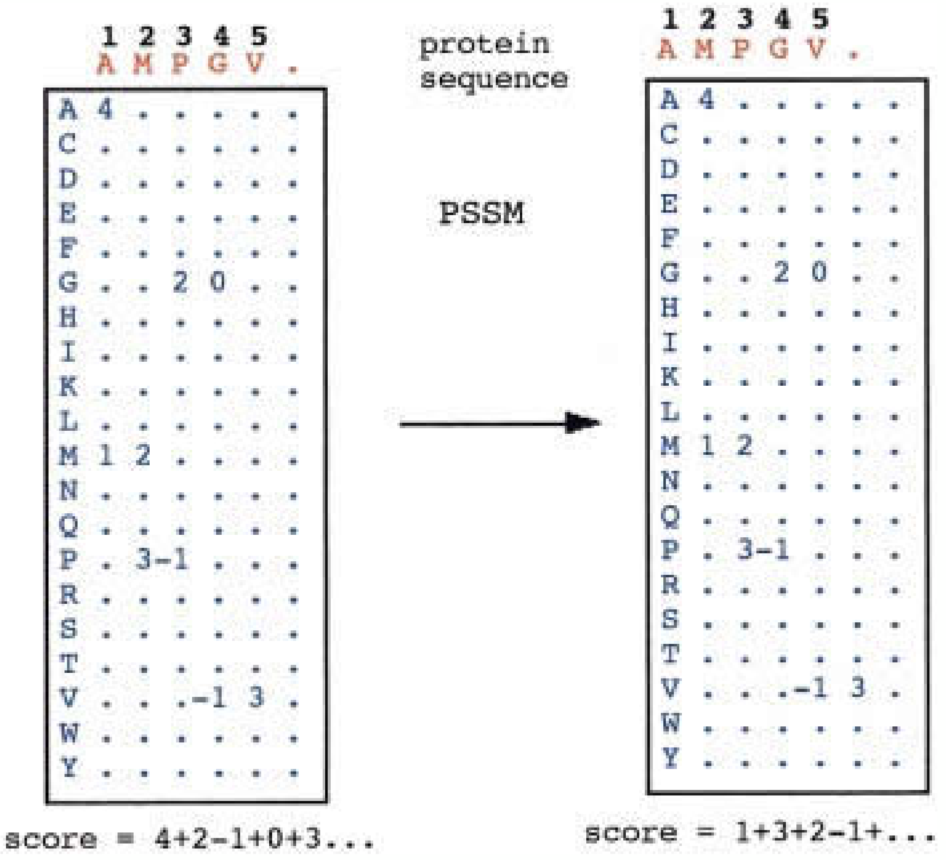


Figure 7.10. Scoring an alignment of a PSSM with a sequence. Only a few matrix values are shown for simplicity. The PSSM is first aligned with sequence position 1 so that columns 1–10 match positions 1–10 in the sequence. The column values that match each sequence position are then added to give a total log odds score for sequence position 1. The PSSM is then moved to sequence position 2 and a new score for matches with sequence positions 2–11 is calculated. This process if repeated until the last 10 sequence positions are matched. The highest scores represent the best matches of the motif represented by the PSSM to the sequence. This type of scoring system preferentially rewards matches to columns that have a conserved amino acid more than to matches with variable columns, and penalizes mismatches to columns that have a conserved amino acid more heavily than mismatches to variable columns.

Scoring matrices that correspond to a sequence profile also include two extra rows for gap penalty scores (sometimes these scores may be found in extra columns if the labeling of the rows and columns is reversed). When aligning this type of scoring matrix with a sequence, a similar procedure to the above is followed in that the score for matching the profile scoring matrix to each sequence character is calculated. In addition, a gap of any length may be inserted into the sequence or profile at that position, and the gap penalties are those given in the relevant column of the profile. The gap penalties are usually quite high with respect to the match scores, but are less when gaps were present in the original multiple sequence alignment. The problem of finding the best alignment between the profile and a given start position in the sequence is similar to the problem of aligning two sequences. As with alignment of sequences, the dynamic programming algorithm is used, except that the match scores and gap penalties are site-specific and are the values given in the profile columns.

Web sites and programs for finding common motifs and profiles in a set of related sequences or for searching a protein sequence database with these patterns are listed in Table 7.8. Also shown are sites that can be given an ambiguous pattern, called a regular expression, to use in a protein sequence database search. The first programs available for producing profiles and using these for sequence searches were Profilemake for making profiles from a multiple sequence alignment, Profilegap for aligning a profile with one or more sequences, and Profilesearch for searching a protein sequence database with a profile (Gribskov and Veretnik 1996). These programs are best known as components of the Genetics Computer Group suite of programs. Profiles produced by newer versions of these programs use evolutionary predictions of the amino acid changes in each column, which

Table 7.8. *Programs and Web sites for database similarity searches with a regular expression, motif, block, or profile*

Program	Database searched	Source or location of analysis
1. Regular expressions and motifs[a]		
EMOTIF Scan	SwissProt and Genpept	http://dna.stanford.edu/scan/
Prosite patterns	SwissProt and TrEMBL	http://www.expasy.ch/tools/scnpsit2.html
ISREC pattern-finding service	SwissProt and non-redundant EMBL database	http://www.isrec.isb-sib.ch/software/PATFND_form.html
fpat	PDB SwissProt Genpept	http://www.ibc.wustl.edu/fpat/
PHI-BLAST	BLAST databases	http://www.ncbi.nlm.nih.gov/
MOTIF	SwissProt, PDB, PIR, PRF, Genes	http://www.motif.genome.ad.jp/MOTIF2.html
2. Blocks		
BLOCKS[b]	most databases	http://www.blocks.fhcrc.org/blockmkr/make_blocks.html
MAST[c]	most databases	http://meme.sdsc.edu/meme/website/
BLIMPS[d]	locally available databases	anonymous FTP ncbi.nlm.nih.gov/repository/blocks/unix/blimps
Probe[e]	BLAST databases	anonymous FTP ncbi.nlm.nih.gov/pub/neuwald/probe1.0
Genefind[f]	PIR	http://pir.georgetown.edu/gfserver
3. Profiles		
Profilesearch[g]	locally available databases	anonymous FTP ftp.sdsc.edu/pub/sdsc/biology/profile_programs
Profile-SS[h]	most databases	http://www.psc.edu/general/software/packages/profiless/profiless.html

These resources search for similarity to a sequence pattern. Resources for producing patterns from aligned or unaligned sequences are described in Chapter 4. An individual sequence may also be searched for matches to a motif database, and this procedure is discussed in Chapter 9. Additional resources for database searching are listed in Bork and Gibson (1996).

A statistical estimate of finding the site by random chance in a sequence is sometimes but not always given. Reading how these estimates are derived by the individual programs is strongly recommended. The statistical theory for sequence alignments described in Chapter 3 can be used in these types of analyses (Bailey and Gribskov 1998) but may not always be implemented.

[a] The Scan Web page shows how to compile a regular expression. Mismatches within the expression are allowed. The Prosite form of a regular expression is at http://www.expasy.ch/tools/scnpsit3.html. PHI-BLAST is a BLAST derivative that searches a given sequence for a regular expression and then searches iteratively for other sequences matching the pattern found, at each iteration including the newly found sequences to expand the search.

[b] The BLOCKS server will send a new block analysis to the MAST server.

[c] MAST is the Motif Alignment and Search Tool (Bailey and Gribskov 1998). Available protein databases are similar to those on the BLAST server. It is also possible to search translated nucleotide sequence databases.

[d] BLIMPS will prepare a PSSM from a motif and perform a database search with the PSSM (see README file on FTP site).

[e] PROBE (Neuwald et al. 1997) is described in the text.

[f] The GENEFIND site has the program MOTIFIND for Motif Identification by Neural Design (Wu et al. 1996). This motif finder uses a neural network design to generate motifs and a search strategy for those motifs. The method performed favorably in sensitivity and selectivity with others such as Blimps and Profilesearch and is in addition very fast. Neural networks are described in Chapters 8 and 9.

[g] Profilesearch is one of a set of programs in the GCG suite (see text). It is important to review the parameters of the program which if used inappropriately can lead to incomplete or low-efficiency searches (Bork and Gibson 1996).

[h] A version of Profilesearch running at the University of Pittsburgh Supercomputing Center.

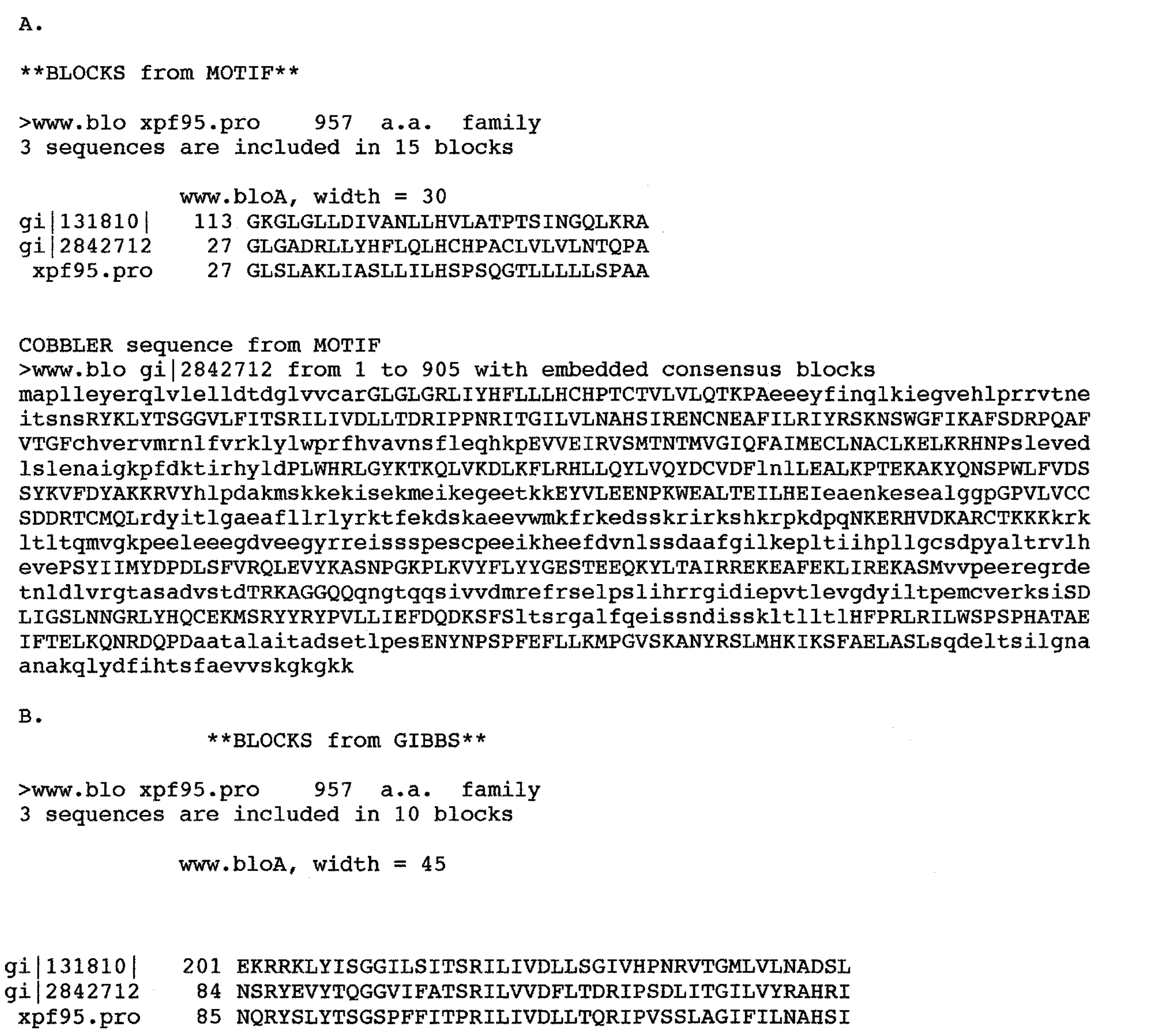
improves the ability of the profile to find related proteins in a database search. Methods for making evolutionary profiles and for using them are discussed in Chapter 4 (p. 163). Profile searches may be performed at two supercomputer centers (Table 7.8). The standard Genetics Computer Group multiple sequence alignment format, the MSF file (described in Chapter 2), is used as input to these programs. READSEQ and other sequence reformatting programs can be used to change the sequence format to the MSF format (see Chapter 2), which can then be used with Profilemake.

There is a difference in the way PSSM and the profile matrix are generated that should influence the results of a database search. The PSSM treats all amino acids as being equal, so that matching an Ala with an Ala is as significant as matching a Cys with a Cys. Scores for amino acid substitutions are based on the distribution of amino acids in each column of the alignment on which the PSSM is based. Profile scores are also based on the distribution of amino acids in each column of the alignment, but the matrix values are also derived from an amino acid substitution matrix, such as the Dayhoff PAM matrices. Hence, the PSSM and profile methods should give different results.

To illustrate these methods, the results of finding blocks by the BLOCKS server, and of using motifs found by the MEME server, for a search of the SwissProt database by MAST are given below. The terms “blocks” and “motifs” are used interchangeably by these sites in that both mean a reasonably long ungapped pattern in a family of protein sequences. Once matching sequences have been found, other searches can be performed with sets of blocks or motifs of shorter or longer length and of more occurrences. Use of the program MACAW, which runs on many computer platforms with a Windows interface, is also very useful for exploring motif size and number (see Chapter 4, p. 177). This program can find motifs either by an alignment method using an amino acid scoring matrix or by the statistical Gibbs sampling method. The relative positions of the found motifs are shown on a graphical representation of the sequences.

Example: BLOCKS Server

This Web server takes a set of unaligned sequences and finds blocks of sequence (matching ungapped patterns) that are present in the sequences. A request to the BLOCKS Web site to find blocks in three sequences similar to the human *XPF* DNA repair gene was made, and an example of the program output is shown in the following example (Fig. 7.11). The sequences were input into a Web form in FASTA format. The server finds blocks by two methods, a pattern-searching method called MOTIF and a statistical method called the Gibbs sampler. These two methods are described in detail in Chapter 4 (pp. 171 and 177). The blocks found by each method may not be the same because each method uses a different algorithm. Examples of a representative block found by MOTIF, www.bloA is shown in Figure 7.11A. Also shown is a portion of a Cobbler sequence of one of the input sequences, xpf95.pro, an *Arabidopsis* gene. In the Cobbler sequence, the sequence in xpf95.pro corresponding to each block location has been replaced by a consensus sequence derived from bloA. These replaced regions are capitalized. In Figure 7.11B, an example of a Gibbs sampler block also called www.bloA is shown. A Cobbler sequence was also produced from these blocks (not shown). There are two options given after each list of blocks: (1) the Cobbler sequence may be used in a BLAST search and (2) the blocks may be sent to the MAST server to search a protein sequence database.



```
A.

**BLOCKS from MOTIF**

>www.blo xpf95.pro    957  a.a.  family
3 sequences are included in 15 blocks

            www.bloA, width = 30
gi|131810|   113 GKGLGLLDIVANLLHVLATPTSINGQLKRA
gi|2842712    27 GLGADRLLYHFLQLHCHPACLVLVLNTQPA
 xpf95.pro    27 GLSLAKLIASLLILHSPSQGTLLLLLSPAA


COBBLER sequence from MOTIF
>www.blo gi|2842712 from 1 to 905 with embedded consensus blocks
maplleyerqlvlelldtdglvvcarGLGLGRLIYHFLLLHCHPTCTVLVLQTKPAeeeyfinqlkiegvehlprrvtne
itsnsRYKLYTSGGVLFITSRILIVDLLTDRIPPNRITGILVLNAHSIRENCNEAFILRIYRSKNSWGFIKAFSDRPQAF
VTGFchvervmrnlfvrklylwprfhvavnsfleqhkpEVVEIRVSMTNTMVGIQFAIMECLNACLKELKRHNPsleved
lslenaigkpfdktirhyldPLWHRLGYKTKQLVKDLKFLRHLLQYLVQYDCVDFlnlLEALKPTEKAKYQNSPWLFVDS
SYKVFDYAKKRVYhlpdakmskkekisekmeikegeetkkEYVLEENPKWEALTEILHEIeaenkesealggpGPVLVCC
SDDRTCMQLrdyitlgaeafllrlyrktfekdskaeevwmkfrkedsskrirkshkrpkdpqNKERHVDKARCTKKKkrk
ltltqmvgkpeeleeegdveegyrreissspescpeeikheefdvnlssdaafgilkepltiihpllgcsdpyaltrvlh
evePSYIIMYDPDLSFVRQLEVYKASNPGKPLKVYFLYYGESTEEQKYLTAIRREKEAFEKLIREKASMvvpeeregrde
tnldlvrgtasadvstdTRKAGGQQqngtqqsivvdmrefrselpslihrrgidiepvtlevgdyiltpemcverksiSD
LIGSLNNGRLYHQCEKMSRYYRYPVLLIEFDQDKSFSltsrgalfqeissndisskltlltlHFPRLRILWSPSPHATAE
IFTELKQNRDQPDaatalaitadsetlpesENYNPSPFEFLLKMPGVSKANYRSLMHKIKSFAELASLsqdeltsilgna
anakqlydfihtsfaevvskgkgkk

B.
              **BLOCKS from GIBBS**

>www.blo xpf95.pro    957  a.a.  family
3 sequences are included in 10 blocks

            www.bloA, width = 45



gi|131810|    201 EKRRKLYISGGILSITSRILIVDLLSGIVHPNRVTGMLVLNADSL
gi|2842712     84 NSRYEVYTQGGVIFATSRILVVDFLTDRIPSDLITGILVYRAHRI
 xpf95.pro     85 NQRYSLYTSGSPFFITPRILIVDLLTQRIPVSSLAGIFILNAHSI
```

Figure 7.11. Example of BLOCK output.

Example: Motif Alignment and Search Tool (MAST) Server

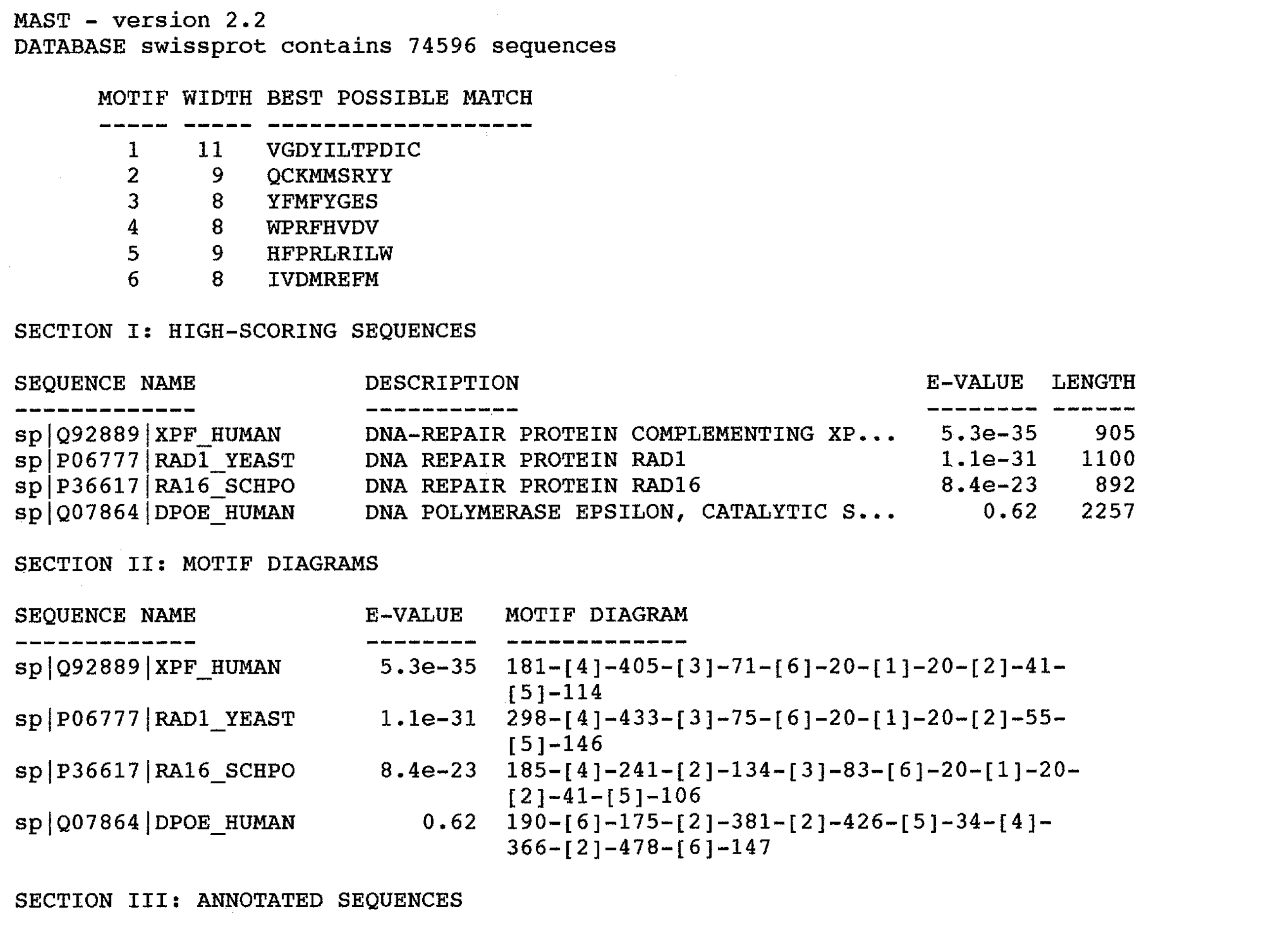
The MAST server searches a protein database for best matches to a set of ungapped motifs or blocks (Bailey et al. 1997). The motifs may also be found by MEME by submitting unaligned sequences to the MEME server for analysis by a statistical method, the expectation maximization method, described in Chapter 4. A MEME output example is shown as Figure 4.15. The same three DNA repair sequences as used above were input into a Web form in FASTA format. To simplify output of many possible choices, MEME was asked for one motif per sequence and for up to six different motifs of short length. Once received by E-mail, the motif messages were saved to a local file, and then this file was submitted to the MAST server (http://www.sdsc.edu/MEME) using the Browse option of the Web page to read the newly made local file of the MEME output. Using a short file name with a short subdirectory listing was necessary because there is not much space on the Web form. It is also possible to submit an already-found motif in GCG, MEME, or PSSM format (see http://www.sdsc.edu/MEME/meme/website/motif-format.html). Another method for readily accessing MAST is through the BLOCKS server. As shown in Figure 7.11, unaligned sequences are searched for blocks by two methods, and from the BLOCKS Web page, the results may be immediately submitted to the MAST server. MAST uses the method shown in Figure 7.10 to align the blocks with each database sequence. If not specified otherwise, the output files are sent by E-mail as HTML files suitable for viewing by a Web browser. These files have some nice graphical features. These files are first saved to a file and then opened with a Web browser.

Alternatively, the files may be requested in text format, as was done below (Fig. 7.12). The initial list in the MAST output is of the motifs found by MEME. Note that motifs are given an ID number (1–6) that is used later in the MAST report. Section I then lists the scoring matches found in the SwissProt sequence database. The expect value is the number of unrelated sequences in a database of the size of SwissProt that would achieve a score as high as the one shown with the motifs used in the search and is based on the scores of individual motifs with the sequence using the extreme value distribution (Bailey and Gribskov 1998). The highest-scoring matches are with the two input DNA repair proteins but then there are several lower-scoring matches with other proteins that interact with DNA, suggesting a common structural motif; however, caution is necessary in interpreting these kinds of matches (Bork and Gibson 1996). One of the input sequences was that of an *Arabidopsis* DNA repair protein that is not reported because it is not in the database yet. Section II shows the locations of the motifs in each sequence. The motifs are shown in brackets and numbered as at the top of the file. Note that the order in the first three sequences is approximately the same, but that there are more and more variations going down the list, reflecting more divergence. Finally, in section III, the matched motifs are aligned with the matched sequence. At each aligned position, the motif number, the *P* value of each match, the motif sequence giving the best match between sequence and motif, and a plus sign to indicate sequence letters corresponding to a positive match score in the motif column are given. A diagram shows the order of the motifs found and a combined *P* (combined probability for matching all matrices to an unrelated sequence) and *E* score (expectation of finding these matches with an unrelated sequence in a database of the size searched). The combined probabilities are calculated using the extreme value distribution as used for determining the significance of FASTA and BLAST scores (Bailey and Gribskov 1998).

OTHER METHODS FOR COMPARING DATABASES OF SEQUENCES AND PATTERNS

One variation of the method for comparing sequences and patterns is to search a query sequence with a database of patterns (search type E, Table 7.1). If the sequence contains patterns representative of a protein family, the sequence is a candidate for membership in that same family. A large number of protein pattern databases are available (Table 9.5), most of them offering this type of search.

FASTA-pat and FASTA-swap are versions of FASTA that may be used for comparing a query sequence to a database of patterns characteristic of protein families. They are designed to search for remotely related protein sequences by a finely tuned system of



```
MAST - version 2.2
DATABASE swissprot contains 74596 sequences

      MOTIF WIDTH BEST POSSIBLE MATCH
      ----- ----- -------------------
        1    11   VGDYILTPDIC
        2     9   QCKMMSRYY
        3     8   YFMFYGES
        4     8   WPRFHVDV
        5     9   HFPRLRILW
        6     8   IVDMREFM

SECTION I: HIGH-SCORING SEQUENCES

SEQUENCE NAME            DESCRIPTION                            E-VALUE  LENGTH
-------------            -----------                            -------- ------
sp|Q92889|XPF_HUMAN      DNA-REPAIR PROTEIN COMPLEMENTING XP...  5.3e-35    905
sp|P06777|RAD1_YEAST     DNA REPAIR PROTEIN RAD1                 1.1e-31   1100
sp|P36617|RA16_SCHPO     DNA REPAIR PROTEIN RAD16                8.4e-23    892
sp|Q07864|DPOE_HUMAN     DNA POLYMERASE EPSILON, CATALYTIC S...     0.62   2257

SECTION II: MOTIF DIAGRAMS

SEQUENCE NAME            E-VALUE   MOTIF DIAGRAM
-------------            --------  -------------
sp|Q92889|XPF_HUMAN       5.3e-35  181-[4]-405-[3]-71-[6]-20-[1]-20-[2]-41-
                                   [5]-114
sp|P06777|RAD1_YEAST      1.1e-31  298-[4]-433-[3]-75-[6]-20-[1]-20-[2]-55-
                                   [5]-146
sp|P36617|RA16_SCHPO      8.4e-23  185-[4]-241-[2]-134-[3]-83-[6]-20-[1]-20-
                                   [2]-41-[5]-106
sp|Q07864|DPOE_HUMAN         0.62  190-[6]-175-[2]-381-[2]-426-[5]-34-[4]-
                                   366-[2]-478-[6]-147

SECTION III: ANNOTATED SEQUENCES
```

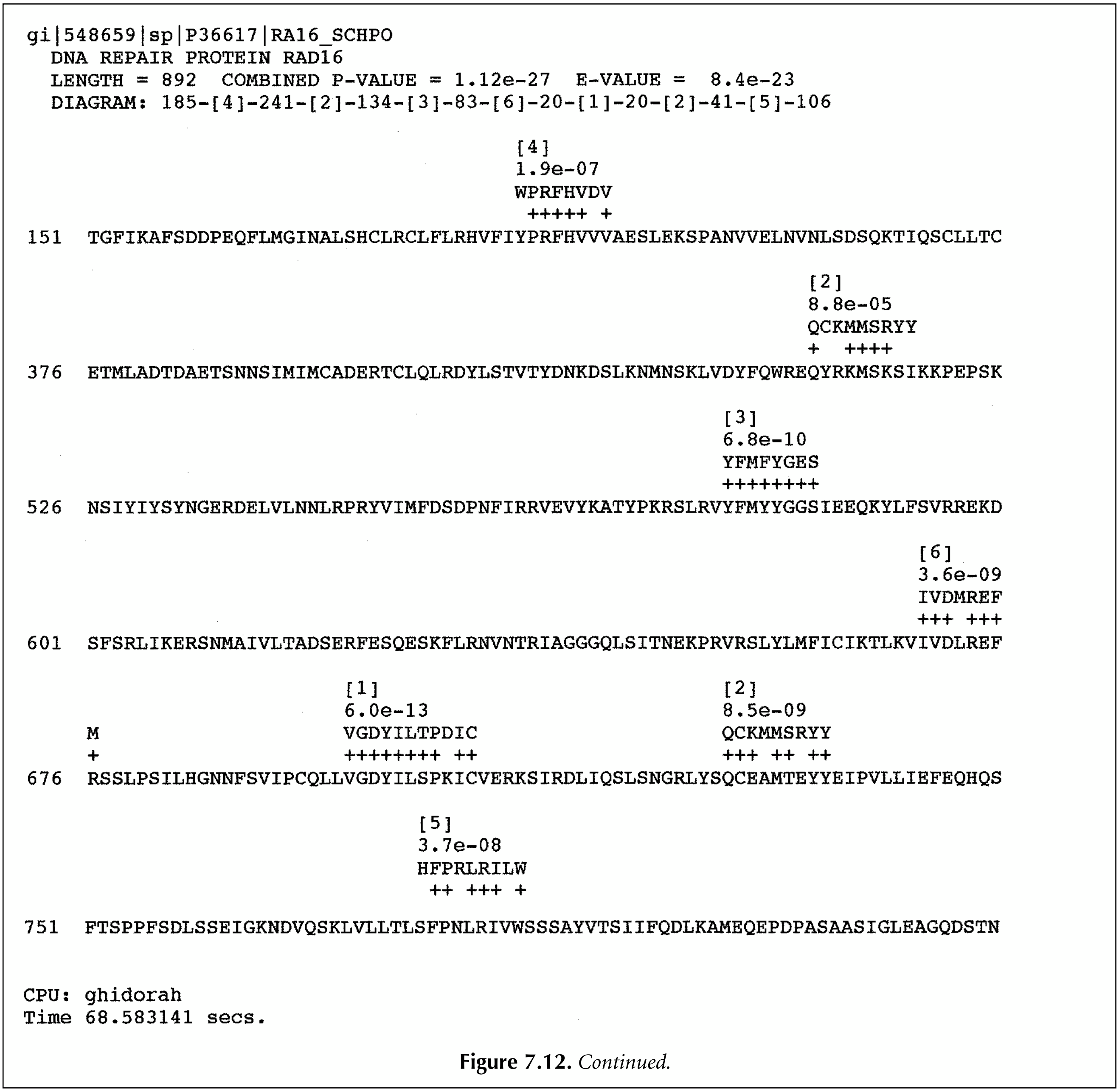
Figure 7.12. Example of MAST output. *Figure continues on next page.*

amino acid matches. The FASTA algorithm normally identifies sequence similarity very rapidly by a method for finding common patterns, or *k*-tuples, in the same order in two sequences. In FASTA-pat and FASTA-swap, the same rapid method is used to find common patterns. FASTA-pat performs a faster method of comparing sequences to patterns by means of a lookup table, as described above (Table 7.3). FASTA-swap performs a more rigorous search for the most significant matches of sequence to patterns.

These programs use databases of patterns found in columns of multiple sequence alignments of related protein sequences. Multiple sequence alignments of a large number of protein families were prepared using the PIMA program (see Chapter 4, p. 160). From these alignments, a large number of conserved patterns were identified, and the pattern was placed in a new type of scoring matrices. Unlike PSSMs, the columns in these matrices only indicate whether or not a given amino acid is present; there is no score indicating frequency.

In addition to these pattern matrices, two log odds scoring matrices, weighted-match minimum average matrix (WMM) and empirical matrix (EMMA), were prepared from the scoring matrices. These scoring matrices are used by FASTA-pat and FASTA-swat for comparing a query sequence with a database of pattern matrices.

The scoring system takes into account the possibility that the substitution of amino acid *a* for amino acid *b* may not be as likely as the substitution of *b* for *a*. An example from



```
gi|548659|sp|P36617|RA16_SCHPO
  DNA REPAIR PROTEIN RAD16
  LENGTH = 892  COMBINED P-VALUE = 1.12e-27  E-VALUE =  8.4e-23
  DIAGRAM: 185-[4]-241-[2]-134-[3]-83-[6]-20-[1]-20-[2]-41-[5]-106

                                         [4]
                                         1.9e-07
                                         WPRFHVDV
                                          +++++ +
151   TGFIKAFSDDPEQFLMGINALSHCLRCLFLRHVFIYPRFHVVVAESLEKSPANVVELNVNLSDSQKTIQSCLLTC

                                                                 [2]
                                                                 8.8e-05
                                                                 QCKMMSRYY
                                                                 +  ++++
376   ETMLADTDAETSNNSIMIMCADERTCLQLRDYLSTVTYDNKDSLKNMNSKLVDYFQWREQYRKMSKSIKKPEPSK

                                                          [3]
                                                          6.8e-10
                                                          YFMFYGES
                                                          ++++++++
526   NSIYIYSYNGERDELVLNNLRPRYVIMFDSDPNFIRRVEVYKATYPKRSLRVYFMYYGGSIEEQKYLFSVRREKD

                                                                          [6]
                                                                          3.6e-09
                                                                          IVDMREF
                                                                          +++ +++
601   SFSRLIKERSNMAIVLTADSERFESQESKFLRNVNTRIAGGGQLSITNEKPRVRSLYLMFICIKTLKVIVDLREF

                           [1]                            [2]
                           6.0e-13                        8.5e-09
      M                    VGDYILTPDIC                    QCKMMSRYY
      +                    ++++++++ ++                    +++ ++ ++
676   RSSLPSILHGNNFSVIPCQLLVGDYILSPKICVERKSIRDLIQSLSNGRLYSQCEAMTEYYEIPVLLIEFEQHQS

                                 [5]
                                 3.7e-08
                                 HFPRLRILW
                                  ++ +++ +

751   FTSPPFSDLSSEIGKNDVQSKLVLLTLSFPNLRIVWSSSAYVTSIIFQDLKAMEQEPDPASAASIGLEAGQDSTN


CPU: ghidorah
Time 68.583141 secs.
```

Figure 7.12. *Continued.*

Ladunga et al. (1996) is informative. On the one hand, if an alignment column has 9 Cys and 1 Ala, the substitution of Ala for Cys in this column would be given a low substitution score because Cys is involved in disulfide bonds and this function cannot be replaced by Ala. Cys-to-Cys substitutions receive a high score for the same reason. On the other hand, if a column has 1 Cys and 9 Ala, then the Cys might readily substitute for the Ala, which has no comparable specific function. The substitution of Cys for Ala is considered to be a random insertion of no particular significance and is therefore given a corresponding likelihood score of zero. When aligning a query sequence to a pattern, a single amino acid in the sequence is matched to a series of possible substitutions in the pattern. WMM uses the minimum of the scores for aligning the amino acid in the query sequence with each of the amino acids in the pattern. WMM gives significantly better results than EMMA, probably because it is more finely tuned for detecting the types of variations in related sequences.

Program outputs of FASTA-pat and FASTA-swap are very similar to those of FASTA described above.

Another type of pattern database searching is to use a pattern query to search a database of patterns. The LAMA (Local Alignment of Multiple Alignments) server at the BLOCKS Web site, described below, performs such a search. A final variation is to use a query sequence, called a Cobbler sequence (see Fig. 7.11), modified by substituting a consensus sequence for the corresponding part of the sequence. The BLOCKS server automatically produces such sequences when generating new blocks from sequences, and they may be used for sequence database searches. Embedding consensus residues has been demonstrated to improve database searches by a query sequence (Henikoff et al. 1995).

LAMA is a type of analysis provided on the BLOCKS server (http://www.blocks.fhcrc.org/blockshelp/LAMA_help.html#LAMA) that compares a query PSSM representing a particular set of proteins with a database of such matrices to find related sets of proteins (Pietrokovski 1996). In this manner, new and larger related sets of proteins not identified previously might be discovered. Because the search is for matching sequence patterns instead of entire sequence alignments, there is an opportunity to analyze the evolution of function in different parts of a protein molecule (Henikoff et al. 1997). For example, a given group of proteins may be found to have two regions, one related to one particular group of proteins and a second related to another group. The LAMA program compares the scores found in each column of one PSSM to those in a second to discover whether there is any correlation. Examples of the procedure are given at http://www.blocks.fhcrc.org/blockshelp/LAMA_help.html#EXAMPLES.

PSI-BLAST, A Version of BLAST for Finding Protein Families

As described above, there are advantages to using a scoring matrix that represents conserved sequence patterns in a protein family instead of a single query sequence to search a sequence database. The search of sequence databases will thereby be expanded to identify additional related sequences that might otherwise be missed. The major difficulty with such an expanded search is that an alignment of related sequences must already be available to know the variations at each position in the query sequence. A new version of BLAST called position-specific-iterated BLAST, or PSI-BLAST, has been designed to provide information on this variation starting with a BLAST search by a single query sequence. A similar program, PHI-BLAST, performs a similar type of search starting with a specified pattern in a query sequence (see below).

The method used by PSI-BLAST involves a series of repeated steps or iterations. First, a database search of a protein sequence database is performed using a query sequence. Second, the results of the search are presented and can be assessed visually to see whether any database sequences that are significantly related to the query sequence are present. Third, if such is the case, the mouse is clicked on a decision box to go through another iteration of the search. The high-scoring sequence matches found in the first step are aligned, and, from the alignment, a type of scoring matrix that indicates the variations at each aligned position is produced. The database is then again searched with this scoring matrix. Thus, the search has been expanded to include sequences that match the variations found in the multiple sequence alignment at each sequence position. The results are again displayed, indicating any newly discovered sequences that are significantly related to the aligned sequences in addition to those found in the previous iteration. Again, an opportunity is given to go through another iteration of the program, but this time including any newly recruited sequences to refine the alignment. In this fashion, a new family of sequences that are significantly similar to the original query sequence can be found.

This new method was made possible by the development of the gapped BLAST program, which increased the speed of the BLAST algorithm by over one-half so that more sophisticated search routines of PSI-BLAST could be added without an overall loss of speed. PSI-BLAST may not be as sensitive as other pattern-generating and searching programs described in Chapter 4 and above, but the simplicity and ease of use of this program are very attractive features for exploring protein family relationships. In a comparison of the ability of PSI-BLAST with the Smith-Waterman dynamic programming program SSEARCH to identify members of 11 protein families defined by sequence similarity, PSI-BLAST found more sequences and, in some cases, many-fold more sequences, than SSEARCH and at a 40-fold greater speed.

A similar program, MAXHOM, has been described previously (Sander and Schneider 1991). The sequence alignment is built up in two steps. Matching sequences found in a database search are aligned by dynamic programming with a query sequence, and a profile is made from the alignment. A new round of sequences that match the updated profile are then picked from the SwissProt database (visit http://www.embl-heidelberg.de/predictprotein/predictprotein.html).

The main difficulty with searching for subtle sequence relationships based on similarity is determining the significance of the alignments that are found. Such similarities may be evidence of structural or evolutionary relationships, but they could also be due to matching of random variations that have no common origin or function (Bork and Gibson 1996). Protein structures are in general composed of a tightly packed core and outside loops. Amino acid substitutions within the core are common, but only certain substitutions will work at a given amino acid position in a given structure. Thus, sequence similarity is not usually a good indicator of structural similarity (see Chapter 9), and the alignments found need to be carefully evaluated before any firm conclusions can be drawn. Another difficulty with the PSI-BLAST approach is that the procedure follows a type of algorithm called a greedy algorithm. Put simply, once additional sequences that match the query are found, these newly found sequences influence the finding of more sequences like themselves, and so on. If a different but also related query sequence was used initially, a different group with possible overlaps with the first may be found. Thus, there is no guarantee that the alignments finally discovered represent the same set of related sequences. Nevertheless, PSI-BLAST potentially offers exciting opportunities to the curious but careful investigator. New types of relationships in the protein databases may be readily discovered and used to infer evolutionary origins of proteins (Tatusov et al. 1997).

The later steps of a PSI-BLAST search use a scoring matrix that represents the alignments found. PSI-BLAST has been engineered to find database matches to this matrix almost as rapidly as BLASTP finds matches to a query sequence. However, there are some differences between the matrix produced by PSI-BLAST and those produced by other matrix programs: (1) The matrix covers the entire length of the aligned sequences whereas other matrices cover only a short stretch of the alignment; (2) the same gap penalties are used throughout the procedure and there is no position-specific penalty as in other programs; (3) each subsequent alignment is based on using the query sequence as a master template for producing a multiple sequence alignment of the same length as the query sequence. Columns in the alignment involve varying numbers of sequences depending on the extent of the local alignment of each sequence with the query, and columns with gaps in the query sequence are ignored. Sequences >98% similar to the query are not included to avoid biasing the matrix. Thus, the multiple sequence alignment is a compilation of the pairwise alignments of each matching database sequence with the query sequence and is not a true multiple sequence alignment, as illustrated below. The resulting alignment provides the columns for the scoring matrix

```
xxxxxxxxxxxxxxxx      query sequence with no gaps
 xx-xxxx              alignment of sequence 1
          xxx-x            alignment of sequence 2
     xxxx-xx          alignment of sequence 3
----------------      columns of the PSI-BLAST
                      alignment
```

Once the alignment has been found, the frequencies of amino acids in each column are adjusted by weighting the sequences to reduce the influence of the more-alike sequences, and by adding more counts (pseudocounts) representing other amino acid substitutions found among the observed types in order to increase the statistical power of the matrix. These procedures are discussed in Chapter 4 (p. 192). The resulting scores in each column of the scoring matrix are scaled using the same scaling factor λ as the BLOSUM62 scoring matrix so that a threshhold value *T* for HSPs and other statistical parameters used by BLASTP may also be used by PSI-BLAST. At each iteration, previously matched sequences with an *E* value less than 0.001 are used to produce the next alignment, but this value may also be changed. PSI-BLAST is in a state of evolution, and the Web page should be consulted for recent improvements. An example of a PSI-BLAST result is shown in Figure 7.13.

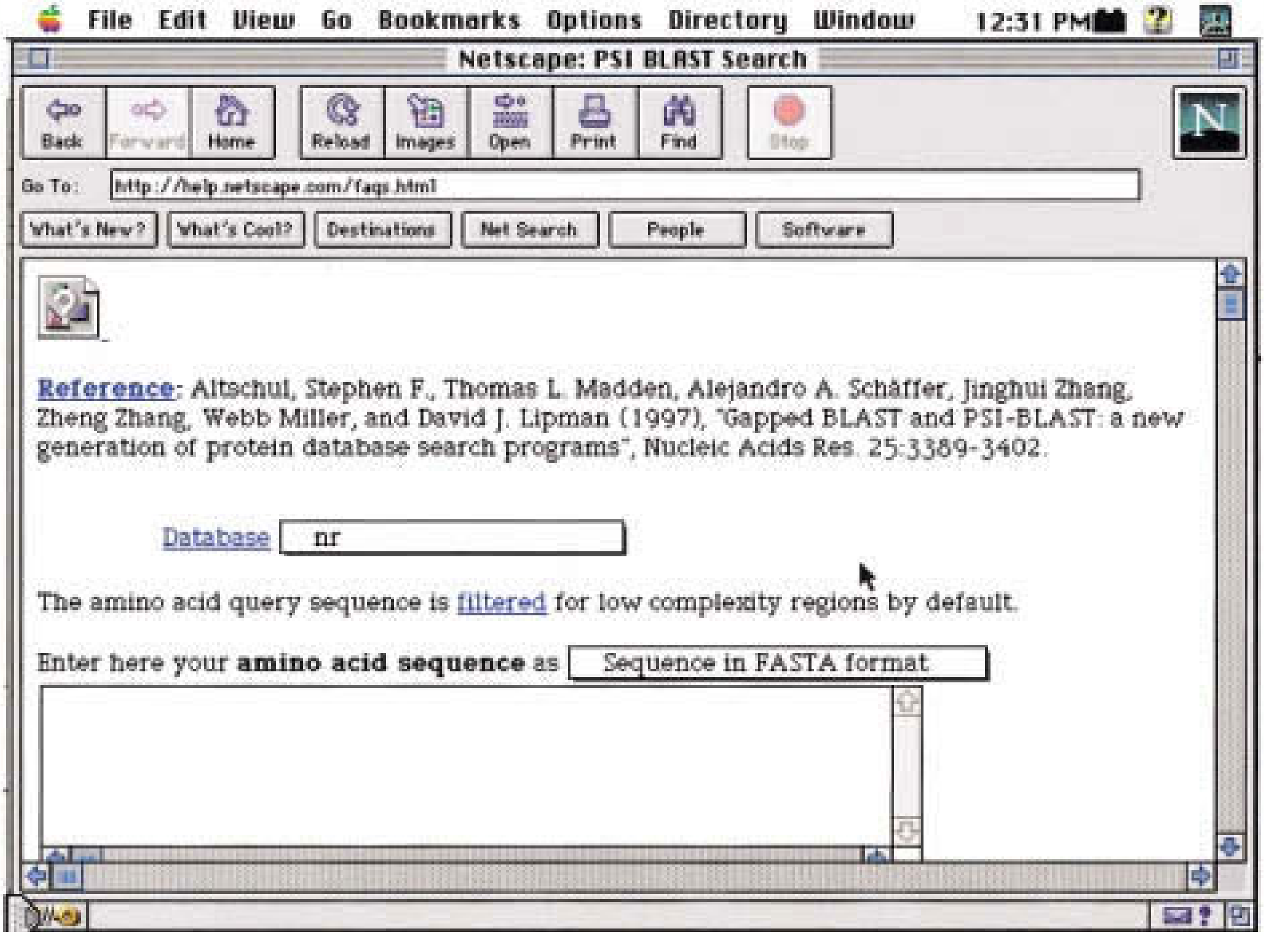
Pattern-Hit Initiated BLAST (PHI-BLAST)

This program functions much like PSI-BLAST except that the query sequence is first searched for a complex pattern provided by the investigator (Zhang et al. 1998). The subsequent search for similarity in the protein sequence database is then focused on regions containing the pattern. Thus, the method provides an opportunity to explore variations of a known pattern in the sequence database. This program is accessible from the BLAST server at http://www.ncbi.nlm.nih.gov/.

The chosen query sequence is first searched for a particular pattern or class of patterns called a regular expression, which allows for a wide range of pattern-matching options. The Prosite catalog also uses regular expressions to describe variability in the amino acid patterns for the active sites of proteins. For example, the expression [LIVMF]-G-E-x-[GAS]-[LIVM]-x(5,11)-R-[STAQ]-A-x-[LIVMA]-x-[STACV] means: one of LIVMF in the first position, followed by G and then E, followed by any single character (indicated by x), followed by one of GAS and then by one of LIVM, followed by any 5–11 characters indicated by x(5,11), then by R, one of STAQ, then A, then any single character, then one of LIVMA, then any single character, and finally by one of STACV. More information about these patterns may be provided by the investigator in a standard file, as described on the Prosite Web site (http://www.expasy.ch/prosite/.

PROBE

PROBE is a database search tool that is similar to PSI-BLAST but performs a more complex and rigorous type of data analysis (Neuwald et al. 1997). Like PSI-BLAST, the program PROBE starts with a single query sequence and searches for family members by a BLASTP search. After removing the most-alike sequences, PROBE constructs an alignment model by means of a Bayesian statistical approach that uses both a Gibbs sampling procedure and the genetic algorithm (both methods are described in Chapter 4) to sort the pat-



http://help.netscape.com/faqs.html

Reference: Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", Nucleic Acids Res. 25:3389-3402.

Database nr

The amino acid query sequence is filtered for low complexity regions by default.

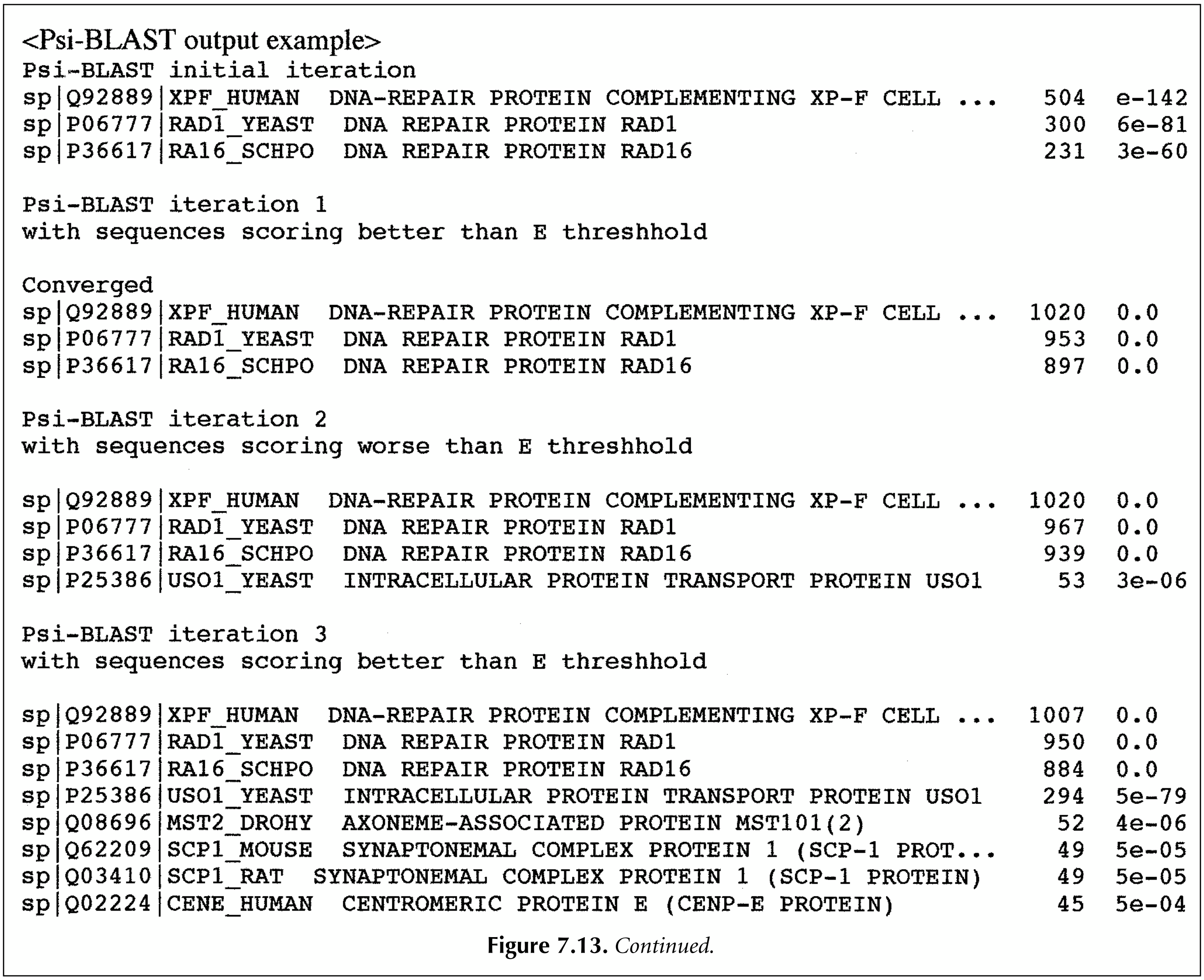
Enter here your **amino acid sequence** as Sequence in FASTA format

Figure 7.13. Example of PSI-BLAST search. The sequence of the *Arabidopsis XPF* DNA repair gene was used to query the SwissProt database, with an *E* setting of 0.01, requesting 10 descriptions and alignments with otherwise the recommended default program settings. The initial iteration found three matching sequences, and these were used to enter iteration 1. Iteration 1 did not produce any additional matches at the chosen level of significance, and the program indicated that the search had converged with no more sequences at the chosen level of significance. Therefore, for iteration 2 the sequences scoring worse than the threshhold were used. Since only those lower-scoring sequences that have an alignment with the query could influence the result, this option could potentially find additional sequences. A yeast transport protein was then reported. With another iteration using the four sequences above threshhold, another set of sequences was now pulled into the high-scoring group. This search therefore revealed that the SwissProt database has three other sequences strongly related to the query sequence but that other sequences of less-significant similarity were also present. *Figure continues on next page.*

terns in all possible combinations in order to find the most significant set. As in PSI-BLAST, the alignment model is then used as a query for additional database sequences. PROBE provides a new and powerful approach toward finding a sequence family and is available by anonymous FTP from ncbi.nlm.nih.gov/pub/neuwald/.

SUMMARY

As the sequence databases continue to increase in size, for the most part with genomic DNA sequences of unknown function, it is important to have a set of computational tools for predicting the functions of these sequences. The first choice is usually to go to the



<Psi-BLAST output example>

```
Psi-BLAST initial iteration
sp|Q92889|XPF_HUMAN  DNA-REPAIR PROTEIN COMPLEMENTING XP-F CELL ...   504  e-142
sp|P06777|RAD1_YEAST  DNA REPAIR PROTEIN RAD1                         300  6e-81
sp|P36617|RA16_SCHPO  DNA REPAIR PROTEIN RAD16                        231  3e-60

Psi-BLAST iteration 1
with sequences scoring better than E threshhold

Converged
sp|Q92889|XPF_HUMAN  DNA-REPAIR PROTEIN COMPLEMENTING XP-F CELL ...  1020  0.0
sp|P06777|RAD1_YEAST  DNA REPAIR PROTEIN RAD1                         953  0.0
sp|P36617|RA16_SCHPO  DNA REPAIR PROTEIN RAD16                        897  0.0

Psi-BLAST iteration 2
with sequences scoring worse than E threshhold

sp|Q92889|XPF_HUMAN  DNA-REPAIR PROTEIN COMPLEMENTING XP-F CELL ...  1020  0.0
sp|P06777|RAD1_YEAST  DNA REPAIR PROTEIN RAD1                         967  0.0
sp|P36617|RA16_SCHPO  DNA REPAIR PROTEIN RAD16                        939  0.0
sp|P25386|USO1_YEAST  INTRACELLULAR PROTEIN TRANSPORT PROTEIN USO1     53  3e-06

Psi-BLAST iteration 3
with sequences scoring better than E threshhold

sp|Q92889|XPF_HUMAN  DNA-REPAIR PROTEIN COMPLEMENTING XP-F CELL ...  1007  0.0
sp|P06777|RAD1_YEAST  DNA REPAIR PROTEIN RAD1                         950  0.0
sp|P36617|RA16_SCHPO  DNA REPAIR PROTEIN RAD16                        884  0.0
sp|P25386|USO1_YEAST  INTRACELLULAR PROTEIN TRANSPORT PROTEIN USO1    294  5e-79
sp|Q08696|MST2_DROHY  AXONEME-ASSOCIATED PROTEIN MST101(2)             52  4e-06
sp|Q62209|SCP1_MOUSE  SYNAPTONEMAL COMPLEX PROTEIN 1 (SCP-1 PROT...    49  5e-05
sp|Q03410|SCP1_RAT  SYNAPTONEMAL COMPLEX PROTEIN 1 (SCP-1 PROTEIN)     49  5e-05
sp|Q02224|CENE_HUMAN  CENTROMERIC PROTEIN E (CENP-E PROTEIN)           45  5e-04
```

Figure 7.13. *Continued.*

BLAST Web site because a variety of database searches are possible against regularly updated databases and can be performed with rapid turnaround time. This chapter has discussed a variety of additional resources for such searches, most available on Web sites or available for setup on a local computer system. For extensive searching, establishment of the databases and programs on a local system is a reasonable and achievable option. It is then possible to set up batch files or scripts that automate the searches. These searches generate large amounts of information that needs to be organized into a database.

Some of the most interesting matches are those to more distantly related sequences. A short alignment region between a query and a database sequence is usually not biologically significant, even though there may be a number of identities in the alignment. If additional sequences can be found that share the same alignment, however, it is possible that the pattern represents a common structure in a family of related proteins. There are, in addition, databases of conserved patterns in protein families, and it has been estimated that about one-half of these patterns can be linked to a protein structural fold. Thus, it is very worthwhile to follow the distant relationships further with the eventual goal of trying to discover a relationship to a protein of known structure. There are some excellent computer tools available to the molecular biologist for finding conserved patterns in protein families and for searching new sequences with these patterns, and it can be anticipated that the number will continue to grow. There are a large number of Web servers for this purpose, and these are described in Chapter 9.

As methods are used to search for related sequences, it is important to keep an eye on the statistical significance of the matches and the plausibility of the observed amino acid substitutions from a structural perspective. It is quite easy to end up with a group of sequences that are related to each other but not to the query sequence. There are presently no guides as to which of the above methods is most likely to work. The best advice is to go further than the basic methods and Web sites by becoming familiar with the range of available methods.

REFERENCES

Agarwal P. and States D.J. 1998. Comparative accuracy of methods for protein sequence similarity search. *Bioinformatics* **14:** 40–47.

Altschul S.F. and Gish W. 1996. Local alignment statistics. *Methods Enzymol.* **266:** 460–480.

Altschul S.F., Boguski M.S., Gish W., and Wootton J.C. 1994. Issues in searching molecular sequence databases. *Nat. Genet.* **6:** 119–129.

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

Altschul S.F., Madden T.L., Schaffer A.A., Zhang J., Zhang Z., Miller W., and Lipman D.J. 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* **25:** 3389–3402.

Bailey T.L. and Gribskov M. 1998. Combining evidence using p-values: Application to sequence homology searches. *Bioinformatics* **14:** 48–54.

Bailey T.L., Baker M.E., and Elkan C.P. 1997. An artificial intelligence approach to motif discovery in protein sequences: Application to steriod dehydrogenases. *J. Steroid Biochem. Mol. Biol.* **62:** 29–44.

Bork P. and Gibson T. 1996. Applying motif and profile searches. *Methods Enzymol.* **266:** 162–184.

Brenner S.E., Chothia C., and Hubbard T.J. 1998. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl. Acad. Sci.* **95:** 6073–6078.

Claverie J.-M. 1996. Effective large-scale sequence similarity searches. *Methods Enzymol.* **266:** 212–227.

Claverie J.-M. and States D.J. 1993. Information enhancement methods for large scale sequence analysis. *Comput. Chem.* **17:** 191–201.

Durbin R., Eddy S., Krogh A., and Mitchison G. 1998. Profile HMMs for sequence families. In *Biological sequence analysis. Probabilistic models of proteins and nucleic acids,* chap. 5. Cambridge University Press, United Kingdom.

Gribskov M. and Veretnik S. 1996. Identification of sequence pattern with profile analysis. *Methods Enzymol.* **266:** 198–212.

Henikoff S. and Henikoff J.G. 1992. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci.* **89:** 10915–10919.

———. 1997. Embedding strategies for effective use of information from multiple sequence alignments. *Protein Sci.* **6:** 698–705.

———. 2000. Amino acid substitution matrices. *Adv. Protein Chem.* **54:** 73–97.

Henikoff S., Henikoff J.G., Alford W.J., and Pietrokovski S. 1995. Automated construction and graphical presentation of protein blocks from unaligned sequences. *Gene* **163:** GC17–26.

Henikoff S., Greene E.A., Pietrokovski S., Bork P., Attwood T.K., and Hood L. 1997. Gene families: The taxonomy of protein paralogs and chimeras. *Science* **278:** 609–614.

Hobohm U., Scharf M., Schneider R., and Sander C. 1992. Selection of representative protein data sets. *Protein Sci.* **1:** 409–417.

Jurka J. 1998. Repeats in genomic DNA, mining and meaning. *Curr. Opin. Struct. Biol.* **8:** 333–337.

Ladunga I., Wiese B.A., and Smith R.F. 1996. FASTA-SWAP and FASTA-PAT: Pattern database searches using combinations of aligned amino acids, and a novel scoring theory. *J. Mol. Biol.* **259:** 840–854.

Lipman D.J. and Pearson W.R. 1985. Rapid and sensitive protein similarity searches. *Science* **227:** 1435–1441.

Madej T., Gibrat J.F., and Bryant S.H. 1995. Threading a database of protein cores. *Proteins* **23:** 356–369.

Murzin A.G., Brenner S.E., Hubbard T., and Chothia C. 1995. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* **247:** 536–540.

Neuwald A.F., Liu J.S., Lipman D.J., and Lawrence C.E. 1997. Extracting protein alignment models from the sequence database. *Nucleic Acids Res.* **25:** 1665–1677.

Pearson W.R. 1990. Rapid and sensitive sequence comparison with FASTP and FASTA. *Methods Enzymol.* **183:** 63–98.

———. 1991. Searching protein sequence libraries: Comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. *Genomics* **11:** 635–650.

———. 1995. Comparison of methods for searching protein sequence databases. *Protein Sci.* **4:** 1150–1160.

———. 1996. Effective protein sequence comparison. *Methods Enzymol.* **266:** 227–258.

———. 1998. Empirical statistical estimates for sequence similarity searches. *J. Mol. Biol.* **276:** 71–84.

———. 2000. Flexible sequence similarity searching with the FASTA3 program package. *Methods Mol. Biol.* **132:** 185–219.

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

Pearson W.R. and Miller W. 1992. Dynamic programming algorithms for biological sequence comparison. *Methods Enzymol.* **210:** 575–601.

Pearson W.R., Wood T., Zhang Z., and Miller W. 1997. Comparison of DNA sequences with protein sequences. *Genomics* **46:** 24–36.

Pietrokovski S. 1996. Searching databases of conserved sequence regions by aligning protein multiple-alignments. *Nucleic Acids Res.* **24:** 3836–3845.

Retief J.D., Lynch K.R., and Pearson W.R. 1999. Panning for genes — A visual strategy for identifying novel gene orthologs and paralogs. *Genome Res.* **9:** 373–382.

Sander C. and Schneider R. 1991. Database of homology derived protein structures and the structural meaning of sequence alignment. *Proteins* **9:** 56–68.

Schäffer A.A., Wolf Y.I., Ponting C.P., Koonin E.V., Aravind L., and Altschul S.F. 1999. IMPALA: Matching a protein sequence against a collection of PSI-BLAST constructed position-specific score matrices. *Bioinformatics* **15:** 1000–1011.

Tatusov R.L., Koonin E.V., and Lipman D.J. 1997. A genomic perspective on protein families. *Science* **278:** 631–637.

Walker D.R. and Koonin E.V. 1997. SEALS: A system for easy analysis of lots of sequences. *Ismb* **5:** 333–339.

Wilbur W.J. and Lipman D.J. 1983. Rapid similarity searches of nucleic acid and protein data banks. *Proc. Natl. Acad. Sci.* **80:** 726–730.

Wootton J.C. and Federhen S. 1993. Statistics of local complexity in amino acid sequences and sequence databases. *Comput. Chem.* **17:** 149–163.

———. 1996. Analysis of compositionally biased regions in sequences. *Methods Enzymol.* **266:** 554–571.

Worley K.C., Wiese B.A., and Smith R.F. 1995. BEAUTY: An enhanced BLAST-based search tool that integrates multiple biological information resources into sequence similarity search results. *Genome Res.* **5:** 173–184.

Wu C.H., Zhao S., Chen H.L., Lo C.J., and McLarty J. 1996. Motif identification neural design for rapid and sensitive protein family search. *Comput. Appl. Biosci.* **12:** 109–118.

Zhang Z., Pearson W., and Miller W. 1997. Aligning a DNA sequence with a protein sequence. *J. Comput. Biol.* **4:** 333–443.

Zhang Z., Schäffer A.A., Miller W., Madden T.L., Lipman D.J., Koonin E.V., and Altschul S.F. 1998. Protein sequence similarity searches using patterns as seeds. *Nucleic Acids Res.* **26:** 3986–3990.

Zhu J., Liu J.S., and Lawrence C.E. 1998. Bayesian adaptive sequence alignment algorithms. *Bioinformatics* **14:** 25–39.

This Page Intentionally Left Blank