Pairwise Sequence Alignment

CHAPTER 3

An accepted point mutation in a protein is a replacement of one amino acid by another, accepted by natural selection. It is the result of two distinct processes: the first is the occurrence of a mutation in the portion of the gene template producing one amino acid of a protein; the second is the acceptance of the mutation by the species as the new predominant form. To be accepted, the new amino acid usually must function in a way similar to the old one: chemical and physical similarities are found between the amino acids that are observed to interchange frequently.

—*Margaret Dayhoff (1978, p. 345)*

LEARNING OBJECTIVES

Upon completion of this chapter, you should be able to:

- define homology as well as orthologs and paralogs;
- explain how PAM (accepted point mutation) matrices are derived;
- contrast the utility of PAM and BLOSUM scoring matrices;
- define dynamic programming and explain how global (Needleman–Wunsch) and local (Smith–Waterman) pairwise alignments are performed; and
- perform pairwise alignment of protein or DNA sequences at the NCBI website.

INTRODUCTION

One of the most basic questions about a gene or protein is whether it is related to any other gene or protein. Relatedness of two proteins at the sequence level suggests that they are homologous. Relatedness also suggests that they may have common functions. By analyzing many DNA and protein sequences, it is possible to identify domains or motifs that are shared among a group of molecules. These analyses of the relatedness of proteins and genes are accomplished by aligning sequences. As we complete the sequencing of the genomes of many organisms, the task of finding out how proteins are related within an organism and between organisms becomes increasingly fundamental to our understanding of life.

Two genes (or proteins) are homologous if they have evolved from a common ancestor.

In this chapter we introduce pairwise sequence alignment. We adopt an evolutionary perspective in our description of how amino acids (or nucleotides) in two sequences can be aligned and compared. We then describe algorithms and programs for pairwise alignment.

Bioinformatics and Functional Genomics, Third Edition, Jonathan Pevsner.
© 2015 John Wiley & Sons, Inc. Published 2015 by John Wiley & Sons, Inc.
Companion Website: www.wiley.com/go/pevsnerbioinformatics

Protein Alignment: Often More Informative than DNA Alignment

Given the choice of aligning a DNA sequence or the sequence of the protein it encodes, it is often more informative to compare protein sequence. There are several reasons for this. Many changes in a DNA sequence (particularly at the third position of a codon) do not change the amino acid that is specified. Furthermore, many amino acids share related biophysical properties (e.g., lysine and arginine are both basic amino acids). The important relationships between related (but mismatched) amino acids in an alignment can be accounted for using scoring systems (described in this chapter). DNA sequences are less informative in this regard. Protein sequence comparisons can identify homologous sequences while the corresponding DNA sequence comparisons cannot (Pearson, 1996).

To see an example of this use human beta globin protein (NP_000509.1) in a DELTA-BLAST query against plant RefSeq proteins; we learn how to do this in Chapter 5. There are many dozens of significant matches. Perform a BLASTN search with the coding region of the corresponding DNA (NM_000518.4); there are no significant matches. When BLASTN is used to query DNA from organisms that last shared a common ancestor with humans more recently, such as fish, there are many significant matches.

When a nucleotide coding sequence is analyzed, it is often preferable to study its translated protein. In Chapter 4 (on BLAST searching), we see that we can move easily between the worlds of DNA and protein. For example, the TBLASTN tool from the NCBI BLAST website allows related proteins derived from a DNA database to be searched for with a protein sequence. This query option is accomplished by translating each DNA sequence into all of the six proteins that it potentially encodes.

Nevertheless, in many cases it is appropriate to compare nucleotide sequences. This comparison can be important in confirming the identity of a DNA sequence in a database search, in searching for polymorphisms, in analyzing the identity of a cloned cDNA fragment, in comparing regulatory regions, or in many other applications.

Definitions: Homology, Similarity, Identity

The website ⊕ http://timetree.org (WebLink 3.1) of Sudhir Kumar and colleagues provides estimates of the divergence times of species across the tree of life (Hedges *et al*., 2006).

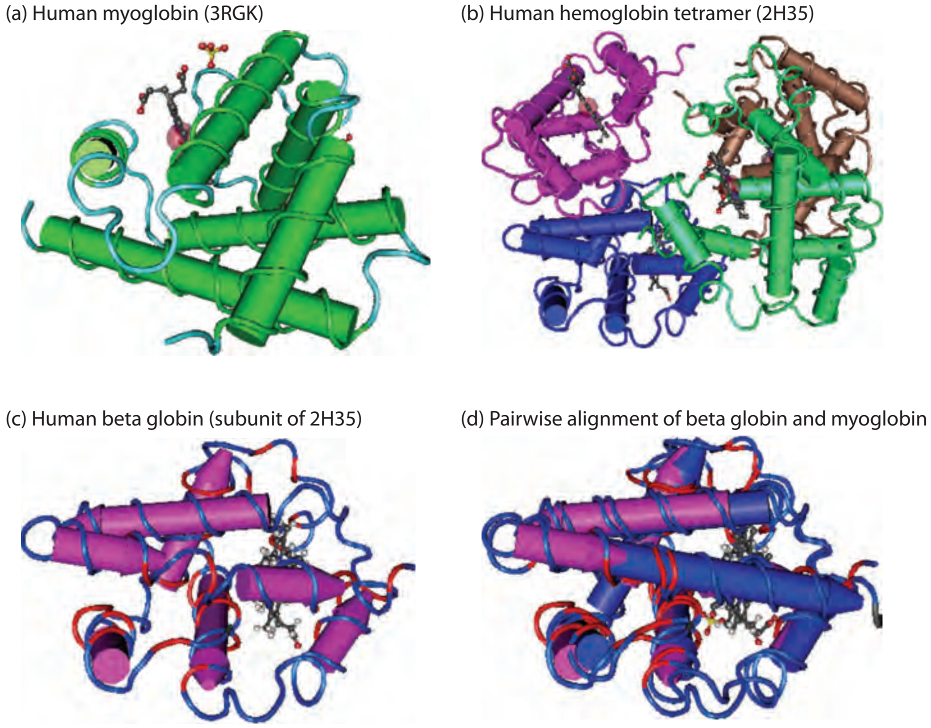
Let us consider the globin family of proteins. We begin with human myoglobin (accession number NP_005359.1) and beta globin (accession number NP_000509.1) as two proteins that are distantly but significantly related. The accession numbers are obtained from Gene at NCBI (Chapter 2). Myoglobin and the hemoglobin chains (alpha, beta, and other) are thought to have diverged some 450 million years ago, near the time that human and cartilagenous fish lineages diverged (**Fig. 19.22**).

Some researchers use the term *analogous* to refer to proteins that are not homologous but share some similarity by chance. Such proteins are presumed not to have descended from a common ancestor.

Two sequences are *homologous* if they share a common evolutionary ancestry. There are no degrees of homology; sequences are either homologous or not (Reeck *et al*., 1987; Tautz, 1998). Homologous proteins almost always share a significantly related three-dimensional structure. Myoglobin and beta globin have very similar structures, as determined by X-ray crystallography (**Fig. 3.1**). When two sequences are homologous, their amino acid or nucleotide sequences usually share significant identity. While homology is a qualitative inference (sequences are homologous or not), identity and similarity are quantities that describe the relatedness of sequences. Notably, two molecules may be homologous without sharing statistically significant amino acid (or nucleotide) identity. In the globin family for example, all the members are homologous but some have sequences that have diverged so greatly that they share no recognizable sequence identity (e.g., human beta globin and human neuroglobin share only 22% amino acid identity). Perutz and colleagues demonstrated that inviduals globin chains share the same overall shape as myoglobin, even though the myoglobin and alpha globin proteins share only about 26% amino acid identity. In general, three-dimensional structures diverge much more slowly than amino acid sequence identity between two proteins (Chothia and Lesk, 1986). Recognizing this type of homology is an especially challenging bioinformatics problem.

You can see the protein sequences used to generate **Figures 3.2** and **3.3** in Web Documents 3.1 and 3.2 and ⊕ http://www.bioinfbook.org/chapter3.

Proteins that are homologous may be orthologous or paralogous. *Orthologs* are homologous sequences in different species that arose from a common ancestral gene during speciation. **Figure 3.2** shows a tree of myoglobin orthologs. There is a human myoglobin gene and a rat gene. Humans and rodents diverged about 90 million years ago (MYA) (see Chapter 19), at which time a single ancestral myoglobin gene diverged by



(a) Human myoglobin (3RGK)

(b) Human hemoglobin tetramer (2H35)

(c) Human beta globin (subunit of 2H35)

(d) Pairwise alignment of beta globin and myoglobin

FIGURE 3.1 Three-dimensional structures of: (a) myoglobin (accession 3RGK); (b) the tetrameric hemoglobin protein (2H35); (c) the beta globin subunit of hemoglobin; and (d) myoglobin and beta globin superimposed. The images were generated with the program Cn3D (see Chapter 13). These proteins are homologous (descended from a common ancestor) and share very similar three-dimensional structures. However, pairwise alignment of the amino acid sequences of these proteins reveals that the proteins share very limited amino acid identity.

Source: Cn3D, NCBI.

speciation. Orthologs are presumed to have similar biological functions; in this example, human and rat myoglobins both transport oxygen in muscle cells. *Paralogs* are homologous sequences that arose by a mechanism such as gene duplication. For example, human alpha 1 globin (NP_000549.1) is paralogous to alpha 2 globin (NP_000508.1); indeed, these two proteins share 100% amino acid identity. Human alpha 1 globin and beta globin are also paralogs (as are all the proteins shown in **Fig. 3.3**). All of the globins have distinct properties, including regional distribution in the body, developmental timing of gene expression, and abundance. They are all thought to have distinct but related functions as oxygen carrier proteins.

The concept of homology has a rich history dating back to the nineteenth century (Box 3.1). Walter M. Fitch (1970, p. 113) provided our current definitions of these terms. He wrote that "there should be two subclasses of homology. Where the homology is the result of gene duplication so that both copies have descended side by side during the history of an organism (for example, α and β hemoglobin) the genes should be called paralogous (para = in parallel). Where the homology is the result of speciation so that the history of the gene reflects the history of the species (for example α hemoglobin in man and mouse) the genes should be called orthologous (ortho = exact)."

Notably, orthologs and paralogs do not necessarily have the same function. We provide various definitions of gene and protein function in Chapters 8–14. Later in the book, we explore genomes across the tree of life (Chapters 15–20). In all genome sequencing projects, orthologs and paralogs are identified based on database searches. Two DNA (or protein) sequences are defined as homologous based on achieving significant alignment

In general, when we consider other paralogous families they are presumed to share common functions. Consider the lipocalins: all are about 20 kilodalton proteins that have a hydrophobic binding pocket that is thought to be used to transport a hydrophobic ligand. Members include retinol binding protein (a retinol transporter), apolipoprotein D (a cholesterol transporter), and odorant-binding protein (an odorant transporter secreted from the lateral nasal gland).

We therefore define homologous genes within the same organism as paralogous. But consider further the case of globins. Human α-globin and β-globin are paralogs, as are mouse α-globin and mouse β-globin. Human α-globin and mouse α-globin are orthologs. What is the relation of human α-globin to mouse β-globin? These could be considered paralogs, because α-globin and β-globin originate from a gene duplication event rather than from a speciation event. However, they are not paralogs because they do not occur in the same species. It may therefore be more appropriate to simply call them "homologs," reflecting their descent from a common ancestor. Fitch (1970, p. 113) notes that phylogenies require the study of orthologs (see also Chapter 7).

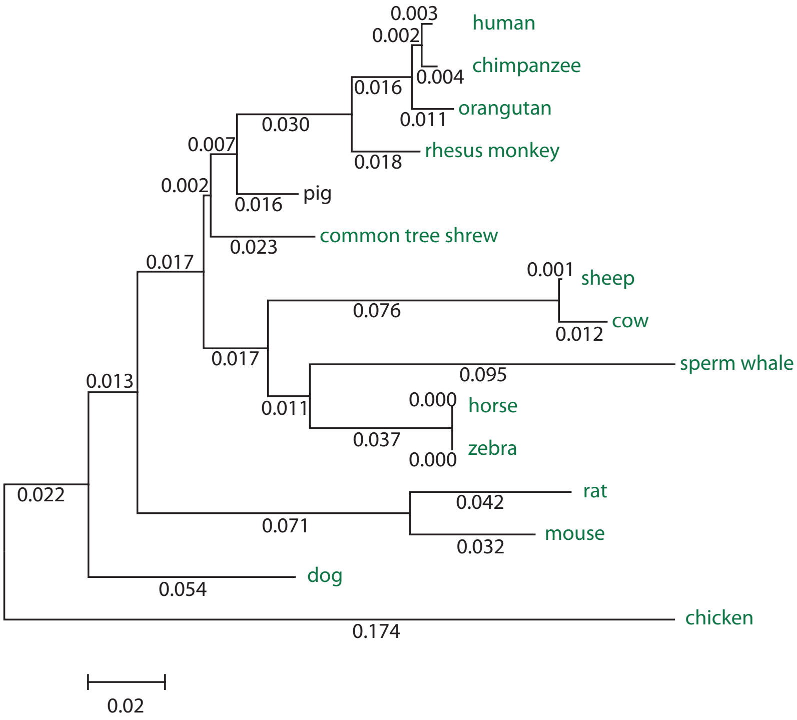


FIGURE 3.2 A group of myoglobin orthologs, visualized by multiply aligning the sequences (Chapter 6) then creating a phylogenetic tree by neighbor-joining (Chapter 7). The accession numbers and species names are as follows: human, NP_005359 (*Homo sapiens*); chimpanzee, XP_001156591 (*Pan troglodytes*); orangutan, P02148 (*Pongo pygmaeus*); rhesus monkey, XP_001082347 (*Macaca mulatta*); pig, NP_999401 (*Sus scrofa*); common tree shrew, P02165 (*Tupaia glis*); horse, P68082 (*Equus caballus*); zebra, P68083 (*Equus burchellii*); dog, XP_850735 (*Canis familiaris*); sperm whale, P02185 (*Physeter catodon*); sheep, P02190 (*Ovis aries*); rat, NP_067599 (*Rattus norvegicus*); mouse, NP_038621 (*Mus musculus*); cow, NP_776306 (*Bos taurus*); chicken_XP_416292 (*Gallus gallus*). The sequences are shown in Web Document 3.1 (🌐 http://www.bioinfbook.org/chapter3). In this tree, sequences that are more closely related to each other are grouped closer together. Note that as entire genomes continue to be sequenced (Chapters 15–20), the number of known orthologs will grow rapidly for most families of orthologous proteins.

scores, as discussed below and in Chapter 4. However, some homologous proteins have entirely distinct functions.

We can assess the relatedness of any two proteins by performing a *pairwise alignment*. In this procedure, we place the two sequences directly next to each other. One

BOX 3.1 A HISTORY OF HOMOLOGY

Richard Owen (1804–1892) was one of the first biologists to use the term homology. He defined homology as “the same organ in different animals under every variety of form and function” (Owen, 1843, p. 379). Charles Darwin (1809–1882) also discussed homology in the 6th edition of *The Origin of Species or The Preservation of Favoured Races in the Struggle for Life* (1872). He wrote:

> That relation between parts which results from their development from corresponding embryonic parts, either in different animals, as in the case of the arm of man, the foreleg of a quadruped, and the wing of a bird; or in the same individual, as in the case of the fore and hind legs in quadrupeds, and the segments or rings and their appendages of which the body of a worm, a centipede, &c., is composed. The latter is called serial homology. The parts which stand in such a relation to each other are said to be homologous, and one such part or organ is called the homologue of the other. In different plants the parts of the flower are homologous, and in general these parts are regarded as homologous with leaves.

For a review of the history of the concept of homology see Hossfeld and Olsson (2005).

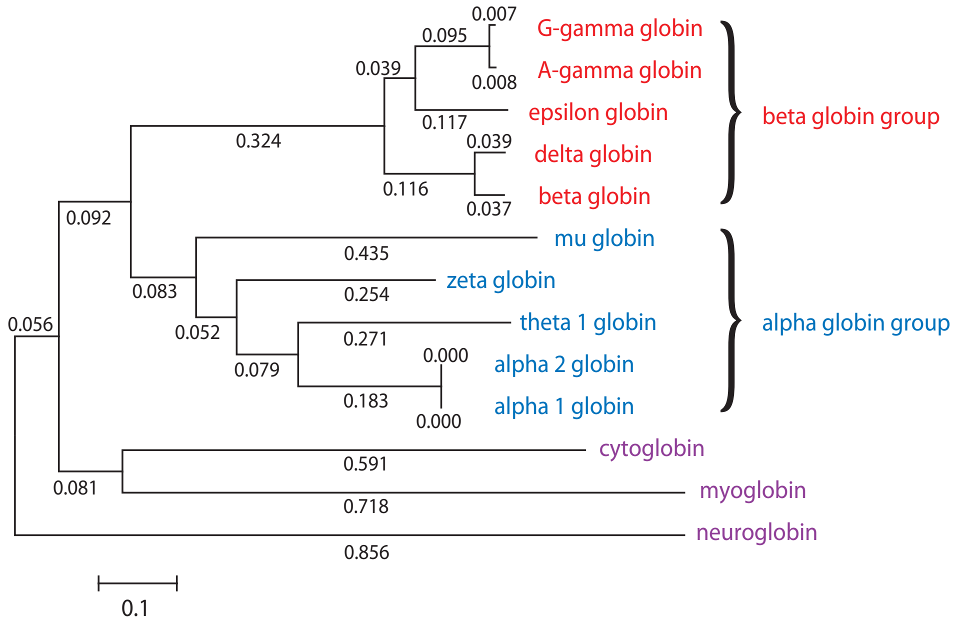


FIGURE 3.3 Paralogous human globins: Each of these proteins is human, and each is a member of the globin family. This unrooted tree was generated using the neighbor-joining algorithm in MEGA (see Chapter 7). The proteins and their RefSeq accession numbers (also shown in Web Document 3.2) are delta globin (NP_000510), G-gamma globin (NP_000175),beta globin (NP_000509), A-gamma globin (NP_000550), epsilon globin (NP_005321), zeta globin (NP_005323), alpha 1 globin (NP_000549), alpha 2 globin (NP_000508), theta 1 globin (NP_005322), hemoglobin mu chain (NP_001003938), cytoglobin (NP_599030), myoglobin (NP_005359), and neuroglobin (NP_067080). A Poisson correction model was used (see Chapter 7).

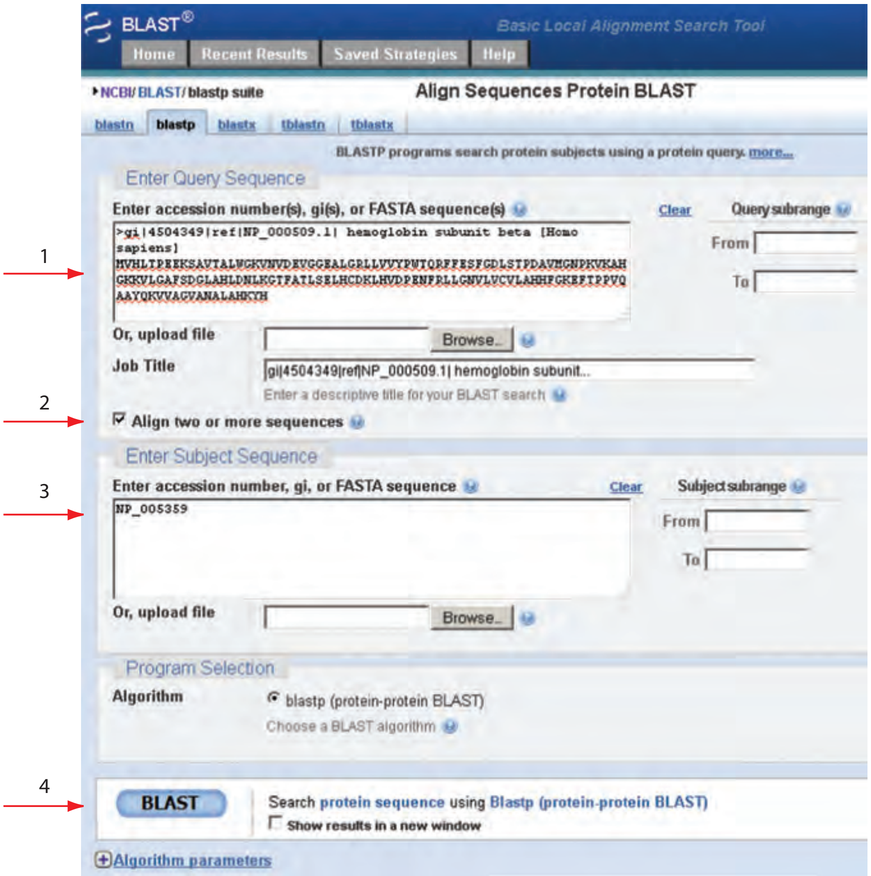
practical way to do this is through the NCBI BLASTP tool (for proteins) or BLASTN (for nucleotides) (Tatusova and Madden, 1999; **Fig. 3.4**). Perform the following steps:

1. Choose the program BLASTP (for "BLAST proteins") for our comparison of two proteins. Check the box "Align two or more sequences."
2. Enter the sequences or their accession numbers. Here we use the sequence of human beta globin in the FASTA format, and for myoglobin we use the accession number (**Fig. 3.4**).
3. Select any optional parameters.
   - You can choose from eight scoring matrices: BLOSUM90, BLOSUM80, BLOSUM62, BLOSUM50, BLOSUM45, PAM250, PAM70, PAM30. Select PAM250.
   - You can change the gap creation penalty and gap extension penalty.
   - For BLASTN searches you can change reward and penalty values.
   - There are other parameters you can change, such as word size, expect value, filtering, and dropoff values. We discuss these in more detail in Chapter 4.
4. Click "align." The output includes a pairwise alignment using the single-letter amino acid code (**Fig. 3.5a**).

The BLAST suite of programs is available at the NCBI site, 🌐 http://www.ncbi.nlm.nih.gov/BLAST/ (WebLink 3.2). We discuss various options for using the Basic Local Alignment Search Tool (BLAST) in Chapter 4.

Note that the FASTA format uses the single-letter amino acid code; those abbreviations are shown in Box 3.2.

It is impractical to align proteins by visual inspection. Also, if we allow gaps in the alignment to account for deletions or insertions in the two sequences, the number of possible alignments rises exponentially. Clearly, we need a computer algorithm to perform


1
2
3
4
BLAST®
Basic Local Alignment Search Tool
Home
Recent Results
Saved Strategies
Help
NCBI/ BLAST/ blastp suite
Align Sequences Protein BLAST
blastn
blastp
blastx
tblastn
tblastx
BLASTP programs search protein subjects using a protein query. more...
Enter Query Sequence
Clear
Query subrange
From
To
Or, upload file
Browse...
Enter a descriptive title for your BLAST search
Enter Subject Sequence
Clear
Subject subrange
From
To
Or, upload file
Browse...
Program Selection
Choose a BLAST algorithm
BLAST
Search protein sequence using Blastp (protein-protein BLAST)
□ Show results in a new window
Algorithm parameters

Enter accession number(s), gi(s), or FASTA sequence(s)

```
>gi|4504349|ref|NP_000509.1| hemoglobin subunit beta [Homo
sapiens]
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAH
GKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQ
AAYQKVVAGVANALAHKYH
```

Job Title: gi|4504349|ref|NP_000509.1| hemoglobin subunit...

☑ Align two or more sequences

Enter accession number, gi, or FASTA sequence

```
NP_005359
```

Algorithm: ● blastp (protein-protein BLAST)

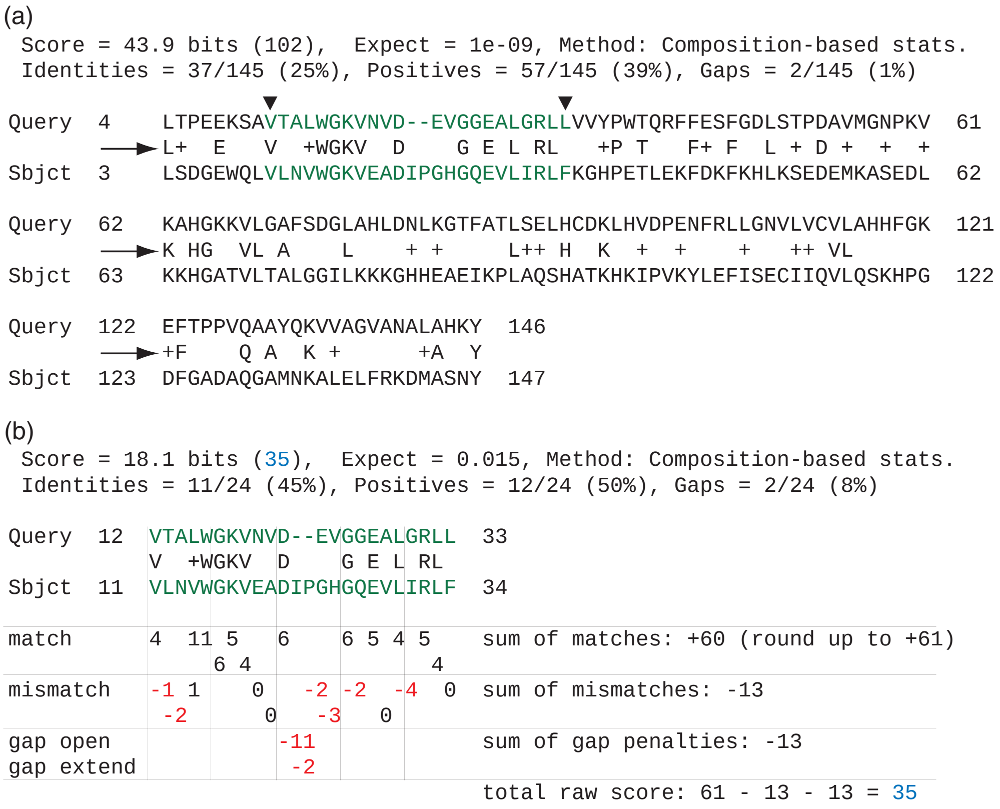
FIGURE 3.4 The BLAST tools at the NCBI website allow the comparison of two DNA or protein sequences. Here the program is set to BLASTP for the comparison of two proteins (arrow 2). Human beta globin (NP_000509) is input in the FASTA format (arrow 1), while human myoglobin (NP_005359) is input as an accession number (arrow 3). Click BLAST to start the search (arrow 4), and note the option at bottom left to view and adjust the algorithm parameters.

Source: BLAST, NCBI.

We discuss global and local alignments in the section "Alighment Algorithms: Global and Local".

an alignment (see Box 3.3). In the pairwise alignments shown in **Figure 3.5a**, beta globin is on top (on the line labeled "query") and myoglobin is below (on the subject line). An intermediate row indicates the presence of *identical* amino acids in the alignment. For example, note that near the beginning of the alignment the residues WGKV are identical between the two proteins. We can count the total number of identical residues; in this case, the two proteins share 25% identity (37 of 145 aligned residues). Identity is the extent to which two amino acid (or nucleotide) sequences are invariant. Note that this particular alignment is called *local* because only a subset of the two proteins is aligned: the first and last few amino acid residues of each protein are not displayed. A global pairwise alignment includes all residues of both sequences.

Another aspect of this pairwise alignment is that some of the aligned residues are similar but not identical; they are related to each other because they share similar biochemical properties. *Similar* pairs of residues are structurally or functionally related. For example, on the first row of the alignment we can find threonine and serine (T and S connected by a + sign in **Fig. 3.5a**); nearby we can see a leucine and a valine residue that are aligned. These are *conservative substitutions*. Amino acids with similar properties include the basic amino acids (K, R, H), acidic amino acids (D, E), hydroxylated amino acids (S, T), and hydrophobic amino acids (W, F, Y, L, I, V, M, A). Later in this chapter we will see how scores are assigned to aligned amino acid residues. In the pairwise alignment



(a)

```
 Score = 43.9 bits (102),  Expect = 1e-09, Method: Composition-based stats.
 Identities = 37/145 (25%), Positives = 57/145 (39%), Gaps = 2/145 (1%)

Query  4     LTPEEKSAVTALWGKVNVD--EVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKV  61
       ───►L+  E   V  +WGKV  D    G E L RL   +P T   F+ F  L + D +  +  +
Sbjct  3     LSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDL  62

Query  62    KAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGK  121
       ───►K HG  VL A    L    + +     L++ H  K  +  +    +   ++ VL
Sbjct  63    KKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKHPG  122

Query  122   EFTPPVQAAYQKVVAGVANALAHKY  146
       ───►+F    Q A  K +      +A  Y
Sbjct  123   DFGADAQGAMNKALELFRKDMASNY  147
```

(b)

```
 Score = 18.1 bits (35),  Expect = 0.015, Method: Composition-based stats.
 Identities = 11/24 (45%), Positives = 12/24 (50%), Gaps = 2/24 (8%)

Query  12  VTALWGKVNVD--EVGGEALGRLL  33
           V  +WGKV  D    G E L RL
Sbjct  11  VLNVWGKVEADIPGHGQEVLIRLF  34

match      4  11 5   6    6 5 4 5    sum of matches: +60 (round up to +61)
                6 4                4
mismatch   -1 1    0   -2 -2  -4  0   sum of mismatches: -13
            -2      0   -3   0
gap open             -11              sum of gap penalties: -13
gap extend            -2
                                      total raw score: 61 - 13 - 13 = 35
```

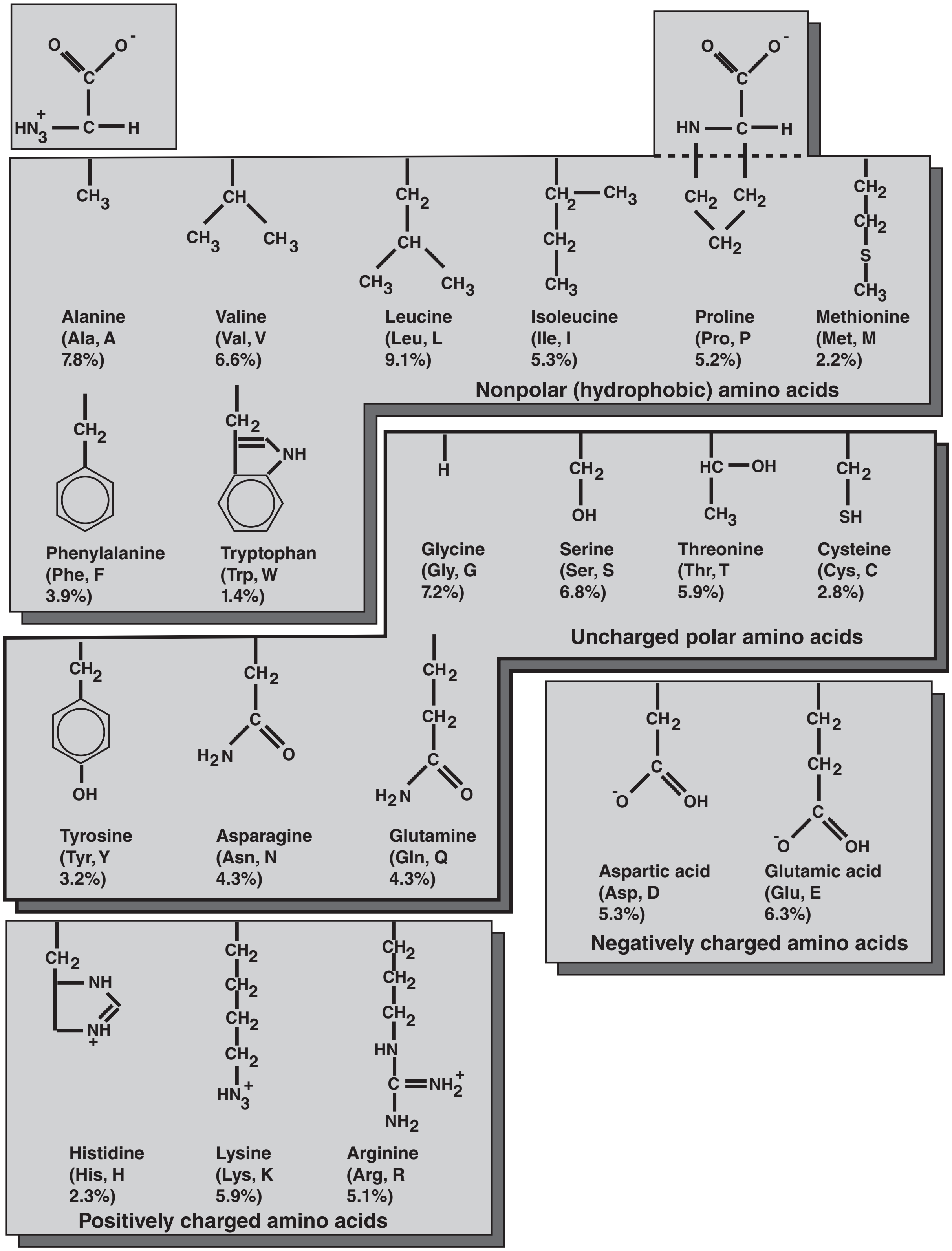
FIGURE 3.5 Pairwise alignment of human beta globin (the "query") and myoglobin (the "subject"). (a) The alignment from the search shown in **Figure 3.4**. Note that this alignment is local (i.e., the entire lengths of each protein are not compared), and there are many positions of identity between the two sequences (indicated with amino acids intervening between the query and subject lines; see rows with arrows). The alignment contains an internal gap (indicated by two dashes). (b) Illustration of how raw scores are calculated, using the result of a separate search with just amino acids 12–33 of HBB (corresponding to the region with green shaded letters between the arrowheads in (a). The raw score is 35, rounded up to 36; this represents the sum of the match scores (from a BLOSUM62 matrix in this case), the mismatch scores, the gap opening penalty (set to −11 for this search), and the gap extension penalty (set to −1). Raw scores are subsequently converted to bit scores.

of a segment of HBB and myoglobin, you can see that each pair of residues is assigned a score that is relatively high for matches and often negative for mismatches.

The *percent similarity* of two protein sequences is the sum of both identical and similar matches. In **Figure 3.5a**, there are 57 aligned amino acid residues which are similar. In general, it is more useful to consider the identity shared by two protein sequences rather than the similarity, because the similarity measure may be based upon a variety of definitions of how related (similar) two amino acid residues are to each other.

In summary, pairwise alignment is the process of lining up two sequences to achieve maximal levels of identity (and maximal levels of conservation in the case of amino acid alignments). The purpose of a pairwise alignment is to assess the degree of similarity and the possibility of homology between two molecules. For example, we may say that two proteins share 25% amino acid identity and 39% similarity. If the amount of sequence identity is sufficient, then the two sequences are probably homologous. It is never correct to say that two proteins share a certain percent homology, because they are either homologous or not. Similarly, it is not appropriate to describe two sequences as "highly homologous;" instead, it can be said that they share a high degree of similiarity.

BOX 3.2 STRUCTURES AND ONE- AND THREE-LETTER ABBREVIATIONS OF 20 COMMON AMINO ACIDS



It is very helpful to memorize these abbreviations and to become familiar with the physical properties of the amino acids. The percentages refer to the relative abundance of each amino acid in proteins.

BOX 3.3 ALGORITHMS AND PROGRAMS

An *algorithm* is a procedure that is structured in a computer program (Sedgewick, 1988). For example, there are many algorithms used for pairwise alignment. A computer *program* is a set of instructions that uses an algorithm (or multiple algorithms) to solve a task. For example, the BLAST program (Chapters 3–5) uses a set of algorithms to perform sequence alignments. Other programs that we will introduce in Chapter 7 use algorithms to generate phylogenetic trees.

Computer programs are essential to solve a variety of bioinformatics problems because millions of operations may need to be performed. The algorithm used by a program provides the means by which the operations of the program are automated. Throughout this book, note how many hundreds of programs have been developed using many hundreds of different algorithms. Each program and algorithm is designed to solve a specific task. An algorithm that is useful to compare one protein sequence to another may not work in a comparison of one sequence to a database of 10 million protein sequences.

Why might an algorithm that is useful for comparing two sequences be less useful to compare millions of sequences? Some problems are so inherently complex that an exhaustive analysis would require a computer with enormous memory or the problem would take an unacceptably long time to complete. A *heuristic algorithm* is one that makes approximations of the best solution without exhaustively considering every possible outcome. The 13 proteins in **Figure 3.2** can be arranged in a tree over a billion distinct ways (see Chapter 7); finding the optimal tree is a problem that a heuristic algorithm can solve in a second.

See the section "The Statistical Significance of Pairwise Alignments" for further discussion, including the use of expect values to assess whether an alignment of two sequences is likely to have occurred by chance (Chapter 4). Such analyses provide evidence to assess the hypothesis that two proteins are homologous. Ultimately, the strongest evidence to determine whether two proteins are homologous comes from structural studies in combination with evolutionary analyses.

Two proteins could have similar structures due to convergent evolution. Molecular evolutionary studies are essential (based on sequence analyses) to assess this possibility.

BOX 3.4 DAYHOFF'S PROTEIN SUPERFAMILIES

Dayhoff (1978, p. 3) studied 34 protein "superfamilies" grouped into 71 phylogenetic trees. These proteins ranged from some that are very well conserved (e.g., histones and glutamate dehydrogenase; see **Fig. 3.10**) to others that have a high rate of mutation acceptance (e.g., immunoglobulin (Ig) chains and kappa casein; see **Fig. 3.11**). Protein families were aligned; then they counted how often any one amino acid in the alignment was replaced by another. Here is a partial list of the proteins they studied, including the rates of mutation acceptance. For a more detailed list, see **Table 7.1**. There is a range of almost 400-fold between the families that evolve fastest and slowest, but within a given family the rate of evolution (measured in PAMs per unit time) varies only two- to three-fold between species. Used with permission.

PROTEIN	PAMS PER 100 MILLION YEARS
Immunoglobulin (Ig) kappa chain C region	37
Kappa casein	33
Epidermal growth factor	26
Serum albumin	19
Hemoglobin alpha chain	12
Myoglobin	8.9
Nerve growth factor	8.5
Trypsin	5.9
Insulin	4.4
Cytochrome c	2.2
Glutamate dehydrogenase	0.9
Histone H3	0.14
Histone H4	0.10

Gaps

Pairwise alignment is useful as a way to identify mutations that have occurred during evolution and have caused divergence of the sequences of the two proteins we are studying. The most common mutations are *substitutions*, *insertions*, and *deletions*. In protein sequences, substitutions occur when a mutation results in the codon for one amino acid being changed into that for another. This results in the alignment of two nonidentical amino acids, such as serine and threonine. Insertions and deletions occur when residues are added or removed and are typically represented by dashes that are added to one or the other sequence. Insertions or deletions (even those just one character long) are referred to as *gaps* in the alignment.

In our alignment of human beta globin and myoglobin there is one gap (**Fig. 3.5a**, between the arrowheads). Gaps can occur at the ends of the proteins or in the middle. Note that one of the effects of adding gaps is to make the overall length of each alignment exactly the same. The addition of gaps can help to create an alignment that models evolutionary changes that have occurred.

For a description of affine gap penalties at NCBI see ⊕ http://www.ncbi.nlm.nih.gov/blast/html/sub_matrix.html (WebLink 3.3).

In a typical scoring scheme there are two gap penalties called affine gap costs. One is a score $-a$ for creating a gap (–11 in the example of **Fig. 3.5b**). A second penalty is $-b$ for each residue that a gap extends. If a gap extends for k residues it is assigned a penalty of $-(a + bk)$. For a gap of length 1, the score is $-(a + b)$.

Pairwise Alignment, Homology, and Evolution of Life

If two proteins are homologous, they share a common ancestor. Generally, we observe the sequence of proteins (and genes) from organisms that are extant. We can compare myoglobins from species such as human, horse, and chicken, and see that the sequences are homologous (**Fig. 3.2**). This implies that an ancestral organism had a myoglobin gene and lived sometime before the divergences of the lineages that gave rise to human and chicken ~310 MYA (see Chapter 19). Descendants of that ancestral organism include many vertebrate species. The study of homologous protein (or DNA) sequences by pairwise alignment involves an investigation of the evolutionary history of that protein (or gene).

It is possible to infer the sequence of the common ancestor (see Chapter 7).

For a brief overview of the time scale of life on Earth, see **Figure 3.6** (refer to Chapter 15 for a more detailed discussion). The divergence of different species is established through the use of many sources of data, especially the fossil record. Fossils of bacteria have been discovered in rocks 3.5 billion years old or even older (Schopf, 2002). Fossils of methane-producing archaea, representative of a second domain of life, are found in rocks over 3 billion years old. The other main domain of life, the eukaryotes, emerged at a similar time. In the case of globins, in addition to the vertebrate proteins represented in **Figure 3.2** there

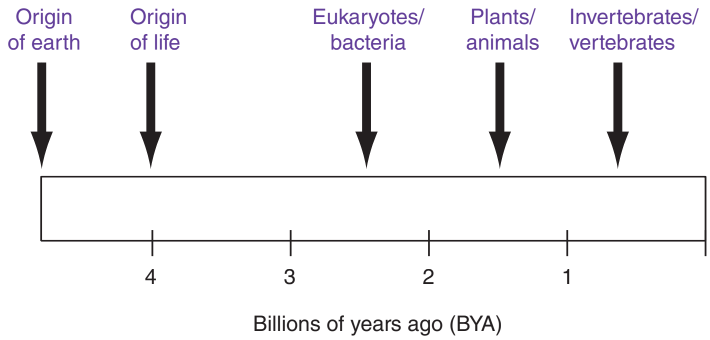


FIGURE 3.6 Overview of the history of life on Earth. See Chapters 15 and 19 for details. Gene/protein sequences are analyzed in the context of evolution. Which organisms have orthologous genes? When did these organisms evolve? How related are human and bacterial globins?

are plant globins that must have shared a common ancestor with the metazoan (animal) globins some 1.5 billion years ago. There are also many bacterial and archaeal globins, suggesting that the globin family arose earlier than two billion years ago.

SCORING MATRICES

When two proteins are aligned, what scores should they be assigned? For the alignment of beta globin and myoglobin in **Figure 3.5a** there were specific scores for matches and mismatches; how were they derived? Margaret Dayhoff (1966, 1978) provided a model of the rules by which evolutionary change occurs in proteins. We now examine the Dayhoff model in seven steps (following the article from Dayhoff, 1978). This provides the basis of a quantitative scoring system for pairwise alignments between any proteins, whether they are closely or distantly related. We then describe the BLOSUM matrices of Steven Henikoff and Jorja G. Henikoff. Most database searching methods such as BLAST and HMMER (Chapters 4 and 5) depend in some form upon the evolutionary insights of the Dayhoff model.

The Dayhoff (1978) reference is to the *Atlas of Protein Sequence and Structure*, a book with 25 chapters (and various co-authors) describing protein families. The 1966 version of the Atlas described the sequences of just several dozen proteins (cytochromes c, other respiratory proteins, globins, some enzymes such as lysozyme and ribonucleases, virus coat proteins, peptide hormones, kinins, and fibrinopeptides). The 1978 edition included about 800 protein sequences.

Dayhoff Model Step 1 (of 7): Accepted Point Mutations

Dayhoff and colleagues considered the problem of how to assign scores to aligned amino acid residues. Their approach was to catalog hundreds of proteins and compare the sequences of closely related proteins in many families. They considered the question of which specific amino acid substitutions are observed to occur when two homologous protein sequences are aligned. They defined an *accepted point mutation* as a replacement of one amino acid in a protein by another residue that has been accepted by natural selection. Accepted point mutation is abbreviated PAM (which is easier to pronounce than APM). An amino acid change that is accepted by natural selection occurs when: (1) a gene undergoes a DNA mutation such that it encodes a different amino acid; and (2) the entire species adopts that change as the predominant form of the protein.

Dayhoff *et al*. focused on proteins sharing 85% or more identity; they could therefore construct their alignments with a high degree of confidence. In the section “Global Sequence Alignment: Algorithm of Needleman and Wunsch” below, we will see how the Needleman and Wunsch algorithm (1970) permits the optimal alignment of protein sequences.

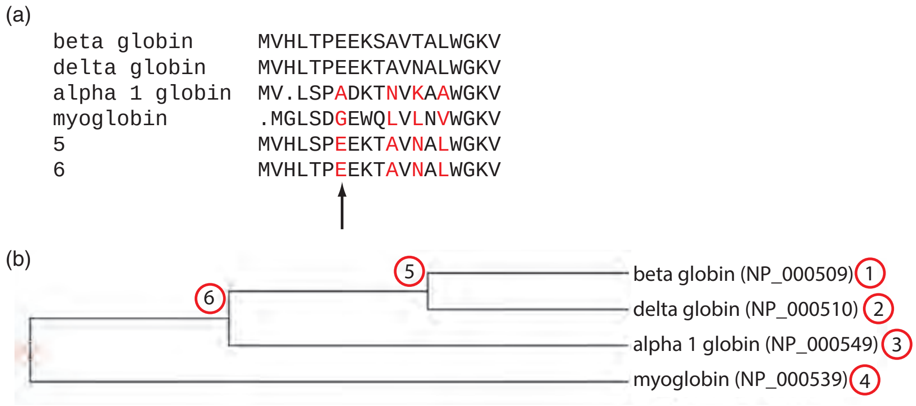
Which point mutations are accepted in protein evolution? Intuitively, conservative replacements such as serine for threonine would be most readily accepted. In order to determine all possible changes, Dayhoff and colleagues examined 1572 changes in 71 groups of closely related proteins (Box 3.4). Their definition of “accepted” mutations was therefore based on empirically observed amino acid substitutions. Their approach involved a phylogenetic analysis: rather than comparing two amino acid residues directly, they compared them to the inferred common ancestor of those sequences (**Fig. 3.7**; Box 3.5).

The empirical results of observed substitutions are shown in **Figure 3.8**, which describes the frequency with which any amino acid pairs *i*, *j* are aligned. Inspection of this table reveals which substitutions are unlikely to occur (for example, cysteine and tryptophan have noticeably few substitutions), while others such as asparagine and serine tolerate replacements quite commonly. Today, we could generate a table like this with vastly more data (refer to **Fig. 2.3** and the explosive growth of DNA sequence repositories). Several groups have produced updated versions of the PAM matrices (Gonnet *et al*., 1992; Jones *et al*., 1992). Nonetheless, the findings from 1978 are essentially correct. The largest inaccuracies in **Figure 3.8** occur for pairs of rarely substituted residues such as cys and asp, for which zero substitutions were observed in the 1978 dataset (35 of 190 total possible exchanges were never observed).

Dayhoff Model Step 2 (of 7): Frequency of Amino Acids

To model the probability that one aligned amino acid in a protein changes to another, we need to know the frequencies of occurrence of each amino acid. **Table 3.1** shows the frequency with which each amino acid is found (f_i).

Look up a recent estimate of the frequency of occurrence of each amino acid at the SwissProt website http://www.expasy.ch/sprot/relnotes/relstat.html (WebLink 3.4). From the UniProtKB/Swiss-Prot protein knowledgebase (release 51.7), the amino acid composition of all proteins is shown in Web Document 3.3 (http://www.bioinfbook.org/chapter3).



```
(a)
    beta globin      MVHLTPEEKSAVTALWGKV
    delta globin     MVHLTPEEKTAVNALWGKV
    alpha 1 globin   MV.LSPADKTNVKAAWGKV
    myoglobin        .MGLSDGEWQLVLNVWGKV
    5                MVHLSPEEKTAVNALWGKV
    6                MVHLTPEEKTAVNALWGKV
```

(b)
5
6
beta globin (NP_000509) 1
delta globin (NP_000510) 2
alpha 1 globin (NP_000549) 3
myoglobin (NP_000539) 4

FIGURE 3.7 Dayhoff's approach to determining amino acid substitutions. (a) Partial multiple sequence alignment of human alpha 1 globin, beta globin, delta globin, and myoglobin. Four columns in which alpha 1 globin and myoglobin have different amino acid residues are indicated in red. For example, A is aligned with G (arrow). (b) Phylogenetic tree that shows the four extant sequences (labeled 1–4) as well as two internal nodes that represent the ancestral sequences (labeled 5 and 6). The inferred ancestral sequences were identified by maximum parsimony analysis using the software PAUP (Chapter 7), and are displayed in (a). From this anaysis it is apparent that at each of the columns labeled in red, there was no direct interchange of two amino acids between alpha 1 globin and myoglobin. Instead, an ancestral residue diverged. For example, the arrow in (a) indicates an ancestral glutamate that evolved to become alanine or glycine, but it would not be correct to suggest that alanine had been converted directly to glycine.

Dayhoff Model Step 3 (of 7): Relative Mutability of Amino Acids

Dayhoff *et al*. calculated the relative mutabilities of the amino acids (**Table 3.2**). This simply describes how often each amino acid is likely to change over a short evolutionary period. (We note that the evolutionary period in question is short because this analysis involves protein sequences that are closely related to each other.) To calculate relative mutability, they divided the number of times each amino acid was observed to mutate (m_i) by the overall frequency of occurrence of that amino acid (f_i).

Why are some amino acids more mutable than others? The less mutable residues probably have important structural or functional roles in proteins, such that the consequence of replacing them with any other residue could be harmful to the organism.

BOX 3.5 A PHYLOGENETIC APPROACH TO ALIGNING AMINO ACIDS

Dayhoff and colleagues did not compare the probability of one residue mutating directly into another. Instead, they constructed phylogenetic trees using parsimony analysis (see Chapter 7). They then described the probability that two aligned residues derived from a common ancestral residue. With this approach, they could minimize the confounding effects of multiple substitutions occurring in an aligned pair of residues. As an example, consider an alignment of the four human proteins alpha 1 globin, beta globin, delta globin, and myoglobin. A direct comparison of alpha 1 globin would suggest several amino acid replacements such as ala↔gly, asn↔leu, lys↔leu, and ala↔val (**Fig. 3.7a**). However, a phylogenetic analysis of these four proteins results in the estimation of internal nodes that represent ancestral sequences. In **Figure 3.7b** the external nodes (corresponding to the four existing proteins) are labeled, as are internal nodes 5 and 6 that correspond to inferred ancestral sequences. In the four cases that are highlighted in **Figure 3.7a**, the ancestral sequences suggest that a glu residue changed to ala and gly in alpha 1 globin and myoglobin, but ala and gly never directly interchanged (**Fig. 3.7a**, arrow). The Dayhoff approach was therefore more accurate by taking an evolutionary perspective.

In a further effort to avoid the complicating factor of multiple substitutions occurring in alignments of protein families, Dayhoff et al. also focused on using multiple sequence alignments of closely related proteins. For example, their analysis of globins considered the alpha globins and beta globins separately.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
A																				
R	30																			
N	109	17																		
D	154	0	532																	
C	33	10	0	0																
Q	93	120	50	76	0															
E	266	0	94	831	0	422														
G	579	10	156	162	10	30	112													
H	21	103	226	43	10	243	23	10												
I	66	30	36	13	17	8	35	0	3											
L	95	17	37	0	y	75	15	17	40	253										
K	57	477	322	85	0	147	104	60	23	43	39									
M	29	17	0	0	0	20	7	7	0	57	207	90								
F	20	7	7	0	0	0	0	17	20	90	167	0	17							
P	345	67	27	10	10	93	40	49	50	7	43	43	4	7						
S	772	137	432	98	117	47	86	450	26	20	32	168	20	40	269					
T	590	20	169	57	10	37	31	50	14	129	52	200	28	10	73	696				
W	0	27	3	0	0	0	0	0	3	0	13	0	0	10	0	17	0			
Y	20	3	36	0	30	0	10	0	40	13	23	10	0	260	0	22	23	6		
V	365	20	13	17	33	27	37	97	30	661	303	17	77	10	50	43	186	0	17	
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val

FIGURE 3.8 Numbers of accepted point mutations, multiplied by 10, in 1572 cases of amino acid substitutions from closely related protein sequences. Amino acids are presented alphabetically according to the three-letter code. Notice that some substitutions (green shaded boxes) are very commonly accepted (such as V and I or S and T). Other amino acids, such as C and W, are rarely substituted by any other residue (orange shaded boxes).

Source: Dayhoff (1972). Reproduced with permission from National Biomedical Research Foundation.

(We will see in Chapter 21 that many human diseases, from cystic fibrosis to the autism-related Rett syndrome to hemoglobinopathies, can be caused by a single amino acid substitution in a protein.) Conversely, the most mutable amino acids – asparagine, serine, aspartic acid, and glutamic acid – have functions in proteins that are easily assumed by other residues. The most common substitutions seen in **Figure 3.8** are glutamic acid for aspartic acid (both are acidic), serine for alanine, serine for threonine (both are hydroxylated), and isoleucine for valine (both are hydrophobic and of a similar size).

TABLE 3.1 Normalized frequencies of amino acid. These values sum to 1. If the 20 amino acids were equally represented in proteins, these values would all be 0.05 (i.e., 5%); instead, amino acids vary in their frequency of occurrence.

Gly	0.089	Arg	0.041
Ala	0.087	Asn	0.040
Leu	0.085	Phe	0.040
Lys	0.081	Gln	0.038
Ser	0.070	Ile	0.037
Val	0.065	His	0.034
Thr	0.058	Cys	0.033
Pro	0.051	Tyr	0.030
Glu	0.050	Met	0.015
Asp	0.047	Trp	0.010

Source: Dayhoff (1972). Reproduced with permission from National Biomedical Research Foundation.

TABLE 3.2 Relative mutabilities of amino acids. The value of alanine is arbitrarily set to 100.

Asn	134	His	66
Ser	120	Arg	65
Asp	106	Lys	56
Glu	102	Pro	56
Ala	100	Gly	49
Thr	97	Tyr	41
Ile	96	Phe	41
Met	94	Leu	40
Gln	93	Cys	20
Val	74	Trp	18

Source: Dayhoff (1972). Reproduced with permission from National Biomedical Research Foundation. Dayhoff (1972). Reproduced with permission from National Biomedical Research Foundation.

The substitutions that occur in proteins can also be understood with reference to the genetic code (Box 3.6). Observe how common amino acid substitutions tend to require only a single-nucleotide change. For example, aspartic acid is encoded by GAU or GAC, and changing the third position to either A or G causes the codon to encode a glutamic acid. Also note that four of the five least mutable amino acids (tryptophan, cysteine, phenylalanine, and tyrosine) are specified by only one or two codons. A mutation of any of the three bases of the W codon is guaranteed to change that amino acid. The low mutability of this amino acid suggests that substitutions are not tolerated by natural selection. Of the eight least mutable amino acids (**Table 3.2**), only one (leucine) is specified by six codons. Dayhoff *et al.* also noted that a fairly large number (20%) of the interchanges observed in **Figure 3.8** required two nucleotide changes. In other cases such as gly and trp, only a single-nucleotide change would be required for the substitution; this was never empirically observed however, presumably because such a change has been rejected by natural selection.

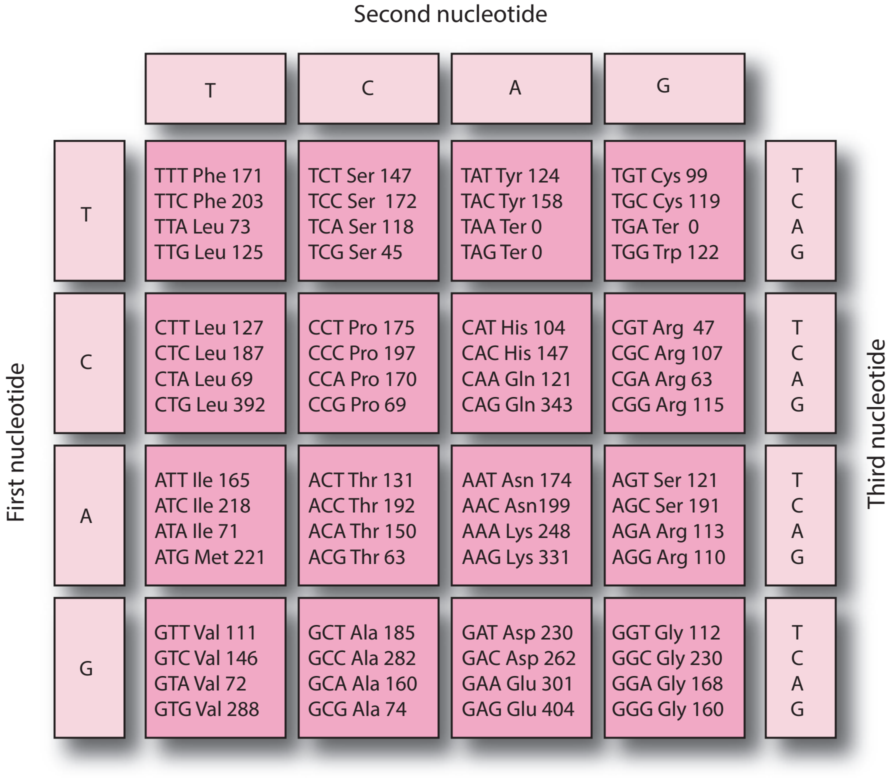
Dayhoff Model Step 4 (of 7): Mutation Probability Matrix for the Evolutionary Distance of 1 PAM

Dayhoff and colleagues next used the data on accepted mutations (**Fig. 3.8**) and the probabilities of occurrence of each amino acid to generate a *mutation probability matrix M* (**Fig. 3.9**). Each element of the matrix M_{ij} shows the probability that an original amino acid j (see the columns) will be replaced by another amino acid i (see the rows) over a defined evolutionary interval. In the case of **Figure 3.9** the interval is one PAM, which is defined as the unit of evolutionary divergence in which 1% of the amino acids have been changed between the two protein sequences. Note that the evolutionary interval of this PAM matrix is defined in terms of percent amino acid divergence and not in units of years. 1% divergence of protein sequence may occur over vastly different time frames for protein families that undergo substitutions at different rates (see **Fig. 7.5** in which we introduce the molecular clock).

Examination of **Figure 3.9** reveals several important features. The highest scores are distributed in a diagonal from top left to bottom right. The values in each column sum to 100%. The value 98.7 at the top left indicates that, when the original sequence consists of an alanine, there is a 98.7% likelihood that the replacement amino acid will also be an alanine over an evolutionary distance of one PAM. There is a 0.3% chance that it will be changed to serine. The most mutable amino acid (from **Table 3.2**), asparagine, has only a

BOX 3.6. THE STANDARD GENETIC CODE

In this table, the 64 possible codons are depicted along with the frequency of codon utilization and the single-letter code of the amino acid that is specified. There are four bases (A, C, G, U) and three bases per codon, so there are $4^3 = 64$ codons.



Second nucleotide

First nucleotide	T	C	A	G	Third nucleotide
T	TTT Phe 171	TCT Ser 147	TAT Tyr 124	TGT Cys 99	T
	TTC Phe 203	TCC Ser 172	TAC Tyr 158	TGC Cys 119	C
	TTA Leu 73	TCA Ser 118	TAA Ter 0	TGA Ter 0	A
	TTG Leu 125	TCG Ser 45	TAG Ter 0	TGG Trp 122	G
C	CTT Leu 127	CCT Pro 175	CAT His 104	CGT Arg 47	T
	CTC Leu 187	CCC Pro 197	CAC His 147	CGC Arg 107	C
	CTA Leu 69	CCA Pro 170	CAA Gln 121	CGA Arg 63	A
	CTG Leu 392	CCG Pro 69	CAG Gln 343	CGG Arg 115	G
A	ATT Ile 165	ACT Thr 131	AAT Asn 174	AGT Ser 121	T
	ATC Ile 218	ACC Thr 192	AAC Asn199	AGC Ser 191	C
	ATA Ile 71	ACA Thr 150	AAA Lys 248	AGA Arg 113	A
	ATG Met 221	ACG Thr 63	AAG Lys 331	AGG Arg 110	G
G	GTT Val 111	GCT Ala 185	GAT Asp 230	GGT Gly 112	T
	GTC Val 146	GCC Ala 282	GAC Asp 262	GGC Gly 230	C
	GTA Val 72	GCA Ala 160	GAA Glu 301	GGA Gly 168	A
	GTG Val 288	GCG Ala 74	GAG Glu 404	GGG Gly 160	G

Adapted from the International Human Genome Sequencing Consortium (2001), **figure 34**. Used with permission.

Several features of the genetic code should be noted. Amino acids may be specified by one codon (M, W), two codons (C, D, E, F, H, K, N, Q, Y), three codons (I), four codons (A, G, P, T, V), or six codons (L, R, S). UGA is rarely read as a selenocysteine (abbreviated sec, and the assigned single-letter abbreviation is U).

For each block of four codons that are grouped together, one is often used dramatically less frequently. For example, for F, L, I, M, and V (i.e., codons with a U in the middle, occupying the first column of the genetic code), adenine is used relatively infrequently in the third-codon position. For codons with a cytosine in the middle position, guanine is strongly under-represented in the third position.

Also note that in many cases mutations cause a conservative change (or no change at all) in the amino acid. Consider threonine (ACX). Any mutation in the third position causes no change in the specified amino acid, because of "wobble." If the first nucleotide of any threonine codon is mutated from A to U, the conservative replacement to a serine occurs. If the second nucleotide C is mutated to a G, a serine replacement occurs. Similar patterns of conservative substitution can be seen along the entire first column of the genetic code, where all of the residues are hydrophobic, and also for the charged residues D, E and K, R.

Codon usage varies between organisms and between genes within organisms. Note also that while this is the standard genetic code, some organisms use alternate genetic codes. A group of two dozen alternate genetic codes are listed at the NCBI Taxonomy website, ⊕ http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/ (WebLink 3.20). As an example of a nonstandard code, vertebrate mitochondrial genomes use AGA and AGG to specify termination (rather than arg in the standard code), ATA to specify met (rather than ile), and TGA to specify trp (rather then termination).

98.22% chance of remaining unchanged; the least mutable amino acid, tryptophan, has a 99.76% chance of remaining the same.

The nondiagonal elements of this matrix have the values:

$$M_{ij} = \frac{\lambda m_j A_{ij}}{\sum_i A_{ij}} \qquad (3.1)$$

		Original amino acid																			
		A Ala	R Arg	N Asn	D Asp	C Cys	Q Gln	E Glu	G Gly	H His	I Ile	L Leu	K Lys	M Met	F Phe	P Pro	S Ser	T Thr	W Trp	Y Tyr	V Val
Replacement amino acid	A	98.7	0.0	0.1	0.1	0.0	0.1	0.2	0.2	0.0	0.1	0.0	0.0	0.1	0.0	0.2	0.4	0.3	0.0	0.0	0.2
	R	0.0	99.1	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.2	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0
	N	0.0	0.0	98.2	0.4	0.0	0.0	0.1	0.1	0.2	0.0	0.0	0.1	0.0	0.0	0.0	0.2	0.1	0.0	0.0	0.0
	D	0.1	0.0	0.4	98.6	0.0	0.1	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
	C	0.0	0.0	0.0	0.0	99.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
	Q	0.0	0.1	0.0	0.1	0.0	98.8	0.3	0.0	0.2	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
	E	0.1	0.0	0.1	0.6	0.0	0.4	98.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	G	0.2	0.0	0.1	0.1	0.0	0.0	0.1	99.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.1
	H	0.0	0.1	0.2	0.0	0.0	0.2	0.0	0.0	99.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	I	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.7	0.1	0.0	0.2	0.1	0.0	0.0	0.1	0.0	0.0	0.3
	L	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.2	99.5	0.0	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.2
	K	0.0	0.4	0.3	0.1	0.0	0.1	0.1	0.0	0.0	0.0	0.0	99.3	0.2	0.0	0.0	0.1	0.1	0.0	0.0	0.0
	M	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	98.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	F	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	99.5	0.0	0.0	0.0	0.0	0.3	0.0
	P	0.1	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	99.3	0.1	0.0	0.0	0.0	0.0
	S	0.3	0.1	0.3	0.1	0.1	0.0	0.1	0.2	0.0	0.0	0.0	0.1	0.0	0.0	0.2	98.4	0.4	0.1	0.0	0.0
	T	0.2	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.1	0.0	0.1	0.3	98.7	0.0	0.0	0.1
	W	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	99.8	0.0	0.0
	Y	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	99.5	0.0
	V	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.1	0.0	0.2	0.0	0.0	0.0	0.1	0.0	0.0	99.0

FIGURE 3.9 The PAM1 mutation probability matrix. The original amino acid *j* is arranged in columns (across the top), while the replacement amino acid *i* is arranged in rows. Dayhoff et al. multiplied values by 10,000 (offering added precision) while here we multiply by 100 so that, for example, the first cell's value of 98.7 corresponds to 98.7% occurrence of ala remaining ala over this evolutionary interval.

Source: Dayhoff (1972). Reproduced with permission from National Biomedical Research Foundation.

where M_{ij} refers to the probability that an original amino acid j will be replaced by an amino acid from row i. A_{ij} is an element of the accepted point mutation matrix of **Figure 3.8**, such as the value corresponding to the original alanine being substituted by an arginine. λ is a proportionality constant (discussed below) and m_j is the mutability of the jth amino acid (from **Table 3.2**). We can further consider the diagonal elements of **Figure 3.9** which have the values:

$$M_{jj} = 1 - \lambda m_j \tag{3.2}$$

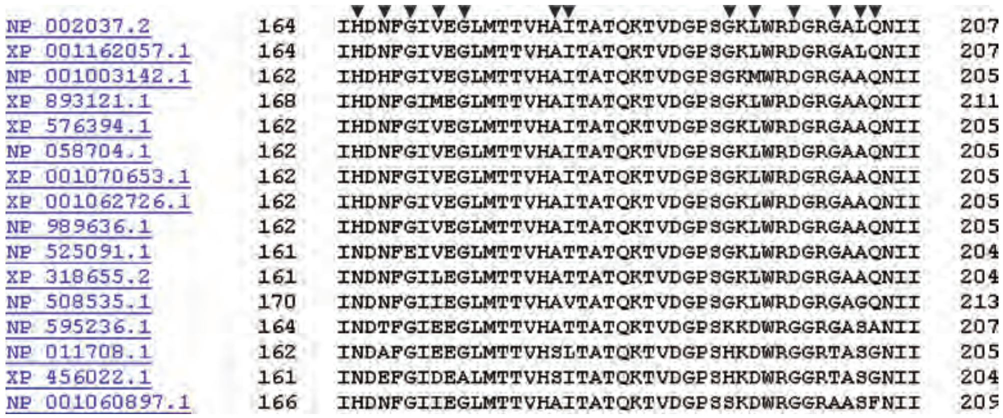
where M_{jj} is the probability that original amino acid j will remain j without undergoing a substitution to another amino acid. Let's understand these two equations by inspecting the first column of the mutation probability matrix in which the original amino acid is alanine. The total probability (the sum of all elements) is 1 or, considering the elements as percentages, the sum of the column is 100%. It is intuitively reasonable that the probability of observing a change to the amino acid – equivalent to the sum of all the elements other than alanine remaining itself, M_{jj} – is proportional to the mutability of alanine.

For each original amino acid, it is easy to observe the amino acids that are most likely to replace it if a change should occur. These data are very relevant to pairwise sequence alignment because they will form the basis of a scoring system (described below in Dayhoff Model Steps 5–7) in which reasonable amino acid substitutions in an alignment are rewarded while unlikely substitutions are penalized.

Almost all molecular sequence data are obtained from extant organisms. We can infer ancestral sequences, as described in Box 3.5 and Chapter 7. In general however, for an aligned pair of residues i, j we do not know which mutated into the other. Dayhoff and colleagues used the assumption that accepted amino acid mutations are undirected, that is, they are equally likely in either direction. In the PAM1 matrix, the close relationship of the proteins makes it unlikely that the ancestral residue is entirely different from both of the observed, aligned residues.

Dayhoff Model Step 5 (of 7): PAM250 and Other PAM Matrices

The PAM1 matrix was based upon the alignment of closely related protein sequences, having an average of 1% change. To ensure that the multiple alignments were valid,



```
                            ▼ ▼ ▼ ▼ ▼      ▼▼           ▼ ▼  ▼  ▼ ▼▼
NP 002037.2         164    IHDNFGIVEGLMTTVHAITATQKTVDGPSGKLWRDGRGALQNII    207
XP 001162057.1      164    IHDNFGIVEGLMTTVHAITATQKTVDGPSGKLWRDGRGALQNII    207
NP 001003142.1      162    IHDHFGIVEGLMTTVHAITATQKTVDGPSGKMWRDGRGAAQNII    205
XP 893121.1         168    IHDNFGIMEGLMTTVHAITATQKTVDGPSGKLWRDGRGAAQNII    211
XP 576394.1         162    IHDNFGIVEGLMTTVHAITATQKTVDGPSGKLWRDGRGAAQNII    205
NP 058704.1         162    IHDNFGIVEGLMTTVHAITATQKTVDGPSGKLWRDGRGAAQNII    205
XP 001070653.1      162    IHDNFGIVEGLMTTVHAITATQKTVDGPSGKLWRDGRGAAQNII    205
XP 001062726.1      162    IHDNFGIVEGLMTTVHAITATQKTVDGPSGKLWRDGRGAAQNII    205
NP 989636.1         162    IHDNFGIVEGLMTTVHAITATQKTVDGPSGKLWRDGRGAAQNII    205
NP 525091.1         161    INDNFEIVEGLMTTVHATTATQKTVDGPSGKLWRDGRGAAQNII    204
XP 318655.2         161    INDNFGILEGLMTTVHATTATQKTVDGPSGKLWRDGRGAAQNII    204
NP 508535.1         170    INDNFGIIEGLMTTVHAVTATQKTVDGPSGKLWRDGRGAGQNII    213
NP 595236.1         164    INDTFGIEEGLMTTVHATTATQKTVDGPSKKDWRGGRGASANII    207
NP 011708.1         162    INDAFGIEEGLMTTVHSLTATQKTVDGPSHKDWRGGRTASGNII    205
XP 456022.1         161    INDEFGIDEALMTTVHSITATQKTVDGPSHKDWRGGRTASGNII    204
NP 001060897.1      166    IHDNFGIIEGLMTTVHAITATQKTVDGPSSKDWRGGRAASFNII    209
```

FIGURE 3.10 Multiple sequence alignment of a portion of the glyceraldehyde 3-phosphate dehydrogenase (GAPDH) protein from 13 organisms: *Homo sapiens* (human), *Pan troglodytes* (chimpanzee), *Canis lupus* (dog), *Mus musculus* (mouse), *Rattus norvegicus* (rat; three variants), *Gallus gallus* (chicken), *Drosophila melanogaster* (fruit fly), *Anopheles gambiae* (mosquito), *Caenorhabditis elegans* (worm), *Schizosaccharomyces pombe* (fission yeast), *Saccharomyces cerevisiae* (baker's yeast), *Kluyveromyces lactis* (a fungus), and *Oryza sativa* (rice). Columns in the alignment having even a single amino acid change are indicated with arrowheads. The accession numbers are given in the figure. The alignment was created by searching HomoloGene at NCBI with the term gapdh.

proteins within a family were at least 85% identical. We are often interested in exploring the relationships of proteins that share far less than 99% amino acid identity. We can accomplish this by constructing probability matrices for proteins that share any degree of amino acid identity. Consider closely related proteins, such as the glyceraldehyde-3-phosphate dehydrogenase (GAPDH) proteins shown in **Figure 3.10**. A mutation from one residue to another is a relatively rare event, and a scoring system used to align two such closely related proteins should reflect this. (In the PAM1 mutation probability matrix of **Fig. 3.9**, some substutions such as tryptophan to threonine are so rare that they were never observed in Dayhoff's dataset.)

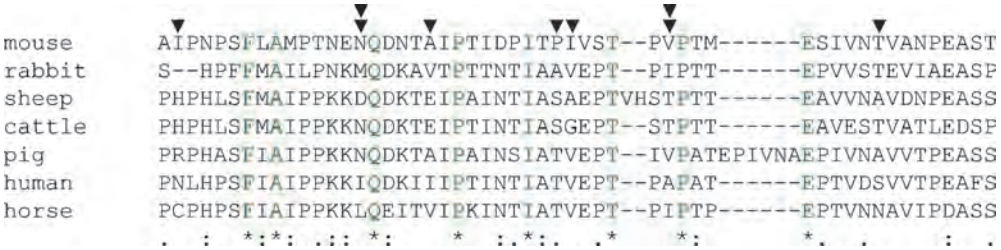
Databases such as Pfam (Chapter 6) summarize the phylogenetic distribution of gene/protein families across the tree of life.

Orthologous kappa caseins from various species provide an example of a less well-conserved family (**Fig. 3.11**). Some columns of residues in this alignment are perfectly conserved among the selected species but most are not, and many gaps need to be introduced. Several positions at which four or even five different residues occur in an aligned column are indicated.

The GAPDH sequences used to generate **Figure 3.10** and the kappa casein sequences used to generate **Figure 3.11** are shown in Web Documents 3.4 and 3.5 at 🌐 http://www.bioinfbook.org/chapter3.

Here, substitutions are likely to be very common. PAM matrices such as PAM100 or PAM250 were generated to reflect the kinds of amino acid substitutions that occur in distantly related proteins.

How are PAM matrices other than PAM1 derived? The proportionality constant λ of Equations (3.1) and (3.2) applies to all columns of the mutation probability matrix of **Figure 3.9**. In that matrix, λ is chosen to correspond to an evolutionary distance of 1 PAM. As we make λ larger, we model a greater evolutionary distance. We could for example make a PAM2, PAM3, or PAM4 matrix by multiplying λ. This approach will fail for greater evolutionary distances (such as PAM250, in which 250 changes occur in two aligned sequences of length 100); the problem is that adjusting λ does not account for multiple substitutions. Dayhoff *et al*. instead used matrix multiplication: they multiplied the PAM1 matrix by itself, up to hundreds of times, to obtain other PAM matrices (see Box 3.7), therefore extrapolating from the PAM1 matrix. Today this approach is considered valid, although it depends on the accuracy of the PAM1 matrix to avoid propagating errors.



```
                         ▼                    ▼
           ▼             ▼   ▼        ▼▼      ▼              ▼
mouse     AIPNPSFLAMPTNENQDNTAIPTIDPITPIVST--PVPTM------ESIVNTVANPEAST
rabbit    S--HPFFMAILPNKMQDKAVTPTTNTIAAVEPT--PIPTT------EPVVSTEVIAEASP
sheep     PHPHLSFMAIPPKKDQDKTEIPAINTIASAEPTVHSTPTT------EAVVNAVDNPEASS
cattle    PHPHLSFMAIPPKKNQDKTEIPTINTIASGEPT--STPTT------EAVESTVATLEDSP
pig       PRPHASFIAIPPKKNQDKTAIPAINSIATVEPT--IVPATEPIVNAEPIVNAVVTPEASS
human     PNLHPSFIAIPPKKIQDKIIIPTINTIATVEPT--PAPAT------EPTVDSVVTPEAFS
horse     PCPHPSFIAIPPKKLQEITVIPKINTIATVEPT--PIPTP------EPTVNNAVIPDASS
          .  :  *:*: .:: *:    *  :.*:.  .*    *:       *.  .     :  .
```

FIGURE 3.11 Multiple sequence alignment of seven kappa caseins, representing a protein family that is relatively poorly conserved. Only a portion of the entire alignment is shown. Note that just eight columns of residues are perfectly conserved (indicated with asterisks), and gaps of varying length form part of the alignment. In several columns, there are four different aligned amino acids (arrowheads); in two instances there are five different residues (double arrowheads). The sequences were aligned with MUSCLE 3.6 (see Chapter 6) and were human (NP_005203), equine (*Equus caballus*; NP_001075353), pig (*Sus scrofa* NP_001004026), ovine (*Ovis aries* NP_001009378), rabbit (*Oryctolagus cuniculus* P33618), bovine (*Bos taurus* NP_776719) and mouse (*Mus musculus* NP_031812).

To make sense of what different PAM matrices mean, consider the extreme cases. When PAM equals zero, the matrix is a unit diagonal (**Fig. 3.12**, upper panel) because no amino acids have changed. PAM can be extremely large (e.g., PAM greater than 2000, or the matrix can even be multiplied by itself an infinite number of times). In the resulting PAM∞ matrix there is an equal likelihood of any amino acid being present and all the values consist of rows of probabilities that approximate the background probability for the frequency occurrence of each amino acid (**Fig. 3.12**, lower panel). We described these background frequencies in **Table 3.1**.

original amino acid

replacement amino acid

PAM0	A	R	N	D	C	Q	E	G
A	100	0	0	0	0	0	0	0
R	0	100	0	0	0	0	0	0
N	0	0	100	0	0	0	0	0
D	0	0	0	100	0	0	0	0
C	0	0	0	0	100	0	0	0
Q	0	0	0	0	0	100	0	0
E	0	0	0	0	0	0	100	0
G	0	0	0	0	0	0	0	100

original amino acid

replacement amino acid

PAM∞	A	R	N	D	C	Q	E	G
A	8.7	8.7	8.7	8.7	8.7	8.7	8.7	8.7
R	4.1	4.1	4.1	4.1	4.1	4.1	4.1	4.1
N	4.0	4.0	4.0	4.0	4.0	4.0	4.0	4.0
D	4.7	4.7	4.7	4.7	4.7	4.7	4.7	4.7
C	3.3	3.3	3.3	3.3	3.3	3.3	3.3	3.3
Q	3.8	3.8	3.8	3.8	3.8	3.8	3.8	3.8
E	5.0	5.0	5.0	5.0	5.0	5.0	5.0	5.0
G	8.9	8.9	8.9	8.9	8.9	8.9	8.9	8.9

FIGURE 3.12 Portion of the matrices for a zero PAM value (PAM0; upper panel) or for an infinite PAM∞ value (lower panel). At PAM∞ (i.e., if the PAM1 matrix is multiplied by itself an infinite number of times), all the entries in each row converge on the normalized frequency of the replacement amino acid (see **Table 3.1**). A PAM2000 matrix has similar values that tend to converge on these same limits. In a PAM2000 matrix, the proteins being compared are at an extreme of unrelatedness. In constrast, at PAM0 no mutations are tolerated and the residues of the proteins are perfectly conserved.

BOX 3.7 MATRIX MULTIPLICATION

A matrix is an orderly array of numbers. An example of a matrix with rows *i* and columns *j* is:

$$\begin{bmatrix} 1 & 2 & 4 \\ 2 & 0 & -3 \\ 4 & -3 & 6 \end{bmatrix}$$

In a symmetric matrix, such as the one above, $a_{ij} = a_{ji}$. This means that all the corresponding nondiagonal elements are equal. Matrices may be added, subtracted, or manipulated in a variety of ways. Two matrices can be multipled together providing that the number of columns in the first matrix M_1 equals the number of rows in the second matrix M_2.

We can view PAM matrices in R. Try working with a PAM1 matrix. Since it is not readily available in R packages or at the NCBI ftp site, we provide the text file pam1.txt at Web Document 3.10 (http://bioinfbook.org). Import it into RStudio, look at its properties, and view its first five rows and columns:

```
> dim(pam1) # this shows the dimensions of the matrix
[1] 20 20
> length(pam1) # this displays the length
[1] 20
> str(pam1) # this displays the structure of pam1; just the first several
# lines are shown here
'data.frame':   20 obs. of 20 variables:
$ A: num 0.9867 0.0001 0.0004 0.0006 0.0001 ...
$ R: num 0.0002 0.9913 0.0001 0 0.0001 ...
...
> pam1 # this shows the full matrix (not shown here)
> pam1[1:5,1:5] # this displays the first five rows and columns
       A      R      N      D      C
1 0.9867 0.0002 0.0009 0.0010 0.0003
2 0.0001 0.9913 0.0001 0.0000 0.0001
3 0.0004 0.0001 0.9822 0.0036 0.0000
4 0.0006 0.0000 0.0042 0.9859 0.0000
5 0.0001 0.0001 0.0000 0.0000 0.9973
```

Next, multiply the PAM1 mutation probability matrix by itself 250 times, creating the data frame called pam250, obtaining a PAM250 matrix.

```
> pam250 <- pam1^250 # we multiply the PAM1 matrix by itself 250 times
> pam250[1:5,1:5] # we view the first five rows and columns
              A         R          N          D         C
[1,] 0.03517888 0.0000000 0.00000000 0.00000000 0.0000000
[2,] 0.00000000 0.1125321 0.00000000 0.00000000 0.0000000
[3,] 0.00000000 0.0000000 0.01121973 0.00000000 0.0000000
[4,] 0.00000000 0.0000000 0.00000000 0.02872213 0.0000000
[5,] 0.00000000 0.0000000 0.00000000 0.00000000 0.5086918
```

The PAM250 matrix is of particular interest (**Fig. 3.13**). It is produced when the PAM1 matrix is multiplied by itself 250 times, and it is one of the common matrices used for BLAST searches of databases (Chapter 4). This matrix applies to an evolutionary distance where proteins share about 20% amino acid identity. Compare this matrix to the PAM1 mutation probability matrix (**Fig. 3.9**), and note that much of the information content is lost. The diagonal from top left to bottom right tends to contain higher values than elsewhere in the matrix, but not in the dramatic fashion of the PAM1 matrix. As an example of how to read the PAM250 matrix, if the original amino acid is an alanine there is just a 13% chance that the second sequence will also have

		Original amino acid																			
		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
Replacement amino acid	A	13	6	9	9	5	8	9	12	6	8	6	7	7	4	11	11	11	2	4	9
	R	3	17	4	3	2	5	3	2	6	3	2	9	4	1	4	4	3	7	2	2
	N	4	4	6	7	2	5	6	4	6	3	2	5	3	2	4	5	4	2	3	3
	D	5	4	8	11	1	7	10	5	6	3	2	5	3	1	4	5	5	1	2	3
	C	2	1	1	1	52	1	1	2	2	2	1	1	1	1	2	3	2	1	4	2
	Q	3	5	5	6	1	10	7	3	7	2	3	5	3	1	4	3	3	1	2	3
	E	5	4	7	11	1	9	12	5	6	3	2	5	3	1	4	5	5	1	2	3
	G	12	5	10	10	4	7	9	27	5	5	4	6	5	3	8	11	9	2	3	7
	H	2	5	5	4	2	7	4	2	15	2	2	3	2	2	3	3	2	2	3	2
	I	3	2	2	2	2	2	2	2	2	10	6	2	6	5	2	3	4	1	3	9
	L	6	4	4	3	2	6	4	3	5	15	34	4	20	13	5	4	6	6	7	13
	K	6	18	10	8	2	10	8	5	8	5	4	24	9	2	6	8	8	4	3	5
	M	1	1	1	1	0	1	1	1	1	2	3	2	6	2	1	1	1	1	1	2
	F	2	1	2	1	1	1	1	1	3	5	6	1	4	32	1	2	2	4	20	3
	P	7	5	5	4	3	5	4	5	5	3	3	4	3	2	20	6	5	1	2	4
	S	9	6	8	7	7	6	7	9	6	5	4	7	5	3	9	10	9	4	4	6
	T	8	5	6	6	4	5	5	6	4	6	4	6	5	3	6	8	11	2	3	6
	W	0	2	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	55	1	0
	Y	1	1	2	1	3	1	1	1	3	2	2	1	2	15	1	2	2	3	31	2
	V	7	4	4	4	4	4	4	5	4	15	10	4	10	5	5	5	7	2	4	17

FIGURE 3.13 The PAM250 mutation probability matrix. At this evolutionary distance, only one in five amino acid residues remains unchanged from an original amino acid sequence (columns) to a replacement amino acid (rows). Note that the scale has changed relative to **Figure 3.11**, and the columns sum to 100.

Source: Dayhoff (1972). Reproduced with permission from National Biomedical Research Foundation.

an alanine. In fact, there is a nearly equal probability (12%) that the alanine will have been replaced by a glycine. For the least mutable amino acids, tryptophan and cysteine, there is more than a 50% probability that those residues will remain unchanged at this evolutionary distance.

Dayhoff Model Step 6 (of 7): From a Mutation Probability Matrix to a Relatedness Odds Matrix

Dayhoff *et al*. defined a relatedness odds matix. For the elements M_{ij} of any given mutation probability matrix, what is the probability that amino acid j will change to i in a homologous sequence?

$$R_{ij} = \frac{M_{ij}}{f_i}. \tag{3.3}$$

Equation (3.3) describes an odds ratio (Box 3.8). For the numerator, Dayhoff *et al*. considered an entire spectrum of models for evolutionary change in determining target frequencies. For the denominator, the normalized frequency f_i is the probability of amino acid residue i occurring in the second sequence by chance.

For the relatedness odds matrix, a value for R_{ij} of 1 means that the substitution (e.g., alanine replaced by asparagine) occurs as often as can be expected by chance. Values greater than 1 indicate that the alignment of two residues occurs more often than expected by chance (e.g., a conservative substitution of serine for threonine). Values less than 1 suggest that the alignment is not favored. For a comparison of two proteins, it is necessary to determine the values for R_{ij} at each aligned position and then multiply the resulting probabilities to achieve an overall score for an alignment.

BOX 3.8. STATISTICAL CONCEPT: THE ODDS RATIO

Dayhoff *et al*. (1972) developed their scoring matrix by using odds ratios. The mutation probability matrix has elements M_{ij} that give the probability that amino acid j changes to amino acid i in a given evolutionary interval. The normalized frequency f_i gives the probability that amino acid i will occur at that given amino acid position by chance. The relatedness odds matrix in Equation (3.3) may also be expressed $R_{ij} = M_{ij}/f_i$, where R_{ij} is the relatedness odds ratio.

Equation (3.3) may also be written:

$$\text{Probability of an authentic alignment} = \frac{\text{P(aligned } | \text{authentic)}}{\text{P(aligned } | \text{random)}}.$$

The right side of this equation can be read: "the probability of an alignment given that it is authentic (i.e., the substitution of amino acid j with amino acid i) divided by the probability that the alignment occurs given that it happened by chance." An odds ratio can be any positive ratio. The probability that an event will occur is the fraction of times it is expected to be observed over many trials; probabilities have values ranging from 0 to 1. Odds and probability are closely related concepts. A probability of 0 corresponds to an odds of 0; a probability of 0.5 corresponds to an odds of 1.0; a probability of 0.75 corresponds to odds of 75:25 or 3. Odds and probabilities may be converted as follows:

$$\text{odds} = \frac{\text{probability}}{1 - \text{probability}} \text{ and probability} = \frac{\text{odds}}{1 + \text{odds}}.$$

Dayhoff Model Step 7 (of 7): Log-Odds Scoring Matrix

The logarithmic form of the relatedness odds matrix is called a log-odds matrix. The log-odds form is given by:

$$s_{ij} = 10 \times \log_{10}\left(\frac{M_{ij}}{f_i}\right). \tag{3.4}$$

The cells in a log-odds matrix consist of scores (s_{ij}) for aligning any two residues (including an amino acid with itself) along the length of a pairwise alignment. M_{ij} (also written as q_{ij}) is the observed frequency of substitution for each pair of amino acids. The values for q_{ij}, also called the "target frequencies," are derived from a mutation probability matrix such as those shown in **Figures 3.9** (for PAM1) and 3.13 (for PAM250). These values consist of positive numbers that sum to 1. The background frequency f_i refers to the independent, background probability of replacement amino acid i occurring in this position.

The log-odds matrix for PAM250 is shown in **Figure 3.14**. The values have been rounded off to the nearest integer. Using the logarithm here is convenient because it allows us to sum the scores of the aligned residues when we perform an overall alignment of two sequences. (If we did not take the logarithm we would need to multiply the ratios at all the aligned positions, and this is computationally more cumbersome.)

Try using Equation (3.4) to make sure you understand how the mutation probability matrix (**Fig. 3.13**) is converted into the log-odds scoring matrix (**Fig. 3.14**). As an example, to determine the score assigned to a substitution from cysteine to leucine, the PAM250 mutation probability matrix value is 0.02 (**Fig. 3.13**) and the normalized frequency of leucine is 0.085 (**Table 3.1**). We therefore have:

$$s_{(\text{cysteine, leucine})} = 10 \times \log_{10}\left(\frac{0.02}{0.085}\right) = -6.3. \tag{3.5}$$

Note that this log-odds scoring matrix is symmetric, in contrast to the mutation probability matrix in **Figure 3.13**. In a comparison of two sequences it does not matter which is given first. As another example, an original lysine replaced by an arginine (frequency 4.1%) has a mutation probability matrix score of 0.09, and employing Equation (3.4) yields a log-odds score of 3.4 (matching the score of 3 in **Fig. 3.14**). The values in the matrix are rounded off.

A	2																			
R	-2	6																		
N	0	0	2																	
D	0	-1	2	4																
C	-2	-4	-4	-5	12															
Q	0	1	1	2	-5	4														
E	0	-1	1	3	-5	2	4													
G	1	-3	0	1	-3	-1	0	5												
H	-1	2	2	1	-3	3	1	-2	6											
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5										
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	-2	6									
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5								
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6							
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9						
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6					
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2				
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3			
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17		
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4
	A	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**

FIGURE 3.14 Log-odds matrix for PAM250. High PAM values (e.g., PAM250) are useful for aligning very divergent sequences. A variety of algorithms for pairwise alignment, multiple sequence alignment, and database searching (e.g., BLAST) allow you to select an assortment of PAM matrices such as PAM250, PAM70, and PAM30. Adapted from NCBI, ftp://ftp.ncbi.nlm.nih.gov/blast/matrices/.

What do the scores in the PAM250 matrix signify? A score of +17 for tryptophan matching tryptophan indicates that this correspondence is 50 times more frequent than the chance alignment of this residue in a pairwise alignment. From Equation (3.4), let $s_{\mathrm{i}}j = +17$ and let the probability of replacement $q_{ij}/p_i = x$. Then $+17 = 10 \log_{10} x$; $+ 1.7 = \log_{10} x$; and $10^{1.7} = x = 50$.

A score of −10 indicates that the correspondence of two amino acids in an alignment that accurately represents homology (evolutionary descent) is one-tenth as frequent as the chance alignment of these amino acids. A score of zero is neutral. A score of +2 indicates that the amino acid replacement occurs 1.6 times as frequently as expected by chance ($+2 = 10 \log_{10} x$; $x = 10^{0.2} = 1.6$).

The highest values in this particular log-odds scoring matrix (**Fig. 3.14**) are for tryptophan (17 for an identity) and cysteine (12), while the most severe penalties are associated with substitutions for those two residues. When two sequences are aligned and a score is given, that score is simply the sum of the scores for all the aligned residues across the alignment.

The “target frequencies” q_{ij} are estimated in reference to a particular amount of evolutionary change. For example, in a comparison of human beta globin versus the closely related chimpanzee beta globin, the likelihood of any particular residue matching another in a pairwise alignment is extremely high; in a comparison of human beta globin and a bacterial globin, the likelihood of a match is low. If in a particular comparison of closely related proteins a serine were aligned to a threonine 5% of the time, then that target frequency $q_{\mathrm{S,T}}$ would be 0.05. If in a different comparison of differently related proteins serine were aligned to threonine more often, say 40% of the time, then that target frequency $q_{\mathrm{S,T}}$ would be 0.4.

It is easy to see how different PAM matrices score amino acid substitutions by comparing the PAM250 matrix (**Fig. 3.14**) with a PAM10 matrix (**Fig. 3.15**). In the PAM10 matrix, identical amino acid residue pairs tend to produce a higher score than in the PAM250 matrix; for example, a match of alanine to alanine scores 7 versus 2, respectively.

A	7																			
R	-10	9																		
N	-7	-9	9																	
D	-6	-17	-1	8																
C	-10	-11	-17	-21	10															
Q	-7	-4	-7	-6	-20	9														
E	-5	-15	-5	0	-20	-1	8													
G	-4	-13	-6	-6	-13	-10	-7	7												
H	-11	-4	-2	-7	-10	-2	-9	-13	10											
I	-8	-8	-8	-11	-9	-11	-8	-17	-13	9										
L	-9	-12	-10	-19	-21	-8	-13	-14	-9	-4	7									
K	-10	-2	-4	-8	-20	-6	-7	-10	-10	-9	-11	7								
M	-8	-7	-15	-17	-20	-7	-10	-12	-17	-3	-2	-4	12							
F	-12	-12	-12	-21	-19	-19	-20	-12	-9	-5	-5	-20	-7	9						
P	-4	-7	-9	-12	-11	-6	-9	-10	-7	-12	-10	-10	-11	-13	8					
S	-3	-6	-2	-7	-6	-8	-7	-4	-9	-10	-12	-7	-8	-9	-4	7				
T	-3	-10	-5	-8	-11	-9	-9	-10	-11	-5	-10	-6	-7	-12	-7	-2	8			
W	-2	-5	-11	-21	-22	-19	-23	-21	-10	-20	-9	-18	-19	-7	-20	-8	-19	13		
Y	-11	-14	-7	-17	-7	-18	-11	-20	-6	-9	-10	-12	-17	-1	-20	-10	-9	-8	10	
V	-5	-11	-12	-11	-9	-10	-10	-9	-9	-1	-5	-13	-4	-12	-9	-10	-6	-22	-10	8
	A	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**

FIGURE 3.15 Log-odds matrix for PAM10. Low PAM values such as this are useful for aligning very closely related sequences. Compare this with the PAM250 matrix (**Fig. 3.14**) and note that there are larger positive scores for identical matches in this PAM10 matrix and larger penalties for mismatches. Adapted from NCBI, ftp://ftp.ncbi.nlm.nih.gov/blast/matrices/.

The penalties for mismatches are greater in the PAM10 matrix; for example, a mutation of aspartate to arginine scores –17 (PAM10) versus –1 (PAM250). PAM10 even has negative scores for substitutions (such as glutamate to asparagine: –5) that are scored positively in the PAM250 matrix (+1).

Practical Usefulness of PAM Matrices in Pairwise Alignment

We can demonstrate the usefulness of PAM matrices by performing a series of global pairwise alignments of both closely related proteins and distantly related proteins. For the closely related proteins we will use human beta globin (NP_000509.1) and beta globin from the chimpanzee *Pan troglodytes* (XP_508242.1); these proteins share 100% amino acid identity. The bit scores proceed in a fairly linear, decreasing fashion from about 590 bits using the PAM10 matrix to 200 bits using the PAM250 matrix and 100 bits using the PAM500 matrix (**Fig. 3.16**, black line). In this pairwise alignment there are no mismatches or gaps, and the high bit scores associated with low PAM matrices (such as PAM10) are accounted for by the higher relative entropy (defined in "Percent Identity and Relative Entropy"). The PAM10 matrix is therefore appropriate for comparisons of closely related proteins. Next consider pairwise alignments of two relatively divergent proteins, human beta globin and alpha globin (NP_000549.1; **Fig. 3.16**, red line). The PAM70 matrix yields the highest score. Lower PAM matrices (e.g., PAM10 to PAM60) produce lower bit scores because the sequences share only 42% amino acid identity, and mismatches are assigned large negative scores. We conclude that different scoring matrices vary in their sensitivity to protein sequences (or DNA sequences) of varying relatedness. When comparing two sequences, it may be necessary to repeat the search using several different scoring matrices. Alignment programs cannot be preset to choose the right matrix for each pair of sequences. Instead, they begin with the most broadly useful scoring matrix such as BLOSUM62, which we describe in the following section.

Important Alternative to PAM: BLOSUM Scoring Matrices

In addition to the PAM matrices, another very common set of scoring matrices is the blocks substitution matrix (BLOSUM) series. Henikoff and Henikoff (1992, 1996) used

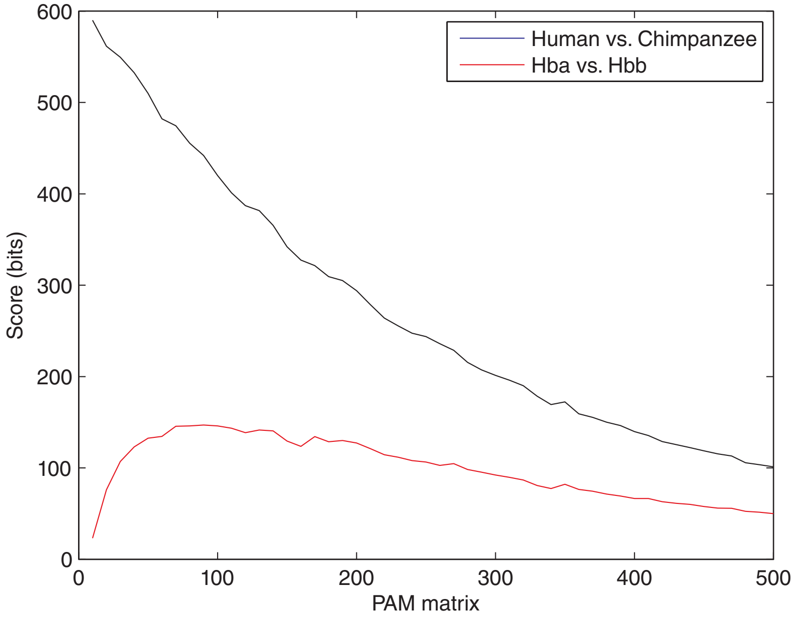


FIGURE 3.16 Global pairwise alignment scores using a series of PAM matrices. Two closely related globins (human and chimpanzee beta globin; black line) were aligned using a series of PAM matrices (*x* axis) and the bit scores were measured (*y* axis). For two distantly related globins (human alpha versus beta globin; red line) the bit scores are smaller for low PAM matrices (such as PAM1 to PAM20) because mismatches are severely penalized.

Note that the denominator in Equations (3.6) and (3.7) includes $p_i\,p_j$, reflecting the background probabilities of the two aligned amino acids. This is given by Henikoff and Henikoff (1992) and Karlin and Altschul (1990) and others (reviewed by Altschul *et al.*, 2005).

the BLOCKS database, which consisted of over 500 groups of local multiple alignments (blocks) of distantly related protein sequences. The Henikoffs therefore focused on conserved regions (blocks) of proteins that are distantly related to each other. The BLOSUM scoring scheme employs a log-odds ratio using the base 2 logarithm:

$$S_{ij} = 2 \times \log_2\left(\frac{q_{ij}}{p_{ij}}\right). \tag{3.6}$$

Equation (3.6) resembles Equation (3.4) in its format. Karlin and Altschul (1990) and Altschul (1991) have shown that substitution matrices can be decribed in general in a log-odds form as follows:

$$S_{ij} = \left(\frac{1}{\lambda}\right)\ln\left(\frac{q_{ij}}{p_i p_j}\right) \tag{3.7}$$

The PAM matrix is given as 10 times the log base 10 of the odds ratio. The BLOSUM matrix is given as 2 times the log base 2 of the odds ratio. BLOSUM scores are therefore not quite as large as they would be if given on the same scale as PAM scores. Practically, this difference in scales is not important because alignment scores are typically converted from raw scores to normalized bit scores (Chapter 4).

where S_{ij} refers to the score of amino acid *i* aligning with *j* and q_{ij} are the positive target frequencies; these sum to 1. λ is a positive parameter that provides a scale for the matrix. We will again encounter λ when we describe the basic statistical measure of a BLAST result (Chapter 4, Equation (4.5)).

The BLOSUM62 matrix is the default scoring matrix for the BLAST protein search programs at NCBI. It merges all proteins in an alignment that has 62% amino acid identity or greater into one sequence. If a block of aligned globin orthologs includes several that have 62, 80, and 95% amino acid identity, these would all be weighted (grouped) as one sequence. Substitution frequencies for the BLOSUM62 matrix are weighted more heavily by blocks of protein sequences having less than 62% identity. (This matrix is therefore useful for scoring proteins that share less than 62% identity.) The BLOSUM62 matrix is shown in **Fig. 3.17**.

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

FIGURE 3.17 The BLOSUM62 scoring matrix of Henikoff and Henikoff (1992). This matrix merges all proteins in an alignment that have 62% amino acid identity or greater into one sequence. BLOSUM62 performs better than alternative BLOSUM matrices or a variety of PAM matrices at detecting distant relationships between proteins. It is therefore the default scoring matrix for most database search programs such as BLAST (Chapter 4).

Source: Henikoff & Henikoff (1992). Reproduced with permission from S. Henikoff.

Henikoff and Henikoff (1992) tested the ability of a series of BLOSUM and PAM matrices to detect proteins in BLAST searches of databases. They found that BLOSUM62 performed slightly better than BLOSUM60 or BLOSUM70 and dramatically better than PAM matrices at identifying various proteins. Their matrices were especially useful for identifying weakly scoring alignments. BLOSUM50 and BLOSUM90 are other commonly used scoring matrices in BLAST searches. (For an alignment of two proteins sharing about 50% identity, try using the BLOSUM50 matrix. The FASTA family of sequence comparison programs use BLOSUM50 as a default.)

The relationships of the PAM and BLOSUM matrices are depicted in **Figure 3.18**. To summarize, BLOSUM and PAM matrices both use log-odds values in their scoring systems. In each case, when performing a pairwise sequence alignment (or when searching a query sequence against a database), specify the exact matrix to use based on the suspected degree of identity between the query and its matches. PAM matrices are based on data from the alignment of closely related protein families, and they involve the assumption that substitution probabilities for highly related proteins (e.g., PAM40) can be extrapolated to probabilities for distantly related proteins (e.g., PAM250). In contrast, the BLOSUM matrices are based on empirical observations of more distantly related protein alignments. Note that a PAM30 matrix, which is available as an option on standard BLASTP searches at NCBI (Chapter 4), may be useful for identifying significant conservation between two closely related proteins. However a BLOSUM matrix with a high value (such as the BLOSUM80 matrix, available from the NCBI BLASTP site) is not necessarily suitable for scoring closely related sequences. This is because the BLOSUM80 matrix is adapted to regions of sequences that share up to 80% identity, but beyond that limited region two proteins may share dramatically less amino acid identity (Pearson and Wood, 2001).

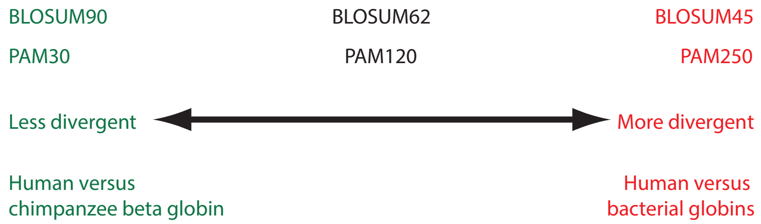


FIGURE 3.18 Summary of PAM and BLOSUM matrices. High-value BLOSUM matrices and low-value PAM matrices are best suited to study well-conserved proteins such as mouse and rat beta globin. BLOSUM matrices with low numbers (e.g., BLOSUM45) or high PAM numbers are best suited to detect distantly related proteins. Remember that in a BLOSUM45 matrix all members of a protein family with greater than 45% amino acid identity are grouped together, allowing the matrix to focus on proteins with less than 45% identity.

Pairwise Alignment and Limits of Detection: The "Twilight Zone"

When we compare two protein sequences, how many mutations can occur between them before their differences make them unrecognizable? When we compared glyceraldehyde 3-phosphate dehydrogenase proteins, it was easy to see their relationship (**Fig. 3.10**). However, when we compared human beta globin and myoglobin, the relationship was much less obvious (**Fig. 3.5**). Intuitively, at some point two homologous proteins are too divergent for their alignment to be recognized as significant.

A hit is a change in an amino acid residue that occurs by mutation. We discuss mutations (including multiple hits at a nucleotide position) in Chapter 7 (see **Fig. 7.15**). We discuss mutations associated with human disease in Chapter 21.

The best way to determine the detection limits of pairwise alignments is through statistical tests that assess the likelihood of finding a match by chance. These are described in "The Statistical Significance of Pairwise Alignments" below and in Chapter 4. In particular we will focus on the expect (E) value. It can also be helpful to compare the percent identity (and percent divergence) of two sequences versus their evolutionary distance. Consider two protein sequences, each 100 amino acids in length, in which one sequence is fixed and various numbers of mutations are introduced into the other sequence. A plot of the two diverging sequences has the form of a negative exponential (**Fig. 3.19**) (Dayhoff, 1978; Doolittle, 1987). If the two sequences have 100% amino acid identity, they have zero changes per 100 residues. If they share 50% amino acid identity, they have sustained an average of 80 changes per 100 residues. One might have expected 50 changes per 100 residues in the case of two proteins that share 50% amino acid identity. However, any position can be subject to multiple hits. Percent identity is therefore not an exact indicator of the number of mutations that have occurred across a protein sequence. When a protein sustains about 250 hits per 100 aligned amino acids (as characterized by the PAM250 matrix), it may have about 20% identity with the original protein and can still be recognizable as significantly related. If a protein sustains 360 changes per 100 residues (PAM360), it evolves to a point at which the two proteins share about 15% amino acid identity and are no longer recognizable as significantly related in a direct pairwise comparison.

The plot in **Figure 3.19** reaches an asymptote below about 15% amino acid identity. This asymptote would reach about 5% (or the average background frequency of the amino acids) if no gaps were allowed in the comparison between the proteins.

The PAM250 matrix assumes the occurrence of 250 point mutations per 100 amino acids. As shown in **Figure 3.19**, this corresponds to the "twilight zone." At this level of divergence, it is usually difficult to assess whether the two proteins are homologous. Other techniques, including multiple sequence alignment (Chapter 6) and structural predictions (Chapter 13), are often very useful to assess homology in these cases. PAM matrices are available from PAM1 to PAM250 or higher, and a specific number of observed amino acid differences per 100 residues is associated with each PAM matrix (**Table 3.3**; **Fig. 3.19**). Consider the case of the human alpha globin compared to myoglobin. These proteins are approximately 150 amino acid residues in length, and they may have undergone over

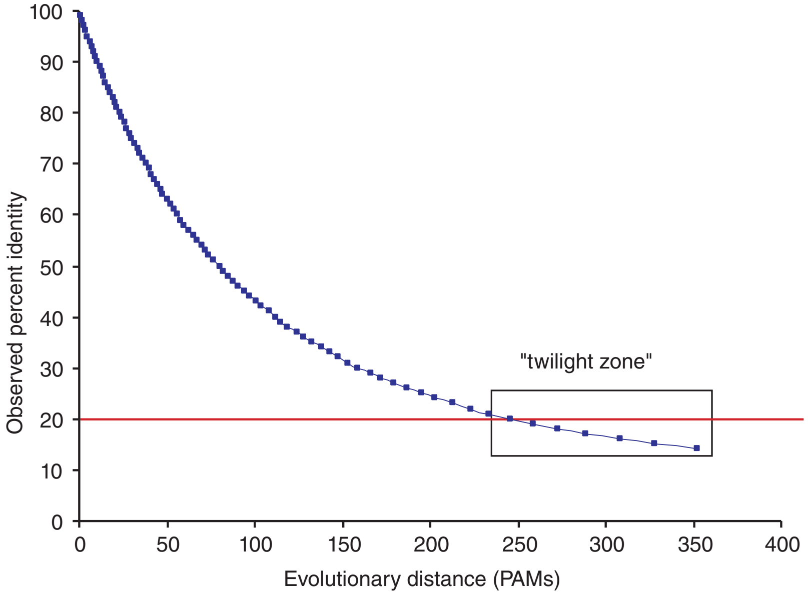


FIGURE 3.19 Two randomly diverging protein sequences change in a negatively exponential fashion. This plot shows the observed number of amino acid identities per 100 residues of two sequences (y axis) versus the number of changes that must have occurred (the evolutionary distance in PAM units). The twilight zone (Doolittle, 1987) refers to the evolutionary distance corresponding to about 20% identity between two proteins. Proteins with this degree of amino acid sequence identity may be homologous, but such homology is difficult to detect. Data from Dayhoff (1978; see **Table 3.3**).

TABLE 3.3 Relationship between observed number of amino acid differences per 100 residues of two aligned protein sequences and evolutionary difference. The number of changes that must have occurred, in PAM units.

Observed differences in 100 residues	Evolutionary distance in PAMs
1	1.0
5	5.1
10	10.7
15	16.6
20	23.1
25	30.2
30	38.0
35	47
40	56
45	67
50	80
55	94
60	112
65	133
70	159
75	195
80	246

Source: Dayhoff (1972). Reproduced with permission from National Biomedical Research Foundation.

300 amino acid substitutions since their divergence (Dayhoff *et al*., 1972, p. 19). Suppose there are 345 changes per 150 amino acids (this corresponds to 230 changes per 100 amino acids). An additional 100 changes would result in only 10 more observable differences (Dayhoff *et al*., 1972.

There are about $2^{2n}/\sqrt{\pi n}$ possible global alignments between two sequences of length *n* (Durbin *et al*., 2000; Ewins and Grant, 2001). For two sequences of length 1000, there are about 10^{600} possible alignments. For two proteins of length 200 amino acid residues, the number of possible alignments is over 6×10^{58}.

ALIGNMENT ALGORITHMS: GLOBAL AND LOCAL

Our discussion so far has focused on matrices that allow us to score an alignment between two proteins. This involves the generation of scores for identical matches, mismatches, and gaps. We also need an appropriate algorithm to perform the alignment. When two proteins are aligned, there is an enormous number of possible alignments.

There are two main types of alignment: global and local. We explore these approaches next. A *global alignment*, such as the method of Needleman and Wunsch (1970), contains the entire sequence of each protein or DNA molecule. A *local alignment*, such as the method of Smith and Waterman (1981), focuses on the regions of greatest similarity between two sequences. We saw a local alignment of human beta globin and myoglobin in **Figure 3.5** above. For many purposes, a local alignment is preferred, because only a portion of two proteins aligns. (We study the modular nature of proteins in Chapter 12.) Most database search algorithms, such as BLAST (Chapter 4), use local alignments.

Each of these methods is guaranteed to find one or more optimal solutions to the alignment of two protein or DNA sequences. We then describe two rapid-search algorithms, BLAST and FASTA. BLAST represents a simplified form of local alignment that is popular because the algorithm is very fast and easily accessible.

Global Sequence Alignment: Algorithm of Needleman and Wunsch

One of the first and most important algorithms for aligning two protein sequences was described by Needleman and Wunsch (1970). This algorithm is important because it produces an optimal alignment of protein or DNA sequences, even allowing the introduction of gaps. The result is optimal, but not all possible alignments need to be evaluated. An exhaustive pairwise comparison would be too computationally expensive to perform.

The Needleman and Wunsch approach is an example of a dynamic programming algorithm. It is called "dynamic" because the alignment is created on a residue-by-residue basis in a search for the optimal alignment. The word "programming" refers to the use of a set of rules to determine the alignment.

We can describe the Needleman–Wunsch approach to global sequence alignment in three steps: (1) setting up a matrix; (2) scoring the matrix; and (3) identifying the optimal alignment.

Step 1: Setting Up a Matrix

First, we compare two sequences in a two-dimensional matrix (**Fig. 3.20**). The first sequence, of length *m*, is listed horizontally along the *x* axis so that its amino acid residues correspond to the columns. The second sequence, of length *n*, is listed vertically along the *y* axis, with its amino acid residues corresponding to rows.

This algorithm is also sometimes called the Needleman–Wunsch–Sellers algorithm. Sellers (1974) provided a related alignment algorithm (one that focuses on minimizing differences, rather than on maximizing similarities). Smith *et al.* (1981) showed that the Needleman–Wunsch and Sellers approaches are mathematically equivalent.

We will describe rules for tracing a diagonal path through this matrix in the following section; the path describes the alignment of the two sequences. A perfect alignment between two identical sequences would simply be represented by a diagonal line extending from the top left to the bottom right (**Fig. 3.20a, b**). Any mismatches between two sequences would still be represented on this diagonal path (**Fig. 3.20c**). However, the score that is assigned might be adjusted according to some scoring system. In the example of **Figure 3.20c**, the mismatch of V and M residues might be assigned a score lower than the perfect match of M and M shown in **Figure 3.20b**.

Gaps are represented in this matrix using horizontal or vertical paths, as shown in **Figure 3.20a, d, e**. Any gap in the top sequence is represented as a vertical line (**Fig. 3.20a, d**),

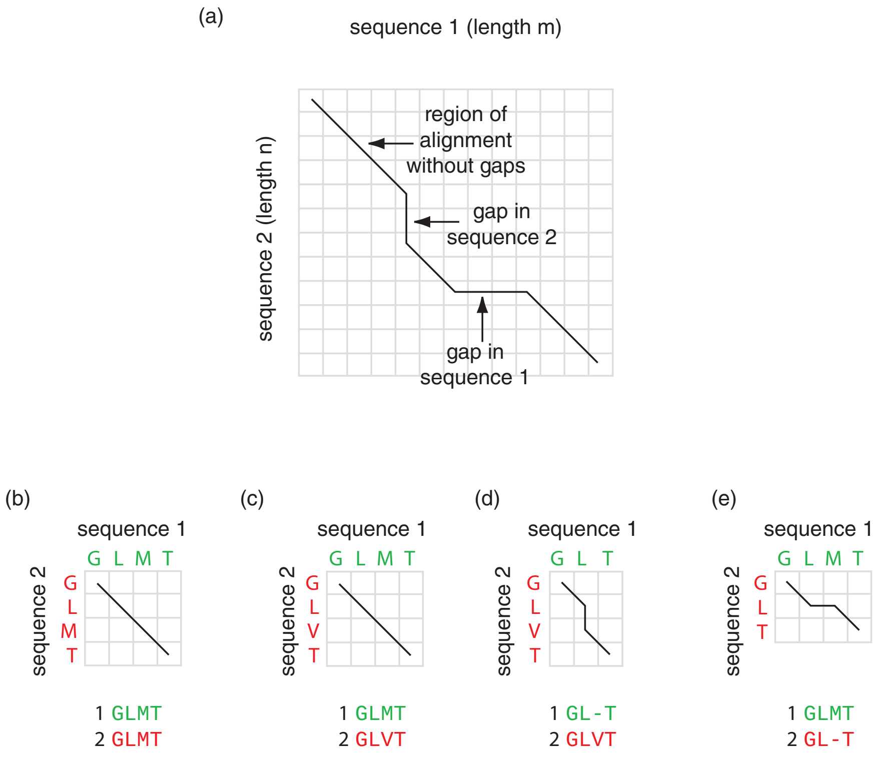


FIGURE 3.20 Pairwise alignment of two amino acid sequences using a dynamic programming algorithm of Needleman and Wunsch (1970) for global alignment. (a) Two sequences can be assigned a diagonal path through the matrix and, when necessary, the path can deviate horizontally or vertically, reflecting gaps that are introduced into the alignment. (b) Two identical sequences form a path on the matrix that fits a diagonal line. (c) If there is a mismatch (or multiple mismatches), the path still follows a diagonal, although a scoring system may penalize the presence of mismatches. If the alignment includes a gap in (d) the first sequence or (e) the second sequence, the path includes a vertical or horizontal line.

while any gap in the bottom sequence is drawn as a horizontal line (**Fig. 3.20a, e**). These gaps can be of any length. Gaps can be internal or terminal.

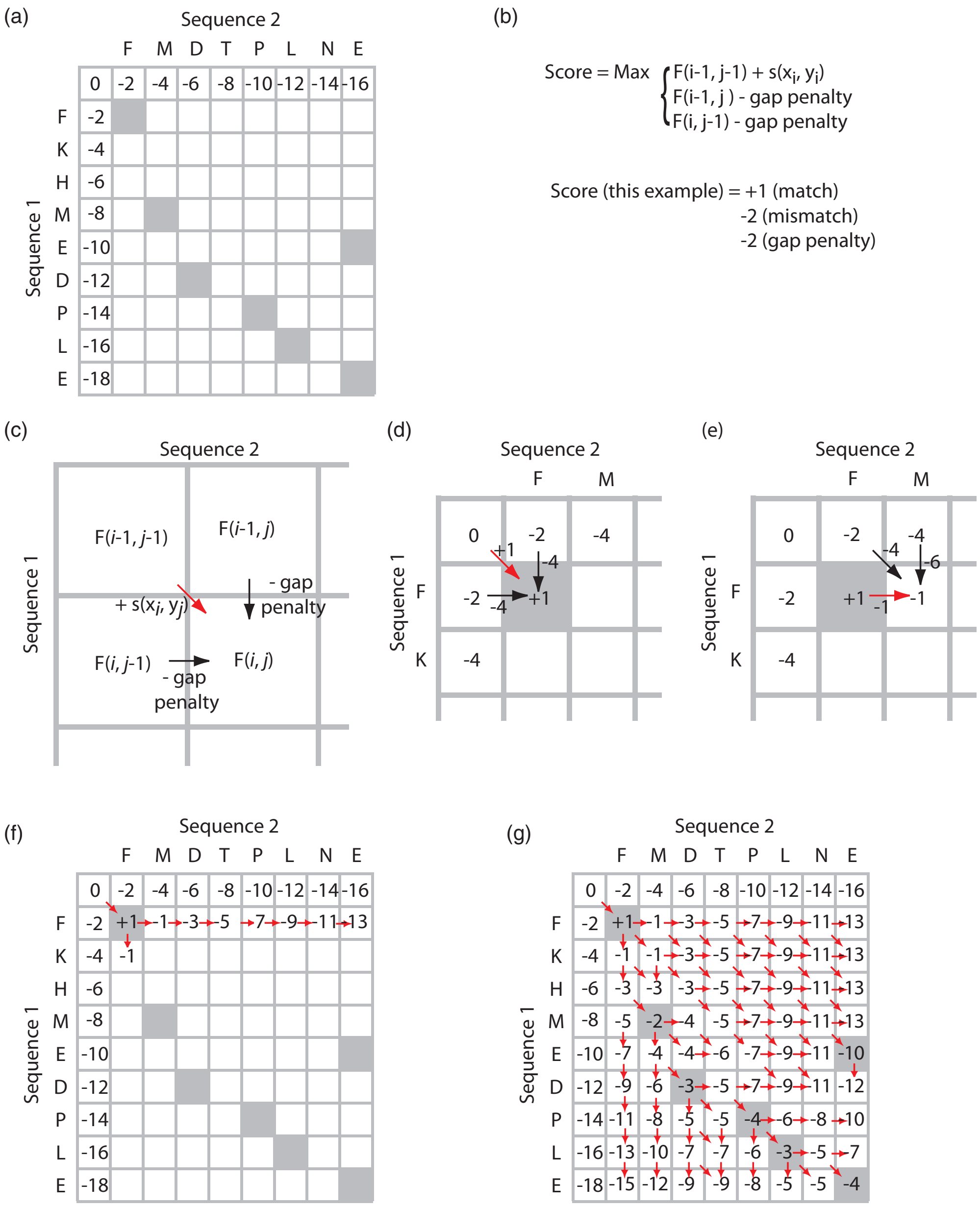
Step 2: Scoring the Matrix

The goal of this algorithm is to identify an optimal alignment. We set up two matrices: an amino acid identity matrix and then a scoring matrix. We create a matrix of dimensions $m + 1$ by $n + 1$ (for the first and second sequences on the x- and y-axes respectively; **Fig. 3.21a**). Gap penalties (here having a value of –2 for each gap position) are placed along the first row and column. This will allow us to introduce a terminal gap of any length. We fill in positions of identity (**Fig. 3.21a**, gray-filled cells); this is called an identity matrix. For two identical sequences this would include a series of gray-filled cells along the diagonal.

Note that in linear algebra an identity matrix is a special kind of number matrix that has the number 1 from top left to bottom right. For sequence alignments, the amino acid identity matrix is simply a matrix showing all the positions of shared amino acid identity between two sequences, as shown in **Fig. 3.20b**.

Next, we define a scoring system (**Fig. 3.21b**). Our goal in finding an optimal alignment is to determine the path through the matrix that maximizes the score. This entails finding a path through as many positions of identity as possible while introducing as few gaps as possible. There are four possible occurrences at each position i, j (i.e., in each cell in the matrix; **Fig. 3.21b**):

1. two residues may be perfectly matched (i.e., identical); in this example the score is +1;
2. they may be mismatched; here we assign a score of –2;



(a)

		F	M	D	T	P	L	N	E
	0	-2	-4	-6	-8	-10	-12	-14	-16
F	-2	▨							
K	-4								
H	-6								
M	-8		▨						
E	-10								▨
D	-12			▨					
P	-14					▨			
L	-16						▨		
E	-18								▨

(b)

$$\text{Score} = \text{Max} \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - \text{gap penalty} \\ F(i, j-1) - \text{gap penalty} \end{cases}$$

Score (this example) = +1 (match)
-2 (mismatch)
-2 (gap penalty)

(f)

		F	M	D	T	P	L	N	E
	0	-2	-4	-6	-8	-10	-12	-14	-16
F	-2	+1	-1	-3	-5	-7	-9	-11	-13
K	-4	-1							
H	-6								
M	-8		▨						
E	-10								▨
D	-12			▨					
P	-14					▨			
L	-16						▨		
E	-18								▨

(g)

		F	M	D	T	P	L	N	E
	0	-2	-4	-6	-8	-10	-12	-14	-16
F	-2	+1	-1	-3	-5	-7	-9	-11	-13
K	-4	-1	-1	-3	-5	-7	-9	-11	-13
H	-6	-3	-3	-3	-5	-7	-9	-11	-13
M	-8	-5	-2	-4	-5	-7	-9	-11	-13
E	-10	-7	-4	-4	-6	-7	-9	-11	-10
D	-12	-9	-6	-3	-5	-7	-9	-11	-12
P	-14	-11	-8	-5	-5	-4	-6	-8	-10
L	-16	-13	-10	-7	-7	-6	-3	-5	-7
E	-18	-15	-12	-9	-9	-8	-5	-5	-4

FIGURE 3.21 Pairwise alignment of two amino acid sequences using the dynamic programming algorithm of Needleman and Wunsch (1970) for global alignment. (a) For sequences of length m and n we form a matrix of dimensions *m* + 1 by *n* + 1 and add gap penalties in the first row and column. Each gap position receives a score of –2. The cells having identity are shaded gray. (b) The scoring system in this example is +1 for a match, –2 for a mismatch, and –2 for a gap penalty. In each cell, the score is assigned using the recursive algorithm that identifies the highest score from three calculations. (c) In each cell F(*i*, *j*) we calculate the scores derived from following a path from the upper left cell (we add the score of that cell + the score of F(*i*, *j*)), the cell to the left (including a gap penalty), and the cell directly above (again including a gap penalty). (d) To calculate the score in the cell of the second row and column, we take the maximum of the three scores +1, –4, –4. This best score (+1) follows the path of the red arrow, and we maintain the information of the best path, resulting in each cell's score in order to later reconstruct the pairwise alignment. (e) To calculate the score in the second row, third column we again take the maximum of the three scores –4, –1, –4. The best score follows from the left cell (red arrow). (f) We proceed to fill in scores across the first row of the matrix. (g) The completed matrix includes the overall score of the optimal alignment (–4; see cell at bottom right, corresponding to the carboxy terminus of each protein). Red arrows indicate the path(s) by which the highest score for each cell was obtained.

3. a gap may be introduced from the first sequence, for which we assign a score of −2; or
4. a gap may be introduced from the second sequence, also resulting in −2.

The Needleman and Wunsch algorithm provides a score corresponding to each of these possible outcomes for each position of the aligned sequences. The algorithm also specifies a set of rules describing how we can move through the matrix.

Consider the cell at the lower right-hand corner of **Figure 3.21c**. There are several rules for deciding the optimal score:

- First, both *i* and *j* must increase. We therefore evaluate scores from three positions (top, left, upper left), moving towards a given cell F(*i*, *j*). It would not make sense to be able to violate the linear arrangement of amino acids (or nucleotides) in a sequence.
- It is acceptable for a gap to extend an arbitrary number of positions; a scoring system may include separate gap creation and gap extension penalties.
- The particular score that is assigned may come from a scoring matrix such as BLOSUM62.

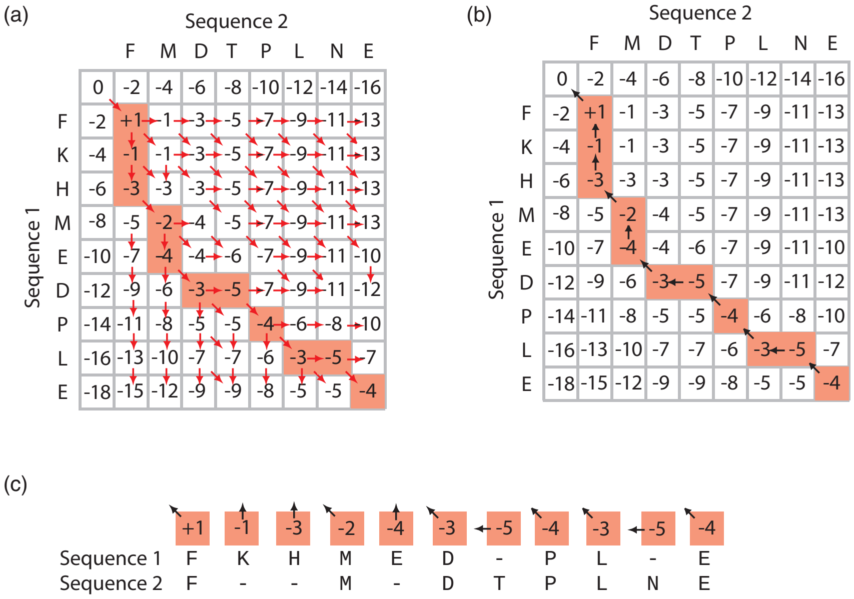
As we begin to align the two sequences in our example we fill in a cell with the value +1 because of the alignment of two F residues (**Fig. 3.21d**). The alternative options of introducing a gap in either sequence would necessitate a gap penalty and a poorer score. We indicate the preferred (highest-scoring) path with a red arrow throughout **Figure 3.21**. We proceed to the next cell to the right, selecting the score of −1 (coming from the left, consisting of +1 (from the previous cell) −2 (for introducing a gap) = −1) as better than the alternative scores of −4 and −6 (**Fig. 3.21e**). This process of analyzing possible scores for each cell continues across each row (**Fig. 3.21f**) until the entire matrix is filled in (**Fig. 3.21g**).

Step 3: Identifying the Optimal Alignment

After the matrix is filled, the alignment is determined by a trace-back procedure. Begin with the cell at the lower right of the matrix (carboxy termini of the proteins or 3’ end of the nucleic acid sequences). In our example, this has a score of −4 and corresponds to an alignment of two glutamate residues. For this and every cell we can determine from which of the three adjacent cells the best score was derived. This procedure is outlined in **Figure 3.22a**, in which red arrows indicate the paths from which the best scores were obtained for each cell. We therefore define a path (see pink-shaded cells) that will correspond to the actual alignment. In **Figure 3.22b**, we show just the arrows indicating from which cell each best score was derived. This is a different way of defining the optimal path of the pairwise alignment. We build that alignment, including gaps in either sequence, proceeding from the carboxy to the amino terminus. The final alignment (**Fig. 3.22c**) is guaranteed to be optimal, given this scoring system. There may be multiple alignments that share an optimal score, although this rarely occurs when scoring matrices such as BLOSUM62 are implemented.

A variety of programs implement global alignment algorithms (see Web Resources at the end of this chapter). An example is the Needle program from EMBOSS, which can be accessed via Galaxy (Box 3.9). Two bacterial globin family sequences are entered: one from *Streptomyces avermitilis* MA-4680 (NP_824492.1, 260 amino acids); and another from *Mycobacterium tuberculosis* CDC1551 (NP_337032.1, 134 amino acids). Penalties are selected for gap creation and extension, and each sequence is pasted into an input box in the FASTA format. The resulting global alignment includes descriptions of the percent identity and similarity shared by the two proteins, the length of the alignment, and the number of gaps introduced (**Fig. 3.23a**).

The Needle program for global pairwise alignment is part of the EMBOSS package available online at the European Bioinformatics Institute (⊕ http://www.ebi.ac.uk/emboss/align/, WebLink 3.5) or at Galaxy (⊕ http://usegalaxy.org, WebLink 3.6). It is further described at the EMBOSS website under applications (⊕ http://emboss.sourceforge.net/, WebLink 3.7). The *E. coli* and *S. cerevisiae* proteins are available in the FASTA format, as well as globally and locally aligned in Web Document 3.6 (⊕ http://www.bioinfbook.org/chapter3).



(a)

		F	M	D	T	P	L	N	E
	0	-2	-4	-6	-8	-10	-12	-14	-16
F	-2	+1	-1	-3	-5	-7	-9	-11	-13
K	-4	-1	-1	-3	-5	-7	-9	-11	-13
H	-6	-3	-3	-3	-5	-7	-9	-11	-13
M	-8	-5	-2	-4	-5	-7	-9	-11	-13
E	-10	-7	-4	-4	-6	-7	-9	-11	-10
D	-12	-9	-6	-3	-5	-7	-9	-11	-12
P	-14	-11	-8	-5	-5	-4	-6	-8	-10
L	-16	-13	-10	-7	-7	-6	-3	-5	-7
E	-18	-15	-12	-9	-9	-8	-5	-5	-4

(b)

		F	M	D	T	P	L	N	E
	0	-2	-4	-6	-8	-10	-12	-14	-16
F	-2	+1	-1	-3	-5	-7	-9	-11	-13
K	-4	-1	-1	-3	-5	-7	-9	-11	-13
H	-6	-3	-3	-3	-5	-7	-9	-11	-13
M	-8	-5	-2	-4	-5	-7	-9	-11	-13
E	-10	-7	-4	-4	-6	-7	-9	-11	-10
D	-12	-9	-6	-3	-5	-7	-9	-11	-12
P	-14	-11	-8	-5	-5	-4	-6	-8	-10
L	-16	-13	-10	-7	-7	-6	-3	-5	-7
E	-18	-15	-12	-9	-9	-8	-5	-5	-4

(c)

	+1	-1	-3	-2	-4	-3	-5	-4	-3	-5	-4
Sequence 1	F	K	H	M	E	D	-	P	L	-	E
Sequence 2	F	-	-	M	-	D	T	P	L	N	E

FIGURE 3.22 Global pairwise alignment of two amino acid sequences using a dynamic programming algorithm: scoring the matrix and using the trace-back procedure to obtain the alignments. (a) The alignment of **Figure 3.21(g)** is shown. The cells highlighted in pink represent the source of the optimal scores. (b) In an equivalent representation, arrows point back to the source of each cell's optimal score. (c) This trace-back allows us to determine the sequence of the optimal alignment. Vertical or horizontal arrows correspond to the positions of gap insertions, while diagonal lines correspond to exact matches (or mismatches). Note that the final score (–4) equals the sum of matches (6 × 1 = 6), mismatches (none in this example), and gaps (5 × –2 = –10).

The Needleman–Wunsch algorithm is an example of dynamic programming (Sedgewick, 1988). This means that an optimal path (i.e., an optimal alignment) is detected by incrementally extending optimal subpaths, that is, by making a series of decisions at each step of the alignment as to which pair of residues corresponds to the best score. The overall goal is to find the path moving along the diagonal of the matrix that lets us obtain the maximal score. This path specifies the optimal alignment.

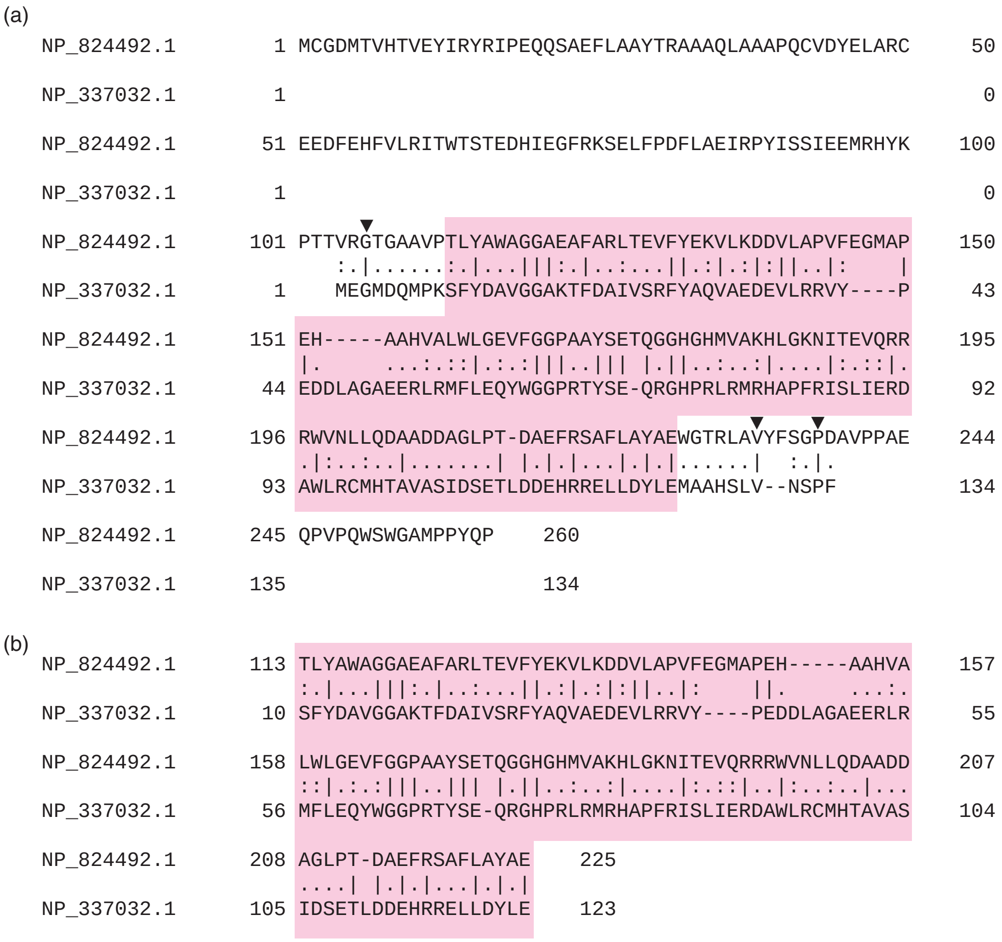
BOX 3.9 EMBOSS

EMBOSS (European Molecular Biology Open Software Suite) is a collection of freely available programs for DNA, RNA, or protein sequence analysis (Rice et al., 2000). There are over 200 available programs in three dozen categories. The home page of EMBOSS (⊕ http://emboss.sourceforge.net/, WebLink 3.21) describes the various packages. A variety of web servers offer EMBOSS, including Galaxy. You can also visit sites such as ⊕ http://emboss.bioinformatics.nl/ (WebLink 3.22) and ⊕ http://www.bioinformatics2.wsu.edu/emboss/ (WebLink 3.23).

To perform pairwise sequence alignment using EMBOSS at Galaxy, try the following steps:

1. Visit Galaxy at ⊕ https://main.g2.bx.psu.edu/ (WebLink 3.24) and sign in.
2. On the left sidebar (Tools menu) select Get Data and choose UCSC Main. From the human genome (hg19) select the RefGenes table, enter hbb for the position (upon clicking "lookup" the coordinates chr11:5246696-5248301 are added), set the output format to "sequence" and check the box to send to Galaxy. When you click "Get output" select protein and submit.
3. Repeat step (2) to import the HBA2 protein. For both proteins in Galaxy, use Edit Attributes (the pencil icon in the history panel) to rename the sequences hbb and hba2.
4. In the tools panel choose EMBOSS, and scroll to find the water tool for Smith–Waterman local alignment. Alternatively, enter "water" into the Tools search box. Select the two proteins, use default settings, and click Execute. The pairwise alignment is returned.

Once you have entered one or more sequences into Galaxy, explore some of the >100 other EMBOSS tools!



(a)

```
NP_824492.1        1 MCGDMTVHTVEYIRYRIPEQQSAEFLAAYTRAAAQLAAAPQCVDYELARC     50

NP_337032.1        1                                                        0

NP_824492.1       51 EEDFEHFVLRITWTSTEDHIEGFRKSELFPDFLAEIRPYISSIEEMRHYK    100

NP_337032.1        1                                                        0
                          ▼
NP_824492.1      101 PTTVRGTGAAVPTLYAWAGGAEAFARLTEVFYEKVLKDDVLAPVFEGMAP    150
                        :.|......:.|...|||:.|..:...||.:|.:|:||..|:    |
NP_337032.1        1    MEGMDQMPKSFYDAVGGAKTFDAIVSRFYAQVAEDEVLRRVY----P     43

NP_824492.1      151 EH-----AAHVALWLGEVFGGPAAYSETQGGHGHMVAKHLGKNITEVQRR    195
                     |.     ...:.::|.:.:|||..||| |.||..:..:|....|:.::|.
NP_337032.1       44 EDDLAGAEERLRMFLEQYWGGPRTYSE-QRGHPRLRMRHAPFRISLIERD     92
                                                          ▼    ▼
NP_824492.1      196 RWVNLLQDAADDAGLPT-DAEFRSAFLAYAEWGTRLAVYFSGPDAVPPAE    244
                     .|:..:..|.......| |.|.|...|.|.|......|  :.|.
NP_337032.1       93 AWLRCMHTAVASIDSETLDDEHRRELLDYLEMAAHSLV--NSPF          134

NP_824492.1      245 QPVPQWSWGAMPPYQP    260

NP_337032.1      135                     134
```

(b)

```
NP_824492.1      113 TLYAWAGGAEAFARLTEVFYEKVLKDDVLAPVFEGMAPEH-----AAHVA    157
                     :.|...|||:.|..:...||.:|.:|:||..|:    ||.     ...:.
NP_337032.1       10 SFYDAVGGAKTFDAIVSRFYAQVAEDEVLRRVY----PEDDLAGAEERLR     55

NP_824492.1      158 LWLGEVFGGPAAYSETQGGHGHMVAKHLGKNITEVQRRRWVNLLQDAADD    207
                     ::|.:.:|||..||| |.||..:..:|....|:.::|..|:..:..|...
NP_337032.1       56 MFLEQYWGGPRTYSE-QRGHPRLRMRHAPFRISLIERDAWLRCMHTAVAS    104

NP_824492.1      208 AGLPT-DAEFRSAFLAYAE    225
                     ....| |.|.|...|.|.|
NP_337032.1      105 IDSETLDDEHRRELLDYLE    123
```

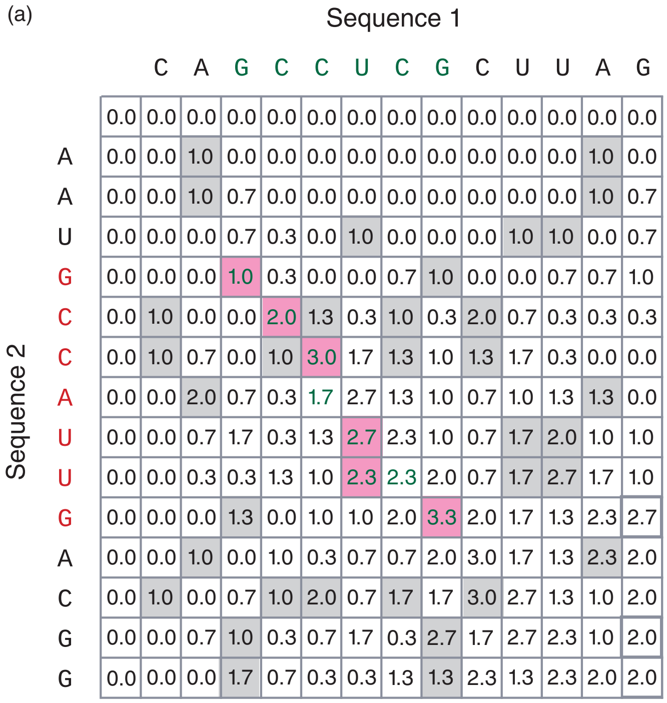
FIGURE 3.23 (a) Global pairwise alignment of bacterial proteins containing globin domains from *Streptomyces avermitilis* MA-4680 (NP_824492) and *Mycobacterium tuberculosis* CDC1551 (NP_337032). The scoring matrix was BLOSUM62. The aligned proteins share 14.7% identity (39/266 aligned residues), 22.6% similarity (60.266), and 51.9% gaps (138/266). (b) A local pairwise alignment of these two sequences lacks the unpaired amino- and carboxy-terminal extensions and shows 30% identity (35/115 aligned residues). The alignment in (b) corresponds to the shaded region of (a). The arrowheads in (a) indicate aligned residues that were not seen in the local alignment. In performing local alignments (as is done in BLAST, Chapter 4) some authentically aligned regions may therefore be missed.

Local Sequence Alignment: Smith and Waterman Algorithm

The local alignment algorithm of Smith and Waterman (1981) is the most rigorous method by which subsets of two protein or DNA sequences can be aligned. Local alignment is extremely useful in a variety of applications such as database searching in which we may wish to align domains of proteins (but not the entire sequences). A local sequence alignment algorithm resembles that for global alignment in that two proteins are arranged in a matrix and an optimal path along a diagonal is sought. However, there is no penalty for starting the alignment at some internal position, and the alignment does not necessarily extend to the ends of the two sequences.

For the Smith–Waterman algorithm a matrix is constructed with an extra row along the top and an extra column on the left side. For sequences of lengths m and n, the matrix has dimensions $m + 1$ and $n + 1$. The rules for defining the value at each position of the matrix differ slightly from those used in the Needleman–Wunsch algorithm. The score in

(a)


Sequence 1
Sequence 2

		C	A	G	C	C	U	C	G	C	U	U	A	G
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
A	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
A	0.0	0.0	1.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.7
U	0.0	0.0	0.0	0.7	0.3	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.7
G	0.0	0.0	0.0	1.0	0.3	0.0	0.0	0.7	1.0	0.0	0.0	0.7	0.7	1.0
C	0.0	1.0	0.0	0.0	2.0	1.3	0.3	1.0	0.3	2.0	0.7	0.3	0.3	0.3
C	0.0	1.0	0.7	0.0	1.0	3.0	1.7	1.3	1.0	1.3	1.7	0.3	0.0	0.0
A	0.0	0.0	2.0	0.7	0.3	1.7	2.7	1.3	1.0	0.7	1.0	1.3	1.3	0.0
U	0.0	0.0	0.7	1.7	0.3	1.3	2.7	2.3	1.0	0.7	1.7	2.0	1.0	1.0
U	0.0	0.0	0.3	0.3	1.3	1.0	2.3	2.3	2.0	0.7	1.7	2.7	1.7	1.0
G	0.0	0.0	0.0	1.3	0.0	1.0	1.0	2.0	3.3	2.0	1.7	1.3	2.3	2.7
A	0.0	0.0	1.0	0.0	1.0	0.3	0.7	0.7	2.0	3.0	1.7	1.3	2.3	2.0
C	0.0	1.0	0.0	0.7	1.0	2.0	0.7	1.7	1.7	3.0	2.7	1.3	1.0	2.0
G	0.0	0.0	0.7	1.0	0.3	0.7	1.7	0.3	2.7	1.7	2.7	2.3	1.0	2.0
G	0.0	0.0	0.0	1.7	0.7	0.3	0.3	1.3	1.3	2.3	1.3	2.3	2.0	2.0

(b)

```
sequence 1   GCC-UCG
sequence 2   GCCAUUG
```

(c)

```
sequence 1   CA-GCC-UCGCUUAG
sequence 2   AAUGCCAUUGACG-G
```

FIGURE 3.24 Local sequence alignment method of Smith and Waterman (1981). (a) In this example, the matrix is formed from two RNA sequences (CAGCCUCGCUUAG and AAUGCCAUUGACGG). While this is not an identity matrix (such as that shown in **Fig. 3.21a**), positions of nucleotide identity are shaded gray (or shaded pink in the region of local alignment). They scoring system here is +1 for a match, minus one-third for a mismatch, and a gap penalty of the difference between a match and a mismatch (−1.3 for a gap of length one). The matrix is scored based on finding the maximum of four possible non-negative values. The highest value in the matrix (3.3) corresponds to the beginning of the optimal local alignment, and the aligned residues (green font) extend up and to the left until a value of zero is reached. (b) The local alignment derived from this matrix is shown. Note that this alignment includes identities, a mismatch, and a gap. (c) A global alignment of the two sequences is shown for comparison to the local alignment. Note that it encompasses the entirety of both sequences.

Source: Adapted from Smith and Waterman (1981). Reproduced with permissions from Elsevier.

each cell is selected as the maximum of the preceding diagonal or the score obtained from the introduction of a gap. However, the score cannot be negative: a rule introduced by the Smith–Waterman algorithm is that, if all other score options produce a negative value, then a zero must be inserted in the cell. The score $S(i,j)$ is given as the maximum of four possible values (**Fig. 3.24**):

1. The score from the cell at position $i - 1, j - 1$, that is, the score diagonally up to the left. To this score, add the new score at position $s[i, j]$, which consists of either a match or a mismatch.
2. $S(i, j - 1)$ (i.e., the score one cell to the left) minus a gap penalty.

3. $S(i-1, j)$ (i.e., the score immediately above the new cell) minus a gap penalty.
4. The number zero.

This condition ensures that there are no negative values in the matrix. In contrast, negative numbers commonly occur in global alignments because of gap or mismatch penalties (note the log-odds matrices in this chapter).

An example of the use of a local alignment algorithm to align two nucleic acid sequences adapted from Smith and Waterman (1981) is shown in **Figure 3.24**. The topmost row and the leftmost column of the matrix are filled with zeros. The maximal alignment can begin and end anywhere in the matrix (within reason; the linear order of the two amino acid sequences cannot be violated). The procedure is to identify the highest value in the matrix (this value is 3.3 in **Fig. 3.24a**). This represents the end (3' end for nucleic acids, or carboxy-terminal portion proteins) of the alignment. This position is not necessarily at the lower right corner as it must be for a global alignment. The trace-back procedure begins with this highest-value position and proceeds diagonally up to the left until a cell is reached with a value of zero. This defines the start of the alignment, and it is not necessarily at the extreme top left of the matrix.

An example of a local alignment of two proteins using the Smith–Waterman algorithm is shown in **Figure 3.23b**. Compare this with the global alignment of **Figure 3.23a** and note that the aligned region is shorter for the local alignment, while the percent identity and similarity are higher. Note also that the local alignment ignores several identically matching residues (**Fig. 3.23a**, arrowheads). Since database searches such as BLAST (Chapter 4) rely on local alignments, there may be conserved regions that are not reported as aligned, depending on the chosen search parameters.

Rapid, Heuristic Versions of Smith–Waterman: FASTA and BLAST

While the Smith–Waterman algorithm is guaranteed to find the optimal alignment(s) between two sequences, it suffers from the fact that it is relatively slow. For pairwise alignment, speed is usually not a problem. When a pairwise alignment algorithm is applied to the problem of comparing one sequence (a "query") to an entire database however, the speed of the algorithm becomes a significant issue and may vary by orders of magnitude.

Most algorithms have a parameter N that refers to the number of data items to be processed (see Sedgewick, 1988). This parameter can greatly affect the time required for the algorithm to perform a task. If the running time is proportional to N, then doubling N doubles the running time. If the running time is quadratic (N^2), then for $N = 1000$ the running time is one million. For both the Needleman–Wunsch and the Smith–Waterman algorithms, both the computer space and the time required to align two sequences is proportional to at least the length of the two query sequences multiplied by each other ($m \times n$). For the search of a database of size N, this is $m \times N$.

The modified alignment algorithms introduced by Gotoh (1982) and Myers and Miller (1988) require only $O(nm)$ time and occupy $O(n)$ in space. Instead of committing the entire matrix to memory, the algorithms ignore scores below a threshold in order to focus on the maximum scores that are achieved during the search.

Another useful descriptor is O-notation (called "big-Oh notation") which provides an approximation of the upper bounds of the running time of an algorithm. The Needleman–Wunsch algorithm requires $O(mn)$ steps, while the Smith–Waterman algorithm requires $O(m^2n)$ steps. Subsequently, Gotoh (1982) and Myers and Miller (1988) improved the algorithms so they require less time and space.

FASTA stands for FAST-All, referring to its ability to perform a fast alignment of all sequences (i.e., proteins or nucleotides).

Two popular local alignment algorithms have been developed that provide rapid alternatives to Smith–Waterman: FASTA (Pearson and Lipman, 1988) and BLAST (Basic Local Alignment Search Tool; Altschul *et al*., 1990). Each of these algorithms requires less time to perform an alignment. The time saving occurs because FASTA and BLAST restrict the search by scanning a database for likely matches before performing more

The parameter *ktup* refers to multiples such as duplicate, triplicate, or quadruplicate (for $k=2$, $k=3$, $k=4$). The *ktup* values are usually 3–6 for nucleotide sequences and 1–2 for amino acid sequences. A small *ktup* value yields a more sensitive search but requires more time to complete.

William Pearson of the University of Virginia provides FASTA online. Visit ⊕ http://fasta.bioch.virginia.edu/fasta_www2/fasta_list2.shtml (WebLink 3.8). Another place to try FASTA is at the European Bioinformatics Institute website, ⊕ http://www.ebi.ac.uk/fasta33/ (WebLink 3.9).

Dotlet is a web-based diagonal plot tool available from the Swiss Institute of Bioinformatics (⊕ http://myhits.isb-sib.ch/cgi-bin/dotlet, WebLink 3.10). It was written by Marco Pagni and Thomas Junier. The website provides examples of the use of Dotlet to visualize repeated domains, conserved domains, exons and introns, terminators, frameshifts, and low-complexity regions.

The accession number of the snail globin is CAJ44466.1,while the accession of human cytoglobin is NP_599030.1.

rigorous alignments. These are heuristic algorithms (Box 3.3) that sacrifice some sensitivity in exchange for speed; in contrast to Smith–Waterman, they are not guaranteed to find optimal alignments.

The FASTA search algorithm introduced by Pearson and Lipman (1988) proceeds in four steps.

1. A lookup table is generated consisting of short stretches of amino acids or nucleotides from a database. The size of these stretches is determined from the *ktup* parameter. If *ktup* = 3 for a protein search, then the query sequence is examined in blocks of three amino acids against matches of three amino acids found in the lookup table. The FASTA program identifies the 10 highest scoring segments that align for a given *ktup*.
2. These 10 aligned regions are rescored, allowing for conservative replacements, using a scoring matrix such as PAM250.
3. High-scoring regions are joined together if they are part of the same proteins.
4. FASTA then performs a global (Needleman–Wunsch) or local (Smith–Waterman) alignment on the highest scoring sequences, thus optimizing the alignments of the query sequence with the best database matches.

Dynamic programming is therefore applied to the database search in a limited fashion, allowing FASTA to return its results very rapidly because it evaluates only a portion of the potential alignments.

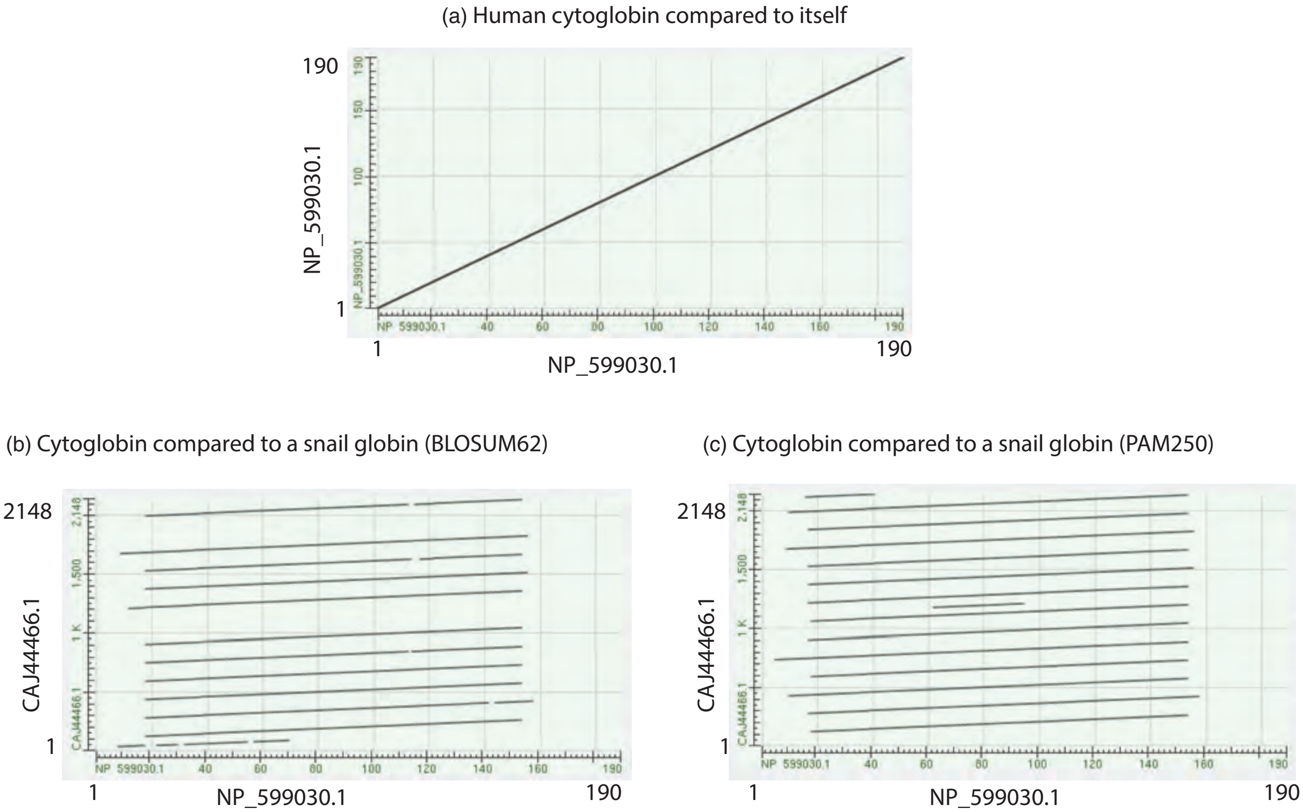
Basic Local Alignment Search Tool (BLAST)

BLAST was introduced as a local alignment search tool that identifies alignments between a query sequence and a database without the introduction of gaps (Altschul *et al*., 1990). The version of BLAST that is available today allows gaps in the alignment. We provided an example of an alignment of two proteins (**Figs 3.4** and **3.5**) and introduce BLAST in more detail in Chapter 4, where we describe its heuristic algorithm.

Pairwise Alignment with Dotplots

In addition to displaying a pairwise alignment, the BLAST output includes a dotplot (or dot matrix), which is a graphical method for comparing two sequences. One protein or nucleic acid sequence is placed along the *x* axis and the other is placed along the *y* axis. Positions of identity are scored with a dot. A region of identity between two sequences results in the formation of a diagonal line. This is illustrated for an alignment of human cytoglobin with itself as part of the BLAST output (**Fig. 3.25a**). We also illustrate a dotplot using the web-based Dotlet program of Junier and Pagni (2000; Web Document 3.7). Dotlet features an adjustable sliding window size, a zoom feature, a variety of scoring matrices, and a histogram window to adjust the pixel intensities in order to manually optimize the signal-to-noise ratio.

We can further illustrate the usefulness of dotplots by examining an unusual hemoglobin protein of 2148 amino acids from the snail *Biomphalaria glabrata*. It consists of 13 globin repeats (Lieb *et al*., 2006). When we compare it to human cytoglobin (190 amino acids) with a default BLOSUM62 matrix, the BLAST output shows cytoglobin (*x* axis) matching the snail protein 12 times (*y* axis) (**Fig. 3.25b**); one repeat is missed. By changing the scoring matrix to BLOSUM45 we can now see all 13 snail hemoglobin repeats (**Fig. 3.25c**). The gap at the start of the dotplot (**Fig. 3.25c**, position 1 to the first red arrowhead on *x* axis) is evident in the pairwise alignment of that region (**Fig. 3.25d**): the first 128 amino acids of the snail protein are unrelated and therefore not aligned with cytoglobin. Using Dotlet, all 13 globin repeats are evident in a comparison of the snail protein with itself or with cytoglobin (Web Document 3.7).

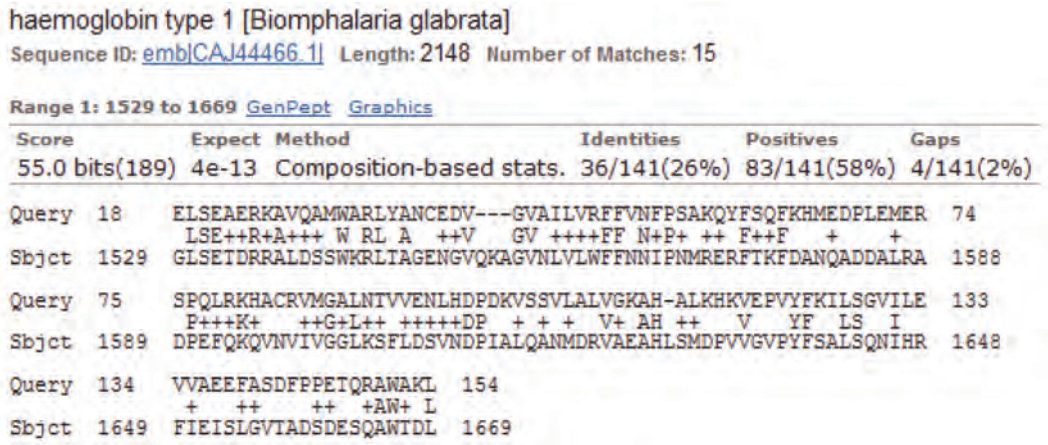


(a) Human cytoglobin compared to itself

(b) Cytoglobin compared to a snail globin (BLOSUM62)

(c) Cytoglobin compared to a snail globin (PAM250)

(d) Pairwise alignment (human cytoglobin to one repeat of a snail globin)



haemoglobin type 1 [Biomphalaria glabrata]

Sequence ID: emb|CAJ44466.1| Length: 2148 Number of Matches: 15

Range 1: 1529 to 1669 GenPept Graphics

Score	Expect	Method	Identities	Positives	Gaps
55.0 bits(189)	4e-13	Composition-based stats.	36/141(26%)	83/141(58%)	4/141(2%)

```
Query  18    ELSEAERKAVQAMWARLYANCEDV---GVAILVRFFVNFPSAKQYFSQFKHMEDPLEMER  74
              LSE++R+A+++ W RL A  ++V   GV ++++FF N+P+ ++ F++F    +    +
Sbjct  1529  GLSETDRRALDSSWKRLTAGENGVQKAGVNLVLWFFNNIPNMRERFTKFDANQADDALRA  1588

Query  75    SPQLRKHACRVMGALNTVVENLHDPDKVSSVLALVGKAH-ALKHKVEPVYFKILSGVILE  133
              P+++K+    ++G+L++ +++++DP  + + +  V+ AH ++    V   YF  LS  I
Sbjct  1589  DPEFQKQVNVIVGGLKSFLDSVNDPIALQANMDRVAEAHLSMDPVVGVPYFSALSQNIHR  1648

Query  134   VVAEEFASDFPPETQRAWAKL  154
              +   ++     ++  +AW+ L
Sbjct  1649  FIEISLGVTADSDESQAWTDL  1669
```

FIGURE 3.25 Dot matrix plots in the output of the NCBI BLASTP program permit visualization of matching domains in pairwise protein alignments. The program is used as described in **Figure 3.4**. (a) For a comparison of human cytoglobin (NP_599030.1, length 190 amino acids) with itself, the output includes a dotplot shown with sequences 1 and 2 (both cytoglobin) on the *x* and *y* axes, and the data points showing amino acid identities appear as a diagonal line. (b) For a comparison of cytoglobin with a globin from the snail *Biomphalaria glabrata* (accession CAJ44466.1, length 2148 amino acids), the cytoglobin sequence (*x* axis) matches 12 times with internal globin repeats in the snail protein. This search uses the default BLOSUM62 scoring matrix. (c) Changing the scoring matrix to PAM250 enables all 13 globin repeats of the snail protein to be aligned with cytoglobin. (d) A pairwise alignment of the sequences shows that the snail globin repeats align with residues 18–154 of cytoglobin. This is reflected in the dotplots, where the portion on the x axis corresponding to cytoglobin residues 1–17 and 155–190 (see red arrowheads in (c)) do not align to the snail sequence. The BLASTP output produces a set of all the pairwise alignments of which the first is shown here.

Source: BLASTP, NCBI.

We encounter dotplots in Chapter 16 when we compare viral genome sequences to each other. We also see a dotplot in Chapter 18 (on fungi). Protein sequences from *Saccharomyces cerevisiae* chromosomes were systematically BLASTP searched against each other. The resulting dotplot (**Fig. 18.10**) showed many diagonal lines, indicating homologous regions. This provided evidence that, surprisingly, the entire yeast genome duplicated over 100 million years ago.

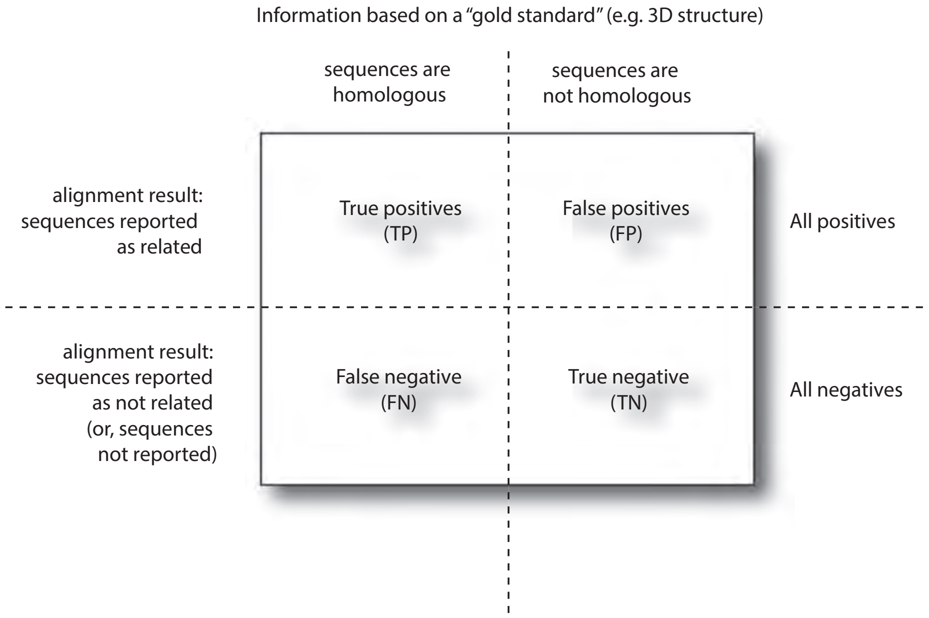

Information based on a "gold standard" (e.g. 3D structure)
sequences are homologous
sequences are not homologous
alignment result: sequences reported as related
True positives (TP)
False positives (FP)
All positives
alignment result: sequences reported as not related (or, sequences not reported)
False negative (FN)
True negative (TN)
All negatives

FIGURE 3.26 Sequences alignments, whether pairwise (this chapter) or from a database search (Chapter 4), can be classified as true or false and positives or negatives. Statistical analyses of alignments provide the main method of evaluating whether an alignment represents a true positive, that is, an alignment of homologous sequences. Ideally, an alignment algorithm can maximize both sensitivity and specificity.

THE STATISTICAL SIGNIFICANCE OF PAIRWISE ALIGNMENTS

How can we decide whether the alignment of two sequences is statistically significant? We address this question for local alignments and then for global alignments.

Consider two proteins that share limited amino acid identity (e.g., 20–25%). Alignment algorithms report the score of a pairwise alignment or the score of the best alignments of a query sequence against an entire database of sequences (Chapter 4). We need statistical tests to decide whether the matches are true positives (i.e., whether the two aligned proteins are genuinely homologous) or whether they are false positives (i.e., whether they have been aligned by the algorithm by chance; **Fig. 3.26**). For the alignments that are not reported by an algorithm, for instance because the score falls below some threshold, we would like to evaluate whether the sequences are true negatives (i.e., genuinely unrelated) or whether they are false negatives, that is, homologous sequences that receive a score suggesting that they are not homologous.

A main goal of alignment algorithms is therefore to maximize the sensitivity and specificity of sequence alignments (**Fig. 3.26**). Sensitivity is the number of true positives divided by the sum of true positive and false negative results. This is a measure of the ability of an algorithm to correctly identify genuinely related sequences. Specificity is the number of true negative results divided by the sum of true negative and false positive results. This describes the sequence alignments that are not homologous.

Statistical Significance of Global Alignments

When we align two proteins, such as human beta globin and myoglobin, we obtain a score. We can use hypothesis testing to assess whether that score is likely to have occurred by chance. To do this, we first state a null hypothesis (H_0) that the two sequences are not related. According to this hypothesis, the score S of beta globin and myoglobin represents a chance occurrence. We then state an alternative hypothesis (H_1) that they are indeed

BOX 3.10 STATISTICAL CONCEPTS: Z-SCORES

The familiar bell-shaped curve is a Gaussian distribution or normal distribution. The *x* axis corresponds to some measured values, such as the alignment score of beta globin versus 100 randomly shuffled versions of myoglobin. The *y* axis corresponds to the probability density (when considering measurements of an exhaustive set of shuffled myoglobins) or to the number of trials (when considering a number of shuffled myoglobins). The mean value is obtained simply by adding all the scores and dividing by the number of pairwise alignments; it is apparent at the center of a Gaussian distribution. For a set of data points $x_1, x_2, x_3, \ldots x_n$ the mean $\overline{x}$ is the sum divided by *n*, or:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}.$$

The sample variance s^2 describes the spread of the data points from the mean. It is related to the squares of the distances of the data points from the mean, and it is given by:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_{\mathrm{i}} - \overline{x})^2.$$

The sample standard deviation *s* is the square root of the variance, so its units match those of the data points. It is defined:

$$s = \sqrt{\frac{\sum_{i=1}^{N} (Y_i - m)^2}{N-1}}.$$

Note that *s* is the sample standard deviation (rather than the population standard deviation, σ) and s^2 is the sample variance. Population variance refers to the average of the square of the deviations of each value from the mean, while the sample variance includes an adjustment from number of measurements *N*; *m* is the sample mean (rather than the population mean, μ). Z-scores (also called standardized scores) describe the distance from the mean per standard deviation:

$$Z_i = \frac{x_{\mathrm{i}} - \overline{x}}{s}.$$

If you compare beta globin to myoglobin, you can get a score (such as 43.9 as shown in **Fig. 3.5a**) based on some scoring system. Randomly scramble the sequence of myoglobin 1000 times (maintaining the length and composition of the myoglobin), and measure the 1000 scores of beta globin to these scrambled sequences. You can obtain a mean and standard deviation of the comparison to shuffled sequences. For more information on statistical concepts, see Motulsky (1995) and Cumming *et al*. (2007).

related. We choose a significance value α, often set to 0.05, as a threshold for defining statistical significance. One approach to determining whether our score occurred by chance is to compare it to the scores of beta globin or myoglobin relative to a large number of other proteins (or DNA sequences) known to be not homologous. Another approach is to compare the query to a set of randomly generated sequences. A third approach is to randomly scramble the sequence of one of the two query proteins (e.g., myoglobin) and obtain a score relative to beta globin; by repeating this process 100 times, we can obtain the sample mean ($\overline{x}$) and sample standard deviation (*s*) of the scores for the randomly shuffled myoglobin relative to beta globin. We can express the authentic score in terms of how many standard deviations above the mean it is. A *Z* score (Box 3.10) is calculated as:

$$Z = \frac{x - \mu}{s} \tag{3.8}$$

where *x* is the score of two aligned sequences, μ is the mean score of many sequence comparisons using a scrambled sequence, and *s* is the standard deviation of those measurements obtained with random sequences. We can do the shuffle test using an algorithm such as PRSS. This calculates the score of a global pairwise alignment, and also performs comparisons of one protein to a randomized (jumbled) version of the other.

If the scores are normally distributed, then the *Z* statistic can be converted to a probability value. If *Z* = 3, then we can refer to a table in a standard statistics resource to see

PRSS, written by William Pearson, is available online at http://fasta.bioch.virginia.edu/fasta_www2/fasta_www.cgi?rm=shuffle (WebLink 3.11). For an example of PRSS output for a comparison of human beta globin and myoglobin, see Web Document 3.8 at http://www.bioinfbook.org/chapter3.

that 99.73% of the population (i.e., of the scores) are within three standard deviations of the mean, and the fraction of scores that are greater than three standard deviations beyond the mean is only 0.13%. We can expect to see this particular score by chance about 1 time in 750 (i.e., 0.13% of the time). The problem in adopting this approach is that if the distribution of scores deviates from a Gaussian distribution the estimated significance level will be wrong. For global (but not local) pairwise alignments, the distribution is generally not Gaussian; there is therefore not a strong statistical basis for assigning significance values to pairwise alignments. What can we conclude from a *Z* score? If 100 alignments of shuffled proteins all have a score less than the authentic score of two aligned proteins, this indicates that the probability (*p*) that this occurred by chance is less than 0.01. (We can therefore reject the null hypothesis that the two protein sequences are not significantly related.) However, because of the concerns about the applicability of the *Z* score to sequence scores, conclusions about statistical significance should be made with caution.

For local pairwise alignments, the best approach to defining statistical significance is to estimate an expect value (*E* value) which is closely related to a probability value (*p* value). In contrast to the situation with global alignment, for local alignment there is a thorough understanding of the distribution of scores. An *E* value describes the number of matches having a particular score (or better) that are expected to occur by chance. For example, if a pairwise alignment of a beta globin and a myoglobin has some score with an associated *E* value of 10^{-3}, that particular score (or better) can be expected one time in one thousand by chance. This is the approach taken by the BLAST family of programs; we discuss *E* values in detail in Chapter 4.

Another consideration involves the problem of multiple comparisons. If we compare a query such as beta globin to one million proteins in a database, we have a million opportunities to find a high-scoring match between the query and some database entry. In such cases it is appropriate to adjust the significance level α, that is, the probability at which the null hypothesis is rejected, to a more stringent level. One approach, called a Bonferroni correction, is to divide α (nominally $p < 0.05$) by the number of trials (10^6) to set a new threshold for defining statistical significance of level of $0.05/10^6$, or 5×10^{-8}. The equivalent of a Bonferroni correction is applied to the probability value calculation of BLAST statistics (see Chapter 4), and we also encounter multiple comparison corrections in microarray data analysis (see Chapter 11).

Statistical Significance of Local Alignments

Most database search programs such as BLAST (Chapter 4) depend on local alignments. Additionally, many pairwise alignment programs compare two sequences using local alignment.

Percent Identity and Relative Entropy

One approach to deciding whether two sequences are significantly related from an evolutionary point of view is to consider their percent identity. It is very useful to consider the percent identity that two proteins share in order to obtain a sense of their degree of relatedness. As an example, a global pairwise alignment of odorant-binding protein from rat and cow reveals only 30% identity, although both are functionally able to bind odorants with similar affinities (Pevsner *et al*., 1985). The rat protein shares just 26% identity to its closest human ortholog. From a statistical perspective the inspection of percent identities has limited usefulness in the "twilight zone;" it does not provide a rigorous set of rules for inferring homology and it is associated with false positive or false negative results. A high degree of identity over a short region might sometimes not be evolutionarily significant, and conversely a low percent identity could reflect homology. Percent amino acid identity alone is not sufficient to demonstrate (or rule out) homology.

The accession numbers of rat and bovine odorant-binding proteins are NP_620258.1 and P07435.2; the human protein closest to rat has accession EAW50553.1. The alignments of these proteins are shown in Web Document 3.9 at 🌐 http://www.bioinfbook.org/chapter3.

Still, it may be useful to consider percent identity. Some researchers have suggested that if two proteins share 25% or more amino acid identity over a span of 150 or more amino acids, they are probably significantly related (Brenner *et al*., 1998). If we consider an alignment of just 70 amino acids, it is popular to consider the two sequences "significantly related" if they share 25% amino acid identity. However, Brenner *et al*. (1998) have shown that this may be erroneous, partly because the enormous size of today's molecular sequence databases increases the likelihood that such alignments occur by chance. For an alignment of 70 amino acid residues, 40% amino acid identity is a reasonable threshold to estimate that two proteins are homologous (Brenner *et al*., 1998). If two proteins share

BOX 3.11 RELATIVE ENTROPY

Altschul (1991) estimated that about 30 bits of information are required to distinguish an authentic alignment from a chance alignment of two proteins of average size (given that one protein is used against a database of a particular size). For each substitution matrix with its unique target frequencies q_{ij} and background distributions $p_i\, p_j$, it is possible to derive the relative entropy H as follows (Altschul, 1991):

$$H = \sum_{i,j} q_{i,j} s_{i,j} = \sum_{i,j} q_{i,j} \log_2 \frac{q_{ij}}{p_i\, p_j}$$

where H corresponds to the information content of the target and background distributions associated with a particular scoring matrix (units nats). As shown in **Figure 3.27**, for higher H values it is easier to distinguish the target from background frequencies. This analysis is consistent with the analysis of the diagonals for the PAM1 and PAM250 mutation probability matrices (**Figs 3.9** and **3.13**) in which there is far less signal apparent in the PAM250 matrix.

about 20–25% identity over a reasonably long stretch (e.g., 70–100 amino acid residues), they are in the "twilight zone" (**Fig. 3.19**) and it is more difficult to be sure. Two proteins that are completely unrelated often share about 10–20% identity when aligned. This is especially true because the insertion of gaps can greatly improve the alignment of any two sequences.

Altschul (1991) evaluated alignment scores from an information theory perspective. Target frequencies vary as a function of evolutionary distance. Recall that an alignment of alanine with threonine is assigned a different score in a PAM10 matrix (–3; see **Fig. 3.15**) than in a PAM250 matrix (+1; see **Fig. 3.14**). The relative entropy (H) of the target and background distributions measures the information that is available per aligned amino acid position that, on average, distinguishes a true alignment from a chance alignment (Box 3.11). For a PAM10 matrix, the value of H is 3.43 bits. Assuming that 30 bits of information are sufficient to distinguish a true rather than a chance alignment in a database search, an alignment of at least 9 residues is needed using a PAM10 matrix (**Fig. 3.27**).

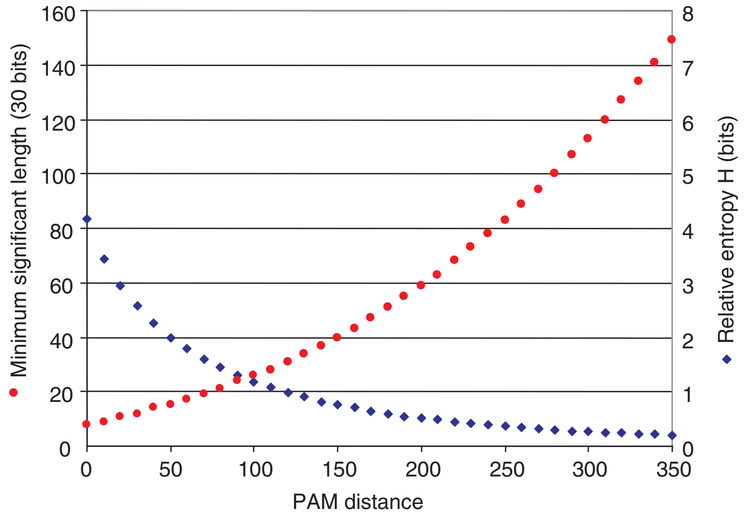


FIGURE 3.27 Relative entropy (H) as a function of PAM distance. For PAM matrices with low value (e.g., PAM10), the relative entropy in bits is high and the minimum length required to detect a significantly aligned pair of sequences is short (e.g., about 10 amino acids). Using a PAM10 matrix, two very closely related proteins can therefore be detected as homologous even if only a relatively short region of amino acid residues is compared. For PAM250 and other PAM matrices with high values, the relative entropy (or information content in the sequence) is low, and it is necessary to have a longer region of amino acids (e.g., 80 residues) aligned in order to detect significant relationships between two proteins. Adapted from Altschul (1991).

TABLE 3.4 Global pairwise alignment algorithms.

Program	Site	URL
BLAST	NCBI	🌐 http://www.ncbi.nlm.nih.gov/BLAST/
Needle EMBOSS package (global pairwise alignment)	EBI	🌐 http://www.ebi.ac.uk/Tools/emboss/
Water EMBOSS package (local pairwise alignment)	EBI	🌐 http://www.ebi.ac.uk/emboss/align/
Pairwise	Two Sequence Alignment Tool (global and local options)	🌐 http://informagen.com/Applets/Pairwise/
Stretcher	Institut Pasteur; global alignment	🌐 http://bioweb2.pasteur.fr/docs/EMBOSS/ stretcher.html

For a PAM250 matrix however, the relative entropy is 0.36 and an alignment of at least 83 residues are needed to distinguish authentic alignments.

We see in Chapter 5 that scoring matrices ("profiles") can be customized to a sequence alignment, greatly increasing the sensitivity of a search. We also see in Chapters 5 and 6 that multiple sequence alignments can offer far greater sensitivity than pairwise sequence alignment.

PERSPECTIVE

The pairwise alignment of DNA or protein sequences is one of the most fundamental operations of bioinformatics. Pairwise alignment allows the relationship between any two sequences to be determined, and the degree of relatedness that is observed helps in the forming of a hypothesis about whether they are homologous (descended from a common evolutionary ancestor). Almost all of the topics in the rest of this book are heavily dependent upon sequence alignment. In Chapter 4, we introduce the searching of large DNA and/or protein databases with a query sequence. Database searching typically involves an extremely large series of local pairwise alignments, with results returned as a rank order beginning with most related sequences.

TABLE 3.5 Local pairwise alignment algorithms.

Resource	Description	URL
BLAST	At NCBI	http://www.ncbi.nlm.nih.gov/BLAST/
est2genome	EMBOSS program from the Institut Pasteur; aligns expressed sequence tags to genomic DNA	http://bioweb.pasteur.fr/docs/EMBOSS/est2genome.html
LALIGN	Finds multiple matching subsegments in two sequences	http://www.ch.embnet.org/software/LALIGN_form.html
Pairwise	Two sequence alignment tool (global and local options)	http://informagen.com/Applets/Pairwise/
PRSS	From the University of Virginia (Bill Pearson)	http://fasta.bioch.virginia.edu/fasta_www2/fasta_www.cgi?rm=shuffle
SIM	Alignment tool for protein sequences from ExPASy	http://web.expasy.org/sim/
SSEARCH	At the Protein Information Resource	http://pir.georgetown.edu/pirwww/search/pairwise.shtml

Substituent residue
(Percentage of total residue sites at which the substituent occurs)

Sequence (original amino acid)

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	■			28			31	33								31				
R		■			🟥			🟥	50			58	🟥	🟥	🟥	25		🟥	🟥	
N	33		■	47	🟥				33	🟥		33	🟥	🟥		33	33	🟥		
D	44		22	■	🟥		47	34	22			28	🟥	🟥		25		🟥	🟥	
C	(66)	🟥	🟥	🟥	■	🟥	🟥	🟥		🟥			🟥		🟥			🟥	🟥	
Q				56	🟥	■	30	🟥	40	🟥		70	🟥	🟥				🟥	🟥	
E	50			44	🟥		■	38				41	🟥	🟥	24				🟥	
G	51	🟥		33	🟥	🟥	30	■				27	🟥	🟥		36				
H				26					■		26	30				22	22	🟥		
I	39		🟥		🟥	🟥				■	58			🟥				🟥	🟥	46
L	21									23	■	23		28	🟥					30
K	23	21		28			31	23			21	■				21				
M	22	🟥	🟥	🟥	🟥	🟥	🟥	🟥		22	89		■	22	🟥	🟥				45
F		🟥	🟥	🟥		🟥	🟥	🟥	22	🟥	61			■	🟥					
P	50	🟥		43	🟥		57	43			🟥	21	🟥	🟥	■			🟥	🟥	🟥
S	49			24			24	36				24	🟥			■	40	🟥	🟥	
T	32						28	24				24				52	■	🟥	🟥	
W	(40)	🟥	🟥	🟥	🟥	🟥			🟥	🟥	(40)			(60)	🟥	🟥	🟥	■		🟥
Y		🟥		🟥	🟥	🟥	🟥		(33)	🟥				(50)	🟥	🟥	🟥		■	
V	36									21	43	21			🟥			🟥		■

FIGURE 3.28 Substitution frequencies of globins (adapted from Zuckerkandl and Pauling, 1965, p. 118). Amino acids are presented alphabetically according to the three-letter abbreviations. The rows correspond to an original amino acid in an alignment of several dozen hemoglobin and myoglobin protein sequences from human, other primates, horse, cattle, pig, lamprey, and carp. Numbers represent the percentages of residue sites at which a given substitution occurs. For example, a glycine substitution was observed to occur in 33% of all the alanine sites. Substitutions that were never observed to occur are indicated by squares colored red. Rarely occurring substitutions (percentages <20%) are indicated by empty white squares (numerical values are not given). "Very conservative" substitutions (percentages ≥40%) are in boxes shaded gray. For example, in 89% of the sites containing a methionine, leucine was also observed to be present. Identities are indicated by black solid squares. Values in parentheses indicate a very small available sample size, suggesting that conclusions about those data should be made cautiously.

Source: Zuckerkandl and Pauling (1965).

The algorithms used to perform pairwise alignment were developed in the 1970s, beginning with the global alignment procedure of Needleman and Wunsch (1970). Dayhoff (1978) introduced PAM scoring matrices that permit the comparison and evaluation of distantly related molecular sequences. Scoring matrices are an integral part of all pairwise (or multiple) sequence alignments, and the choice of a scoring matrix can strongly influence the outcome of a comparison. By the 1980s, local alignment algorithms were introduced (see the work of Sellers, 1974; Smith and Waterman, 1981; Smith *et al*., 1981). Practically, pairwise alignment is performed today with a limited group of software packages, most of which are freely available.

The sensitivity and specificity of the available pairwise sequence alignment algorithms continue to be assessed. Recent areas in which pairwise alignment has been further developed include methods of masking low-complexity sequences (to be discussed in Chapter 4) and theoretical models for penalizing gaps in alignments.

PITFALLS

The optional parameters that accompany a pairwise alignment algorithm can greatly influence the results. A comparison of the homologs human RBP4 and bovine β-lactoglobulin using BLAST 2 Sequences results in no match detected if the default parameters are used.

Any two sequences can be aligned, even if they are unrelated. In some cases, two proteins that share even greater than 30% amino acid identity over a stretch of 100 amino acids are not homologous (evolutionarily related). It is always important to assess the biological significance of a sequence alignment. This may involve searching for evidence for a common cellular function, a common overall structure or, if possible, a similar three-dimensional structure.

Consider two aligned proteins, each of length 100 amino acids. When they share 50% amino acid identity, then on average 80 changes have occurred. I have found that this concept confuses many students. The explanation is that the observed number of changes (50 differences per 100 aligned residues) does not reflect the multiple substitutions that have occurred. For example, the proteins might be a mouse and human globin. About 90 million years ago a species of furry little creatures separated into two groups, eventually leading to speciation and the emergence of primate and rodent lineages. At one position the protein might have had an alanine in the common ancestor that mutated to a threonine and then to an asparagine in the rodent lineage. Two changes occurred at that particular position over a period of millions of years, although we observe only one. We further explore this concept in Chapter 7 on phylogeny and evolution.

When two proteins share 20% amino acid identity (and are in the "twilight zone") they have 80 observed differences. However, Dayhoff (1978) estimated that 250 changes (on average) had occurred. The PAM250 matrix was therefore considered useful for detecting distantly related proteins.

ADVICE FOR STUDENTS

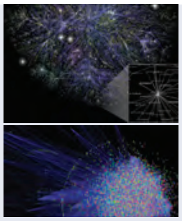
Begin using the BLAST website at NCBI to compare two sequences. Choose two closely related proteins and two that are very distantly related; what are the effects of changing scoring matrices or other parameters? For each topic we discussed, try to gain practical experience. For example, select members of a protein family that are locally aligned because they share a region of homology, and perform global alignment as well. Can you change the local alignment search parameters to include larger or smaller aligned regions? Also try different alignment tools, from various websites to R or Python. In Chapter 4 we introduce BLAST+ for performing any BLAST search on the command line, and you can also use BLAST+ for pairwise alignment.

Joshua Lederberg helped Zuckerkandl and Pauling (1965) make the matrix of **Figure 3.28.** They used an IBM 7090 computer, one of the first commercial computers based on transistor technology. The computer cost about US$3 million. Its memory consisted of 32,768 binary words or about 131,000 bytes. To read about Lederberg's Nobel Prize from 1958, see 🌐 http://nobelprize.org/nobel_prizes/medicine/laureates/1958/ (WebLink 3.12).

WEB RESOURCES

Pairwise sequence alignment can be performed using software packages that implement global or local alignment algorithms. In all cases, two protein or two nucleic acid sequences are directly compared.

Many websites offer web-based pairwise local alignment algorithms based upon global alignment (**Table 3.4**) or local alignment (**Table 3.5**). These sites include EBI and NCBI, the Baylor College of Medicine (BCM) launcher, the SIM program at ExPASy, and SSEARCH at the Protein Information Resource (PIR) at Georgetown University. Computer lab problem (3.4) introduces pairwise alignment in R.



Discussion Questions

[3-1] If you want to compare any two proteins, is there any one "correct" scoring matrix to choose? Is there any way to know which scoring matrix is best to try?

[3-2] Many protein (or DNA) sequences have separate domains. (We discuss domains in Chapter 12.) Consider a protein that has one domain that evolves rapidly and a second domain that evolves slowly. In performing a pairwise alignment with another protein (or DNA) sequence, would you use two separate alignments with scoring matrices such as PAM40 and PAM250 or would you select one "intermediate" matrix? Why?

[3-3] Years before Margaret Dayhoff and colleagues published a protein atlas with scoring matrices, Emile Zuckerkandl and Linus Pauling (1965) produced a scoring matrix for several dozen available globin sequences (**Fig. 3.28**). The rows (*y* axis) of this figure show the original globin amino acid, and the columns show substitutions that were observed to occur. Numerical values are entered in cells for which the substitutions occur in at least 20% of the sites. Note that, for cells shaded red, these amino acid substitutions were never observed; for cells shaded gray the amino acid substitutions were defined as very conservative. How do the data in this matrix compare to those described by Dayhoff and colleagues? Which substitutions occur most rarely, and which most frequently? How would you go about filling in this table today?

[3-4] The first five computer lab problems (below) guide you to perform pairwise alignment using five different methods. If you want to align two protein or DNA sequences, how can you decide which tool(s) are most appropriate? In other words, what are some of the strengths and limitations of these various methods?

[3-5] The PAM1 matrix (**Fig. 3.9**) is nonreciprocal: the probability of changing an amino acid such as alanine to arginine is not equal to the probability of changing an arginine to an alanine. Why? Log-odds matrices such as PAM10 (**Figure 3.15**) are reciprocal.

PROBLEMS/COMPUTER LAB

For problems (3.1)–(3.3) and (3.5) we perform pairwise alignments of globins using complementary approaches.

[3-1] Obtain the human HBA and HBB protein sequences. Perform pairwise alignment at the NCBI BLAST website. Then use a comparison tool from the EBI website. Vary the scoring matrix (e.g., try different PAM and BLOSUM matrices) and record the effects on the score, the number of gaps, the percent identity, and the length of the aligned region. For the NCBI BLASTP program note that the output of a pairwise alignment includes a dot matrix view.

[3-2] Perform pairwise alignment at the UCSC website. (1) Go to ⊕ http://genome.ucsc.edu (WebLink 3.13). follow the link to the genome browser, select the human genome hg19 build, and enter a query of hbb. This should direct you to chr11:5,246,696–5,248,301 (a region of 1606 base pairs encompassing the beta globin gene, *HBB*. (2) Click the box to set the view to default tracks. (3) Under "Comparative Genomics" select Placental Chain/Net and set the display to full. By clicking the Placental Chain/Net header you can view a series of options. Set Chains to full view and Nets to full view. Set the species to horse (deselect other species). Click submit. (4) The display now shows human/horse chained alignments and and alignment nets.

[3-3] Perform pairwise alignment using EMBOSS tools via Galaxy and UCSC. In this exercise we perform global alignment with the EMBOSS package needle and local alignment with the EMBOSS package water. Both of these are available at the Galaxy public web server (along with over 100 other EMBOSS tools). Box 3.9 introduces EMBOSS and explains how to import beta globin (HBB) and alpha globin (HBA2) proteins from the UCSC Table Browser using Galaxy, and to then align them. This history is saved at ⊕ https://main.g2.bx.psu.edu/u/pevsner/h/pairwise-alignment-via-ucsc-and-emboss (WebLink 3.14). Note that Galaxy is a web-based platform for using hundreds of bioinformatics tools, including next-generation sequence data analysis software. To use it visit ⊕ http://usegalaxy.org then go to the public server. Be sure to create a username and log in. This will allow you to continue your work over time and at different work stations.

[3-4] View scoring matrices and perform pairwise alignment using R. In this exercise we begin by installing the `Biostrings` package. Instructions for installing R and RStudio are given in Chapter 2.

```
> getwd() # Get (show) the working directory
# Use setwd() to change it to any location
> source("http://bioconductor.org/biocLite.R")
> biocLite("Biostrings")
> library(Biostrings)
# Install the Biostrings library
> data(BLOSUM50)
# load the data for the BLOSUM50 matrix
> BLOSUM50[1:4,1:4]
# view the first four rows and
# columns of this matrix
> nw <- pairwiseAlignment(AAString("PAWHEAE"),
AAString("HEAGAWGHEE"), substitutionMatrix =
BLOSUM50, gapOpening = 0, gapExtension = -8)
# create object
# nw aligning two amino acid strings with the
# specified matrix and gap penalties
> nw # view the result.
# Try repeating this alignment with
# different gap penalties and scoring matrices.
# Biostrings includes 10 matrices (PAM30 PAM40,
# PAM70, PAM120, PAM250, BLOSUM45, BLOSUM50,
# BLOSUM62, BLOSUM80, and BLOSUM100).
> compareStrings(nwdemo) # view the alignment
```

[3-5] Perform pairwise alignment using Python, a freely available programming language. When implemented with Biopython it offers a broad range of computational tools (Cock *et al*., 2009). You will need to install three programs: (1) Python; (2) Numpy (a package for scientific computing with Python); and (3) Biopython (this provides particular bioinformatics applications within the Python framework). The downloads can be obtained from ⊕ http://www.python.org (WebLink 3.15), ⊕ http://www.numpy.org/ (WebLink 3.16), and ⊕ http://biopython.org (WebLink 3.17).If you are working on a PC launch a user-friendly interface called IDLE (Python's Integrated DeveLopment Environment). From a Mac, open a terminal window and type python to see the command prompt (>>>).n For information on installing Biopython, and for a "cookbook" with many basic bioinformatics applications including pairwise alignment, visit ⊕ http://biopython.org/DIST/docs/tutorial/Tutorial.html (WebLink 3.18). Try the following commands; my comments follow a hash (#) and are in green text.

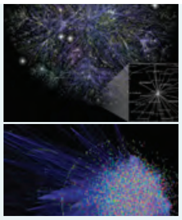
```
$ python # launch python from a terminal
Python 2.7.5 (default, Mar  9 2014, 22:15:05)
Type "copyright", "credits" or "license()" for
more information.
>>> from Bio import pairwise2
>>> from Bio.SubsMat import MatrixInfo as
matlist
>>> matrix = matlist.blosum62
# specify the scoring matrix
>>> help(matlist)
```

```
# This shows a list of available matrices
>>> gap_open = -10 # set the affine gap penal-
ties
>>> gap_extend = -1
>>> hbb = "VTALWGKVNVDEVGGEALGRLL"
# This is part of beta globin from Fig. 3.5b
>>> mb = "VLNVWGKVEADIPGHGQEVLIRLF"
# This is part of myoglobin from Fig. 3.5b
>>> alns = pairwise2.align.globalds(hbb, mb, ma-
trix, gap_open, gap_extend)
>>> top_aln = alns[0]
>>> aln_hbb, aln_mb, score, begin, end = top_aln
>>> print aln_hbb+'\n'+aln_mb
# the '\n' command inserts a line break
VTALWGKVNVDEVGG--EALGRLL
VLNVWGKVEADIPGHGQEVLIRLF
```

We have used the pairwise2 module from Python. It is capable of both global and local pairwise alignments. Compare the result to **Fig. 3.5b** and note that the gap placement differs. Try raising the gap extend penalty from –0.5 to –2. What happens to the alignment? Documentation is available for the pairwise2 Python module (http://biopython.org/DIST/docs/api/Bio.pairwise2-module.html).

[3-6] Using the amino acid explorer tool from NCBI. (1) Visit ⊕ http://www.ncbi.nlm.nih.gov/Class/Structure/aa/aa_explorer.cgi (WebLink 3.19). (2) Select the Biochemical Properties table. Which amino acid is most abundant? (Is it leucine, at 9.94%?). Use this table to test yourself and make sure you know the one- and three-letter abbreviations for all 20 amino acids, as well as their structures. (3) Is tyrosine a hydrophobic amino acid? To decide, use the Common Substitutions table. Explore valine (a hydrophobic residue), sort the results by hydrophobicity, and see where tyrosine is located. You can also explore the Structure and Chemistry table.

[3-7] Many tools are available to manipulate sequences. Visit the Sequence Manipulation Suite (⊕ http://www.bioinformatics.org/sms2/index.html) (Weblink 3.20) to access a large number of tools. (Compare its tools to those in EMBOSS) What is the reverse complement of the sequence GGAATTCC?



Self-Test Quiz

[3-1] Match the following amino acids with their single-letter codes:

Asparagine	Q
Glutamine	W
Tryptophan	Y
Tyrosine	N
Phenylalanine	F

[3-2] Orthologs are defined as:

(a) Homologous sequences in different species that share an ancestral gene.

(b) Homologous sequences that share little amino acid identity but share great structural similarity.

(c) Homologous sequences in the same species that arose through gene duplication.

(d) Homologous sequences in the same species which have similar and often redundant functions.

[3-3] Which of the following amino acids is least mutable according to the PAM scoring matrix?

(a) alanine;

(b) glutamine;

(c) methionine; or

(d) cysteine.

[3-4] The PAM250 matrix is defined as having an evolutionary divergence in which what percentage of amino acids between two homologous sequences have changed over time?

(a) 1%;

(b) 20%;

(c) 80%; or

(d) 250%.

[3-5] Which of the following sentences best describes the difference between a global alignment and a local alignment between two sequences?

(a) Global alignment is usually used for DNA sequences, while local alignment is usually used for protein sequences.

(b) Global alignment has gaps, while local alignment does not have gaps.

(c) Global alignment finds the global maximum, while local alignment finds the local maximum.

(d) Global alignment aligns the whole sequence, while local alignment finds the best subsequence that aligns.

[3-6] You have two distantly related proteins. Which BLOSUM or PAM matrix is best suited to compare them?

(a) BLOSUM45 or PAM250;

(b) BLOSUM45 or PAM1;

(c) BLOSUM80 or PAM250; or

(d) BLOSUM80 or PAM1.

[3-7] How does the BLOSUM scoring matrix differ most notably from the PAM scoring matrix?

(a) It is best used for aligning very closely related proteins.

(b) It is based on global multiple alignments from closely related proteins.

(c) It is based on local multiple alignments from distantly related proteins.

(d) It combines local and global alignment information.

[3-8] True or false: Two proteins that share 30% amino acid identity are 30% homologous.

[3-9] A global alignment algorithm (such as the Needleman–Wunsch algorithm) is guaranteed to find an optimal alignment. Such an algorithm:

(a) Puts the two proteins being compared into a matrix and finds the optimal score by exhaustively searching every possible combination of alignments.

(b) Puts the two proteins being compared into a matrix and finds the optimal score by iterative recursions.

(c) Puts the two proteins being compared into a matrix and finds the optimal alignment by finding optimal subpaths that define the best alignment(s).

(d) Can be used for proteins but not for DNA sequences.

[3-10] In a database search or in a pairwise alignment, sensitivity is defined as:

(a) The ability of a search algorithm to find true positives (i.e., homologous sequences) and to avoid false positives (i.e., unrelated sequences having high similarity scores).

(b) The ability of a search algorithm to find true positives (i.e., homologous sequences) and to avoid false positives (i.e., homologous sequences that are not reported).

(c) The ability of a search algorithm to find true positives (i.e., homologous sequences) and to avoid false negatives (i.e., unrelated sequences having high similarity scores).

(d) The ability of a search algorithm to find true positives (i.e., homologous sequences) and to avoid false negatives (i.e., homologous sequences that are not reported).

SUGGESTED READING

We introduced this chapter with the concept of homology, an often misused term. A one-page article by Reeck *et al*. (1987) provides authoritative, standard definitions of the terms homology and similarity. Other discussions of homology in relation to phylogeny are provided by Tautz (1998) and Pearson (2013). The William Pearson article provides an excellent introduction to sequence alignment (including *E* values, which we describe in Chapter 4). His earlier article (Pearson, 1996) provides descriptions of the statistics of similarity scores, sensitivity and selectivity, and search programs such as Smith–Waterman and FASTA.

For studies of pairwise sequence alignment algorithms, an important historical starting point is the 1978 book by Margaret O. Dayhoff and colleagues (Dayhoff, 1978). Most of this book consists of an atlas of protein sequences with accompanying phylogenetic reconstructions. Chapter 22 of the *Atlas of Protein Sequence and Structure* introduces the concept of accepted point mutations, while chapter 23 describes various PAM matrices. Russell F. Doolittle (1981) also wrote a clear, thoughtful overview of sequence alignment. By the early 1990s, when far more protein sequence data were available, Steven and Jorja Henikoff (1992) described the BLOSUM matrices. This article provides an excellent technical introduction to the use of scoring matrices, usefully contrasting the performance of PAM and BLOSUM matrices. Later (in Chapters 4 and 5) we will use these matrices extensively in database searching.

The algorithms originally describing global alignment are presented technically by Needleman and Wunsch (1970) and later local alignment algorithms were introduced by Smith and Waterman (1981) and Smith *et al*. (1981). The problem of both sensitivity (the ability to identify distantly related sequences) and selectivity (the avoidance of unrelated sequences) of pairwise alignments was addressed by Pearson and Lipman in a 1988 paper introducing the FASTA program.

Marco Pagni and C. Victor Jongeneel (2001) of the Swiss Institute of Bioinformatics provide an excellent overview of sequence-scoring statistics. This includes a discussion of BLAST scoring statistics that is relevant to Chapters 4 and 5.

Finally, Steven Brenner, Cyrus Chothia, and Tim Hubbard (1998) have compared several pairwise sequence methods. This article is highly recommended as a way to learn how different algorithms can be assessed (we will see similar approaches for multiple sequence alignment in Chapter 6, for example). Reading this paper can help to show why statistical scores are more effective than other search parameters such as raw scores or percent identity in interpreting pairwise alignment results. For a more recent overview of sequence alignment, see Stormo (2009).

REFERENCES

Altschul, S.F. 1991. Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology* **219**(3), 555–565. PMID: 2051488.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., Lipman, D. J. 1990. Basic local alignment search tool. *Journal of Molecular Biology* **215**, 403–410.

Altschul, S.F., Wootton, J.C., Gertz, E.M. *et al*. 2005. Protein database searches using compositionally adjusted substitution matrices. *FEBS Journal* **272**(20), 5101–5109. PMID: 16218944.

Anfinsen, C. 1959. *The Molecular Basis of Evolution*. John Wiley & Sons, Inc., New York.

Brenner, S. E., Chothia, C., Hubbard, T. J. 1998. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of National Academy of Sciences, USA* **95**, 6073–6078.

Chothia, C., Lesk, A. M. 1986. The relation between the divergence of sequence and structure in proteins. *EMBO Journal* **5**, 823–826.

Cock, P.J., Antao, T., Chang, J.T. *et al*. 2009. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**(11), 1422–1423. PMID: 19304878.

Cumming, G., Fidler, F., Vaux, D.L. 2007. Error bars in experimental biology. *Journal of Cell Biology* **177**, 7–11.

Darwin, C. 1872. *The Origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life*. John Murray, London.

Dayhoff, M.O. (ed.) 1966. *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, Silver Spring, MD.

Dayhoff, M. O. (ed.) 1978. *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, Silver Spring, MD.

Dayhoff, M.O., Hunt, L.T., McLaughlin, P.J., Jones, D.D. 1972. Gene duplications in evolution: the globins. In: *Atlas of Protein Sequence and Structure*, volume **5** (ed. Dayhoff, M.O.). National Biomedical Research Foundation, Washington, DC.

Doolittle, R. F. 1981. Similar amino acid sequences: Chance or common ancestry? *Science* **214**, 149–159.

Doolittle, R. F. 1987. *OF URFS AND ORFS: A Primer on How to Analyze Derived Amino Acid Sequences*. University of Science Books, Mill Valley, CA.

Durbin, R., Eddy, S., Krogh, A., Mitchison, G. 2000. *Biological Sequence Analysis*. Cambridge University Press, Cambridge.

Ewins, W.J., Grant, G.R. 2001. *Statistical Methods in Bioinformatics: An Introduction*. Springer-Verlag, New York.

Fitch, W.M. 1970. Distinguishing homologous from analogous proteins. *Systematic Zoology* **19**(2), 99–113. PMID: 5449325.

Gonnet, G. H., Cohen, M. A., Benner, S. A. 1992. Exhaustive matching of the entire protein sequence database. *Science* **256**, 1443–1445.

Gotoh, O. 1982. An improved algorithm for matching biological sequences. *Journal of Molecular Biology* **162**, 705–708.

Hedges, S.B., Dudley, J., Kumar, S. 2006. TimeTree: a public knowledge-base of divergence times among organisms. *Bioinformatics* **22**, 2971–2972.

Henikoff, S., Henikoff, J. G. 1992. Amino acid substitution matrices from protein blocks. *Proceedings of National Academy of Sciences, USA* **89**, 10915–10919.

Henikoff, J. G., Henikoff, S. 1996. Blocks database and its applications. *Methods in Enzymology* **266**, 88–105.

Hossfeld, U., Olsson, L. 2005. The history of the homology concept and the "Phylogenetisches Symposium". *Theory in Biosciences* **124**(2), 243–253. PMID: 1704635.

International Human Genome Sequencing Consortium. 2001. Initial sequencing and analysis of the human genome. *Nature* **409**, 860–921.

Jones, D.T., Taylor, W.R., Thornton, J.M. 1992. The rapid generation of mutation data matrices from protein sequences. *Computer Applications in the Biosciences* **8**, 275–282.

Junier, T., Pagni, M. 2000. Dotlet: diagonal plots in a web browser. *Bioinformatics* **16**(2), 178–179. PMID: 10842741.

Karlin, S., Altschul, S.F. 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences, USA* **87**, 2264–2268.

Lieb, B., Dimitrova, K., Kang, H.S. *et al*. 2006. Red blood with blue-blood ancestry: intriguing structure of a snail hemoglobin. *Proceedings of the National Academy of Sciences, USA* **103**(32), 12011–12016. PMID: 16877545.

Motulsky, H. 1995. *Intuitive Biostatistics*. Oxford University Press, New York.

Myers, E. W., Miller, W. 1988. Optimal alignments in linear space. *Computer Applications in the Biosciences* **4**, 11–17.

Needleman, S. B., Wunsch, C. D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* **48**, 443–453.

Owen, R. 1843. *Lectures on the Comparative Anatomy and Physiology of the Invertebrate Animals, Delivered at the Royal College of Surgeons in 1843*. Longman Brown Green and Longmans, London.

Pagni, M., Jongeneel, C. V. 2001. Making sense of score statistics for sequence alignments. *Briefings in Bioinformatics* **2**, 51–67.

Pearson, W. R. 1996. Effective protein sequence comparison. *Methods in Enzymology* **266**, 227–258.

Pearson, W.R. 2013. An introduction to sequence similarity ("homology") searching. *Current Protocols in Bioinformatics* **Chapter** 3, Unit 3.1. PMID:23749753.

Pearson, W. R., Lipman, D. J. 1988. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences, USA* **85**, 2444–2448.

Pearson, W. R., Wood, T. C. 2001. Statistical significance in biological sequence comparison. In *Handbook of Statistical Genetics* (eds D. J.Balding, M.Bishop, C.Cannings). Wiley, London, pp. 39–65.

Pevsner, J., Trifiletti, R.R., Strittmatter, S.M., Snyder, S.H. 1985. Isolation and characterization of an olfactory receptor protein for odorant pyrazines. *Proceedings of the National Academy of Sciences, USA* **82**, 3050–3054.

Reeck, G. R., de Haën, C., Teller, D.C. *et al.* 1987. "Homology" in proteins and nucleic acids: A terminology muddle and a way out of it. *Cell* **50**, 667.

Rice, P., Longden, I., Bleasby, A. 2000. EMBOSS: The European molecular biology open software suite. *Trends in Genetics* **16**, 276–277.

Sedgewick, R. 1988. *Algorithms*. Addison-Wesley Longman, Reading, MA.

Schopf, J.W. 2002. When did life begin? In: *Life's Origin: The Beginnings of Biological Evolution* (ed. Schopf, J.W.). University of California Press, Berkeley.

Sellers, P. H. 1974. On the theory and computation of evolutionary distances. *SIAM Journal of Applied Mathematics* **26**, 787–793.

Smith, T. F., Waterman, M. S. 1981. Identification of common molecular subsequences. *Journal of Molecular Biology* **147**, 195–197.

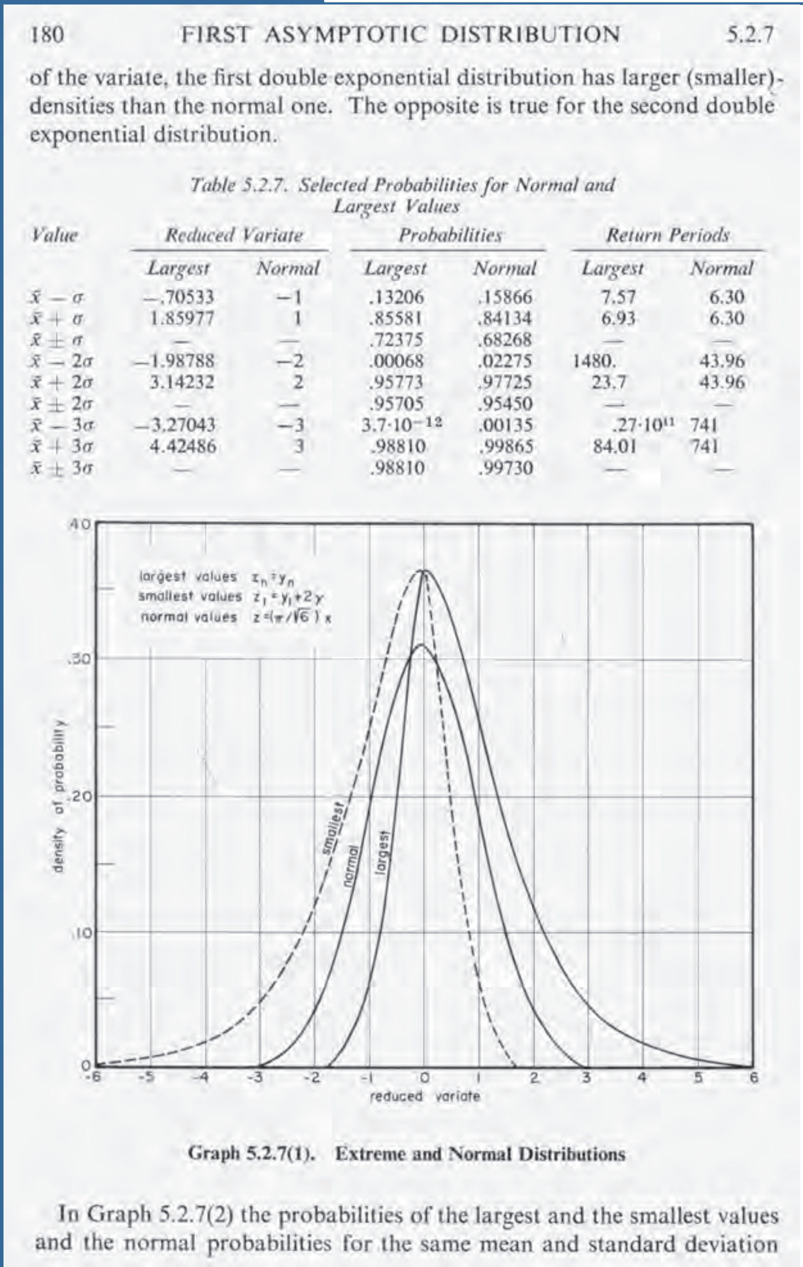
Smith, T. F., Waterman, M. S., Fitch, W. M. 1981. Comparative biosequence metrics. *Journal of Molecular Evolution* **18**, 38–46.

Stormo, G.D. 2009. An introduction to sequence similarity ("homology") searching. *Current Protocols in Bioinformatics* **Chapter** 3, Unit 3.1 3.1.1-7. PMID: 19728288.

Tatusova, T. A., Madden, T. L. 1999. BLAST 2 Sequences, a new tool for comparing protein and nucleotide sequences. *FEMS Microbiology Letters* **174**, 247–250.

Tautz, D. 1998. Evolutionary biology. Debatable homologies. *Nature* **395**, 17, 19.

Zuckerkandl, E., Pauling, L. 1965. Evolutionary divergence and convergence in proteins. In: *Evolving Genes and Proteins* (eds Bryson, V., Vogel, H.J.). Academic Press, New York, pp. 97–166.



180 FIRST ASYMPTOTIC DISTRIBUTION 5.2.7

of the variate, the first double exponential distribution has larger (smaller)-densities than the normal one. The opposite is true for the second double exponential distribution.

Table 5.2.7. Selected Probabilities for Normal and Largest Values

Value	*Reduced Variate*		*Probabilities*		*Return Periods*	
	Largest	*Normal*	*Largest*	*Normal*	*Largest*	*Normal*
$\bar{x} - \sigma$	−.70533	−1	.13206	.15866	7.57	6.30
$\bar{x} + \sigma$	1.85977	1	.85581	.84134	6.93	6.30
$\bar{x} \pm \sigma$	—	—	.72375	.68268	—	—
$\bar{x} - 2\sigma$	−1.98788	−2	.00068	.02275	1480.	43.96
$\bar{x} + 2\sigma$	3.14232	2	.95773	.97725	23.7	43.96
$\bar{x} \pm 2\sigma$	—	—	.95705	.95450	—	—
$\bar{x} - 3\sigma$	−3.27043	−3	$3.7 \cdot 10^{-12}$	.00135	$.27 \cdot 10^{11}$	741
$\bar{x} + 3\sigma$	4.42486	3	.98810	.99865	84.01	741
$\bar{x} \pm 3\sigma$	—	—	.98810	.99730	—	—

40
.30
.20
.10
0
density of probability
largest values $z_n = y_n$
smallest values $z_1 = y_1 + 2\gamma$
normal values $z = (\pi/\sqrt{6})\,x$
smallest
normal
largest
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
reduced variate

Graph 5.2.7(1). Extreme and Normal Distributions

In Graph 5.2.7(2) the probabilities of the largest and the smallest values and the normal probabilities for the same mean and standard deviation

Chapter 4 describes the principal database search tool, BLAST. While BLAST was first described by Altschul *et al*. in 1990, the statistical interpretation of the scores obtained from a BLAST search are based on mathematical models developed in the 1950s. In many instances, the distribution of values in a population assumes a normal (Gaussian) distribution, as shown in this figure (see curve labeled "normal"). However, for a wide variety of natural phenomena the distribution of extreme values is not normal. Such is the case for database searches in which you search with a protein or DNA sequence of interest (the query) against a large database, as described in this chapter. The maximum scores fit an extreme value distribution (EVD) rather than a normal distribution.

In 1958 Emil Gumbel described the statistical basis of the EVD in his book *Statistics of Extremes*. This figure (Gumbel, 1958, p. 180) shows the EVD. Note that for the curve marked "largest" the tail is skewed to the right. Also, as shown in the table, for a normal distribution values that are up to three standard deviations above the mean occupy 99.865% of the area under the curve; for the EVD, values up to three standard deviations occupy only 98.810%. In other words, the EVD is characterized by a larger area under the curve at the extreme right portion of the plot. We see how this analysis applied to BLAST search results allows you to assess whether a query sequence is significantly related to a match in the database.

Source: Gumbel (1958).