4

APPLICATIONS CHAPTER

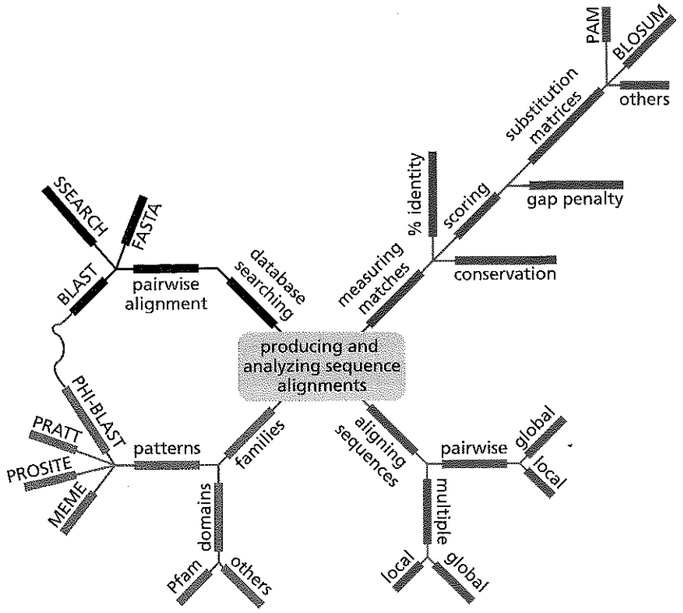
PRODUCING AND ANALYZING SEQUENCE ALIGNMENTS

When you have read Chapter 4, you should be able to:

Determine homology by sequence alignment.
Describe different uses of protein and DNA sequence alignments.
Define scoring alignments.
Make alignments between two sequences.
Make multiple alignments between many sequences.
Compare local alignment techniques for finding limited areas of similarity.
Explain global alignment techniques for matching whole sequences.
Search databases for homologous sequences.
Look for patterns and motifs in a protein sequence.
Use patterns and motifs to locate proteins of similar function.

The revolution in genetic analysis that began with recombinant DNA technology and the invention of DNA sequencing techniques in the 1970s has, 30 years later, filled vast databases with nucleotide and protein sequences from a wide variety of organisms. Genomes that have now been completely sequenced include human, mouse, chimpanzee, the fruit fly *Drosophila*, the nematode *Caenorhabditis*, and the yeast *Saccharomyces*, as well as numerous bacteria, archaea, and viruses. Although entries for nucleotide and protein sequences in databases such as GenBank, dbEST, and UniProt KB now number many millions, nothing is known about the structure or function of the proteins specified by many of them. Converting this sequence information into useful biological knowledge is now the main challenge.

To find out more about a newly determined sequence, it is subjected to the process of sequence analysis. There are many aspects to this, depending on the source of the sequence and what you ultimately want to find out about it. In this chapter, we will focus on one of the key stages in most sequence analyses: the alignment of different sequences to detect homology and the comparison of a novel sequence with those in the databases to see whether there is any similarity between them. The practical use of techniques and programs for general alignment, database searching, and pattern searching will be described in this chapter, with the main focus on the alignment and analysis of protein sequences. The theory underlying programs for pairwise alignment is described in Chapter 5 and that dealing with multiple alignments in Chapter 6, for both nucleic acid and protein sequences. Techniques and programs for detecting genes and other sequence features in genomic DNA are dealt with in Chapters 9 and 10.



Mind Map 4.1
A mind map of the four major sections relating to sequence analysis and alignment: aligning sequences, searching through databases, measuring how well sequences match, and looking for families of proteins.

The identification of similar sequences has a multitude of applications. For raw, uncharacterized genomic DNA sequences, comparison with sequences in a database can often tell you whether the sequence is likely to contain, or be part of, a protein-coding gene. The similarity search may retrieve a known gene or family of genes with a strong similarity to the new sequence. This will provide the first clues to the type of protein the new gene encodes and its possible function. Similarities in sequence can also help in making predictions about a protein's structure (see Chapters 11–14). Sequences of proteins or DNAs from different organisms can also be compared in order to construct phylogenetic trees, which trace the evolutionary relationships between species or within a family of proteins (see Chapters 7 and 8).

As well as many general and specialized databases of DNA and protein sequences, the fully sequenced genomes of various organisms are now available (see Chapter 3), providing vast amounts of information for comparison. It is, however, important to remember that although many newly discovered sequences will share some or considerable similarity to sequences in the databases, there will still be many that are unique.

4.1 Principles of Sequence Alignment

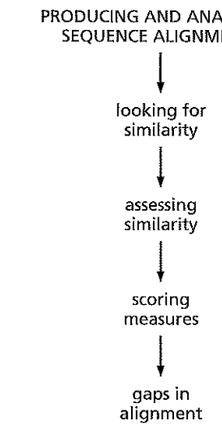
Devising ways of comparing sequences has never been straightforward, not just because of the vast amounts of information now available for searching. The difficulties arise because of the many ways DNA and protein sequences can change during evolution. Mutation and selection over millions of years can result in considerable divergence between present-day sequences derived from the same ancestral gene. Bases at originally corresponding positions, and the amino acids they encode, can change as a result of point mutation, and the sequence lengths can be quite different as a result of insertions and deletions. Even more dramatic changes may have occurred; for example, the fusion of sequences from two different genes. Gene

Box 4.1 Genes and pseudogenes

Pseudogenes are sequences in genomic DNA that have a similar sequence to known protein-coding genes but do not produce a functional protein. They are assumed to arise after gene duplication, when one of the gene copies undergoes mutation that either prevents its transcription or disrupts its protein-coding sequence. The human genome is estimated to contain up to 20,000 pseudogenes. As the pseudogene sequence is no longer under selection to retain protein function, it will generally accumulate further mutations at a higher rate than the functional gene. Despite this, many pseudogenes retain considerable sequence similarity to their active counterparts. One case has even been found in which the RNA from a transcribed pseudogene regulates the expression of the corresponding functional gene.

duplications are common in eukaryotic genomes, and in many cases mutation has disabled one copy of a gene so that it is either no longer expressed or, if transcribed, does not produce a functional protein. Such genes are called pseudogenes (see Box 4.1) and can be found in homology searches.

On superficial inspection, such changes in gene sequence and length can effectively mask any underlying sequence similarity. To reveal it, the sequences have to be aligned with each other to maximize their similarities. This crucial step in sequence comparison is the main topic of the first half of this chapter (Sections 4.1 to 4.5). Alignment methods are at the core of many of the software tools used to search the databases, and in the second half of the chapter we will describe some of these tools and how they can be used to retrieve similar sequences from the databases (Sections 4.6 to 4.10). The first steps to consider are shown in Flow Diagram 4.1.



Flow Diagram 4.1
The key concept introduced in these first four sections is that in order to assess the similarity of two sequences it is necessary to have a quantitative measure of their alignment, which includes the degree of similarity of two aligned residues as well as accounting for insertions and deletions.

Alignment is the task of locating equivalent regions of two or more sequences to maximize their similarity

As the result of mutation, even the sequences of the same protein or gene from two closely related species are rarely identical. Ideally, what we want to achieve when comparing sequences is to line them up in such a way that, when they do derive from a common ancestor, bases or amino acids derived from the same ancestral base are aligned. Without information to the contrary, this is best achieved by maximizing the similarity of aligned regions.

To illustrate the general principle, take the two hypothetical amino acid sequences THISSEQUENCE and THATSEQUENCE. If we align them so that as many identical letters as possible pair up we get

```
T H I S S E Q U E N C E
| |     | | | | | | | |
T H A T S E Q U E N C E
```

where the letters in red type are identical. As we can easily see with such short and similar sequences, this alignment clearly identifies their strong similarity to each other.

So far so good, but when sequences become more different from each other, they become more difficult to compare. How would we go about comparing the two sequences THATSEQUENCE and THISISASEQUENCE, in which a mutation has led to the insertion of the three amino acids I, S, A into one of the original sequences? Simply lining them up from the beginning loses much of the similarity we can see exists. More subtly, because of the difference in length, it also creates false matches between noncorresponding positions.

```
T H A T S E Q U E N C E
| |       |       |
T H I S I S A S E Q U E N C E
```

To get round this problem, **gaps** are introduced into one or both of the sequences so that maximum similarity is preserved.

```
T H I S I S A - S E Q U E N C E
| |         |   | | | | | | |
T H - - - - A T S E Q U E N C E
```

There is never just one possible alignment between any two sequences, and the best one is not always obvious, especially when the sequences are not very similar to each other. At the heart of sequence-comparison and database-searching methods are algorithms for testing the fit of each alignment generated, giving it a quantitative score, and filtering out the unsatisfactory ones according to preset criteria.

Alignment can reveal homology between sequences

In all methods of sequence comparison, the fundamental question is whether the similarities perceived between two sequences are due to chance, and are thus of little biological significance, or whether they are due to the derivation of the sequences from a common ancestral sequence, and are thus homologous. The terms "homology" and "similarity" are sometimes used interchangeably, but each has a distinct meaning. **Similarity** is simply a descriptive term telling you that the sequences in question show some degree of match. Homology, in contrast, has distinct evolutionary and biological implications. In the molecular biological context, it is generally defined as referring specifically to similarity in sequence or structure due to descent from a common ancestor. Homologous genes are therefore genes derived from the same ancestral gene. During their evolutionary history they will have diverged in sequence as a result of accumulating different mutations.

Because homology implies a common ancestor, it can also imply a common function or structure for two homologous proteins, which can be a useful pointer to function if one of the proteins is known only from its sequence. The operation of natural selection tends to result in the acceptance of mutations that preserve the folding and function of a protein, whereas those that destroy folding or function will be eliminated. However, similar or identical aligned residues may simply be due to relatively recent divergence of the two sequences, and so care must be taken not to overestimate their functional importance. Moreover, mutation and selection can generate proteins with new functions but relatively little change in sequence. Therefore, sequence similarity does not always imply a common function.

Conversely, there are proteins with very little sequence similarity to each other but in which a common protein fold and function are preserved. Consequently, low sequence similarity does not necessarily rule out common function or homology. Such cases require extra information, such as structural or biochemical knowledge, to demonstrate their true relationship.

Sequences can also be significantly similar to each other, and yet not be evolutionarily homologous, as a result of **convergent evolution** for similar function (see Box 4.2). In this case, identical or very similar aligned residues can be argued to have an important functional role. Convergent evolution does not, however, usually produce highly similar sequences of any great length.

All these considerations have to be taken into account when analyzing the results of a database search. An alignment of two sequences is, in effect, a hypothesis about which pairs of residues have evolved from the same ancestral residue. But an alignment in itself does not imply an evolutionary order of events, so that the two

Box 4.2 Convergent and divergent evolution

Convergent evolution is the evolutionary process in which organs, proteins, or DNA sequences that are unrelated in their evolutionary origin independently acquire the same structure or function. This usually reflects a response to similar environmental and selective pressures. Convergent evolution is contrasted with the process of **divergent evolution**, which produces different structures or sequences from a common ancestor. An example of convergent evolution for function can be seen in the wings of insects and bats. Although adapted to the same function—that of flight—insect wings and bat wings do not derive from the same ancestral structure.

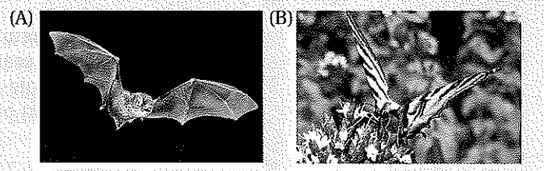


Figure B4.1
(A) Bat wings and (B) butterfly wings. (A, courtesy of Ron Austing/Science Photo Library.)

alternatives of homology and convergent evolution cannot usually be distinguished without additional information.

Sequence comparison methods have to take account of such factors as the types of mutation that occur over evolutionary time, differences in the physicochemical properties of amino acids and their role in determining protein structure and function, and the selective pressures that result in some mutations being accepted and others being eliminated. One has to consider the evolutionary processes that are responsible for sequence divergence and find a way to include the salient features in practicable schemes for testing the goodness of fit of the alignment. These must be quantitative and hence involve a score. Such **scoring schemes** can then be incorporated in algorithms designed to generate the best possible alignments. Finally, ways must be found to discriminate between fortuitously good alignments and those due to a real evolutionary relationship.

As we shall see in this chapter, all computational methods of sequence comparison take account of these factors in some way.

It is easier to detect homology when comparing protein sequences than when comparing nucleic acid sequences

For most purposes, comparisons of protein sequences show up homology more easily than comparisons of the corresponding DNA sequences. There are many reasons for this greater sensitivity. First, there are only four letters in the DNA alphabet compared to the 20 letters in the protein alphabet, and so a DNA sequence, of necessity, provides less information at each sequence position than does a protein sequence. In other words, there is a much greater probability that a match at any one position between two DNA sequences will have occurred by chance. Therefore, the degree of similarity, as judged by some appropriate quantitative score, needs to be greater between DNA sequences than between protein sequences for the alignment to be of importance. As we shall see later in this chapter, ways have been devised of determining the likelihood that one amino acid can be substituted for another during evolution, and this provides additional information beyond simple identity for scoring an alignment and determining homology.

Second, as we saw in Chapter 1, the genetic code is redundant; that is, there are two or more different codons for most amino acids (see Table 1.1). This means that identical amino acid sequences can be encoded by different nucleotide sequences. Finally, the complex three-dimensional structure of a protein, and hence its function, is determined by the amino acid sequence. The importance of maintaining

protein function usually leads to amino acid sequences changing less over evolutionary time than homologous DNA sequences. In this chapter we will concentrate for the most part on protein sequence analysis.

There are many circumstances, however, in which it is necessary to compare DNA sequences: when searching for promoters and other regulatory sequences, for example, or in whole-genome comparisons. DNA alignment is also performed, to some extent, as part of gene identification (see Chapters 9 and 10).

4.2 Scoring Alignments

The quality of an alignment is measured by giving it a quantitative score

Two homologous sequences are often so different that a correct or best alignment is not obvious by visual inspection. Furthermore, the large numbers of sequences that can be examined for similarity nowadays oblige us to use automated computational methods to judge the quality of an alignment, at least as an initial filter.

Because it is possible for two sequences to be aligned in a variety of different ways, including the insertion of gaps to improve the number of matched positions, how does one objectively determine which is the best possible alignment for any given pair of sequences? In practice, this is done by calculating a numerical value or **score** for the overall similarity of each possible alignment so that the alignments can be ranked in some order.

We can then work on the basis that alignments of related sequences will give good scores compared with alignments of randomly chosen sequences, and that the correct alignment of two related sequences will ideally be the one that gives the best score. The alignment giving the best score is referred to as the **optimal alignment**, while others with only slightly worse scores are often called **suboptimal alignments**. No one has yet devised a scoring scheme that perfectly models the evolutionary process, which is so complex that it defies any practical method of modeling. The implication of this is that the best-scoring alignment will not necessarily be the correct one, and conversely, that the correct alignment will not necessarily have the best score. However, the scoring schemes now in common use, and which are described in this chapter, are generally reliable and useful in most circumstances, as long as the results are treated with due caution and regard for biological plausibility. In principle, a scoring scheme can either measure similarity or difference, the best score being a maximum in the former case and a minimum in the latter.

The simplest way of quantifying similarity between two sequences is percentage identity

Identity describes the degree to which two or more sequences are actually identical at each position, and is simply measured by counting the number of identical bases or amino acids matched between the aligned sequences. Identity is an objective measure and can be precisely defined. **Percentage** or **percent identity** is obtained by dividing the number of identical matches by the total length of the aligned region and multiplying by 100. For the THATSEQUENCE/THISISASEQUENCE comparison, for example, the alignment given on page 74 is the best that can be achieved, and has a percentage identity score of 68.75% (11 matches over a total length of 16 positions, including the gaps).

One might think that an alignment of completely unrelated sequences would have a percentage identity of zero. However, as there are only four different nucleotides in nucleic acid sequences, and only 20 different amino acids in protein sequences,

there is always a small but finite probability for any aligned sequences that identical residues will be matched at some positions. Because there are often hundreds of residues in a protein sequence and thousands in a nucleotide sequence, unrelated sequences are expected to align matches at several positions. The length of the sequence matters: a 30% identity over a long alignment is less likely to have arisen by chance than a 30% identity over a very short alignment. Statistically rigorous methods have been devised to measure the significance of an alignment, which will be discussed later in connection with database searches and in Section 5.4.

The dot-plot gives a visual assessment of similarity based on identity

A dot matrix or **dot-plot** is one of the simplest ways to compare sequence similarity graphically, and can be used for both nucleotide and protein sequences. To compare two sequences X and Y, one sequence is written out vertically, with each residue in the sequence represented by a row, while the other is written horizontally, with each residue represented by a column. Each residue of X is compared to each residue of Y (row to column comparison) and a dot is placed where the residues are identical. In the simplest scoring system, identical residues are scored as 1 and nonidentical residues as 0, and dots are placed at all positions that contain a 1. For example, if we take the pair THISSEQUENCE/THISISASEQUENCE pair, then a simple dot-plot will look like that illustrated in Figure 4.1. The dots in red, which form diagonal lines, represent runs of matched residues. The pink dots scattered either side of the diagonals are the same residues found elsewhere in the sequence. The diagonals are interrupted by a few cells, where a gap has been inserted.

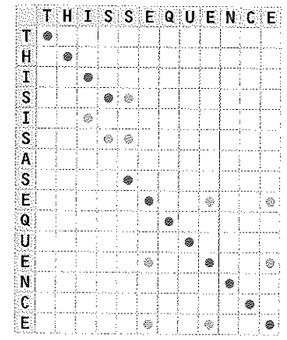
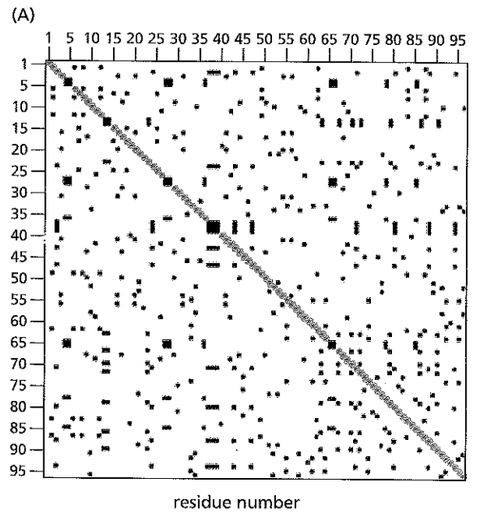
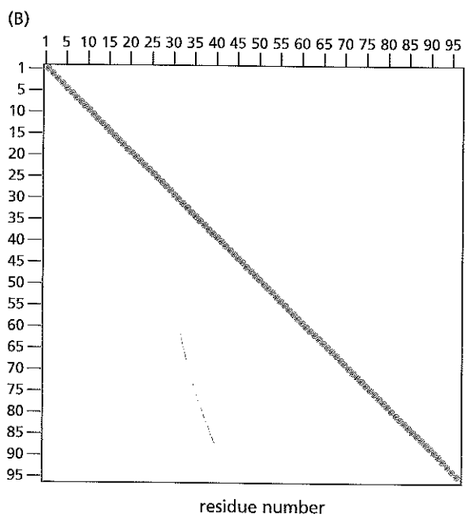


Figure 4.1
Dot-plot representations. A dot-plot matrix of the THISSEQUENCE/THISISASEQUENCE example where red dots represent identities that are due to true matching of identical residue-pairs and pink dots represent identities that are due to noise; that is, matching of random identical residue-pairs.

Dot-plots can be useful for identifying intrasequence repeats in either proteins or nucleic acids. However, dot-plots suffer from background noise. To distinguish dot-patterns arising from background noise from significant dot-patterns it is usually necessary to apply a filter. The most widely used filtering procedure uses

Figure 4.2
Two views of dot-plot representations of an SH2 sequence compared with itself. (A) Unfiltered dot-plot (window length = 1 residue). The identity between the two sequences is shown by the unbroken identity diagonal. Nevertheless, there is still background noise. (B) Dot-plot of the same sequence comparison with a window of 10 residues and a minimum identity score within each window set to 3. The background noise has all been removed, leaving only the identity diagonal.





overlapping fixed-length windows and requires that the comparison achieve some minimum identity score summed over that window before being considered; that is, only diagonals of a certain length will survive the filter. Figure 4.2 shows a dot-plot between two identical SH2 sequences (see Box 4.3).

Figure 4.2A has a window length of 1; in other words, every residue is considered individually. Although the diagonal line indicating matched identical residues is clear and unbroken, as one would expect from a comparison of two identical sequences, there is still a certain amount of background noise detracting from the result, as most types of amino acid occur more than once in the sequence. Figure 4.2B shows the same comparison with a window of 10 residues and a minimum score for each window set to 3. Only the main diagonal is now seen, representing the one-to-one matching of the identical sequences.

Most dot-plot software provides a default window length and this is sufficient for an initial analysis. But one can use the window length to greater effect by varying it depending on what one is searching for. Window length can be set, for example, to the length of an exon when comparing coding sequences, or to the size of an average secondary structure within a protein when looking for structural motifs. When searching for internal repeats, the length of the repeat can be used to cut out background noise. In addition, rather than using 0 and 1 as the scores for nonidentical and identical residues, other values can be used and the score can be varied depending on the type of residues involved.

Figure 4.3 illustrates how a dot-plot can be used to identify repeats within a sequence. It shows two dot-plot calculations on the protein BRCA2 encoded by the breast cancer susceptibility gene *BRCA2*. This protein contains eight repeats of a short sequence of around 39 amino acids, called the BRC repeat (see Box 4.4). Figure 4.3A shows an unfiltered version of a self-comparison dot-plot of a region of BRCA2 containing two BRC repeats. The background noise is so strong that it is very difficult to pick out the repeats. Figure 4.3B shows a highly filtered dot-plot of the same comparison in which a diagonal line is now visible. This is the identity diagonal, where the one-to-one alignment of the sequence with itself is highlighted. But two other runs of dots are now also visible; these represent the internal BRC repeats.

Box 4.3 The SH2 protein-interaction domain

The SH2 or Src-homology 2 domain is a small domain of about 100 residues found in many proteins involved in intracellular signaling in mammalian cells. It gets its name from the protein tyrosine kinase Src, where it was first found. It is one of numerous protein-interaction domains found in signaling proteins, which recognize and bind to particular features on other proteins to help pass the signal onward. SH2 domains bind specifically to phosphotyrosines on proteins; these are formed by the phosphorylation—the modification by covalent addition of a phosphate group—of tyrosine residues in specific peptide motifs by protein tyrosine kinases. This type of kinase is often part of, or associated with, cell-surface receptors, and is activated in response to an extracellular signal. The phosphotyrosine-binding site on SH2 domains consists of two pockets. One is conserved and binds the phosphotyrosine residue (pY); the other is more variable in sequence between different SH2 domains and binds residues located downstream from the pY, thereby conferring specificity on the protein–protein interaction. Because of its role in intracellular signaling, the SH2 domain is an important potential drug target for a number of diseases, including cancer and osteoporosis.

Figure B4.2
A ribbon representation of an SH2 domain.

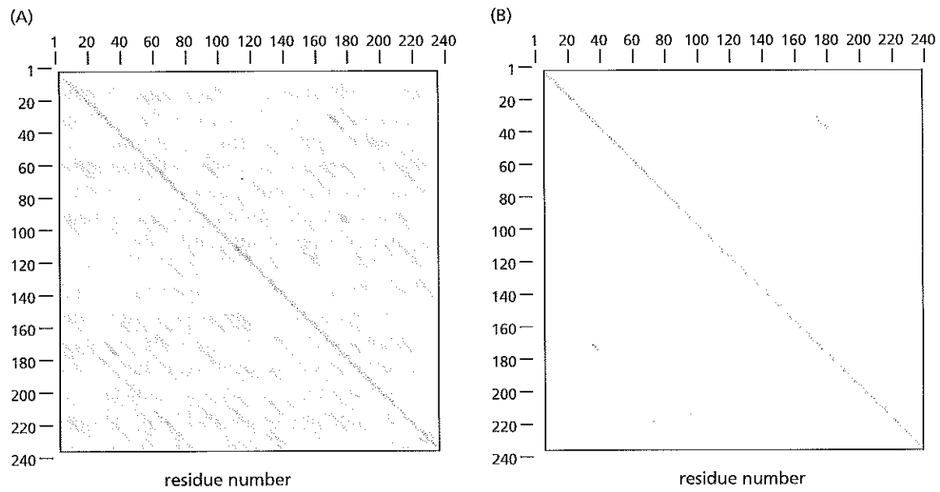


Figure 4.3
Two dot-plots involving the breast cancer susceptibility gene protein BRCA2, which contains the small BRCA2 repeat domain. (A) An unfiltered self-comparison dot-plot of part of the human BRCA2 sequence containing two BRCA2 repeats (the first and second BRCA repeat in the sequence). The background noise is so strong that it is very difficult to pick out the repeats. (B) The same dot-plot with a window length of 30 and a minimum score of 5. In addition to the identity diagonal there are two other clear diagonal runs of dots that represent the two internal BRCA2 repeats.

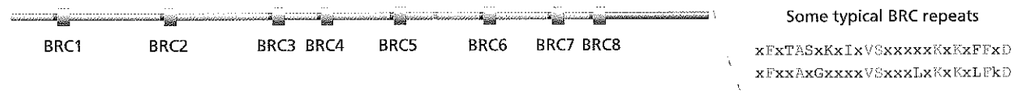
Genuine matches do not have to be identical

Although it is the simplest alignment score to obtain, and can be very useful as a quick test of the quality of an alignment, percentage identity is a relatively crude measure and does not give a complete picture of the degree of similarity of two sequences to each other, especially in regard to protein sequences. For example,

Box 4.4 The breast cancer susceptibility genes *BRCA1* and *BRCA2*

Two genes that confer increased susceptibility to breast cancer have been identified: the *BRCA1* gene on chromosome 17 in 1994 and the *BRCA2* gene on chromosome 13 in 1995. Women with a mutation in either *BRCA1* or *BRCA2* are at increased risk of developing breast, ovarian, and some other cancers by a given age than those without a mutation. The normal role of the BRCA1 and BRCA2 proteins, which are not structurally related, is to associate with the protein RAD51, a protein essential for the repair of double-strand breaks in DNA. Mutations in *BRCA1* or *BRCA2* can thus partly disable this repair mechanism, leading to more errors in DNA repair than usual, an increased mutation rate, and, ultimately, a greater risk of tumorigenesis. The BRCA2 protein has a number of repeats of 39 amino acids, the BRC repeats. Eight BRC repeats in BRCA2 are defined in the Pfam database, of which six are highly conserved and are involved in binding RAD51.

Figure B4.3
BRC repeats of the BRCA2 protein as defined by the Pfam database.



simply scoring identical matches as 1 and mismatches as 0 ignores the fact that the type of amino acid involved is highly significant. In particular, certain nonidentical amino acids are very likely to be present in the same functional position in two related sequences, and thus are likely to represent genuine matches. This is chiefly because certain amino acids resemble each other closely in their physical and/or chemical properties (see Figure 2.3) and can thus substitute functionally for each other. Mutational changes that replace one amino acid with another having similar physicochemical properties are therefore more likely to have been accepted during evolution. So pairs of amino acids with similar properties will often represent genuine matches rather than matches occurring randomly.

The simplest way of taking this into account is simply to count such similar pairs of amino acids as matches, and to refer to the score as **percent similarity.** In the now familiar example sequences below, red is used to indicate residues that are similar but not identical. Here the sequences have been realigned to take into account similarity as well as identity. Isoleucine (I) and alanine (A) are similar as they are both hydrophobic, whereas serine (S) and threonine (T) both have an -OH group in their side chain and are polar.

```
T H I S I S A S E Q U E N C E
| |       | | | | | | | |
T H A T - - - S E Q U E N C E
```

Not all similar amino acid pairs are equally likely to occur, however, and more sophisticated measures of assessing similarity are more commonly used. In these, each aligned pair of amino acids is given a numerical score based on the probability of the relevant change occurring during evolution. In such scoring schemes, pairs of identical amino acids are assigned the highest score; then, pairs of amino acids with similar properties (such as isoleucine and leucine) score more highly than those with quite different properties (such as isoleucine and lysine), which are rarely found in corresponding positions in known homologous protein sequences.

Other properties of amino acids can be added into scoring schemes for greater accuracy. For example, the type of residue involved should be taken into account. Many cysteine residues are highly conserved because of their important structural role in forming disulfide bonds, and tryptophan residues are usually key components of the hydrophobic cores of proteins. To mimic this, the scores for matching residues can be varied according to the type, with pairs of cysteines and tryptophans, for example, being assigned particularly high values. When aligned amino acid pairs are given varying scores in this way, summing the values at all positions gives the **overall alignment score**.

Most currently used alignment-scoring schemes for protein sequences measure the relative likelihood of an evolutionary relationship compared to chance. The theory behind such assessments is explained further in Section 5.1. With such schemes, the higher the alignment score, the more likely it is that the aligned sequences are homologous.

Ideally, it would be possible to decide unequivocally whether two sequences are homologous by simply looking at their best alignment score. This turns out to be more difficult than might be imagined, as the significance of the score will depend on the length of the sequences, their amino acid composition, and the number of sequences being compared—for example when searching a large database. We shall return to this topic later in the chapter.

The concept of similarity, rather than identity, has little relevance to comparisons of nucleotide sequences, especially in generating alignments. Purines tend to mutate to purines (A ↔ G) and pyrimidines to pyrimidines (C ↔ T). This information can be used to help construct phylogenetic trees (see Sections 7.2 and 8.1), but is not

helpful for sequence alignment. In the case of an alignment of nucleotide sequences, the scoring scheme is almost always very simple. For example, in the database-searching program FASTA, which is discussed later and in Section 5.3, a score of +5 for matching bases and –4 for mismatches has been found to be effective for DNA database searches. This simpler scoring scheme is sufficiently sensitive to be useful in part because of the much higher percentage identity expected if there is significant homology between the sequences, since there are only four types of bases as compared to 20 amino acids.

There is a minimum percentage identity that can be accepted as significant

What is the minimum percentage identity that can reasonably be accepted as significant? Burkhard Rost analyzed more than a million alignments of pairs of protein sequences for which structural information was available to find a cut-off for the level of sequence identity below which alignment becomes unreliable as a measure of homology. He found that 90% of sequence pairs with identity at or greater than 30% over their whole length were pairs of structurally similar proteins. Given both sequence and structural similarity, one can usually be confident that two sequences are homologous, so 30% sequence identity is generally taken as the threshold for an initial presumption of homology. Below about 25% sequence identity, however, Rost found that only 10% of the aligned pairs represented structural similarity. The region between 30% and 20% sequence identity has been called the **twilight zone,** where homology may exist but cannot be reliably assumed in the absence of other evidence. Even lower sequence identity (<20%) is referred to as the **midnight zone.**

There are many different ways of scoring an alignment

The function of an alignment score is to provide a single numerical value for the degree of similarity or difference between two sequences. Most current applications measure similarity, and in this case the highest scores are best. A few applications, particularly those used for generating phylogenetic trees (see Chapters 7 and 8), use a score related to sequence difference, usually known as a **distance**, in which case the most closely related sequences give alignments with the lowest scores. The measure of difference between two homologous sequences from different species is sometimes called the **genetic** or **evolutionary distance.**

There is no a priori reason why residue pair alignment scores cannot be negative, for example to represent especially unlikely alignments. In fact, some of the popular techniques require scores that can be negative, and most commonly used schemes have both positive and negative scores for pairs of residues.

Scoring schemes have to represent two salient features of an alignment. On the one hand, they must reflect the degree of similarity of each pair of residues; that is, the likelihood that both are derived from the same residue in the presumed common ancestral sequence. On the other hand, they must assess the validity of inserted gaps. Ways of quantifying these two features will be described separately here, although in fact they are used together to arrive at the final score. We will first go through the ways of assessing the degree of similarity for pairs of aligned residues.

4.3 Substitution Matrices

Substitution matrices are used to assign individual scores to aligned sequence positions

For alignments of protein sequences, the score is assigned to each aligned pair of amino acids is generally determined by reference to a **substitution matrix**, which

defines values for all possible pairs of residues. Various types of substitution matrices have been used over the years. Some were based on theoretical considerations, such as the number of mutations that are needed to convert one amino acid into another, or similarities in physicochemical properties. The most successful, however, use actual evidence of what has happened during evolution, and are based on analysis of alignments of numerous homologs of well-studied proteins from many different species.

The choice of which substitution matrix to use is not trivial because there is no one correct scoring scheme for all circumstances. There is a wide range of variation in the similarity of sequences, from almost complete identity to a few percent. On one occasion we may need to align and score closely related sequences, whereas on another we may want to identify very distant relationships reliably. In the first case, the scoring scheme should be strongly biased toward giving high values to perfect matches and highly conserved substitutions. In the second case, a wider range of substitutions should be treated favorably.

Most scoring schemes for amino acid sequences use as reference a 20 × 20 substitution matrix, representing the 20 amino acids found in proteins. Each cell of the matrix is occupied by a score representing the likelihood that that particular pair of amino acids will occupy the same position through true homology, compared to the likelihood of their occurring as a random match. The most important scoring matrices will be described below, with general guidance as to which one to use when. A more comprehensive description of the theory underlying the scoring schemes discussed here is given in Section 5.1.

When an alignment is made, each aligned amino acid pair is given a score from the substitution matrix. These scores are then summed to give the overall score (*S*) of the alignment. For example, using the BLOSUM-62 matrix (see Figure 4.4A) we would score our example alignment as follows (in this case "U" represents an unknown residue; that is, a residue that could not be identified by sequencing techniques and is thus not given a score).

Seq1:	T	H	I	S	S	E	Q	U	E	N	C	E
Seq2:	T	H	A	T	S	E	Q	U	E	N	C	E
Score:	5	8	–1	1	4	5	5	0	5	6	9	5

Therefore the overall score *S* for this alignment equals 52. The BLOSUM matrices are described in more detail below.

The PAM substitution matrices use substitution frequencies derived from sets of closely related protein sequences

A commonly used set of substitution matrices is based on the observed amino acid substitution frequencies in alignments of homologous protein sequences. These matrices were first developed by Margaret Dayhoff and her co-workers in the 1960s and 1970s, and have been found to be superior to substitution schemes that use only the physicochemical similarities of amino acids, as they use real data to model the evolutionary process. The sequences used to generate these matrices were all very similar, allowing the alignment to be made with confidence. In addition, the high similarity meant that there was a high probability that amino acid differences at an alignment position were due to just a single mutation event, over a short period of time, since it is unlikely that more than one mutation would occur at the same site. A phylogenetic tree (see Section 7.1) was constructed for the protein sequences, from which the individual mutations that had occurred could be deduced. From this tree, the researchers calculated the ratio of the number of

changes undergone by each type of amino acid to the total number of occurrences of that amino acid in the sequence set.

From these ratios it was possible to calculate the probabilities that any one amino acid would mutate into any other over a given period of evolutionary time. The final matrix of substitution scores is a logarithmic matrix of the mutation probabilities. Probabilities are converted to logarithms so that the final alignment score can be calculated by summation of the individual scores from aligned pairs of amino acids, rather than by multiplication of probabilities.

(A)

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	−1	4																		
T	−1	1	5																	
P	−3	−1	−1	7																
A	0	1	0	−1	4															
G	−3	0	−2	−2	0	6														
N	−3	1	0	−2	−2	0	6													
D	−3	0	−1	−1	−2	−1	1	6												
E	−4	0	−1	−1	−1	−2	0	2	5											
Q	−3	0	−1	−1	−1	−2	0	0	2	5										
H	−3	−1	−2	−2	−2	−2	1	−1	0	0	8									
R	−3	−1	−1	−2	−1	−2	0	−2	0	1	0	5								
K	−3	0	−1	−1	−1	−2	0	−1	1	1	−1	2	5							
M	−1	−1	−1	−2	−1	−3	−2	−3	−2	0	−2	−1	−1	5						
I	−1	−2	−1	−3	−1	−4	−3	−3	−3	−3	−3	−3	−3	1	4					
L	−1	−2	−1	−3	−1	−4	−3	−4	−3	−2	−3	−2	−2	2	2	4				
V	−1	−2	0	−2	0	−3	−3	−3	−2	−2	−3	3	2	1	3	1	4			
F	−2	−2	−2	−4	−2	−3	−3	−3	−3	−3	−1	−3	−3	0	0	0	−1	6		
Y	−2	−2	−2	−3	−2	−3	−2	−3	−2	−1	2	−2	−2	−1	−1	−1	−1	3	7	
W	−2	−3	−2	−4	−3	−2	−4	−4	−3	−2	−2	−3	−3	−1	−3	−2	−3	1	2	11

(B)

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	−1	3																		
T	−3	2	4																	
P	−3	1	−1	6																
A	−3	1	1	1	3															
G	−5	1	−1	−2	1	5														
N	−5	1	0	−2	0	0	4													
D	−7	0	−1	−2	0	0	2	5												
E	−7	−1	−2	−1	0	−1	1	3	5											
Q	−7	−2	−2	0	−1	−3	0	1	2	6										
H	−4	−2	−3	−1	−3	−4	2	0	−1	3	7									
R	−4	−1	−2	−1	−3	−4	−1	−3	−3	1	1	6								
K	−7	−1	−1	−2	−2	−3	1	−1	−1	0	−2	2	5							
M	−6	−2	−1	−3	−2	−4	−3	−4	−4	−1	−4	−1	0	8						
I	−3	−2	0	−3	−1	−4	−2	−3	−3	−3	−4	−2	−2	1	6					
L	−7	−4	−3	−3	−3	−5	−4	−5	−4	−2	−3	−4	−4	3	1	5				
V	−2	−2	0	−2	0	−2	−3	−3	−3	−3	−3	−3	−4	1	3	1	5			
F	−6	−3	−4	−5	−4	−5	−4	−7	−6	−6	−2	−4	−6	−1	0	0	−3	8		
Y	−1	−3	−3	−6	−4	−6	−2	−5	−4	−5	−1	−6	−6	−4	−2	−3	−3	4	8	
W	−8	−2	−6	−7	−7	−8	−5	−8	−8	−6	−5	1	−5	−7	−7	−5	−8	−1	−1	12

Figure 4.4
Amino acid substitution scoring matrices. (A) The BLOSUM-62 matrix and (B) the PAM120 substitution matrix. Each cell represents the score given to a residue paired with another residue (row × column). The values are given in half-bits, as discussed in Section 5.1. The colored shading indicates different physicochemical properties of the residues (see Figure 2.3): small and polar, yellow; small and nonpolar, white; polar or acidic, red; basic, blue; large and hydrophobic, green; aromatic, orange.

There is more than one such matrix and each matrix corresponds to a particular quantity of **accepted mutations** — mutations that have been retained in the sequence. This quantity is measured in PAM units, where PAM stands for Point Accepted Mutations (accepted point mutations per 100 residues), and these matrices are generally called **PAM matrices**. One of the more frequently used substitution matrices corresponds to 250 PAM, which means that 250 mutations have been fixed on average per 100 residues; that is, many residues have been subject to more than one mutation. The matrix itself is called PAM250. This amount of change is near the limit of detection of distant relationships. Other matrices, such as PAM120, correspond to a smaller amount of mutation (see Figure 4.4B)

The currently used PAM matrices, also known as Dayhoff mutation data matrices (MDMs), were originally created in 1978. More recent matrices have also been constructed using newer and larger data sets. The PET91 matrix, for example, represents a new generation of Dayhoff-type matrices.

The BLOSUM substitution matrices use mutation data from highly conserved local regions of sequence

The **BLOSUM matrix** is another very commonly used amino acid substitution matrix that depends on data from actual substitutions. It was derived much more recently than the Dayhoff matrices, in the early 1990s, using local multiple alignments rather than global alignments. First, a large set of aligned highly conserved short regions was generated from analysis of the protein-sequence database SWISS-PROT. The sequences were then clustered into groups according to similarity, so that sequences were grouped together if they exceeded a specified threshold for percentage identity. Substitution frequencies for all possible pairs of amino acids were then calculated between the clustered groups (without the construction of phylogenetic trees) and used to compute BLOSUM (BLOck SUbstitution Matrix) scores. Various BLOSUM matrices are obtained by varying the percentage cut-off for clustering into similarity groups. For example, the commonly used BLOSUM-62 matrix was derived using a threshold of 62% identity (see Figure 4.4).

The choice of substitution matrix depends on the problem to be solved

With many scoring matrices available, it is hard to know which one to use. Within a group of matrices such as the PAM or,BLOSUM series, different ones, for example PAM250 versus PAM120 or BLOSUM-50 versus BLOSUM-80, are more suitable for different types of problem. The PAM matrix number indicates evolutionary distance whereas the BLOSUM matrix number refers to percentage identity. When aligning sequences that are anticipated to be very distantly related, matrices such as PAM250 and BLOSUM-50 may therefore be preferable. PAM120 and BLOSUM-80 may perform better for more closely related sequences.

Some matrices have been derived using additional information; the STR matrix, for example, includes information from known protein structures. Because protein structure is more conserved than sequence, more distantly related proteins can be compared using such methods, even when sequence alignment alone would not pick up any significant relationship.

Some scoring matrices have been designed to work well in special situations. For example, the matrices SLIM (ScoreMatrix Leading to Intra-Membrane) and PHAT (Predicted Hydrophobic And Transmembrane matrix) are especially designed for membrane proteins, where the characteristic amino acid composition and the selective forces for acceptable mutations are different from those for soluble proteins. In 2006, there were 94 matrices collected in a database list called AAINDEX and searchable at GenomeNet.

As well as the degree of evolutionary distance, the length of the sequences to be aligned must be taken into account when choosing a suitable matrix. This is especially relevant when searching databases against a query sequence, as the length of the sequence is taken into account when assessing the significance of the score: the shorter the sequence, the higher the score needs to be in order to be judged significant. Short sequences need to use matrices designed for short evolutionary time scales, such as PAM40 or BLOSUM-80. Longer sequences of 100 residues or more can use matrices intended for use with longer evolutionary time scales (such as PAM250 and BLOSUM-50). The reasons why the significance of a score depends on the length of the sequences to be aligned are discussed in more detail in Section 5.4.

4.4 Inserting Gaps

Gaps inserted in a sequence to maximize similarity with another require a scoring penalty

Homologous sequences are often of different lengths as the result of insertions and deletions (**indels**) that have occurred in the sequences as they diverged from the ancestral sequence. Their alignment is generally dealt with by inserting **gaps** in the sequences to achieve as correct a match as possible. To signify that an insertion or deletion has occurred, a letter or stretch of letters in one sequence is paired up with blank spaces (usually indicated by hyphens) inserted into the other sequence to achieve a better match.

Gaps must be introduced judiciously: forcing two sequences to match up simply by inserting large numbers of gaps will not reflect reality and will produce a meaningless alignment. To place limits on the introduction of gaps, alignment programs use a **gap penalty**: each time a gap is introduced, the penalty is subtracted from the score, decreasing the overall score of the alignment. Structural analysis has shown that fewer insertions and deletions occur in sequences of structural importance, and that insertions tend to be several residues long rather than just a single residue long. This information can be included in the scoring scheme by placing a smaller penalty on lengthening an existing gap (**gap extension penalty**) than on introducing a new gap, thus penalizing single-residue gaps relatively more. The best alignment is thus the one that returns the maximum score for the smallest number of introduced gaps.

Gap penalties can usually be varied in an alignment program, so the user has to decide what gap penalty to use. It should be kept in mind that the insertion of a gap must improve the quality of the alignment and therefore the maximum-match value. If a gap penalty is set high, then fewer gaps will be inserted into the alignment, as their inclusion will radically decrease the maximum-match value. If a low gap penalty is chosen, then more and larger gaps will be inserted. Therefore, if you are searching for sequences that are a strict match for your query sequence, the gap penalty should be set high. This will often retrieve a region, or regions, of very closely related sequence. If you are searching for similarity between distantly related sequences, the gap penalty should be set low. Note that suitable gap-penalty values may be different with different substitution matrices. It is advisable to start, when possible, with a combination of matrix and gap penalties that have been reported to give optimal performance.

In some alignment programs, a gap score depends on the type of residue with which the gap is aligned. Some types of residues are more likely to be conserved than others because their side chains tend to be more important in determining structure or function. An example is tryptophan, and so a gap aligned with a tryptophan will exact a larger gap penalty than a gap aligned with a glycine, for example.

(A)

```
Bovine PI-3Kinase p110a        LNWENPDIMSELLFQNNEIIFKNGDDLRQDMLTLQIIRIMENIWQNQGLDLRMLPYGCLSIGDCVGLIEVVRNSHTIMQIQCKGGLKGAL
cAMP-dependent protein kinase --WENPAQNTAHLDQFERIKTLGTGSFGRVMLVKHMETGNHYAMKILDKQKVVKLKQIEHTLNEKRILQAVNFPFLVKLEFSFKDNSNLY

Bovine PI-3Kinase p110a        QFNSHTLHQWLKDKNKGEIYDAAIDLFTRSCAGYCVATFILGIGDRHNSNIMVKDDGQLFHIDFGHFLDHKKKKFGYKRERVPFVLTQDF
cAMP-dependent protein kinase MVMEYVPGGEMFSHLRRIGRFSEPHARFYAAQIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCGTPEYLAP

Bovine PI-3Kinase p110a        LIVISKGAQECTKTREFERFQEMCYKAYLAIRQHANLFINLFSMMLGSGMPELQSFDDIAYIRKTLALDKTEQEALEYFMKQMNDAHHGG
cAMP-dependent protein kinase EIILSKGYNKAVDWWALGVLIYEMAAGYPPFFADQPIQIYEKIVSGKVRFPSHFSSDLKDLLRNLLQVDLTKRFGNLKNGVNDIKNHKWF

Bovine PI-3Kinase p110a        WTTKMDWIFHTIKQHALN-----------------------------------
cAMP-dependent protein kinase ATTDWIAIYQRKVEAPFIPKFKGPGDTSNFDDYEEEEIRVXINEKCGKEFSEF
```

(B)

```
Bovine PI-3Kinase p110a        LNWENPDIMSELLFQNNEIIFKNGDDLRQDMLTLQIIRIMENIWQNQGLDLRMLPYGCLSIGDCVGLIEVVRNSHTIMQIQCKGGLKGAL
cAMP-dependent protein kinase ?-WENPAQNTAHLDQFERIKTLGTGSFGRVMLVKHM--ETGNHYAMKILDKQKV-VKLKQIEHTLNEKRILQAVNFPFLVKLEFSFKDN-

Bovine PI-3Kinase p110a        QFNSHTLHQWLKDKNKGEIYDAAIDLFTRSCAGYCVATFILGIGDRHNSNIMVKD-DGQLFHIDFGHFLDHKKKKFGYKRERVPFVL--I
cAMP-dependent protein kinase -SNLYMVMEYVPGGEMFSHLRR-IGRFSEPHARFYAAQIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCGT

Bovine PI-3Kinase p110a        QDFL---IVISKGAQECTKTREFERF-QEMC--YKAYLAIRQHANLFINLFSMMLGSGMPELQSFDDIAYIRKTLALDKTEQEALEYFMK
cAMP-dependent protein kinase PEYLAPEIILSKGYNKAVDWWALGVLIYEMAAGYPPFFA-DQPIQIYEKIVSGKVRF--PSHFSSDLKDLLRNLLQVDLTKR--FGNLKN

Bovine PI-3Kinase p110a        QMNDAHHGGWTTKMDWI----------------------FHTIKQHAL----N----------
cAMP-dependent protein kinase GVNDIKNHKWFATTDWIAIYQRKVEAPFIPKFKGPGDTSNFDDYEEEEIRVXINEKCGKEFSEF
```

Figure 4.5
Pairwise alignments of the PI3-kinase p110α and a cAMP-dependent protein kinase. Note that the protein kinase sequence is considerably longer than the p110α sequence. (A) An alignment where the gap penalty has been set very high. Gaps have therefore only been inserted at the beginning and end of the sequences. The percentage identity of this alignment is 10%. (B) An alignment with a very low gap penalty. Many more gaps have been inserted to maximize the number of matched residues. Especially apparent is the lone matched pair of asparagine (N) residues in the carboxy-terminal region. The percentage identity of this alignment is 18%. Green shading, identical amino acids.

It is best to start with the default values given by the program you are using and then raise or lower the penalty to obtain a desired alignment. However, the number of gaps should always be kept to the minimum possible. Figure 4.5 shows two pairwise alignments of a **phosphatidylinositol-3-OH** kinase sequence (from bovine PI3-kinase p110α) and a **protein kinase** sequence from a cyclic AMP (cAMP)-dependent protein kinase (see Box 4.5), which have only limited similarity to each other.

In the first alignment (see Figure 4.5A) the gap penalty was set very high; therefore the program inserts as few gaps as possible. Any inserted gaps are found at the ends of the sequence, as often, unless there is an obvious relationship between the terminal amino acids, end gaps are not penalized. In the second alignment (see Figure 4.5B) the gap penalty was set very low; the effect is that many more gaps are inserted and the number of matched amino acids is increased (identities are shown in green). Although there are more matched residues in the alignment with low gap penalties, this does not necessarily mean that it is more accurate. In sequences that share such low homology as these, expert knowledge, such as the location of active-site residues, has to be used to decide if the alignment is accurate.

Dynamic programming algorithms can determine the optimal introduction of gaps

In practice, it is nearly always necessary to insert gaps into sequences when aligning them. The most obvious way of finding the best alignment with gaps would be to generate all possible gapped alignments, find the score for each, and select the highest-scoring alignment. This would be enormously time consuming,

Box 4.5 Protein kinases and phospholipid kinases

Phosphorylation is one of the commonest ways of rapidly altering a protein's activity. The enzymes that phosphorylate proteins are known as protein kinases and add phosphate groups to specific amino acid residues in the protein. Most, such as the cAMP-dependent protein kinases, phosphorylate serine or threonine residues, whereas others phosphorylate tyrosine residues. The effect of protein phosphorylation can be reversed by phosphoprotein phosphatases, which specifically remove the phosphate group. Because of their important roles as regulators of cellular activity and behavior, the activity of protein kinases is, in general, tightly controlled. The cAMP-dependent protein kinases, for example, are activated by binding the intracellular second messenger cAMP, which is specifically generated in response to a variety of extracellular signals acting at cell-surface receptors.

The phosphatidylinositol-3-OH kinases (PI3-kinases) phosphorylate inositol phospholipids in the cytoplasmic surface of the cell membrane, adding a phosphate group to position 3 on the inositol ring. Other members of this family, the PI4-kinases, phosphorylate the inositol ring on position 4. The phosphorylated lipids then specifically bind and activate other proteins, such as protein kinases, to initiate intracellular signal transduction cascades. PI3-kinases are involved in initiating the pathway by which the hormone insulin controls carbohydrate metabolism. PI3-kinases and protein kinases have very little sequence similarity to each other except in the enzymatic kinase domain.

however. For example, approximately 10^{75} alignments would need to be generated for a sequence of only 100 residues. It only became practicable to incorporate gaps into an alignment with the development of **dynamic programming algorithms**. These avoid unnecessary exploration of the bulk of alignments that can be shown to be nonoptimal. The name "dynamic programming" reflects the fact that the precise behavior of the algorithm is established only when it runs (in other words, dynamically) because it depends on the sequences being aligned.

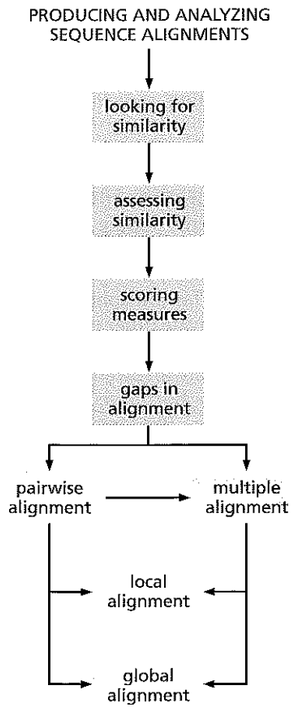
The first algorithm to use dynamic programming for sequence comparison was that of S. B. Needleman and C. D. Wunsch, published in 1970. Their technique is still the core of many present-day alignment and sequence-searching methods. In their method, gaps, regardless of length, have an associated penalty score; newer methods use more complicated gap penalties. The actual values of the gap scores can be varied depending on the type of scoring matrix being used. One rule always followed is that gaps can never be aligned with each other.

The basic concept of a Needleman–Wunsch-type algorithm is that comparisons are made on the basis of all possible pairs of amino acids that could be made between the two sequences. All possible pairs are represented as a two-dimensional matrix, in which one of the sequences to be aligned runs down the vertical axis and the other along the horizontal axis. All possible comparisons between any number of pairs are given by pathways through the array, each of which can be scored. The principles and method of the algorithm are dealt with in detail in Section 5.2. The general idea is to grow the alignment from the amino or carboxy terminus, at each step rejecting all possible alignments except that with the best score.

4.5 Types of Alignment

Different kinds of alignments are useful in different circumstances

The general principles outlined in the previous sections can be used to make different types of alignment (see Flow Diagram 4.2). Two closely related homologous sequences will generally be of approximately the same length, so that their alignment



Flow Diagram 4.2
The key concept introduced in this section is that there are several different types of sequence alignment, one of which will be the most appropriate for a particular problem.

will cover the full range of each sequence. This is referred to as a **global alignment**, and is generally the appropriate one to use when you want to compare or find closely related sequences that are similar over their whole length.

On the other hand, there are many cases where only parts of sequences are related. A simple example is the amino acid sequences of two proteins each consisting of two domains, with only one domain common to both proteins and the other domains completely unrelated. In this case, the only meaningful alignment will be a **local alignment** of the shared domain. Looking only at global alignments may not reveal the limited but important similarity between the sequences. This is particularly the case for comparisons between multidomain proteins, such as PI3-kinases, which consist of a number of small protein domains strung together (see Figure 4.6). Local alignment programs are therefore useful for detecting shared domains in such proteins.

When searching through a sequence database with a query sequence from an unknown protein, local alignment is a very useful tool to use initially. Once sequences with regions of high similarity are found using local alignment, global alignment can be used to align the rest of the sequence that is not so similar. Local alignment is also a good tool for identifying particular functional sites from which sequence patterns and motifs can be derived.

A widely used local alignment algorithm is the Smith–Waterman algorithm, which is a modification of the Needleman–Wunsch algorithm. Instead of looking at each sequence in its entirety, which is what the Needleman–Wunsch algorithm does, the Smith–Waterman method compares segments of all possible lengths and chooses the segment that optimizes the similarity measure. The scoring matrix used must include both positive and negative scores, and only alignments with a positive total score are considered. Therefore, if on extending the alignment at a particular step none of the possible alignments has a positive score, all previous alignments are

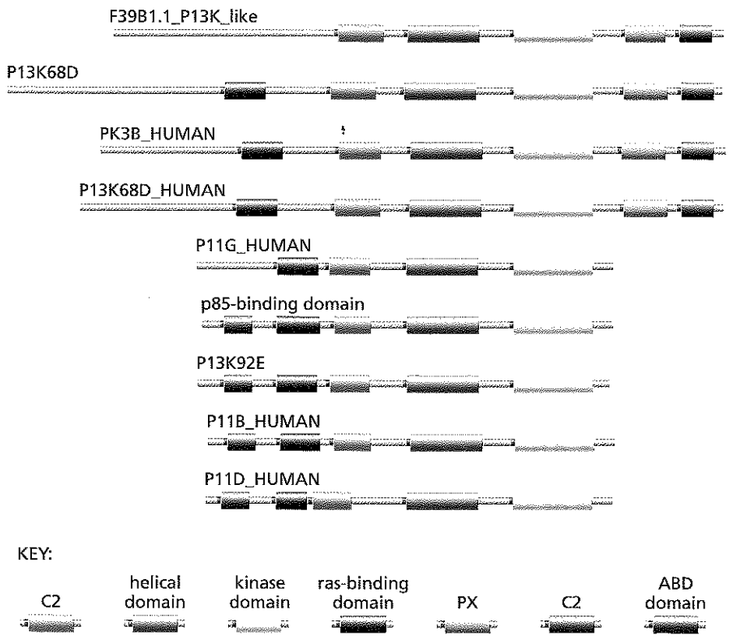


Figure 4.6
PI3-kinase is a multidomain protein. One possible output from a search of the Pfam database with the p100α PI3-kinase catalytic domain (yellow bar) is shown here. The figure also shows the complete domain structure of the protein family comprising the PI3-kinases and the related PI4-kinases, which catalyze phosphorylation of position 4 of the inositol ring of inositol phospholipids. The other domains and their arrangement are represented by the other colored bars.

rejected, and new ones are considered starting from that point. This makes the calculation sensitive to the precise match and mismatch scores and gap penalties. Section 5.2 describes the algorithm in detail.

Figure 4.7 shows an example of local versus global alignment of the complete protein sequences of the bovine PI3-kinase p110α and the cAMP-dependent protein kinase shown in Figure 4.5, using the Web-based programs ALIGN (global) and LALIGN (local). Although these proteins share structural homology within the core kinase catalytic domain, there is very little sequence homology. Figure 4.7A shows that local alignment of the catalytic domains has identified one important conserved region, out of five regions that were aligned. This region is involved in catalysis and also contains the three-residue motif DFG, which is conserved between many kinases. Figure 4.7B shows that, in this case, a global alignment fails to identify this region. The percentage sequence identity for these two sequences is very low (17.8%), well into the midnight zone of sequence alignment.

For both global and local alignments, methods exist for making **pairwise alignments**, that is, the alignment of just two sequences, and for making **multiple alignments**, in which more than two sequences are aligned with each other. In this part of the chapter, we have mainly used examples of pairwise alignments to illustrate the general principles of alignment scoring and quality assessment. Multiple alignment introduces yet another dimension to the computational problems of alignment. The theory is dealt with in detail in Chapter 6, but a few general points are described here.

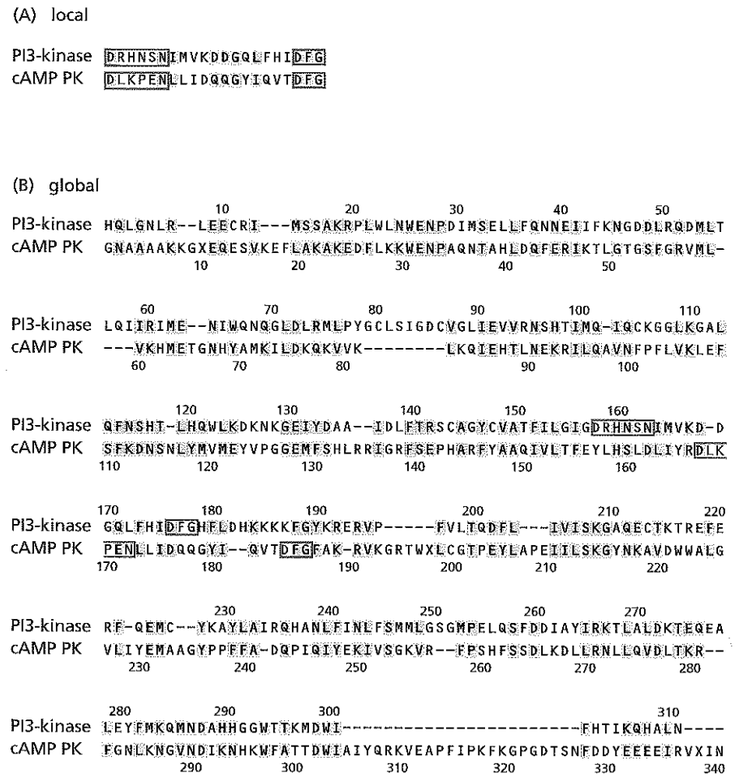


Figure 4.7
Local and global alignments. The complete sequences of PI3-kinase p110α and the cAMP-dependent protein kinase (cAMP PK) shown in Figure 4.5 were compared. (A) Local alignment using the program LALIGN (a subset of the FASTA package) has matched a short conserved region in the kinase domains that contains the functionally important residues D and N in the DLKPEN sequence and the DFG repeat common to nearly all kinases. (B) Because of the low overall sequence similarity, a standard global alignment of these two sequences using the program ClustalW has not matched these functionally important residues (boxed in each sequence). Green shading, identical amino acids; gray shading, similar amino acids.

Multiple sequence alignments enable the simultaneous comparison of a set of similar sequences

Multiple alignments can be used to find interesting patterns characteristic of specific protein families, to build phylogenetic trees, to detect homology between new sequences and existing families, and to help predict the secondary and tertiary structures of new sequences, as we shall see in more detail in Chapters 11 to 14.

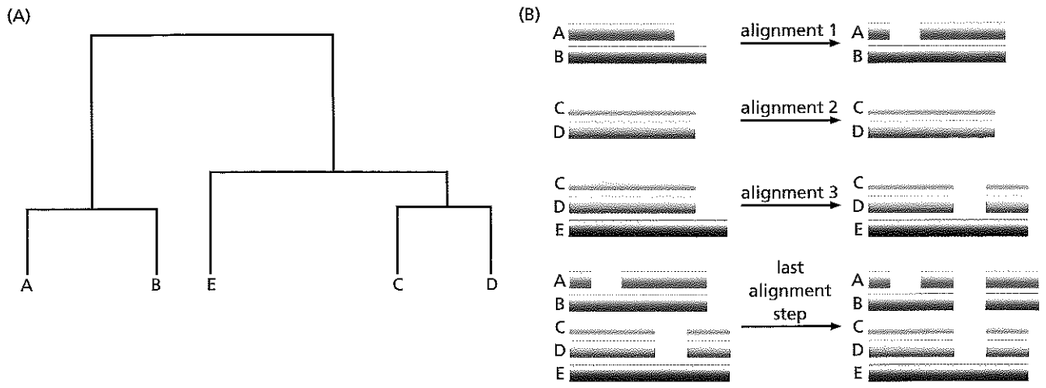
In general, the alignment of multiple sequences will give a more reliable assessment of similarity than a pairwise alignment. The reason for this is that ambiguities in a pairwise comparison can often be resolved when further sequences are compared. Multiple alignment provides more information than pairwise alignment on the individual amino acid positions, such as the overall similarity and evolutionary relationships. This is especially important when using sequence-comparison methods to construct taxonomic phylogenetic trees. Multiple alignment is especially useful for illustrating sequence conservation throughout the aligned sequences. Such conservation over many sequences can identify amino acids that are important for function or for the structural integrity of the protein fold.

Multiple alignments can be constructed by several different techniques

A number of methods are available for generating multiple alignments. One of these is an extension of the dynamic programming method, so that instead of a two-dimensional matrix for a pair of sequences, an alignment of n protein sequences uses an n-dimensional matrix. However, this is limited by the prohibitively large computational requirement of the algorithm, and none of the examples discussed below uses this technique.

Other methods, while often using dynamic programming to align pairs of sequences, use other techniques to combine these together into one multiple alignment. Tree or hierarchical methods of multiple alignment are widely used, for example in the multiple alignment program ClustalW. This method first compares all the sequences in a pairwise fashion, then performs a **cluster analysis** on the pairwise data to generate a hierarchy of sequences in order of their similarity (see Figure 4.8A). The hierarchy is a simple phylogenetic tree and is often referred to as the **guide tree**. A multiple alignment is then built based on the guide tree by first aligning the most similar pairs, then aligning the other sequences with these pairs until all the sequences have been aligned (Figure 4.8B). However accurate this method is, there are problems with it in that any errors in the initial alignments cannot be corrected later as new information from other sequences is added. This

Figure 4.8
The tree method for the multiple alignment of sequences A, B, C, D, and E. Pairwise alignments are first made between all possible pairs of sequences—that is, AB, AC, AD, and so on—to determine their relative similarity to each other (not shown). (A) A cluster analysis is performed on this preliminary round of alignments, and the individual sequences are ranked in a tree according to their similarity to each other. (B) In the next step, the most similar sequences are aligned in pairs as far as possible. These are then aligned to the next closest sequence. This is repeated until all sequences or groups of sequences are aligned.



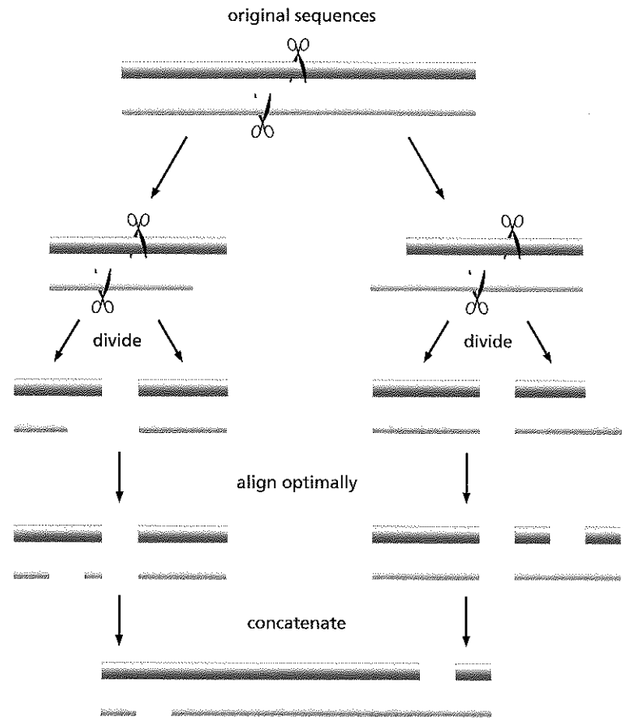


Figure 4.9
The divide-and-conquer method of multiple alignment. The sequences to be aligned are divided into two regions, then into four, and so on until the segments are considered small enough for accurate optimal alignment. The segments are then aligned and in the last step the alignments are concatenated to form the final complete multiple alignment.

difficulty has been avoided in iterative or stochastic sampling procedures as in the Barton and Sternberg program (see Chapter 6).

Other methods for building multiple alignments include the segment method, the consensus method, and the **divide-and-conquer method**. In the divide-and-conquer alignment, the sequences are first cut several times to reduce the length of the sequences to be aligned, the cut sequences are then aligned, and they are finally concatenated into a multiple alignment (see Figure 4.9). Initially, each sequence is divided into two segments at a suitable cut-position somewhere close to the midpoint of the sequences. This procedure is repeated until the sequences are shorter than a predetermined size, which is set as a parameter of the divide-and-conquer algorithm. Therefore the problem of aligning one family of long sequences is divided into several smaller alignment tasks. The segments are then aligned. The last step concatenates the short alignments, giving a multiple alignment of the original sequences.

Multiple alignments can improve the accuracy of alignment for sequences of low similarity

The same proteins with which we illustrated local versus global alignment—a cAMP-dependent protein kinase and a PI3-kinase—will be used to illustrate the improvement multiple alignment can make to the alignment of sequences of low similarity. Figure 4.10A shows part of a pairwise alignment between the protein kinase and the PI3-kinase. The active-site region and the DFG pattern are not aligned. Figure 4.10B shows the result of a multiple alignment between five different PI3-kinases and the protein kinase made using the program ClustalW with the default settings. The effect of the multiple alignment is to give added weight to

```
(A) p110α          TFILGIGDRHNSNIMVKDDG-QLFHIDFGHFLDHKKKKFGYKRERVPFVLT--QDFLIVI 142
    cAMP-kinase    QIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCGTPEYLAPE 179

(B) p110β          SYVLGIG----------DRHSDNINVKKTGQLFHIDFGHILGNFKSKFGIKRERVPFILT 136
    p110δ          TYVLGIG----------DRHSDNIMIRESGQLFHIDFGHFLGNFKTKFGINRERVPFILT 136
    p110α          TFILGIG----------DRHNSNIMVKDDGQLFHIDFGHFLDHKKKKFGYKRERVPFVLT 135
    p110γ          TFVLGIG----------DRHNDNIMITETGNLFHIDFGHILGNYKSFLGINKERVPFVLT 135
    p110_dicti     TYVLGIG----------DRHNDNLMVTKGGRLFHIDFGHFLGNYKKKFGFKRERAPFVFT 135
    cAMP-kinase    QIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCG--TPEYLA 177
```

Figure 4.10
Pairwise and multiple alignments of part of the catalytic domains of five PI3-kinases and a cAMP-dependent protein kinase. (A) Pairwise alignment of PI3-kinase p110α and the protein kinase does not align the important active-site residues and the DFG motif (in green). (B) Multiple alignment of the protein kinase with a set of five PI3-kinases (which have considerable overall homology to each other) has the effect of forcing the best-conserved regions to be matched. Here the DFG motif and the important N and D (green) residues are aligned correctly in all the sequences. In addition it is apparent that a G (green) is also totally conserved (identical) and that three more residues are conserved in their physicochemical properties (blue).

the conserved residues within the PI3-kinases, resulting in a better alignment for that region of the kinase domain.

ClustalW can make global multiple alignments of both DNA and protein sequences

ClustalW uses a tree method of multiple alignment as described briefly above. The program is easy to use with the default settings and can be accessed from a number of Web sites. To use it, one must have collected a set of sequences, perhaps from a database search. Either protein or DNA sequences can be used. The sequences are cut and pasted into a dialog box; you can then run the program immediately with the default settings (for gap penalties and type of scoring matrix, for example). All the settings can be changed if required.

Multiple alignments can be made by combining a series of local alignments

DIALIGN is a relatively recent method for multiple alignment developed by Burkhard Morgenstern and colleagues. Whereas standard alignment programs such as ClustalW compare residues one pair at a time and impose gap penalties, DIALIGN constructs pairwise and multiple alignments by comparing whole ungapped segments several residues long. The alignment is then constructed from pairs of equal-length gap-free segments, which are termed diagonals because they would show up as diagonal lines in the respective pairwise comparison matrices. The segment length varies between diagonals. Many diagonals overlap, and the program has to find a set that can be combined into one consistent alignment (see Section 6.5). As the segments are gap-free there is no need to use a gap-penalty parameter. Every diagonal is given a weight reflecting the degree of similarity between the two segments involved. The overall score of an alignment is the sum of the weights of all the diagonals, and the program finds the alignment with the maximum score. A threshold can be set so that diagonals are considered only if their weights exceed this threshold, so that regions of lower similarity are ignored. As DIALIGN is a local alignment method it may not align the whole sequence, and may align several blocks of residues with unaligned regions between them.

Figure 4.11 illustrates the alignment of five SH2 domain sequences using ClustalW, DIALIGN, and the divide-and-conquer algorithm (DCA) methods compared with the structural/functional alignment from BAliBase, which can be considered accurate. All three methods fail to some extent to align the residues of the first helix correctly, inserting a gap. ClustalW does slightly worse in this region by splitting the helix, but is better in conserving the integrity of the second core block around the FLVR region important for binding. DCA does not align the last helix as well as ClustalW or DIALIGN. However, all the alignment programs are generally good and useful in that they often produce alignments very close to the correct ones based on extra information, such as those found in BALiBase.

(A) structural/functional alignment from BAliBase

```
1csy  SHEKMPWFHGKISREESEQIVLIGSKTNGKFLIRARD--NNGSYALCLLHEGKVLHYRIDKDKTGKLSIPEGK-KFDTLWQLVEHYSYKA------DGLLRVL-TVPCQK
1gri  EMKPHPWFFGKIPRAKAEEML-SKQRHDGAFLIRESES-APGDFSLSVKFGNDVQHFKVLRDGAGKYFL-WVV-KFNSLNELVDYHRSTS-VSRNQQIFLRDIEQVPQQ-
1aya  ---MRRWFHPNITGVEAENLLLTRG-VDGSFLARPSKS-NPGDFTLSVRRNGAVTHIKIQN--TGDYYDLYGGEKFATLAELVQYYMEHHGQLKEKNGDVIELK-YPLN-
2pna  -LQDAEWYWGDISREEVNEKLRDT--ADGTFLVRDASTKMHGDYTLTLRKGGNNKLIKIFH-RDGKYGFSDPL-TFNSVVELINHYRNES-LAQYNPKLDVKL-LYPVS-
1bfi  HHDEKTWNVGSSNRNKAENLLRGK--RDGTFLVRESS--KQGCYACSVVVDGEVKHCVINKTATG-YGFAEPYNLYSSLKELVLHYQHTS-LVQHNDSLNVTL-AYPVYA
```

(B) DIALIGN multiple sequence alignment

```
1csy  SHEKMPWFHGKISREESEQIVLIGSKT-NGKFLIRAR-DN--NGSYALCLLHEGKVLHYRIDKDKTGKLSIPEGKK-FDTLWQLVEHYSYKA-------DGLLRVLT-VPCQK
1gri  EMKPHPWFFGKIPRAKAEEML--SKQRHDGAFLIRESESA--PGDFSLSVKFGNDVQHFKVLRDGAGKYFLWVV-K-FNSLNELVDYHRST--SVSRNQQIFLRDIEQVPQQ-
1aya  M---RRWFHPNITGVEAENLLLTRGV--DGSFLARPSKSN--PGDFTLSVRRNGAVTHIKIQNTGDYYDLYG-GEK-FATLAELVQYYMEHHGQLKEKNGDV-IELK-YPLN-
2pna  LQDAE-WYWGDISREEVNEKL--RDTA-DGTFLVRDA-STKMHGDYYTLTLRKGGNNKLIKIFHRDGKYGFSD-PLT-FNSVVELINHYRNE--SLAQYNPKLDVKLL-YPVS-
1bfi  HHDEKTWNVGSSNRNKAENLL--RGKR-DGTFLVRES-SK--QGCYACSVVVDGEVKHCVLNKTATGYGFAE-PYNLYSSLKELVLHYQHT--SLVQHNDSLNVTLA-YPVYA
```

(C) ClustalW multiple sequence alignment

```
1csy  SHEKMPWFHGKISREESEQIVLIGSKTNGKFLIRARDN--NGSYALCLLHEGKVLHYRIDKDKTGKLSIPEGKKFD-TLWQLVEHYSYK------ADGLLRVLTVPCQK
1gri  EMKPHPWFFGKIPRAKAEE-MLSKQRHDGAFLIRESES-APGDFSLSVKFGNDVQHFKVLRDGAGKY-FLWVVKFN-SLNELVDYHRSTS-VSRNQQIFLRDIEQVPQQ
1aya  ---MRRWFHPNITGVEAEN-LLLTRGVDGSFLARPSKS-NPGDFTLSVRRNGAVTHIKIQNT-GDYYDLYGGEKFA-TLAELVQYYMEHHGQLKEKNGDVIELKYPLN-
2pna  -LQDAEWYWGDISREEVN--EKLRDTADGTFLVRDASTKMHGDYTLTLRKGGNNKLIKIFHR-DGKYGFSDPLTFN-SVVELINHYRNES-LAQYNPKLDVKLLYPVS-
1bfi  HHDEKTWNVGSSNRNKAE--NLLRGKRDGTFLVRESSK--QGCYACSVVVDGEVKHCVINKT-ATGYGFAEPYNLYSSLKELVLHYQHTS-LVQHNDSLNVTLAYPVYA
```

(D) divide-and-conquer multiple sequence alignment

```
1csy  SHEKMPWFHGKISREESEQIVLIGSKTNGKFLIRA-RDNN-GSYALCLLHEGKVLHYRIDKDKTGKLSIPEGKK-FDTLWQLVEHY-SY----KADGLLRV-L-TVPCQK
1gri  EMKPHPWFFGKIPRAKAEEMLS-KQRHDGAFLIRE-SESAPGDFSLSVKFGNDVQHFKVLRDGAGK-YFLWVVK-FNSLNELVDYH-RSTSVSRNQQIFLRDIEQVPQQ-
1aya  ---MRRWFHPNITGVEAENLLL-TRGVDGSFLARP-SKSNPGDFTLSVRRNGAVTHIKIQNTGDYY-DLYGGEK-FATLAELVQYYMEHHGQLKEKNGDVIEL-KYPLN-
2pna  -LQDAEWYWGDISREEVNEKL--RDTADGTFLVRDASTKMHGDYTLTLRKGGNNKLIKIFHRDGKY-GFSDPLT-FNSVVELINHY-RNESLAQYNPKLDVKL-LYPVS-
1bfi  HHDEKTWNVGSSNRNKAENLL--RGKRDGTFLVRE-SSKQ-GCYACSVVVDGEVKHCVINKTATGY-GFAEPYNLYSSLKELVLHY-QHTSLVQHNDSLNVTL-AYPVYA
```

Once a satisfactory alignment has been obtained, there are now numerous programs available through the Web that allow you to view, analyze, and even edit alignments. AMAS (Analyze Multiply Aligned Sequences), CINEMA (Colour Interactive Editor for Multiple Alignments), and ESPript (Easy Sequencing in Postscript) are but a few.

Alignment can be improved by incorporating additional information

The alignment of two or more sequences can be improved by incorporating expert knowledge such as known structural properties of one or more sequences. For example, if the structure of one of the proteins to be aligned is known, then the gap penalty can be increased for regions of known secondary structures such as α-helices or β-strands, as these regions are less likely to suffer insertions or deletions. This will mean that few or no gaps are introduced into these regions. On the other hand, gap penalties can be decreased for loop regions, in which insertions and deletions are better tolerated.

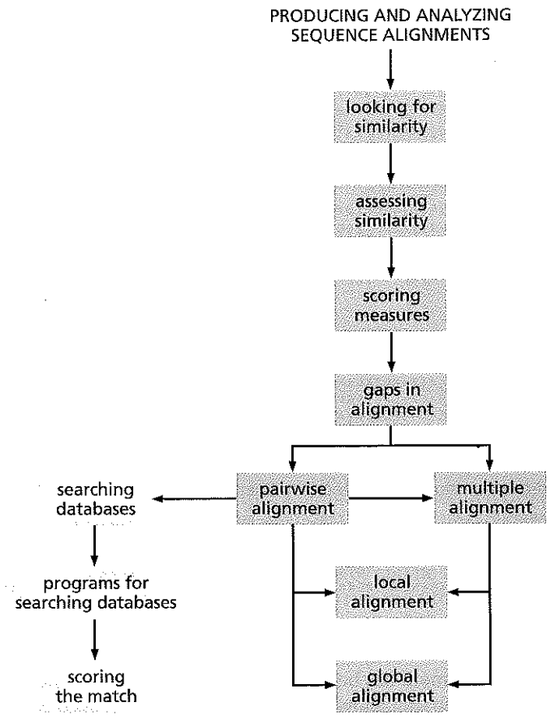
Often the results of an automatic alignment program benefit from manual final adjustment. For example, if specific residues are known to be important for structure, function, or ligand binding, then manual realignment may be necessary to match these residues.

Figure 4.11
Known structural alignments can be useful in checking sequence alignments. (A) Multiple alignment of the sequences of five SH2 domains according to their sequence/structure alignment in BAliBase. α-Helices are shown in red and β-strands in yellow. (B) Multiple alignment for the same set of sequences obtained by DIALIGN. (C) Alignment obtained by ClustalW. (D) Alignment obtained by the divide-and-conquer method. There is not much difference in performance between the algorithms (all were run with the default settings), although some alignment programs break the secondary structure element indicated by dashes. The coded names of the domains on the left are their identification numbers in the Protein Data Bank (PDB).

4.6 Searching Databases

Searching sequence databases now has a part to play in nearly every branch of molecular biology, and is crucial for making sense of the sequence data becoming available from the genome projects. For example, one may wish to search the database with a DNA sequence to locate and identify a gene in a new genome. When a protein sequence is available, then searching through the database can be used to identify the potential function. Sometimes one wishes to find the gene for

Flow Diagram 4.3
The key concept introduced in this and the following section is that applications have been designed to overcome the problems associated with searching a database for sequences that are similar to a query sequence, including the need to pay special attention to the statistical significance of the alignment scores obtained.



a particular protein in a genome, which can be done by searching with a homologous protein or DNA sequence.

We will now discuss the practical task of searching sequence databases to find sequences that are similar to the query sequence or search sequence that we submit to them (Flow Diagram 4.3). When searching a database with a newly determined DNA or protein query sequence, one does not usually know whether an expected similarity might span the entire query sequence or just part of it; similarly, one does not know if the match will extend along the full length of a database sequence or only part of it. Therefore, one initially needs to look for local alignments between the query sequence and any sequence in the database. The top-scoring database sequences are then candidates for further analysis.

Database searching needs to be both sensitive, in order to detect distantly related homologs and avoid **false-negative** searches, and also specific, in order to reject unrelated sequences with fortuitous similarity (**false-positive** hits). This is not an easy balance to achieve, and search results should be scrutinized with care.

In general, it is not possible to decide from a visual inspection of the alignment whether the database and query sequences are truly homologous. However, analysis of the score statistics has provided us with useful measures to estimate the validity of a hit. This important aspect of database searching, which is required to interpret any database search correctly, is discussed later in this chapter and in more detail in Section 5.4.

Fast yet accurate search algorithms have been developed

The sequence databases are now extremely large and growing daily. This means that aligning a query sequence with sequences in a database requires considerable

computer resources. In the past, this exceeded the available computing power and so great effort was put into developing fast yet accurate alignment methods. Almost all database search programs currently in use are modifications of the rigorous methods discussed earlier. The Needleman–Wunsch and Smith–Waterman methods are rigorous in the sense that given a scoring scheme they are guaranteed to find the best-scoring alignments between two sequences. Two suites of programs are in common use for database searching: FASTA and BLAST. These use dynamic programming, but only for database entries that have a segment sufficiently similar to the query sequences. The methods used to find these entries are purely heuristic; that is, not rigorous.

FASTA is a fast database-search method based on matching short identical segments

FASTA is a popular database-searching program that increases the speed of a search at the expense of some sensitivity. It speeds up the searching process by using **k-tuples**, short stretches of k contiguous residues. In protein searches k can equal 1 or 2, while 6 is a typical value for DNA. The program makes up a dictionary of all possible k-tuples within the query sequence. Each entry contains a list of numbers that describe the location of the k-tuple in the query sequence. This is called **hashing**, and the theory behind it is described in Section 5.3. Therefore, for each k-tuple in the searched sequences, FASTA only has to consult the dictionary to find out if it occurs in the query sequence. However, sensitivity is reduced because a partial match of a k-tuple (for example, AC to AG in DNA) is ignored. Therefore, although speed increases with the length of a k-tuple, sensitivity will decrease.

In the first step of the FASTA method all possible pairwise k-tuples are identified: these can be considered as diagonals in a set of dot-plots. In the second step, alignments of these diagonals are rescored using a scoring matrix such as one of those described above. In this step, the k-tuple regions are also extended without including gaps, and only those that score above a given threshold are retained. In the third step, the program checks to see if some of the highest-scoring diagonals can be joined together. Finally, the search sequences with the highest scores are aligned to the query sequence using dynamic programming. The final alignment score ranks the database entries and the highest-scoring set is reported.

BLAST is based on finding very similar short segments

BLAST (Basic Local Alignment Search Tool) or Wu-BLAST (a version of BLAST developed at Washington University, St Louis) is one of the most widely used database-search program suites. It relies on finding core similarity, which is defined by a window of preset size (called a "word") with a certain minimum density of matches (for DNA) or with an amino-acid similarity score above a given threshold (for proteins). Note that these amino acid word-matches do not only include identities and that they are scored with a standard substitution matrix. In the first step, all suitable matches are located in each database sequence. Subsequently, matches are extended without including gaps, and on this basis the database sequences are ranked. The highest-scoring sequences are then subjected to dynamic programming to obtain the final alignments and scores. BLAST and Wu-BLAST can be run with or without the use of gaps. The gapped setting of BLAST, which is usually the default setting, reports the best local alignments and is suitable for most applications. Both the FASTA and BLAST methods are described in detail in Section 5.3.

Different versions of BLAST and FASTA are used for different problems

Many of the search algorithms can be used to search either nucleic acid or protein sequences, or even to search a protein-sequence database using a nucleic acid

Table 4.1
The various algorithms within the FASTA package are given with descriptions of their uses.
Equivalent programs in the BLAST package are highlighted. The ktup parameter of fasta defines the length of a k-tuple, as explained in Section 5.3.

Program	Description	BLAST equivalent
fasta	Protein compared to protein database or DNA to DNA database. For protein, ktup = 2 by default (ktup = 1 is more sensitive); default for DNA is 6; 4 or 3 is more sensitive. 1 should be used for short DNA stretches.	blastp/blastn
ssearch	Uses Smith–Waterman algorithm. Can search protein to protein or DNA to DNA. Can be more sensitive than fasta with protein sequences.	
fastx/fasty	DNA compared to protein database. DNA translated into all three frames. fasty slower than fastx but better. Used to see if DNA encodes a protein.	blastx
tfastx/tfasta	Protein compared to DNA database. Mainly used to identify EST sequences. This is preferred over fastx as protein comparison is more sensitive than DNA.	tblastn (tblastx compares translated DNA to translated DNA database)
fastf	Mixed peptide sequence (such as obtained by Edman degradation) compared to protein database.	
tfastf	Mixed peptide sequence compared to DNA database.	

sequence and vice versa. However, you need to choose the correct program for the required type of search. In BLAST, for example, one can choose among blastp, which compares an amino acid query sequence against a protein-sequence database; blastn, which compares a nucleotide query sequence against a nucleic acid sequence database; blastx, which compares a nucleotide query sequence translated in all reading frames against a protein-sequence database; tblastn, which compares a protein query sequence against a nucleotide-sequence database dynamically translated in all reading frames; and finally, tblastx, which compares the six possible translations of a nucleotide query sequence against the six frame translations of a nucleotide-sequence database. The FASTA suite has similar versions of these search programs (see Table 4.1).

PSI-BLAST enables profile-based database searches

Variations of BLAST such as PSI-BLAST (Position-Specific Iterative BLAST) have been devised. This suite of programs makes use of features characteristic of a particular protein family to identify related sequences in a protein database, and can identify related sequences that are too dissimilar to be found in a straightforward BLAST search. In PSI-BLAST, a **profile**, or **position-specific scoring matrix (PSSM)**, of a set of sequences is constructed from a multiple alignment of the highest-scoring hits returned in an initial BLAST search (see Section 6.1). The PSSM is created by calculating new scores for each position in this alignment. A highly conserved residue at a particular position is assigned a high positive score, while other residues at that position are assigned high negative scores. At positions that are weakly conserved throughout the alignment, all residues are given scores near zero. The profile generated is used to replace the substitution matrix in a subsequent BLAST search. This process can be repeated many times; each time, the results from the search are used to refine the profile. This type of iterative searching results in increased sensitivity and has been used to good effect in protein-fold recognition programs such as 3D-PSSM (see Chapter 13).

Ways of extracting more distantly related homologous sequences and finding links between known families are now being explored. Such methods include, for example, the use of **intermediate sequences**; that is, sequences that are found in more than one family. Suppose we submit an unknown sequence A to a database search and among the significant hits there is a protein called, for example, mediator protein. We then submit an unknown sequence B to the same database search, and this also returns mediator protein with a significant score. Then, especially if more than one such intermediate sequence is found, we can deduce that sequences A and B are homologous, as their families are related. Such ideas can be automated for ease of application.

SSEARCH is a rigorous alignment method

Despite the computational requirements, some programs have been written that use rigorous methods to search databases. SSEARCH is a search program based on the Smith–Waterman algorithm and is therefore slower than either BLAST or FASTA. SSEARCH performs a rigorous search for similarity between a query sequence and the database. Other search algorithms based on the Smith–Waterman method have been written and are gaining in popularity as computer power increases.

4.7 Searching with Nucleic Acid or Protein Sequences

DNA or RNA sequences can be used either directly or after translation

In general, nucleic acid sequence searches are more difficult to handle and analyze than protein sequence searches. However, most primary data will be in the form of nucleic acid sequences. If you have an untranslated DNA or RNA sequence and you want to know if the DNA codes for a protein, you can use fasta, ssearch, or blastn (see Table 4.1) to search the EST (expressed sequence tag), EMBL, or nr (nonredundant) databases, or one of the species-specific genome EST databases, such as EST-Rodent. The results may well be confusing, in that a lot of partial sequence matches will be found. Many retrieved sequences will also be unknown sequences. An easier search can be made using fastx/fasty (or blastx), which will translate the DNA in all three reading frames on both strands—six translations in all—and search a protein database of choice. More details and examples of dealing with DNA sequences can be found in Chapter 9.

The quality of a database match has to be tested to ensure that it could not have arisen by chance

How good is an alignment and how believable are the results of a database search? These vital questions must be answered before any further use can be made of the results. Every alignment reported will have been selected on the basis of its score. What we need to know is whether the score is greater than we would expect from the alignment of the sequence with a random (unrelated) sequence. However, there is a complex relationship between the score and the significance of the sequence similarity. For one thing, as each pair of aligned residues contributes to the score, longer sequences are expected to give higher alignment scores, assuming the same degree of similarity.

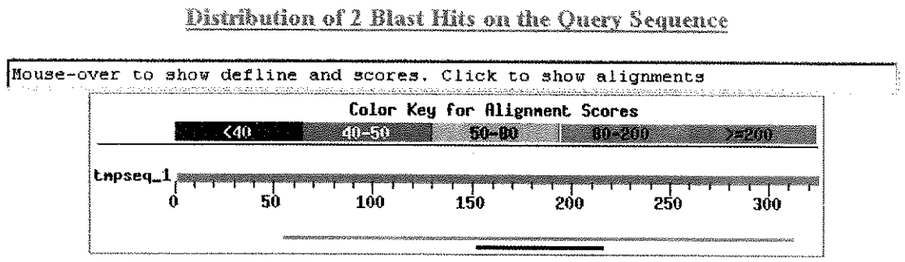
If a large number of random sequences are generated and aligned with the query sequence, the resulting alignment scores will follow a particular distribution. Because we always choose the best-scoring alignment, the distribution will be related to the extreme-value distribution (see Section 5.4). Through application of

Figure 4.12
The results of a search of the SWISS-PROT protein sequence database using BLAST with PI3-kinase p100α as the query sequence. (A) Output from a standard BLAST search. Each line reports a separate database sequence. The penultimate column gives the alignment score, and the last column the *E*-value. Hits before the arrow are significant, while below the arrow the hit does not have a significant score. (B) A BLAST search on one month's new sequences, using the same query sequence as in (A), finds only two matches. One is a PI4-kinase, which has most of its sequence aligned to the query sequence (magenta line). The other has only a small region aligned (black line) and a borderline score. (C) Output from a Conserved Domain Database (CDD) search.

(A)

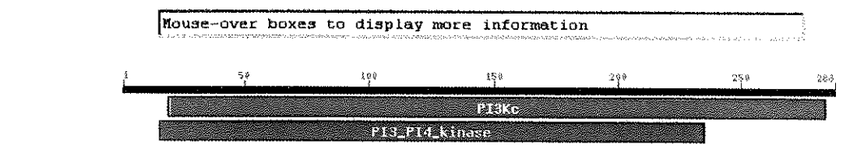
```
sp|P32871|P11A_BOVIN  PHOSPHATIDYLINOSITOL 3-KINASE CATALYTI...   680  0.0
sp|P42336|P11A_HUMAN   PHOSPHATIDYLINOSITOL 3-KINASE CATALYT...   676  0.0
sp|P42337|P11A_MOUSE   PHOSPHATIDYLINOSITOL 3-KINASE CATALYT...   674  0.0
sp|P42338|P11B_HUMAN   PHOSPHATIDYLINOSITOL 3-KINASE CATALYT...   338  9e-93
sp|O35904|P11D_MOUSE   PHOSPHATIDYLINOSITOL 3-KINASE CATALYT...   332  7e-91
sp|O00329|P11D_HUMAN   PHOSPHATIDYLINOSITOL 3-KINASE CATALYT...   331  2e-90

sp|P47473|RIR1_MYCGE  RIBONUCLEOSIDE-DIPHOSPHATE REDUCTASE A...    34  0.59
```

(B)




```
dbj|BAB10275.1|  (AB008266) phosphatidylinositol 4-kinase [A...   111  3e-25
dbj|BAB11344.1|  (AB011477) AtRAD3 [Arabidopsis thaliana]          38  0.008
```

(C)



Sequences producing significant alignments: Score (bits) E value

gnl|Smart|PI3Kc Phosphoinositide 3-kinase, catalytic domain; Phosphoinositide ... 301 3e-83
gnl|Pfam|pfam00454 PI3_PI4_kinase, Phosphatidylinositol 3- and 4-kinases 263 9e-72

gnl|Smart|PI3Kc, Phosphoinositide 3-kinase, catalytic domain; Phosphoinositide 3-kinase isoforms participate in a variety of processes, including cell motility, the Ras pathway, vesicle trafficking and secretion, and apoptosis. These homologues may be either lipid kinases and/or protein kinases: the former phosphorylate the 3-position in the inositol ring of inositol phospholipids. The ataxia telangiectesia-mutated gene produced, the targets of rapamycin (TOR) and the DNA-dependent kinase have not been found to possess lipid kinase activity. Some of this family possess PI-4 kinase activities.

Add query to multiple alignment, display up to 10 sequences most similar to the query

```
Length = 265
Score =  301 bits (763), Expect = 3e-83

Query:  19  IIFKNGDDLRQDMLTLQIIRIMENIWQNQGLDLRMLPYGCLSIGDCVGLIEVVRNSHTIM  78
Sbjct:  2   IIFKHGDDLRQDMLILQILRIMESIWETESLDLCLLPYGCISTGDKIGMIEIVKDATTIA  61
```


this distribution it is possible to estimate the probability of two random sequences aligning with a score greater than or equal to *S*. This is usually reported as an **expectation value** or ***E*-value**, and is used to order the database search results.

The programs BLAST and FASTA calculate an *E*-value, which is the number of alignments with a score of at least *S* that would be expected by chance alone in searching a complete database of *n* sequences. These *E*-values can vary from 0 to *n*. For example, by chance alone, you would expect to find three sequence alignments with an *E*-value of 3.0 or less in a database search, so an *E*-value of 3.0 suggests that the database sequence is not related to the query sequence. Quite closely related sequences often give very small *E*-values of 10^{-20} or less, and such scores clearly indicate a significant similarity of the database and query sequences. However, we really need to know how large an *E*-value can be while still reliably indicating a significant sequence similarity. It is important to remember that the *E*-value depends on the sequence length and the number of sequences in the database as well as on the alignment score.

In general, the smaller the *E*-value the better the alignment, and the higher the percentage identity the more secure the assessment of the significance of the similarity between the database sequence and the query sequence. The default *E*-value threshold in many search packages is set to either 0.01 or 0.001. However, most programs permit the user to set the *E*-value threshold, and matches above that threshold will not be included in the output.

To test new or existing sequence-alignment programs and their scoring schemes one can compare the alignment obtained by the program against carefully constructed alignments that are based on known structural features or biological function. There are databases of such accurately aligned sequences, such as BAliBase.

Choosing an appropriate *E*-value threshold helps to limit a database search

To illustrate some of the possible sequence searches, alignments, and analyses that can be carried out via the Web, we will use two examples: the catalytic domain from a PI3-kinase and the protein-interaction domain SH2. Structural information and an accurate alignment in the BAliBase database are available for the family of SH2 domains.

The human *Syk* tyrosine kinase carboxy-terminal SH2 domain is the first query sequence. Protein searches with BLAST through the SWISS-PROT database gave 149 sequences below the default *E*-value cut-off. All these were SH2-related domains. That is a lot of information to cope with. All the *E*-values were very low, indicating that all the hits were significant. This is a case of result overload. Decreasing the *E*-value cut-off would have no effect in this case, as all the hits were far below the threshold used.

To reduce the large number of hits one could search a subset of the data, for example only the newest sequences in the database (the "month" option; that is, those deposited in the last month) or a specific genome database. A search through sequences released in the past month detected eight sequences, all with significant scores, of which three had not been identified. A search through the *Drosophila* genome data yielded three hits, all of which are unknown. Taking one of the regions that matched our SH2 (a section of the *Drosophila* 3R chromosome arm) and searching with this sequence through SWISS-PROT yielded a highly significant hit to an SH2 domain of a rat protein. So we may have identified a previously unknown *Drosophila* sequence as containing an SH2 domain.

This example illustrates a search through the database with a family that is very well represented and shows the problems that can arise. We will now look at a family that is not so well represented—the PI3-kinases.

First the SWISS-PROT database was searched using the catalytic domain protein sequence from the PI3-kinase p110α using BLAST with the *E*-value cut-off set to 1. Thirty-two hits were found. In this list there are three near-identical isoforms of p110α which have an *E*-value of 0.0; that is, the chance of obtaining such a match with random sequence is taken to be zero. There is one match that is not significant: ribonucleoside-diphosphate reductase, with an *E*-value of 0.59 (see Figure 4.12A). From the assigned function this is clearly a different enzyme, but the enzymatic reactions of both this reductase and the kinases involve a nucleotide, which might have led to some small degree of similarity between the sequences. Any such speculation would need further and more thorough investigation. If we rerun the search with the *E*-value cut-off set to 0.01 (the advised setting) only the significant matches are retrieved.

Searching with BLAST through a subset of sequences such as those that have only been released in one month found two hits: one is a homolog of PI3-kinase, a PI4-kinase with a significant *E*-value, and the other is a segment of an *Arabidopsis thaliana* protein, atRad3, with an *E*-value score of borderline significance. From the length of the matched sequence illustrated in the search output (see Figure 4.12B) the segment seems far too short to be of interest; compare the length of the matched PI4-kinase, in magenta. For this reason the hit can now be reclassified as not significant.

Another useful option available within the BLAST search server is a concurrent search of the Conserved Domain Database (CDD) entries. Figure 4.12C shows the results of using this option on the PI3-kinase sequence. Proteins often contain several domains, and the program CD-Search can potentially identify domains present in a protein sequence. CDD contains domains derived mainly from the SMART and Pfam protein-family databases. To identify conserved domains in a protein sequence, the CD method uses the BLAST algorithm where the query sequence is matched with a PSSM designed from the conserved domain alignments. Matches are shown as either a pairwise alignment of the query sequence to a representative domain sequence or as a multiple alignment.

A FASTA search with the p110α sequence through SWISS-PROT with default settings (k-tuple = 2) yielded 36 hits, of which eight had a nonsignificant score (see Figure 4.13). Of these eight, ribonucleoside-diphosphate reductase was also found by BLAST. Although both FASTA and BLAST report an *E*-value, the actual values are different, which reflects subtle differences in the methods used. An SSEARCH search of the SWISS-PROT database with default settings found 29 significant hits. SSEARCH, a more rigorous method, found fewer hits than BLAST or FASTA.

Low-complexity regions can complicate homology searches

Among the many features that can complicate a sequence-similarity search is the occurrence of **low-complexity regions** in protein sequences. These are regions with a highly biased amino acid composition, often runs of prolines or acidic amino


```
the best scores are:                                                    E(86391)
SW:P11A_BOVIN P32871 PHOSPHATIDYLINOSITOL 3-KINAS  (1068) 2228 493 1.2e-138
SW:P11A_HUMAN P42336 PHOSPHATIDYLINOSITOL 3-KINAS  (1068) 2216 490 7.4e-138
SW:P11A_MOUSE P42337 PHOSPHATIDYLINOSITOL 3-KINAS  (1068) 2204 488 4.5e-137
SW:P11B_HUMAN P42338 PHOSPHATIDYLINOSITOL 3-KINAS  (1070) 1126 254 1.1e-66
                                  ↓ other sequences
SW:ESR1_YEAST P38111 ESR1 PROTEIN.                 (2368)  144  41     0.028
SW:PRA2_USTMA P31303 PHEROMONE RECEPTOR 2.         ( 346)  116  35      0.35
SW:TEL1_YEAST P38110 TELOMER LENGTH REGULATION PR  (2787)  127  37      0.42
SW:YA51_METJA Q58451 HYPOTHETICAL PROTEIN MJ1051.  ( 513)  112  34      0.91
SW:RIR1_MYCGE P47473 RIBONUCLEOSIDE-DIPHOSPHATE R  ( 721)  106  33         3
SW:YAY1_SCHPO Q10209 HYPOTHETICAL 44.8 KDA PROTEI  ( 392)   99  31       5.1
SW:PAFA_CAVPO P70683 PLATELET-ACTIVATING FACTOR A  ( 436)   96  30       8.8
SW:KC47_ORYSA P29620 CDC2+/CDC28-RELATED PROTEIN   ( 424)   95  30       9.9
```

Figure 4.13
Output from a search of the SWISS-PROT protein sequence database using FASTA with PI3-kinase p110α as the query sequence. Thirty-six hits were obtained. Eight of these have a nonsignificant score (below the arrow). One of these, ribonucleoside-diphosphate reductase, was also found by BLAST. The *E*-values in FASTA are different from those in BLAST.

acids. In some cases, self-comparison dot-plots (see page 77) can identify low-complexity regions in a protein sequence. Alignments of such regions in different proteins can achieve high scores, but these can be misleading and can obscure the biologically significant hits. It is better to exclude low-complexity regions when constructing the alignment. By default, the BLAST program filters the query sequence for low-complexity regions. In the BLAST output file, an X marks regions that have been filtered out (using SEG for proteins and DUST for DNA) (see Box 5.2).

Figure 4.14A shows a self-comparison dot-plot of human prion protein precursor (PrP), an abnormal form of which is found in large amounts in the brains of people with neurodegenerative diseases such as Creutzfeldt–Jakob disease (CJD) and kuru (see Box 4.6). It has several low-complexity regions, which are seen as dark diagonal lines (apart from the main identity diagonal) and the ordered dark-shaded regions. Figure 4.14B shows a search for homologs of the human PrP. The extensive low-complexity regions have been filtered out in the query sequence (as indicated by the strings of Xs). A BLAST search of SWISS-PROT with human PrP with the low-complexity filter turned on gave approximately 40 hits, all prion proteins. One of

Box 4.6 Prions: Proteins that can exist in different conformations

Scrapie in sheep, bovine spongiform encephalopathy (BSE) in cattle, and Creutzfeldt–Jakob disease (CJD), fatal familial insomnia, and kuru in humans are rare, fatal, transmissible, neurodegenerative diseases known generally as the transmissible spongiform encephalopathies, after the characteristic damage they do to the brain. They can arise sporadically, or as a result of the inheritance of a faulty gene, or can be transmitted by ingestion of infected material. Kuru, which was relatively common in people in the Eastern Highlands of Papua New Guinea in the 1950s and 1960s, was found to be caused by the ritual custom of eating the brains of dead relatives, while a variant form of CJD (vCJD), which has appeared only recently, is thought to be caused by people having eaten BSE-infected meat products.

The causal agent in the spongiform encephalopathies is believed to be an infectious protein, a prion, rather than a DNA or RNA virus. Prions are normal proteins that have the property of being able to convert into an alternative stable conformation, which is associated with disease, although the mechanism by which prions cause cell death and neurodegeneration is not yet fully understood. The normal form of the prion protein (PrP^c) is a monomer with a structure consisting mainly of α-helices, and is mainly found at the cell surface, whereas the abnormal form (PrP^{Sc}), is mainly β-sheet and has a tendency to aggregate into clumps. PrP^{Sc} itself appears to be able to induce the conversion of PrP^c into PrP^{Sc}. The prion protein is an example of a metastable protein, where the same or similar sequences can exist in different stable structural forms.

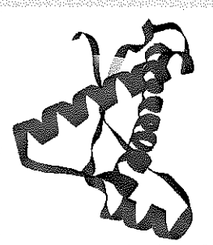
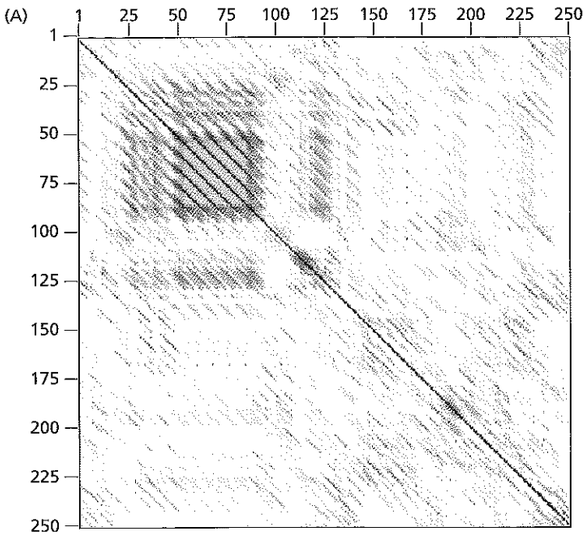


Figure B4.4
A ribbon representation of the normal form of prion protein, PrP^c.

(A)



(B)

```
>sp[P04156]PRIO_HUMAN  MAJOR PRION PROTEIN PRECURSOR (PRP) (PRP27-30) (PRP33-35C) (ASCR)
          Length = 253

Score =  312 bits (792), Expect = 5e-85
Identities = 154/236 (65%), Positives = 154/236 (65%)

Query: 64  MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQGSPGGNRYXXXXXXXXXXX 123
           MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQGSPGGNRY
Sbjct: 1   MANLGCWMLVLFVATWSDLGLCKKRPKPGGWNTGGSRYPGQGSPGGNRYPPQGGGGWGQP 60

Query: 124 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXTHSQWNKPSKPKTNMKHMXXXXXXXX 183
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXTHSQWNKPSKPKTNMKHM
Sbjct: 1   HGGGWGQPHGGGWGQPHGGGWGQPHGGGWGQGGGTHSQWNKPSKPKTNMKHMAGAAAAGA 120

Query: 184 XXXXXXXXXXXXXXXRPIIHFGSDYEDRYYRENMHRYPNQVYYRPMDEYSNQNNFVHDCV 243
                       RPIIHFGSDYEDRYYRENMHRYPNQVYYRPMDEYSNQNNFVHDCV
Sbjct: 121 VVGGLGGYMLGSAMSRPIIHFGSDYEDRYYRENMHRYPNQVYYRPMDEYSNQNNFVHDCV 180

Query: 244 NITIKQHXXXXXXXXXXXXXXDVKMMERVVEQMCITQYERESQAYYQRGSSMVLFS 299
           NITIKQH              DVKMMERVVEQMCITQYERESQAYYQRGSSMVLFS
Sbjct: 181 NITIKQHTVTTTTKGENFTETDVKMMERVVEQMCITQYERESQAYYQRGSSMVLFS 236
```

Figure 4.14
Dealing with low-complexity regions of sequence. (A) The low-complexity regions are clearly visible on a self-comparison dot-plot of a human prion precursor protein (PrP). They are indicated by the black diagonal lines on either side of the identity diagonal and by the ordered dark-shaded regions. (B) Results of a database search with PrP from which sequences of low complexity have been filtered out by application of the program SEG, which marks them with Xs (top row of the alignment).

these is shown aligned with the query sequence in the figure. When the filter was turned off, the number of hits rose to 220, most of which were not homologous.

Sometimes one may wish to study low-complexity regions in particular. For example, in the case of the tubulin and actin gene clusters it is thought that amplification of the protein-coding genes may be related to these regions. There are options in BLAST that allow you to select these regions for study.

Different databases can be used to solve particular problems

To some extent, the choice of which database to search will depend on which databases are provided by the site that runs the search algorithms. Most sites contain a selection of the most popular databases, such as GenBank for DNA sequences, SWISS-PROT for annotated protein sequences, TrEMBL, a translated EMBL DNA-sequence database, and PDB, a database of protein structures with

sequences (see Chapter 3). Some sites also provide access to expressed sequence tag (EST) databases, such as dbEST, and genome-sequence databases from some of the fully sequenced genomes

In general, a first pass should be run on a generic protein- or nucleic acid sequence database. You can also carry out a search on the PDB to see if your query sequence has a homolog with known structure. More specific searches can be performed to answer particular questions. For example, if it is suspected that the query sequence belongs to a family of immune-system proteins, the search could be carried out on the Kabat database, which contains sequences of immunological interest. If the sequence originates from a mouse, you may want to know if a homolog exists in the rat, *Drosophila*, or human genomes, and should therefore search the databases containing sequences from the appropriate species. You also need to check that you are searching a database that is up to date; sites such as those at NCBI and EBI are regularly updated.

If no match is found to the query sequence, it does not necessarily mean that there is no homolog in the databases, just that the similarity is too weak to be picked up by existing techniques. Techniques are continually being improved and the amount of sequence data continues to increase; you should therefore periodically resubmit your sequence.

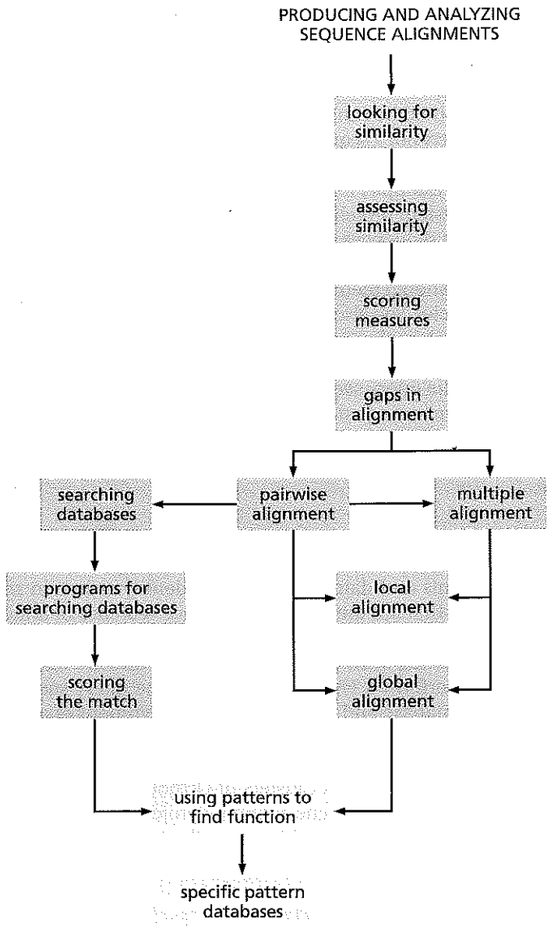
Many other sequence-related databases can usefully be searched and provide additional information. For example the Sequences Annotated by Structure (SAS) server is a collection of programs and data that can help identify a protein sequence by using structural features that are the result of sequence searches of annotated PDB sequences. Residues in the sequences of known structures are colored according to selected structural properties, such as residue similarity, and are displayed using a Web browser. SAS will perform a FASTA alignment of the query sequence against sequences in the PDB database and return a multiple alignment of all hits. Each of the hits is annotated with structural and functional features. That information can be used to annotate the unknown protein sequence. Further links are provided to the separate PDB files. Databases such as Clusters of Orthologous Groups (COGs) and UniGene can help in gene discovery, gene-mapping projects, and large-scale expression analysis. Sites such as Ensembl provide convenient access for gene searches in many different annotated eukaryotic genomes and useful associated information.

4.8 Protein Sequence Motifs or Patterns

If the similarity between an unknown sequence and a sequence of known function is limited to a few critical residues, then standard alignment searches using BLAST or FASTA against general sequence databases such as GenBank, dbEST, or SWISS-PROT will fail to pick up this relationship. What is required is a method of searching for the occurrence of short sequence patterns, or **motifs** (see Flow Diagram 4.4). A motif, in general, is any conserved element of a sequence alignment, whether composed of a short sequence of contiguous residues or a more distributed pattern. Functionally related sequences will share similar distribution patterns of critical functional residues that are not necessarily contiguous. For example, conserved amino acid residues comprising an enzyme's active site may be distant from each other in the protein sequence but will still occur in a recognizable pattern because of the constraints imposed by the requirement for them to come together in a particular spatial configuration to form the active site in the three-dimensional structure.

There are three different types of activity associated with pattern searching. A query sequence can be searched for patterns (from a patterns database) that could help suggest functional activity. A sequence database can be searched with a specific pattern, for example to determine how many gene products in a genome have a

Flow Diagram 4.4
The key concepts introduced in this and the following two sections are that sequence patterns can be very useful in identifying protein function and that special pattern databases and search programs have been designed to assist in identifying patterns in a query sequence.



specific function. Lastly, we may want to define a new pattern specific for a selected set of sequences.

In searches with new sequences, the whole database is searched and expert knowledge, such as the known function of a homologous protein, is then used to extrapolate the function of the new sequence. In contrast, when new patterns and motifs are used to search a database, the expert knowledge is needed right at the beginning to construct the motifs that are intended to identify the specific function or any other physicochemical property.

Pattern and motif searches are mostly used with protein sequences rather than nucleotide sequences, as the greater number of different amino acids makes protein patterns more efficient in discriminating truly significant hits. In addition, many of the patterns identify biological function, which is mediated at the protein level. There are, however, particular problems in DNA- and genome-sequence analysis where searching for motifs is useful (see Chapters 9 and 10).

Creation of pattern databases requires expert knowledge

The construction of patterns or motifs is of prime importance in characterizing a protein family, and much time and energy has gone into constructing pattern and

motif databases that one can search with an unknown sequence. One of the most important steps is careful selection of the sequences used to define the pattern. If these do not all share the same biological properties for which you wish to define a pattern, you will almost certainly encounter problems later. Thus, experimental evidence of function or clear homology is necessary for all the sequences used.

Some pattern databases have been constructed by hand by inspection of large amounts of data. This is very time consuming, but necessary, as the task of extracting a pattern is a complex one, depending on expert knowledge of the structures and/or functions of the sequences involved. For example, analysis of the X-ray structure of a protein can delineate the functional residues involved in an enzyme active site or a regulatory binding site, and an initial pattern can be generated. This pattern can then be refined by multiple alignment of sequences of other members of the same structural or functional protein family. If no structural data are available, multiple sequence alignment of short regions of similarity, assessed alone or in conjunction with experimental data on biochemical and cellular function, can be used to extract a pattern.

The simplest method of constructing a pattern or motif is the consensus method, in which the most similar regions in a global multiple sequence alignment are used to construct a pattern. Those positions in the alignment that are all occupied by the same residue (or a limited subset of residues) are used to define the pattern at these positions, by specifying just the allowed residues at each position. More sophisticated patterns can be generated using scoring tables to assess the similarity of the matched amino acids. In this case, instead of just defining the pattern as requiring, for example, a glutamic or aspartic acid at a given position, different residues at this position have associated scores.

The BLOCKS database contains automatically compiled short blocks of conserved multiply aligned protein sequences

Sequence motifs can also be defined automatically from the multiple alignment of a specified set of sequences. Blocks are multiple alignments of ungapped segments of protein sequence corresponding to the most highly conserved regions of the proteins. The blocks for the BLOCKS database are compiled automatically by looking for the most highly conserved regions in groups of proteins documented in the PROSITE database. The blocks are then calibrated against the SWISS-PROT database to obtain a measure of the chance distribution of matches. The calibrated blocks make up the BLOCKS database, against which a new sequence can be searched. Both protein and DNA sequences (automatically translated into a protein sequence) can be submitted to search the BLOCKS database. The BLOCKS Web site, in addition to providing the BLOCKS database, will align your set of sequences and automatically design a block with which you can search SWISS-PROT. Generating blocks from your set of sequences and searching with them can find sequences that have very weak sequence similarity but are, nevertheless, functionally related. Generating patterns and/or blocks is also a useful method to search for hints to function within an unknown sequence.

Another program that will analyze a set of sequences for similarities and produce a motif for each pattern it discovers is MEME (Multiple Expectation maximization for Motif Elicitation). MEME characterizes motifs as position-dependent probability matrices. The probability of each possible letter occurring at each possible position in the pattern is given in the matrices. Single MEME motifs do not contain gaps and therefore patterns with gaps will be divided by the program into two or more separate motifs.

The program takes the group of DNA or protein sequences provided by the user and creates a number of motifs. The user can choose the number of motifs that MEME will produce. MEME uses statistical techniques to choose the best width,

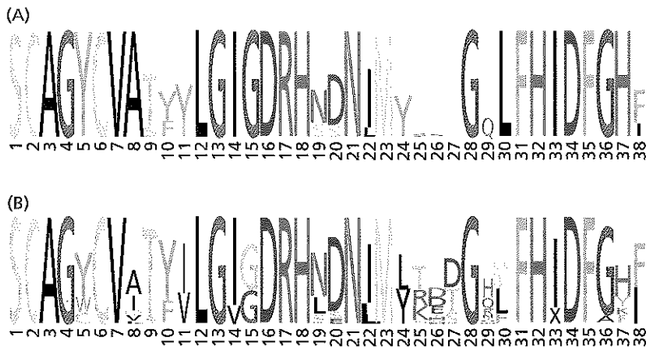


Figure 4.15
Residues that contribute to one of the blocks returned by the BLOCKS database after submission of the PI3-kinase p100α sequence.
(A) A block for four homologous sequences, and (B) for 31 homologous sequences. These representations are called logos, and are computed using a position-specific scoring matrix. This block contains the active-site amino acids and the DFG kinase motif. The size of the letters indicates the level of conservation and the colors indicate physicochemical properties of the residues: acidic, red; basic, blue; small and polar, white; asparagine and glutamine, green; sulfur-containing amino acids, yellow; hydrophobic, black; proline, purple; glycine, gray; aromatic, orange.

number of occurrences, and description for each motif (see Section 6.6). The motifs found by MEME can also be given in BLOCKS format to allow further analysis as described below.

If we submit the PI3-kinase p110α sequence and four homologs to the BLOCKS program it creates six separate blocks of high similarity. Figure 4.15A illustrates the block that contains residues important in PI3-kinase catalysis and ATP binding. Letters that are large and occupy the whole position represent identities in the multiple sequence alignment (see Section 6.1 for further details on this sequence representation). The SCAGY, DRH, and DFG motifs that are the markers for PI3-kinases are identified by the BLOCKS program and form part of the conserved regions. If more distant sequences are submitted, fewer residues will form the highly conserved regions with the largest residues, as shown in Figure 4.15B where 31 sequences were aligned.

The six blocks can now be submitted to a database search using the program LAMA (Local Alignment of Multiple Alignments), which compares multiple alignments of protein sequences with each other. The program searches the BLOCKS database, the PRINTS database (see below), or your own target data, to see if similar blocks or patterns already exist. This is a sensitive search technique, detecting weak sequence relationships between protein families. The LAMA search of the BLOCKS database has identified seven blocks, of which three are significant: these are PI3/4-kinase signatures.

The blocks can also be submitted to a MAST (Motif Alignment and Search Tool) search of one of the online nucleotide- or protein-sequence databases. MAST is a program that searches for motifs—highly conserved regions or blocks. Here we submit the six PI3-kinase blocks to a MAST search of SWISS-PROT (to use this program you need to have an e-mail address to receive the results). Twenty-four sequences were found with significant scores, with the PI3-kinase sequences all scoring more highly than the PI4-kinases.

The same set of PI3-kinase p110α sequences was submitted to the MEME motif-generating program. The number of motifs to be generated was set to six (the same number found by BLOCKS). The top-scoring motif (see Figure 4.16A) describes similar residues as the BLOCK motif described above (see Figure 4.15). The MEME motif starts at the end of the SCAGY motif (Y), contains the active site D, N, and DFG residues, and extends a bit further than the BLOCKS motif. A nice feature of the MEME program is that it generates a figure containing a summary of all motifs (see Figure 4.16B). This illustrates where the motifs are located with respect to each other within all the sequences (only three are shown for clarity). Submitting the MEME motifs to a search through SWISS-PROT finds 21 matches. Matches with significant

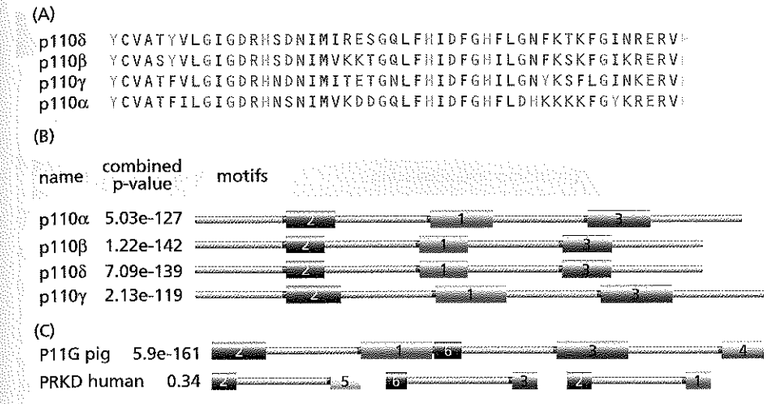


Figure 4.16
MEME generates motifs. (A) The top-scoring patterns are color coded according to the physicochemical properties of the amino acid side chains: dark blue is used for the residues ACFILVM; green for NQST; magenta to indicate DE; red color is used for residues KR; pink for H; orange for G; yellow for P; and light blue shows Y. (B) Summary motif information where each motif is represented by a colored block. The number in a block gives the scored position of the motif. The light blue block, number 1, contains the motif described in (A). The combined p-value of a sequence measures the strength of the match of the sequence to all the motifs. (C) Illustration of how lower-scoring motif matches can still find interesting and true homologs. The distances between the motif blocks are not representative.

scores are all PI3-kinases and PI4-kinases. The significant scores usually match most, if not all, of the motifs submitted. However, lower scores can be informative as well; distant relationships can be found if only a subset of the motifs matches.

For example, a search using the motifs of PI3-kinases finds the DNA-dependent protein kinase catalytic subunit (PRKD), which has shared kinase activity with the PI3-kinases. Four of the six motifs are matched (Figure 4.16C) and some are repeated within the DNA-dependent kinase. Simple sequence-alignment searches through the sequence databases may not pick up this type of relationship, although in this case a blastp search with p110α through SWISS-PROT matches PRKD with a score that would be considered significant, and a search with FASTA gives a borderline score.

4.9 Searching Using Motifs and Patterns

The PROSITE database can be searched for protein motifs and patterns

The PROSITE database is a compilation of motifs and patterns extracted from protein sequences and compiled by inspection of protein families. This database can be searched with an unknown protein sequence to obtain a list of hits to possible patterns or protein signatures. It is also possible to create your own pattern in the manner of a PROSITE pattern to search another sequence database. The syntax of a PROSITE pattern consists of amino acid residues interspersed with characters that denote the rules for the pattern, such as distances between residues, and so on (see Table 4.2).

For example, a pattern for the kinase active site, starting from the conserved DRH and making use of the very conserved DFG region, can be created manually from the 31 sequences used in the BLOCKS example.

D-R-[KH]-X-[DE]-N-[IL]-[MILV](2)-X(3)-G-X-[LI]-X(3)-D-F-G

Inputting this pattern into the ScanProsite Web page and running it against the SWISS-PROT database of protein sequences obtained 92 hits; all were PI3 (PI4)-kinases or protein kinases. If, on the other hand, we submit the catalytic domain of the PI3-kinase p110α sequence to be scanned through the PROSITE database to see if there are any existing patterns, the search retrieves two signature

Code	Explanation	Example explanation	Examples
One-letter codes	Standard amino acids		G-L-L-M-S-A-D-F-F-F
-	All positions must be separated by -		G-L-L-M-S-A-D-F-F-F
X	Any amino acid	Any amino acid allowed in second place	G-X-L-M-S-A-D-F-F-F
[]	Two or more possible amino acids	L or I allowed in second place	G-[LI]-L-M-S-A-D-F-F-F
{}	Disallowed amino acids	R or K not allowed in sixth place	G-[LI]-L-M-S-A-{RK}-F-F-F
(n) n = number	Repetition can be indicated by a number in brackets after the amino acid	F repeated three times	G-[LI]-L-M-S-A-{RK}-F(3)
(n,m)	A range: only allowed with X	One to three positions with any amino acids (X) allowed	G-[LI]-L-M-S-A-{RK}-X(1,3)
<	Pattern at amino-terminal of sequence		
>	Pattern at carboxy-terminal of sequence		

Table 4.2
The various codes used to define a PROSITE protein pattern for a search through a sequence database.

sequences for PI3- and PI4-kinases. These give us a much more specific search signature for the PI3/4-kinase family, but do not tell us, for example, that this kinase family is also similar to the protein kinase family. The patterns for the signatures are:

(1) [LIVMFAC]-K-X(1,3)-[DEA]-[DE]-[LIVMC]-R-Q-[DE]-X(4)-Q

(2) [GS]-X-[AV]-X(3)-[LIVM]-X(2)-[FYH]-[LIVM](2)-X-[LIVMF]-X- D-R-H-X(2)-N

The second signature pattern contains at its right-hand end (underlined) the start of the kinase pattern we created above to scan SWISS-PROT. Pattern 1 and the rest of pattern 2 contain conserved regions within the PI3/4-kinase families that are amino-terminal to our created pattern.

The pattern-based program PHI-BLAST searches for both homology and matching motifs

The BLAST set of programs also has a version that uses motifs in the query sequence as a pattern. PHI-BLAST (Pattern Hit Initiated BLAST) uses the PROSITE pattern syntax shown in Table 4.2 to describe the query protein motif. The specified pattern need not be in the PROSITE database and can be user generated. PHI-BLAST looks for sequences that not only contain the query-specific pattern but are also homologous to the query sequence near the designated pattern. Because PHI-BLAST uses homology as well as motif matching, it generally filters out those sequences where the pattern may have occurred at random. On the NCBI Web server, PHI-BLAST is integrated with PSI-BLAST, enabling one or more subsequent PSI-BLAST database searches using the PHI-BLAST results.

Patterns can be generated from multiple sequences using PRATT

The program PRATT can be used to extract patterns conserved in sets of unaligned protein sequences. The patterns are described using the PROSITE syntax. The power of PRATT is that it requires no knowledge of possible existing patterns in a set of sequences. Figure 4.17 shows the results for the PI3-kinase p110α family. The pattern illustrated in the figure contains the DFG motif which is highlighted in the second PROSITE pattern.


```
PRATT output :

   p110-a:   qlfhi DFGHFLDhkKkkFGykRERVPFVLTqDFLiViskGaQE ctktr
   p110-b:   qlfhi DFGHILGnfKskFGikRERVPFILTyDFIhViqqGkTG ntekf
   p110-d:   qlfhi DFGHFLGnfKtkFGinRERVPFILTyDFVhViqqGkTN nsekf
   p110-g:   nlfhi DFGHILGnyKsfLGinKERVPFVLTpDFLfVm--GtSG kktsp

D-F-G-H-[FI]-L-[DG]-x(2)-K-x(2)-[FL]-G-x(2)-[KR]-E-R-V-P-F-[IV]-L-T-x-D-F-[ILV]-x-V-x(1,3)-G-x-[QST]-[EGN]
```

Figure 4.17
PRATT pattern search. Sequences of the four types of PI3-kinase sequences (α, β, γ, and δ) have been submitted to PRATT to automatically create PROSITE-like patterns from a multiple alignment. This figure shows the alignment block and a PRATT-generated PROSITE pattern of the region that contains the DFG motif (shaded in green and boxed in red).

The PRINTS database consists of fingerprints representing sets of conserved motifs that describe a protein family

The PRINTS database is a next-generation pattern database consisting of fingerprints representing sets of conserved motifs that describe a protein family. The fingerprint is used to predict the occurrence of similar motifs, either in an individual sequence or in a database. The fingerprints were refined by iterative scanning of the OWL composite sequence database: a composite, nonredundant database assembled from sources including SWISS-PROT, sequences extracted from NBRF/PIR protein sequence database, translated sequences from GenBank, and the PDB structural database. A composite, or multiple-motif, fingerprint contains a number of aligned motifs taken from different parts of a multiple alignment. True family members are then easy to identify by virtue of possessing all the elements of the fingerprint; possession of only part of the fingerprint may identify subfamily members. A search of the PRINTS database with our PI3-kinase sequence found no statistically significant results.

The Pfam database defines profiles of protein families

Pfam is a collection of protein families described in a more complex way than is allowed by PROSITE's pattern syntax. It contains a large collection of multiple sequence alignments of protein domains or conserved regions. Hidden Markov model (HMM)-based profiles (see Section 6.2) are used to represent these Pfam families and to construct their multiple alignments. Searching the Pfam database involves scanning the query sequence against each of these HMM profiles. Using these methods, a new protein can often be assigned to a protein family even if the sequence homology is weak. Pfam includes a high proportion of extracellular protein domains. In contrast, the PROSITE collection emphasizes domains in intracellular proteins—proteins involved in signal transduction, DNA repair, cell-cycle regulation, and apoptosis—although there is some overlap. A search of the Pfam database allows you to look at multiple alignments of the matched family, view protein domain organization (see Figure 4.6), follow links to other databases by clicking on the boxed areas, and view known protein structures.

A search in Pfam using the sequence of the PI3-kinase p110α catalytic domain will find the PI3/4-kinase family. You can then retrieve the multiple alignment that has been used to define the family and obtain a diagram of the domain structure of the whole family. (Clicking on a domain will call up another Web page of detailed information.) Figure 4.6 shows a snapshot of the interactive diagram; the yellow boxed area is the catalytic domain upon which the search was based.

Only the most commonly used pattern and profile databases have been described here; links to others are given on the Publisher's Web page.

4.10 Patterns and Protein Function

Searches can be made for particular functional sites in proteins

There are techniques other than simple sequence comparison that can identify functional sites in protein sequences. In contrast to the methods discussed above,

which tend to cover a very wide range of biological functions, these techniques are usually made available in programs which predict only one specific functional site.

For example, signals from the environment are transmitted to the inside of the cell where they induce biochemical reaction cascades called signal transduction pathways. These result in responses such as cell division, proliferation, and differentiation and, if not properly regulated, cancer. During signal transduction, cellular components are chemically modified, often transiently. One of the key modifications used in these pathways is the addition and removal of phosphate groups. Sites susceptible to such modification can be predicted by the NetPhos server, which uses neural network methods to predict serine, threonine, and tyrosine phosphorylation sites in eukaryotic proteins. PROSITE also has patterns describing sites for phosphorylation and other posttranslational modifications, but specific programs such as NetPhos are expected to be more accurate.

Sequence comparison is not the only way of analyzing protein sequences

Apart from sequence comparison and alignment methods, there are various other ways of analyzing protein sequences to detect possible functional features. These techniques can be useful either when you have found a homolog in a database search and want to analyze it further, or when you have failed to find any similar sequence homolog and have no other avenue open. The physicochemical properties of amino acids, such as polarity, can be useful indicators of structural and functional features (see Chapter 2). There are programs available on the Web that plot hydrophobicity profiles, the percentage of residues predicted to be buried, some secondary-structure prediction (see Chapters 11 and 12), and percentage accessibility. ProtScale is one easy-to-use Web site that allows many of the above protein properties to be plotted.

Hydrophobic cluster analysis (HCA) is a protein-sequence comparison method based on α-helical representations of the sequences, where the size, shape, and orientation of the clusters of hydrophobic residues are compared. Hydrophobic cluster analysis can be useful for comparing possible functions of proteins of very low sequence similarity. It can also be used to align protein sequences. The patterns generated by HCA via the online tool drawhca can be compared with any other sequences one is interested in. It has been suggested that the effectiveness of HCA

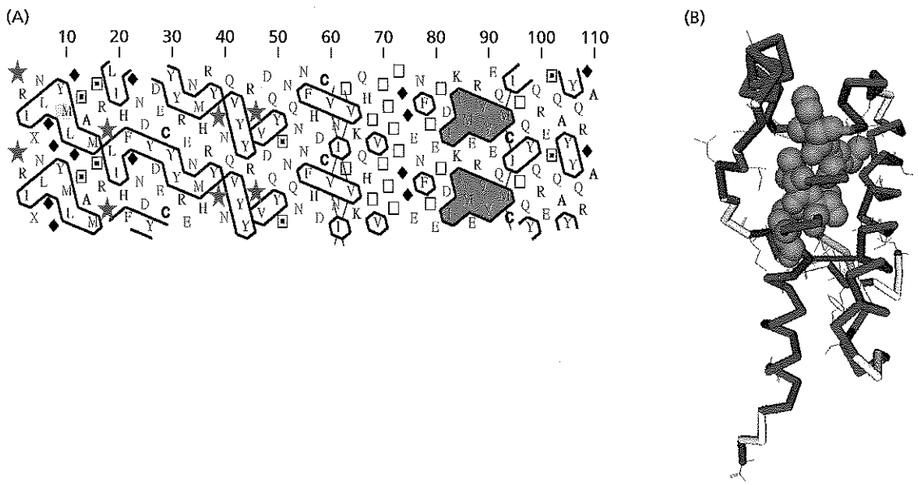


Figure 4.18
(A) Hydrophobic cluster analysis (HCA) of the prion protein using drawhca. Hydrophobic residues are in green, acidic in red, and basic in blue. A star indicates proline, a diamond glycine, an open box threonine, and a box with a dot serine. The same types of residues tend to cluster together, forming hydrophobic or charged patches. One such patch is highlighted in magenta. (B) X-ray structure of the same protein, with the same residues highlighted in magenta. As shown by the X-ray structure, the patch found by HCA forms a hydrophobic core in the interior of the protein.

Box 4.7 Protein localization signals

Proteins are all synthesized on ribosomes in the cytosol but, in eukaryotic cells in particular, have numerous final destinations: the cell membrane, particular organelles, or secretion from the cell. Intrinsic localization signals in the protein itself help to direct it to its destination and these can often be detected by their sequence characteristics. Proteins sorted through the endoplasmic reticulum (ER) for secretion or delivery to the cell membrane and some other organelles usually have a characteristic signal sequence at the amino-terminal end. This interacts with transport machinery in the ER membrane, which delivers the protein into the ER membrane or into the lumen. The signal sequence is often subsequently removed. Signal sequences are characterized by an amino-terminal basic region and a central hydrophobic region, and these features are used to predict their presence.

for comparison originates from its ability to focus on the residues forming the hydrophobic cores of globular proteins. Figure 4.18 shows the prion protein patterns that were generated using the program drawhca.

Information about the possible location of proteins in the cell can sometimes be obtained by sequence analysis. Membrane proteins and proteins destined for organelles such as the endoplasmic reticulum and nucleus contain intrinsic sequence motifs that are involved in their localization. Most secreted proteins, for example and other proteins that enter the endoplasmic reticulum protein-sorting pathway, contain sequences known as signal sequences when they are newly synthesized (see Box 4.7). The PSORT group of programs predicts the presence of signal sequences by looking for a basic region at the amino-terminal end of the protein sequence followed by a hydrophobic region. A score is calculated on the basis of the length of the hydrophobic region, its peak value, and the net charge of the amino-terminal region. A large positive score means that there is a high possibility that the protein contains a signal sequence. More methods of analyzing protein sequences to deduce structure and function are described in Chapters 11 to 14.

Summary

The comparison of different DNA or protein sequences to detect sequence similarity and evolutionary homology is carried out by a process known as sequence alignment. This involves lining up two or more sequences in such a way that the similarities between them are optimized, and then measuring the degree of matching. Alignment is used to find known genes or proteins with sequence similarity to a novel uncharacterized sequence, and forms the basis of programs such as BLAST and FASTA that are used to search sequence databases. Similarities in sequence can help make predictions about a protein's structure and function. Sequences of proteins or DNAs from different organisms can be compared to construct phylogenetic trees, which trace the evolutionary relationships between species or within a family of proteins.

The degree of matching in an alignment is measured by giving the alignment a numerical score, which can be arrived at in several different ways. The simplest scoring method is percentage identity, which counts only the number of matched identical residues, but this relatively crude score will not pick up sequences that are only distantly related. Other scoring methods for protein sequences take into account the likelihood that a given type of amino acid will be substituted for another during evolution, and these methods give pairs of aligned amino acids numerical scores which are summed to obtain a score for the alignment. The probabilities are obtained from reference substitution matrices, which have been compiled from the

analysis of alignments of known homologous proteins. Because insertions or deletions often occur as two sequences diverge during evolution, gaps must usually be inserted in either sequence to maximize matching, and scoring schemes exact a penalty for each gap. As there are 20 amino acids, compared to only four different nucleotides, it is easier to detect homology in alignments of protein sequences than in nucleic acid sequences since chance matches are less likely.

There are several different types of alignment. Global alignments estimate the similarity over the whole sequence, whereas local alignments look for short regions of similarity. Local alignments are particularly useful when comparing multidomain proteins, which may have only one domain in common. Multiple alignments compare a set of similar sequences simultaneously and are more accurate and more powerful than pairwise alignments in detecting proteins with only distant homology to each other.

Algorithms that automate the alignment and scoring process have been devised and are incorporated into various programs. Once an alignment has been scored, the significance of the score has to be tested to determine the likelihood of its arising by chance. Factors such as the length of the alignment and the total number of sequences in the database are taken into account. In database search programs such as BLAST and FASTA, potential matches are evaluated automatically and given a significance score, the *E*-value.

Databases may also be searched to find proteins of similar structure or function by looking for conserved short sequence motifs or discontinuous patterns of residues. These are likely to relate to a functional feature, such as an active site, a binding site, or to structural features. When sufficient members of a protein family have been sequenced, a characteristic profile of the family, summarizing the most typical sequence features, can be derived and can be used to search for additional family members. Database searches can also be widened to include structural information, where available. This is useful for finding homologs which have diverged so much in sequence that their sequence similarity can no longer be detected, but which retain the same overall structure.

Further Reading

4.1 Principles of Sequence Alignment

Bignall G, Micklem G, Stratton MR et al. (1997) The BRC repeats are conserved in mammalian BRCA2 proteins. *Hum. Mol. Genet.* 6, 53–58.

Brenner SE, Chothia C & Hubbard TJP (1998) Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl Acad. Sci. USA* 95, 6073–6078.

Durbin R, Eddy S, Krogh A & Mitchison G (1998) Biological Sequence Analysis. Cambridge: Cambridge University Press.

Higgins D & Taylor W (eds) (2000) Bioinformatics. Sequence, Structure and Databanks, chapters 3–5, 7. Oxford: Oxford University Press.

Shivji MKK, Davies OR, Savill JM et al. (2006) A region of human BRCA2 containing multiple BRC repeats promotes RAD51-mediated strand exchange. *Nucleic Acids Res.* 34, 4000–4011.

4.2 Scoring Alignments

Twilight zone and midnight zone

Doolittle RF (1986) Of URFs and ORFs: A Primer on How to Analyze Derived Amino Acid Sequences. Mill Valley, CA: University Science Books.

Doolittle RF (1994) Convergent evolution: the need to be explicit. *Trends Biochem. Sci.* 19, 15–18.

Rost B (1999) Twilight zone of protein sequence alignments. *Protein Eng.* 12, 85–94.

4.3 Substitution Matrices

Dayhoff MO, Schwartz RM & Orcutt BC (1978) A model of evolutionary change in proteins. In Atlas of Protein Sequence and Structure (MO Dayhoff ed.), vol. 5, suppl. 3, pp. 345–352. Washington, DC: National Biomedical Research Foundation.

Henikoff S & Henikoff JG (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA* 89, 10915–10919.

4.4 Inserting Gaps

Goonesekere NCW & Lee B (2004) Frequency of gaps observed in a structurally aligned protein pair database suggests a simple gap penalty function. *Nucleic Acids Res.* 32, 2838–2843.

4.5 Types of Alignment

Gotoh O (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol.* 162, 705–708.

Needleman SB & Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48, 443–453.

Smith TF & Waterman MS (1981) Identification of common molecular subsequences. *J. Mol. Biol.* 147, 195–197.

ClustalW

Higgins DW, Thompson JD & Gibson TJ (1996) Using CLUSTAL for multiple sequence alignments. *Methods Enzymol.* 266, 383–402.

DIALIGN

Morgenstern B (1999) DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics* 15, 211–218.

Morgenstern B, Frech K, Dress A & Werner T (1998) DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics* 14, 290–294.

4.6 Searching Databases

BLAST

Altschul SF, Gish W, Miller W et al. (1990) Basic Local Alignment Search Tool. *J. Mol. Biol.* 215, 403–410.

FASTA

Pearson WR & Lipman DJ (1988) Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA* 85, 2444–2448.

PSI-BLAST

Altschul SF, Madden TL, Schäffer AA et al. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402.

4.8 Protein Sequence Motifs or Patterns

MEME

Bailey TL & Elkan C (1995) Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Mach. Learn.* 21, 51–80.

4.9 Searching Using Motifs and Patterns

MOTIF

Smith HO, Annau TM & Chandrasegaran S (1990) Finding sequence motifs in groups of functionally related proteins. *Proc. Natl Acad. Sci. USA* 87, 826–830.

PRATT

Jonassen I (1997) Efficient discovery of conserved patterns using a pattern graph. *Comput. Appl. Biosci.* 13, 509–522.

Jonassen I, Collins JF & Higgins DG (1995) Finding flexible patterns in unaligned protein sequences. *Protein Sci.* 4, 1587–1595.

4.10 Patterns and Protein Function

HCA

Lemesle-Varloot L, Henrissat B, Gaboriaud C et al. (1990) Hydrophobic cluster analysis: procedures to derive structural and functional information from 2-D-representation of protein sequences. *Biochimie* 72, 555–574.