Comparison of protein sequences and practical database searching

Golan Yona (1) and Steven Brenner (2)

Department of Structural Biology,
Stanford University, USA.

(1) Present address: Department of Computer Science,
Cornell University, Ithaca, USA.

(2) Present address: Department of Plant and Microbial Biology,
University of California, Berkeley, USA.

Contents

Chapter 1

1.1 Introduction

During the last three decades a considerable effort has been made to develop algorithms that compare sequences of biological macromolecules (proteins, DNA). The purpose of such algorithms is to detect evolutionary, and thus structural and functional relations among sequences. Successful sequence comparison would allow to infer the biological properties of new sequences from data accumulated on related genes. For example, a similarity between a translated nucleotide sequence and a known protein sequence suggests a homologous coding region in the corresponding nucleotide sequence. Significant sequence similarity among proteins may imply that the proteins share the same secondary and tertiary structure, and have close biological functions. The prediction of unknown protein structures is often based on the study of known structures of homologous proteins.

Today, the routine procedure for analysis of a new protein sequence almost always starts with a comparison of the sequence at hand with the sequences in one or more of the main sequence databases. A new sequence is analyzed by extrapolating the properties of its "neighbors" in a database search. Such methods were applied during the last three decades with much success and helped to identify the biological function of many protein sequences, as well as to reveal many distant and interesting relationships between protein families. Actually, more sequences have been putatively characterized by database searches, than by any other single technology.

Detecting homology may often help in determining the function of new proteins. By definition, homologous proteins have evolved from the same ancestor protein. The degree of sequence conservation varies among protein families. Yet, homologous proteins almost always have the same fold [**?**, **?**, **?**]. Although the common evolutionary origin of two proteins is almost never directly observed, we can deduce homology, with a high statistical confidence, given that the sequence similarity is significant.

In principle, similarity does not necessarily imply homology (similarity may be quantified whereas homology is a relation that either holds or does not hold). Therefore, similarity should be used carefully in attempting to deduce homology. The deduction of biological function out of sequence similarity is not straight forward, and sequence comparison procedures may lead to false conclusions when applied simple-mindedly. Today sequence comparison algorithms are accompanied with statistical estimates which provide a measure of statistical significance of the observed sequence similarities. These estimates can further help in assessing the significance of the similarity, and in many cases can lead to deduction of homology. The confidence in the deduction clearly depends on the level of statistical significance. In this view, database searches should be treated as experiments analogous to wet-lab characterization. Their use deserves the same care both in the design of the experiment and in the interpretation of results.

Planning a good experiment requires understanding of the methods being applied. Fundamentally, database searches are a simple operation: a query sequence is aligned with each of the sequences (called targets) in a database. A score is computed from each alignment, and the query/target pairs with the best scores are then reported to the user. Statistics is used to help improve the ability to interpret these scores and distinguish true relations among proteins from chance similarities. A more detailed description of this process, the sequence comparison algorithms, the scoring schemes, and the statistics of sequence alignments is given next.

1.2 Alignment of sequences

During evolution sequences have changed by insertions, deletions and mutations. These evolutionary events may be traced today by applying algorithms for sequence alignment. Suppose that a DNA sequence **a** has evolved to the sequence **b** through substitutions, insertions and deletions. This transformation can be represented by an alignment where **a** is written above **b** with the common (conserved) bases aligned appropriately. For example, say that $\mathbf{a} = ACTTGA$ and **b** is obtained by substituting the second base from C to G, inserting an A between the second and the third bases, and by deleting the fifth base (G). The corresponding alignment will be:

$$\begin{array}{ccccccccc} \mathbf{a} & = & A & C & - & T & T & G & A \\ \mathbf{b} & = & A & G & A & T & T & - & A \\ score & & 1 & 0 & -1 & 1 & 1 & -1 & 1 \end{array}$$

We usually do not actually know which sequence evolved from the other. Therefore the events are not directional and insertion of A in **b** might have been a deletion of A in **a**.

In a typical application we are given two related sequences and we wish to recover the evolutionary events that transformed one to the other. The goal of sequence alignment is to find the *correct* alignment that encodes the true series of evolutionary events which have occurred. The alignment can be assigned a score which accounts for the number of identities (a match of two identical letters), the number of substitutions (a match of two different letters), and the number of gaps (insertions/deletions). For example, in the alignment above a score of 1 was given for each identity, a score of 0 was given for each substitution, and a negative score of -1 was given for each gap. Overall, the alignment scored 2, which is the sum of all pair scores and gap scores. In general, the scores for identities and substitutions which are used to score the alignment are called the **scoring matrix**, and the scores for gaps are called **gap penalties**. Altogether they are called the **scoring scheme** (see section 1.4.5 for details) . With high (positive) scores for identities, and low (or negative) scores for substitutions and gaps, the basic strategy towards tracing the correct alignment seeks the alignment which scores best. In the following sections we describe in detail the common algorithms for sequence comparison. The discussion focuses on the comparison of protein sequences, but it holds for DNA sequences as well.

1.2.1 Rigorous alignment algorithms

There are several different alignment algorithms which have become a standard tool for biologists. The rigorous algorithms use dynamic programming to find the optimal alignment.

Global alignment: The first to propose a dynamic programming algorithm for comparison of macromolecules, were Needleman and Wunsch [**?**]. Their algorithm performs a **global alignment** of the sequences; i.e., an alignment where all letters of **a** and **b** are accounted for. This type of alignment is appropriate when similarity is expected along the whole or most of the sequence.

Formally, let $s(a_i, b_j)$ be the similarity score of a_i, b_j (the scoring matrix) and let $\alpha > 0$ be the penalty for deleting/inserting of one amino acid. The score of an alignment with N_{ij} matches of a_i and b_j and N_{gap} insertions/deletions is defined as

$$\sum_{i,j} N_{ij} \cdot s(a_i, b_j) - N_{gap} \cdot \alpha$$

In sequence evolution, an insertion or deletion of a segment (several adjacent amino acids) usually occurs as a single event. That is, the opening of the gap is the significant event. Therefore, most computational models assign a penalty for a gap of length k that is smaller than the sum of k independent gaps of length 1, by charging large penalty for opening a gap, and a smaller penalty for each extension (**affine** or linear gap penalty). If the penalty for gap of length k is $\alpha(k)$, and N_{k-gap} is the number of gaps of length k in a given alignment, then the score of this alignment is defined in this case as

$$\sum_{i,j} N_{ij} \cdot s(a_i, b_j) - \sum_{k} N_{k-gap} \cdot \alpha(k)$$

The **global similarity** of sequences **a** and **b** is defined as the largest score of any alignment of sequences **a** and **b**, i.e.

$$S(\mathbf{a}, \mathbf{b}) = max_{alignments}\{\sum_{i,j} N_{ij} \cdot s(a_i, b_j) - \sum_{k} N_{k-gap} \cdot \alpha(k)\}$$

In principle, the number of possible alignments is exponentially large, what makes it impossible to perform a direct search. However, a dynamic programming algorithm makes it possible to find the optimal alignment without checking all possible alignments, but only a very small portion of the search space. In brief, the idea is that every subalignment in the optimal alignment should be optimal as well (otherwise it would be possible to improve the overall alignment by improving the subalignments, in contrast with its definition as optimal alignment). Since any optimal subalignment (say, of the substring $a_1 a_2..a_i$ with the substring $b_1 b_2..b_j$) can end only in one of following three ways:

$$\begin{matrix} a_i \\ b_j \end{matrix} \quad \text{or} \quad \begin{matrix} a_i \\ - \end{matrix} \quad \text{or} \quad \begin{matrix} - \\ b_j \end{matrix}$$

every possible subalignment is calculated only once, and in constant time[1], out of its optimal subalignments.

Formally speaking, denote by $S_{i,j}$ the score of the best alignment of the substring $a_1 a_2..a_i$ with the substring $b_1 b_2..b_j$, i.e.

$$S_{i,j} = S(a_1 a_2..a_i \ , \ b_1 b_2..b_j)$$

Assume that the gap penalty is constant and equals α. Then, after an initialization step

$$S_{0,0} = 0 \qquad S_{i,0} = -i \cdot \alpha \ \text{ for } i = 1..n \qquad S_{0,j} = -j \cdot \alpha \ \text{ for } j = 1..m$$

(where n and m are the lengths of the sequences **a** and **b** respectively) define $S_{i,j}$ recursively

$$S_{i,j} = max\{S_{i-1,j-1} + s(a_i, b_j),\ S_{i,j-1} - \alpha,\ S_{i-1,j} - \alpha\}$$

Therefore, the score $S(\mathbf{a}, \mathbf{b})$ can be calculated recursively. Since the subalignment for each i and j has to be calculated, the time complexity of this algorithm is proportional to the product of

[1]This is true with linear gap functions. With non-linear gap penalties, the calculation of this optimal subalignment may need up to $i + j + 1$ operations.

the lengths of the sequences compared (a quadratic time complexity). In practice, the scores are stored in a two-dimensional array of size $(n+1)\cdot(m+1)$. The initialization set the values at row zero and column zero and the computation proceeds row by row so that the value of each matrix cell is calculated from entries which were already calculated (see figure Figure 1.1).

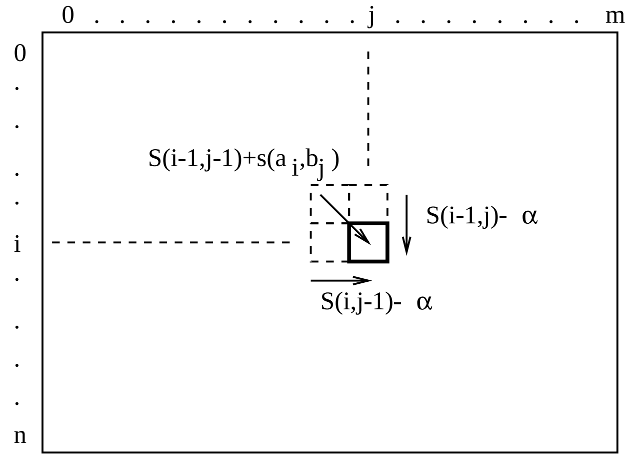

Figure 1.1: **Calculating the global similarity score**. The score of the (i,j) entry in the matrix is calculated from three matrix cells: the one on the left, the one on the top and the one located at the top left corner of the current cell. In case of a non-constant gap penalty we need also to check all the cells in the same row and all the cells in the same column (along the dashed lines).

Local alignment: In many cases the similarity of two sequences is limited to a specific motif or domain, the detection of which may yield valuable structural and functional insights, while outside of this motif/domain the sequences may be essentially unrelated. In such cases global alignment may not be the appropriate tool. In the search for an optimal global alignment, local similarities may be masked by long unrelated regions. Consequently, the score of such an alignment can be as low as for totally unrelated sequences. Moreover, the algorithm may even misalign the common region. Therefore, usually it is better to compare sequences locally. A **local alignment** of **a** and **b** is defined as an alignment between a *substring* of **a** and a *substring* of **b**. The **local similarity** of sequences **a** and **b** is defined as the maximal score over all possible local alignments.

The algorithm which finds the best local alignment is based on a minor modification of the dynamic programming algorithm for global alignment. Specifically, whenever the score of the optimal subalignment of two subsequences becomes negative, the score is set to zero, meaning that the corresponding subsequences should not be aligned. Following the notations of the previous section, $S_{i,j}$ is now defined

$$S_{i,j} = max\{0,\ S_{i-1,j-1} + s(a_i,b_j),\ S_{i,j-1} - \alpha,\ S_{i-1,j} - \alpha\}$$

In the literature, this algorithm is often called the Smith-Waterman (SW) algorithm, after those who introduced this modification [**?**].

There is a lot of literature regarding dynamic programming algorithms in general [**?**], and for sequence comparison specifically [**?**, **?**, **?**]. The interested reader is referred to these books for more details on the algorithmic aspects of this method, as well as its computational aspects.

1.2.2 Heuristic algorithms for sequence comparison

In a typical application new protein sequence is compared with all sequences in the database (**library sequences**), in search of related proteins. Because of its quadratic time complexity, the dynamic programming algorithms may not be suitable for this purpose. For example, the comparison of a sequence, of average length of 350 amino acids, against a typical database (like SWISSPROT [**?**], with more than 80,000 sequences), may take few CPU hours on a standard PC of nowadays (pentium-II 400 MHz).

Several algorithms have been developed to speed up the alignment procedure. The two main algorithms are FASTA [**?**] and BLAST [**?**]. These are heuristic algorithms which are not guaranteed to find the optimal alignment. However, they proved to be very effective for sequence comparison, and they are significantly faster than the rigorous dynamic programming algorithm[2].

BLAST (Basic Local Alignment Search Tool)

BLAST compares two sequences and seeks all pairs of similar segments, whose similarity score exceeds a certain threshold. These pairs of segments are called "high scoring segment pairs" (HSPs). A segment is always a contiguous subsequence of one of the two sequences. Segment pair is a pair of segments of the same length, one from each sequence. Hence the alignment of the segments is without gaps. The score of the match is simply the sum of matches of the amino acids (defined by a scoring matrix) along the segment pair. The segment pair with the highest score is called the "maximum segment pair" (MSP).

To identify the HSPs (and particularly, the MSP), the algorithm starts by locating "seeds" of similarity among the query sequence and the database sequence that score at least T, and then extends them in both direction until the maximum possible score for the extension is reached. The changes in the threshold T permit a tradeoff between speed and sensitivity. A higher value of T yields greater speed, but also an increased probability of missing weak similarities. Finally, multiple MSP regions are combined. For each consistent combination, its probability is calculated using the Poisson or sum statistics [**?**] and the most significant hits (lowest probability) are reported.

The algorithm is an outgrowth of the statistical theory for local alignments without gaps (see section 1.3). This theory gives a framework for assessing the probability that a given similarity between two protein sequences (i.e. the MSP) could have emerged by chance. If the probability is very low, then the similarity is statistically significant and the algorithm reports the similarity along with its statistical significance. Though the algorithm may miss complex similarities which include gaps, the statistical theory of alignments without gaps provided a reliable and efficient way of distinguishing true homologies from chance similarities, thus making this algorithm an important tool for molecular biologists.

Current improvements of BLAST allow gapped alignments, by using dynamic programming to extend a central seed in both directions [**?**]. This is complemented by PSI-BLAST, an iterative version of BLAST, with a position-specific score matrix (see section 1.4.5) that is generated from significant alignments found in round i and used in round $i+1$. The latter may better detect weak similarities that are missed in database searches with a simple sequence query.

[2] In the last few years, biotechnology companies such as Compugen and Paracel, have developed special purpose hardware that accelerates the dynamic programming algorithm [**?**]. This special-purpose hardware has again made the dynamic programming algorithm competitive with FASTA and BLAST, both in speed and in simplicity of use. However, meanwhile, FASTA and BLAST have become standard in this field and are being used extensively by biologists all over the world. Both algorithms are fast, effective, and do not require the purchase of additional hardware. BLAST has an additional advantage, as it may reveal similarities which are missed by the dynamic programming algorithm, for example when two similar regions are separated by a long dissimilar region.

FASTA

FASTA is another heuristic that performs a fast sequence comparison. The algorithm starts by creating a hash table of all k-tuples in the query sequence (usually, k = 1 or 2 for protein sequences, where k=1 gives higher sensitivity). This table stores the k-tuples in a way which enables fast accession, and restoration of each k-tuple. Then, when scanning a library sequence, each k-tuple of the library sequence is looked up in the hash table, and if it is found (this means k-tuple identity) it is marked. At a second stage, the ten regions with the highest density of identities are rescanned. Common k-tuples which are on the same diagonal (same offset in both sequences), and not very far apart (the exact parameters are set heuristically), are joined to form a region (a gapless local alignment, or HSP in BLAST terminology). The regions are scored to account for the matches as well as the mismatches, and the best region is reported (its score is termed "initial score" or "init1"). Then, the algorithm tries to join nearby high scoring regions, even if they are not on the same diagonal (the corresponding score being termed "initn score"). Finally, a bounded dynamic programming is run in a band around the best region, to obtain the "optimized score". If the sequences are related then the optimized score is usually much higher than the initial score.

1.3 Probability and statistics of sequence alignments

In the evolution of protein sequences, not all regions mutate at the same rate. Regions which are essential for the structure and function of proteins, are more conserved. Therefore, significant sequence similarity of two proteins may reflect a close biological function or a common evolutionary origin. The algorithms that were described in the previous section can be used to identify such similarities. However, on any two input protein sequences, even if totally unrelated, the algorithms almost always find some similarity. For unrelated sequences this similarity is essentially random. As the length of the sequences compared increase, this random similarity may increase as well. Therefore, in order to assess the significance of a similarity score it is important to know what score to expect simply by chance.

Naturally, we would like to identify those similarities which are genuine, and biologically meaningful. In the view of the last paragraph, the raw similarity score may not be appropriate for this purpose. However, when the sequence similarity is statistically significant we can deduce, with high confidence level, that the sequences are related[3]. The reverse implication is not always true. We encounter many examples of low sequence similarity despite functional and structural similarity [**?**, **?**, **?**].

Though statistically significant similarity is neither necessary nor sufficient for a biological relationship, it may give us a good indication of such relationship. When comparing a new sequence against the database, in search of close relatives, this is extremely important, as we are interested in reporting only significant hits, and sorting the results according to statistical significance seems reasonable.

To estimate the statistical significance of similarity scores, a statistical theory should be developed. A great effort was made in the last two decades to establish such statistical theory. Currently, there is no complete theory, though some important results were obtained. These results have very practical implications and are very useful for estimating the statistical significance of similarity scores. The statistical significance of similarity scores for "real" sequences is defined by the probability that the same score would have been obtained for random sequences. The statistical results

[3]Two exceptions are segments with unusual amino acid composition, and similarity that is due to convergent evolution.

concern the similarity scores of random sequences, when the similarity scores are defined by ungapped alignments. However, these results have created a framework for assessing the statistical significance of various similarity scores, including gapped sequence alignments, and recently, even structural alignments [**?**].

1.3.1 Statistics of global alignment

Though the distribution of global similarity scores of random sequences has not been characterized yet, some important properties of this distribution were partly determined. The main characteristic of this distribution is the linear growth (or decline, depends on the mean of the scoring matrix) with the sequence length. I.e., the **expected** global similarity score grows **linearly** with the sequences length. However, the growth rate has not been determined.

The statistical significance of a similarity score obtained for "**real**" sequences, which exceeds the expected score by a certain amount, is estimated by the probability that the similarity score of **random** sequences would exceed the expected mean by the same amount. However, since the distribution of scores is unknown, the available estimates give only a rough bound on that probability. The variance of the global similarity score has not been determined either, and the best results give only an upper bound.

In practice, it is possible to empirically approximate the distribution by shuffling the sequences and comparing the shuffled sequences. By repeating this procedure many times it is possible to estimate the mean and the variance of the distribution, and a reasonable measure of statistical significance (e.g. by means of the z-score) can be obtained. Formally, denote by S the global similarity score. Let μ and σ^2 be the mean and the variance of the distribution of scores. Then, the z-score associated with the score S is defined as $\frac{S-\mu}{\sigma}$. This score measures how many units of standard deviation apart the score S is from the mean of the distribution. The larger it is, the more significant is the score S.

1.3.2 Statistics of local alignment without gaps

The statistics of ungapped similarity scores has been studied extensively since the early 90's. The exclusion of gaps allowed a rigorous mathematical treatment, and several important results were obtained. Karlin and Altschul [**?**] have shown that for two random sequences of length n and m, the score of the best ungapped local alignment (the MSP score in BLAST jargon) is centered around $\frac{\ln(n \cdot m)}{\lambda}$, where λ is a parameter that depends on the overall **background distribution** of amino acids in the database, and the scoring matrix. That is, the local similarity score grows **logarithmically** with the length of the sequences, and with the size of the search space $n \cdot m$.

This result in itself is still not enough to obtain a measure of statistical significance for local similarity scores. This can be done only once a concentration of measure result is obtained or the distribution of similarity scores is defined. Indeed, one of the most important results in this field is the characterization of the distribution of local similarity scores without gaps. This distribution was shown to follow the extreme value distribution [**?**, **?**, **?**].

Formally, as the **sum** of many i.i.d random variables is distributed **normally**, then the **maximum** of many i.i.d random variables is distributed as an **extreme value distribution** [**?**]. This distribution is characterized by two parameters: the index value u and the decay constant λ (for $u = 0$ and $\lambda = 1$, the distribution is plotted in Figure 1.2). The distribution is not symmetric. It is positive definite and unimodal with one peak at u. Practically, the score of the best local alignment (the MSP score) is the maximum of the scores of many independent alignments, which explains the observed distribution.

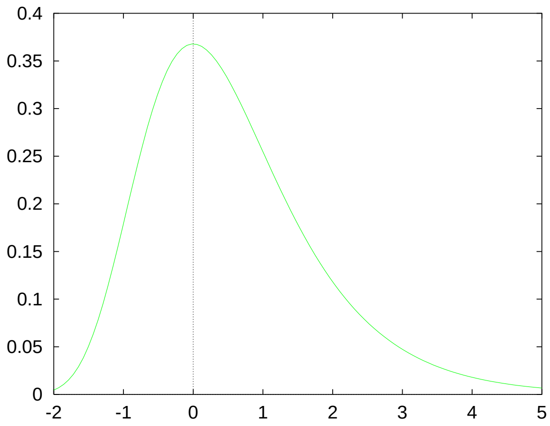

Figure 1.2: **Probability density function for the extreme value distribution with $u = 0$ and $\lambda = 1$.**

Specifically, $\mathcal{S}$, the local similarity score of two random sequences of length n and m, is distributed as an extreme value distribution and

$$Prob(\mathcal{S} \geq x) \sim 1 - \exp(-e^{-\lambda \cdot (x-u)}) \tag{1.1}$$

where $u = \frac{\ln Kmn}{\lambda}$, and K is a constant that can be estimated from the background distribution and the scoring matrix [**?**].

For a large x we can use the approximation $1 - \exp(-e^{-x}) \sim e^{-x}$. Therefore, for a large x,

$$Prob(\mathcal{S} \geq x) \sim e^{-\lambda(x-u)} = e^{-\lambda \cdot x} e^{\lambda \cdot u} = Kmne^{-\lambda \cdot x} \tag{1.2}$$

This result helps to calculate the probability that a given MSP score could have been obtained by chance. The score will be statistically significant at the 1% level if $\mathcal{S} \geq x_0$ where x_0 is determined by the equation $Kmne^{-\lambda \cdot x_0} = 0.01$. In general, a pairwise alignment with score $\mathcal{S}$ has a **p-value** of p where $p = Kmne^{-\lambda \cdot \mathcal{S}}$. I.e., there is a probability p that this score could have happened by chance.

The probability p, that a similarity score $\mathcal{S}$ could have been obtained simply by chance from the comparison of two random sequences, should be adjusted when multiple comparisons are performed. One example of this is when a sequence is compared with each of the sequences in a database with D sequences. Denote by ***p*-match** a match between two sequences that has a p-value $\leq p$ (i.e., its score $\geq \mathcal{S}$). The probability P of observing at least one p-match (i.e., at least one "success"), in a database search follows the Poisson distribution

$$P = Prob(\text{at least one } p\text{-match}) = 1 - e^{-Dp} \tag{1.3}$$

and for $Dp < 0.1$ is well approximated by

$$P \simeq Dp$$

Since not all library sequences have the same probability of sharing a similar region with the query sequence, D should be replaced with the effective size of the database. If the query sequence is of length n, and the (pairwise) alignment of interest involves a library segment of length m, and the database has a total of N amino acids, then D should be replaced with N/m. Thus,

$$P \simeq \frac{N}{m} p = KNne^{-\lambda \cdot \mathcal{S}} \tag{1.4}$$

so the effective size of the search space is Nn (intuitively, this is the number of possible starting positions of a match).

It is very common to use the **expectation value** (e-value) as a measure of statistical significance. The expectation value of the the Poisson distribution is given by

$$E = E(\text{number of } p-\text{matches}) = Dp \tag{1.5}$$

and as discussed above, D should be replaced with N/m. Hence

$$E = KNne^{-\lambda \cdot \mathcal{S}} \tag{1.6}$$

This is the expected number of distinct matches (segment pairs) that would obtain a score $\geq \mathcal{S}$ by chance in a database search, with a database of size N (amino acids) and composition $\mathcal{P}$ (the background distribution of amino acids). The higher it is, the match is less significant. For example, if $E = 0.01$, then the expected number of random hits with a score $\geq \mathcal{S}$ is 0.01. In other words, we may expect a random hit with that score only once in 100 independent searches. If $E = 10$, then we should expect 10 hits with a score $\geq \mathcal{S}$ by chance, in a single database search. This means that such a hit is not significant. (Note that $E \simeq P$ for $P < 0.1$).

Finally, by setting a value for E and solving the equation above for $\mathcal{S}$, it is possible to define a threshold score, above which hits are reported. This is the score above which the number of hits that are expected to occur at random is $< E$. Therefore, we can deduce that a match with this score or above reflects true biological relationship, but we should expect up to E errors per search. The specific value of E affects both the sensitivity of a search (the number of true relationships detected) and its selectivity (the number of errors). A lower value of E would decrease the error rate. However, it would decrease the sensitivity as well. A reasonable choice for E is between 0.1 and 0.001.

1.3.3 Statistics of local alignment with gaps

Though local alignments without gaps may detect most similarities between related proteins, and give a good estimation of the similarity of the two sequences, it is clear that gaps in local alignments are crucial in order to obtain the correct alignment, and for a more accurate measure of similarity. However, no precise model has been proposed yet to explain gaps in alignments. Moreover, introducing gaps in alignments greatly complicates their mathematical tractability. Rigorous results have been obtained only for local alignments without gaps.

Recent studies suggest that the score of local gapped alignments can be characterized in the same manner as the score of local ungapped alignments: As was mentioned in the previous section, the local **ungapped** similarity score grows logarithmically with the sequence's length and the size of the search space. Arratia & Waterman [1994] have shown that for a range of substitution matrices and gap penalties, local **gapped** similarity scores have the same asymptotic characteristic. Furthermore, empirical studies [**?**, **?**] strongly suggest that local gapped similarity scores are distributed according to the extreme value distribution, though some correction factors may apply [**?**].

Based on empirical observations, Pearson [1998] has derived statistical estimates for local alignment with gaps, using the extreme value distribution for scores obtained from a database search. A database search provides tens of thousands of scores from sequences which are unrelated to the query sequence, and therefore are effectively random. As discussed above, these scores are thus expected to follow the extreme-value distribution. This is true as long as the gap penalties are not too low. Otherwise the alignments shift from local to global and the extreme value distribution no longer apply.

Since the logarithmic growth in the sequence length holds in this case, scores are corrected first for the expected effect of sequence length. The correction is done by calculating the regression line $S = a + b \cdot \ln n$ for the scores obtained in a database search, after removing very high scoring sequences (probably related sequences). The process is repeated as many as five times. The regression line and the average variance of the normalized scores are used to define the **z-score**:

$$z\mathsf{-score} = \frac{S - (a + b \cdot \ln n)}{var}$$

and the distribution of z-scores is approximated by the extreme value distribution

$$p = Prob(z\mathsf{-score} > x) = 1 - \exp(-e^{c_1 \cdot x - c_2})$$

where c_1 and c_2 are constants, and the expectation value is defined as before by $E(z\mathsf{-score} > x) = N \cdot p$ where N is the number of sequences in the database (the number of tests).

This empirical approach has the advantage of internal calibration of the accuracy of the estimates, and has proved to be very accurate in estimating the statistical significance of gapped similarity scores [**?**] (see also [**?**, **?**]).

1.4 Practical database searching

1.4.1 Types of comparison

To formulate the database search "experiment", it is first necessary to decide what types of sequences will be compared: DNA, Protein, or DNA as Protein. The algorithms described above may be applied to the comparison of protein sequences as well as to DNA sequences (coding or non-coding regions). However, the comparison of protein sequences has proven to be a much more effective tool [**?**]. Though the evolutionary events occur at the DNA level, the main genetic pressure is on the protein sequence. Moreover, mutations at the DNA level do not necessarily change the encoded amino acid due to the redundancy of the genetic code. Mutations often result in conservative substitutions at the protein level, namely, replacement of an amino acid by another amino acid with similar biochemical properties. Such changes tend to have only a minor effect on the protein's functionality. Therefore, if the sequence under consideration either is a protein or codes for a protein, then it is almost always the case that the search should take place at the protein level, as proteins allow one to detect far more distant homologies than DNA. Another aspect is that in DNA comparisons, there is noise from comparisons of non-coding frames (though this latter issue still arises in DNA as Protein searches). DNA versus DNA comparison is typically only used to find identical regions of sequence in a database. One would do such a search to discover whether another group has sequenced or studied a gene, and to learn where it is expressed or where splice junctions occur. In short, protein-level searches are valuable for detecting evolutionarily related genes, while DNA searches are best for locating nearly identical regions of sequence (see table 1.1 for available comparison programs and the corresponding types of comparison).

1.4.2 Databases

Next, it is necessary to select a database to search against. There are several commonly used databases (e.g., GenBank, SwissProt, ESTs, etc.). For homology searches, it is best to use a comprehensive collection of all known proteins. Two such databases are available. One is the nr database at the NCBI website (http://www.ncbi.nlm.nih.gov/). The nr (which stands for non-redundant)

Programs	Query	DB	Comparison	Common Use
blastn, fasta, ssearch	DNA	DNA	DNA-level	Seek identical DNA sequences, and splicing patterns
blastp, fasta, ssearch	Protein	Protein	Protein-level	Seek homologous proteins
blastx, fastx	DNA	Protein	Protein-level	Query new DNA to find genes and seek homologous proteins
tblastn, tfasta, tfastx	Protein	DNA	Protein-level	Search for genes in un-annotated DNA
tblastx	DNA	DNA	Protein-level	Discover gene structure

Table 1.1: **Comparison programs and types of comparison.**

protein database combines data from several sources (GenPept, SwissProt, PIR, RPF and PDB) removes the redundant identical sequences, and yields a collection with nearly all known proteins. The second nr database is available at the ExPASy website in Switzerland (http://www.expasy.ch/). Both databases are frequently updated, to incorporate as many sequences as possible. Obviously, a search will not identify a sequence that has not been included in the database, and since databases are growing so rapidly, it is essential to use a current database.

The main sources of these non-redundant databases are the SwissProt database and the TrEMBL database [?], the PIR database [?], and the GenPept database [?]. The SwissProt database is maintained at the ExPASy center in Switzerland. This is a non-redundant highly annotated database which offers a lot of valuable biological information on almost all of its entries (more than 80,000 in the latest release, August 1999). Such information may include for example the description of the function of a protein, its domain structure, post-translational modifications, etc. This database is supplemented by TrEMBL, which is a collection of all the translations of EMBL nucleotide sequence entries not yet integrated in SwissProt. For most of these entries some biological information is available, usually based on sequence analysis carried by the ExPASy team. PIR is another database that offers a lot of biological information on entries through an extensive annotation as well as classification to families and superfamilies and links to alignments with other family members. GenPept is a database that contains all translations of DNA sequences in the GenBank database.

Several specialized databases are also available, all of which overlap with the composite non-redundant databases. For example, if one is interested in searching for proteins of known structure, it is best to just search the smaller PDB database. Other specialized databases are available for each of the fully sequenced genomes, as well as for subsets of protein families (such as protein kinases or immunoglobulins), etc. See table 1.2 for a list of the main databases.

Protein Database	Number Entries	Availability	Description
nr (ExPasy)		ftp://www.expasy.ch/databases/sp_tr_nrdb/	consist of SwissProt, TrEMBL
nr (NCBI)		ftp://ncbi.nlm.nih.gov/blast/db/	consist of GenPept, SwissProt, PIR, RPF, PDB
SwissProt	81,581	http://www.expasy.ch/sprot/sprot-top.html	non-redundant, high level of annotation
TrEmbl	197,766	http://www.expasy.ch/sprot/sprot-top.html	non-redundant, computer annotated
PIR	157,586	http://www-nbrf.georgetown.edu/pirwww/pirhome.shtml	non-redundant, annotated, family classification
GenPept		http://www.ncbi.nlm.nih.gov/Entrez/protein.html	translation of DNA sequences in GenBank
PDB	10,963	http://www.rcsb.org/pdb/index.html	repository of all known 3D structures
Genomes		http://www.ncbi.nlm.nih.gov/Entrez/Genome/org.html	protein sequences sorted by organism
DNA Database	Number Entries	Availability	Description
GenBank	4,865,000	http://www.ncbi.nlm.nih.gov/Entrez/protein.html	annotated

Table 1.2: **Sequence databases.** Number of entries is updated to October 1999.

One may also wish to search DNA databases at the protein level. Programs can do so automatically by first translating the DNA in all six reading frames and then making comparisons with each of these conceptual translations. The nr DNA database (containing most known DNA sequence except GSS, EST, STS, or HTGS sequences) is useful to search when hunting new genes; the identified genes in this database would already be in the protein nr database. Searches against the GSS, EST, STS, and HTGS databases can find new homologous genes, and are especially useful to learn about expression data or genome map location.

1.4.3 Algorithms

The choice of the comparison algorithm should be based on the desired comparison type, the available computational resources, and the goals of the search. All standard comparison algorithms can be run over the Web and can be downloaded from the FTP site to run locally (see table 1.3). The rigorous smith-waterman algorithm is available, as well as the FASTA program, within the FASTA package. This algorithm is more sensitive than the others, but it is also much slower. The FASTA program is faster, and with the parameter *ktup* set to 1, is almost as sensitive as the smith-waterman algorithm [**?**, **?**]. The fastest algorithm is BLAST, the newest versions of which support gapped alignments [**?**] and provide a reliable, sensitive and fast option (the older versions are slower, detect fewer homologs, and have problems with some statistics). Iterative programs like PSI-BLAST require extreme care in their operation, as they can provide very misleading results; however, they have the potential to find more homologs than purely pairwise methods.

Program	FTP site	Run over the Web
ssearch	ftp://ftp.virginia.edu/pub/fasta/	http://www2.ebi.ac.uk/bic_sw/ http://genome.dkfz-heidelberg.de/genweb/ http://sgbcd.weizmann.ac.il/genweb/ http://www.ch.embnet.org/software/FDF_form.html
fasta	ftp://ftp.virginia.edu/pub/fasta/	http://www2.ebi.ac.uk/fasta3/
blastp	ftp://ncbi.nlm.nih.gov/blast/	http://www.ncbi.nlm.nih.gov/BLAST/ http://www2.ebi.ac.uk/blastall/ http://www.ch.embnet.org/software/BottomBLAST.html?

Table 1.3: **Availability of sequence comparison programs.**

1.4.4 Filtering

The statistics for database searches assumes that unrelated sequences look essentially random with respect to each other. Specifically, the theoretical results that were obtained for the statistics of local alignments without gaps (see section 1.3.2) are subject to the restriction that the amino acid composition of the two sequences that are compared are not too dissimilar [**?**]. Assuming that both sequences are drawn from the background distribution, the amino acid composition of both should resemble the background distribution. Without this restriction the statistical estimates overestimates the probability of similarity scores, and indeed, this is observed in protein sequences with unusual compositions [**?**, **?**]. The most common exceptions are long runs of a small number of different residues (such as a poly-alanine tract). Such regions of a sequence may spuriously obtain extremely high match scores. For this reason, it is recommended to filter out these regions using programs such as SEG [**?**]. The NCBI BLAST server will automatically remove such sections in proteins, replacing them with X, if default filtering is selected. DNA sequences will be similarly masked by DUST. Though these programs automatically remove the majority of problematic matches, some problems invariably slip through; moreover, valid hits may be missed due to masking of part of the sequence. Therefore, it may be helpful to try using different masking parameters.

Other sorts of filtering are also often desirable; for example, iterative searches are prone to contamination by regions of proteins that resemble coiled-coils or transmembrane helices. Here, one protein that is similar only because it has the general characteristics may match initially. The profile then emphasizes these inappropriate characteristics, eventually causing many spurious hits. Heavily cysteine rich proteins can also obtain anomalous high scores. If these characteristics are not filtered, then it is necessary to carefully review the alignment results to ensure that they have not led to incorrect matches.

1.4.5 Scoring matrices and gap penalties

The next step is to choose the set of parameters for the sequence comparison algorithm. Namely, the scoring matrix and the gap parameters. The default matrices offered with the comparison algorithm (e.g. BLOSUM62 with BLAST, BLOSUM50 with FASTA) are a safe choice. However, it may be fruitful to check other matrices as well. Several different approaches were taken to derive reliable and effective scoring matrices. The most effective matrices are those that are based on actual frequencies of mutations that are observed in closely related proteins. These matrices reflect the biochemical properties of the amino acids, which influence the probability of mutual substitution (exchange occur more frequently among amino acids that share certain properties), and amino acids with similar properties have high pairwise score. Matrices which are based on sequence alignments include the family of PAM matrices [**?**] (and their improvement by [**?**]), the BLOSUM matrices [**?**], and Gonnet matrix [**?**]. Other matrices, which proved to be very effective for protein sequence comparison, are those that are based on structural principles and aligned structures [**?**, **?**].

The two most extensively used families of scoring matrices are the PAM matrices and the BLOSUM matrices. A detailed description of these matrices is given in the next two sections.

The PAM family of scoring matrices

PAM matrices were proposed by Dayhoff et al in 1978 based on observations of hundreds of alignments of closely related proteins. The frequencies of substitution of each pair of amino acids were extracted from alignments of proteins of small evolutionary distance, below 1% divergence, i.e. at most one mutation per 100 amino acids, on average. These frequencies, normalized to account for the frequencies of random occurrences of single amino acids, resulted in the PAM-1 probability transition matrix. The PAM-1 matrix reflects an amount of evolutionary change that yields on average one mutation per 100 amino acids. Accordingly, it is suitable for comparison of proteins which have diverged by 1% or less. The acronym PAM stands for Percent of Accepted Mutations (and hence the distance is in percentages) or for Point Accepted Mutations (and hence the distance in number of mutations per 100 amino acids).

The PAM-1 matrix is then extrapolated to yield the family of PAM-k matrices. Each PAM-k matrix is obtained from PAM-1 by k consecutive multiplication, and is suitable for comparison of sequences which have diverged k%, or are k evolutionary units apart. For example, PAM-250 = $(\text{PAM-1})^{250}$ reflects the frequencies of mutations for proteins which have diverged 250% (250 mutations per 100 amino acids[4]). The actual scoring matrices that are used by search programs are derived from the transition probability matrices and the background probabilities. The score

[4]Though the definition of PAM-250 seems odd, it still make sense, as is subsequently explained.

of each pair $s(a,b)$ is defined as the logarithm of the likelihood ratio of the transition probability M_{ab} (mutation) versus the probability of a random occurrence of the amino acid b in the second sequence, i.e., $s(a,b) = \log \frac{M_{ab}}{p_b}$.

The PAM matrices were later refined by [?] based on much larger data set. The significant differences were detected for substitutions that were hardly observed in the original data set of [?].

The PAM-250 matrix. The PAM-250 matrix is one of the most extensively used matrices in this field. This matrix corresponds to a divergence of 250 mutations per 100 amino acids. Naturally one may ask whether it makes sense to compare sequences which have diverged this much. Surprising as it may seem, when calculating the probability that a sequence remains unchanged after 250 PAMs (this is given by the sum $\sum_a p_a M_{aa}$ where p_a is the probability of a random occurrence of amino acid s and M_{aa} is the diagonal entry in the PAM-250 matrix that corresponds to the amino acid a) the outcome is that such sequences are expected to share about 20% of their amino acids. For reference, note that the expected percentage of identity in a random match is $100 \cdot \sum_a p_a^2$, and for a typical distribution of amino acids (in a large ensemble of protein sequences), we should expect less than 6% identities.

The BLOSUM family of scoring matrices

Unlike PAM matrices, which are extrapolated from a single matrix PAM-1, the BLOSUM series of matrices was constructed by direct observation of sequence alignments of related proteins, at different levels of sequence divergence. The matrices are based on "blocks" - a collection of multiple alignments of similar segments without gaps [?], each block representing a conserved region of a protein family. These blocks provide a list of (accepted) substitutions, and a log-odds scoring matrix can be defined based on the observed relative frequency of aligned pairs of amino acids q_{ab}, and the expected probability of pairs e_{ab} estimated from the population of all observed pairs

$$s_{ab} = \log \frac{q_{ab}}{e_{ab}}$$

To reduce the bias in the amino acid pair frequencies caused by multiple counts from closely related sequences, segments in a block with at least x% identity are clustered and pairs are counted **between** clusters, i.e., pairs are counted only between segments less than x% identical. When counting pairs frequencies between clusters, the contributions of all segments within a cluster are averaged, so that each cluster is weighted as a single sequence. Varying the percentage of identity x within clusters results in a family of matrices BLOSUM-x, where x ranges from 30 to 100. For example, BLOSUM-62 is based on pairs that were counted only between segments less than 62% identical.

Choosing the scoring matrix

When comparing two sequences, the most effective matrix to use is the one which corresponds to the evolutionary distance between them [?]. However, we usually do not know this distance. Therefore, it is recommended to use several scoring matrices which cover a range of evolutionary distances, for example PAM-40, PAM-120 and PAM-250. In general, low PAM matrices are well suited to finding short but strong similarities, while high PAM matrices are best for finding long regions of weak similarity.

Exhaustive evaluations have been carried out to compare the performance of different scoring matrices [?, ?]. These studies show that log-odds matrices derived directly from alignments of highly conserved regions of proteins (such as BLOSUM matrices or the Overington matrix, which is based on structural alignment [?]) outperform extrapolated log-odds matrices based on an evolutionary model, such as PAM matrices. Moreover, the accuracy of alignments based on extrapolated matrices decreases as the evolutionary distance increases. This suggests that extrapolation cannot accurately model distant relationships, and that the PAM evolutionary model is inadequate. BLOSUM matrices were shown to be more effective in detecting homologous proteins. Specifically, BLOSUM-62 and BLOSUM-50 gave superior performance in detecting weak homologies. These matrices offer good overall performance in searching the databases. The best hybrid of matrices for searching in different evolutionary ranges is either BLOSUM 45/62/100 or BLOSUM 45/100 plus the Overington matrix.

Gap penalties

There is no mathematical model to explain the evolution of gaps. Practical considerations (the need for a simple mathematical model, time complexity) have led to the broad use of linear gap functions, where the penalty for a gap of length k is given by $\alpha(k) = \alpha_0 + k \cdot \alpha_1$. Usually a large penalty is charged for opening a gap (α_0), and a smaller penalty is charged for each extension (α_1).

Gonnet et al. [?] have proposed a model for gaps that is based on gaps occurring in pairwise alignments of related proteins. The model suggests an exponentially decreasing gap penalty function. However, a linear penalty function has the advantage of better time complexity, and in most cases the results are satisfactory. Therefore the use of linear gap functions is very common.

The gap parameters that are used as default in the standard comparison programs are usually optimized based on extensive evaluations [?], and it is rarely beneficial to change these from their defaults.

Position dependent scores

In many proteins, mutations are not equally probable along the sequence. Some regions are functionally/structurally important and consequently, the effect of mutation in these regions can be drastic. They may create a nonfunctional protein or even prevent the molecule from folding into its native structure. Such mutations are unlikely to survive, and therefore these regions tend to be more evolutionary conserved than other, less constrained regions (e.g. loops) which can significantly diverge.

Accordingly, it may be appropriate to use position-dependent scores for mismatches and gaps. The incorporation of information about structural preferences can lead to alignments that are more accurate biologically. If a protein's structure is known, the secondary structure should be taken into account. In the absence of such data, general structural criteria, such as the propensities of amino acid for occurring in secondary structures versus loops can be taken into account. For example, the probability of opening a gap in existing secondary structure can be decreased, while the probability for opening/inserting a gap in loop regions can be increased.

Usually position-specific scoring matrices, or **profiles**, are not tailored to a specific sequence. Rather, they are built to utilize the information in a group of related sequences, and provide representations of protein families and domains. These representations are capable of detecting subtle similarities between distantly related proteins. Without going into detail, profiles are usually obtained by applying algorithms for multiple alignment (i.e., a combined alignment of several proteins) to align a group of related sequences. The frequency of each amino acid at each position along the multiple alignment is then calculated. These counts are normalized and transformed to probabilities, so that a probability distribution over amino acids is associated with each position. Finally, the scoring matrix is defined based on these probability distributions as well as on the similarities of pairs of amino acids (taken from a standard scoring matrix). For example, the score for aligning the amino acid a at position i of the profile is given by $s_i(a) = \sum_b prob(b \text{ at position } i) s(a,b)$ where $s(a,b)$ is the similarity of amino acids a and b according to some scoring matrix. For a review on algorithms for multiple alignment and profile techniques see [?, ?, ?, ?].

1.4.6 Command line parameters

The command line parameters of the search programs are generally divided into three groups. The first group is the set of parameters which specify the input and output filenames, and the database name. These are the only mandatory parameters. All other parameters are optional and are set default values otherwise. For example, the basic command line for SSEARCH, FASTA, BLAST and gapped BLAST are:

ssearch -Q query-file -O out-file database
fasta -Q query-file -O out-file database
blastp database query-file
blastpgp -i query-file -o out-file -d database

The second set of parameters affects the comparison algorithm. This set includes the scoring matrix and the gap penalties and the parameters used to control the sensitivity of the search. By altering the later, it is possible to make the program run slower and be more sensitive, or to run faster at the cost of missing more homologs. BLAST has few such parameters. Currently, it is very rare for users to alter these options from the defaults. The FASTA program has one such parameter that a user will often want to set, called *ktup*. Searches with *ktup*=1 are slower, but are more sensitive than BLAST; *ktup* =2 is faster but less effective.

Program	Parameter	Use
ssearch	-Q filename	query file
	-O filename	output file
	-E evalue	evalue threshold (only hits with evalue below this threshold are reported)
	-d number	maximal number of alignments displayed
	-H	supresses histogram of scores
fasta	-Q filename	query file
	-O filename	output file
	-E evalue	evalue threshold (only hits with evalue below this threshold are reported)
	-d number	maximal number of alignments displayed
	-H	suppresses histogram of scores
	ktup number	controls sensitivity (can be either 1 or 2 for proteins and up to 4 for DNA)
blastp	E=evalue	evalue threshold (only hits with evalue below this threshold are reported)
	V=number	maximal number of hits reported
	B=number	maximal number of alignments displayed
	H=1	display histogram of scores
blastpgp	-d database	the database searched
	-i filename	query file
	-o filename	out file
	-e evalue	evalue threshold (only hits with evalue below this threshold are reported)
	-v number	maximal number of hits reported
	-b number	maximal number of alignments displayed
	-j number	maximal number of iterations (PSI-BLAST)
	-C filename	saves a checkpoint profile in a file after each iteration (PSI-BLAST)
	-R filename	reads the initial profile from a file (PSI-BLAST)
	-h evalue	evalue threshold for inclusion in a profile (PSI-BLAST)

Table 1.4: **Parameters for sequence comparison programs.** PSI-BLAST and gapped BLAST are executed by the same program (blastpgp). The default mode is a simple gapped BLAST (i.e., the parameter j is set to 1).

Finally, there is a third set of parameters which controls the output of the program, e.g. how many results are reported, and how many alignments are displayed. The number of hits reported is often controlled by the e-value parameter (see section 1.3.2). For example, by default, the BLAST programs will report only matches with an e-value up to 10 (this parameter also affects the sensitivity of the method, in an indirect manner). The total number of matches is limited to the best 500, and detailed information with the alignment is provided for up to 100 pairs. To retrieve more matches, these numbers can be altered (see table 1.4).

1.5 Interpretation of results

Interpretation of the results of a sequence database search involves first evaluating the matches, to determine whether they are significant and therefore imply homology. The most effective way of doing so is through use of the statistical scores (the e-values). The e-values are more useful than the raw or bit scores, and they are far more powerful than percentage identity (which is best not even considered unless the identity is very high [**?**]). Fortunately, the e-values from FASTA, SSEARCH, and gapped BLAST seem to be accurate and are therefore easy to interpret [**?**, **?**].

The e-value (or expectation-value) of a match should measure the expected number of sequences in the database which would achieve a given score. Therefore, in the average database search, one expects to find ten random matches with e-value score of 10; obviously, such matches are not significant. However, lacking better matches, sequences with these scores may provide hints of function or suggest new experiments. Scores below 0.01 would occur by chance only very rarely, and are therefore likely to indicate homology, unless biased in some way. Scores of near 1e-50 are now seen frequently, and these offer extremely high confidence that the query protein is evolutionarily related to the matched target in the database.

Inferring function from the homologous matched sequences is a process still fraught with difficulty. If the score is extremely good and the alignment covers the whole of both proteins, then there is a good chance that they will share the same or a related function. However, is dangerous to place too much trust in the query having the same function as the matched protein: functions do diverge, and organismal or cellular roles may alter even when biochemical function is unchanged. Moreover, a significant fraction of functional annotations in databases are wrong [**?**], so one needs to be suspicious. There are other complexities; for example, if only a portion of the proteins align, they may share a domain which only contributes an aspect of the overall function. It is often the case that all of the highest-scoring hits align to one region of the query, and matches to other regions need to be sought much lower in the score ranking. For this reason, it is necessary to carefully consider the overlap between the query and each of the targets.

Database search methods are also limited because most homologous sequences have diverged too far to be detected by pairwise sequence comparison methods [**?**, **?**, **?**]. Thus, failure to find a significant match does not necessarily indicate that no homologs exists in the database. In such cases more sophisticated methods must be applied. For example, iterative search programs such as the profile based PSI-BLAST program [**?**] or the HMM based SAM-T98 [**?**] are advanced and sensitive search tools. However, these programs should be carefully used as they can lead to false positives by diverging from the original query sequence, and creating a profile (HMM) which represents unrelated sequences. The most powerful tools today are those that incorporate information from a group of related sequences. This strategy have led to the compilation of databases of protein families and domains. These databases have become an important tool in the analysis of newly discovered protein sequences. They usually offer a lot of biologically valuable information about domains and the domain structure of proteins, through multiple alignments and schematic representations of proteins, and can help to detect weak relationships between remote homologs. Such methods are described in the next chapters.

1.6 Conclusion

One should neither have excessive faith in the results of a database search, nor should they be blithely disregarded. The standard search programs such as FASTA, gapped BLAST and SSEARCH are well-tested and reliable indicators of sequence similarity, and their underlying principles are straightforward. These programs and their parameters have been optimized for the hundreds of thousands of runs every day. If one is careful about posing the database search experiment and interprets the results with care, sequence comparison methods can be trusted to rapidly and easily provide an incomparable wealth of biological information.