AMINO ACID SUBSTITUTION MATRICES

By STEVEN HENIKOFF AND JORJA G. HENIKOFF

Howard Hughes Medical Institute, Fred Hutchinson Cancer Research Center, Seattle, Washington 98109-1024

I. Introduction .............................................................. 73
II. Scoring Basics .......................................................... 74
III. Theory ................................................................. 77
IV. Models for Substitution Scores ................................. 78
   A. MDM (PAM) Model ............................................. 78
   B. BLOSUM Model ................................................. 80
   C. Structure-based Models ....................................... 81
   D. Other Models ................................................... 82
V. Special Substitution Scores .................................... 82
VI. Gap Scores ............................................................. 83
   A. Aligning Residues with Null Characters .................. 83
   B. Leaving Unalignable Regions Unaligned ................... 84
   C. Global Alignments ............................................. 87
VII. Evaluating Scoring Functions ............................... 87
   A. Based on Database Searching Performance .............. 87
   B. Based on Alignment Accuracy ............................... 89
   C. Based on Fold Recognition .................................. 90
VIII. Position-Specific Scores ..................................... 91
IX. Using Multiple Substitution Matrices ....................... 94
   A. Multiple Searches ............................................. 94
   B. Subdividing the Query ...................................... 94
   C. Dispensing with Gap Scores ................................. 94
X. Conclusions .......................................................... 95
References ................................................................ 96

I. INTRODUCTION

Automated sequence comparisons based on the alignment of protein sequences are among the most widely used tools in biological research. Most biologists know how to “BLAST” a sequence. They have at least an intuitive notion that this procedure involves aligning their query sequence with each of the sequences in a databank in an attempt to detect interesting sequence similarities; however, they may not be aware of how a program such as BLAST decides how good an alignment is. All alignment programs use scoring functions, usually a set of “substitution” scores for each aligned amino acid pair and “gap” scores for insertion or deletion of an amino acid in one or the other sequence. This chapter describes scoring functions and their practical use in alignment tasks. Scoring functions have diverse applications because substitution and
gap scores are needed for most protein alignment methods used in biology. These methods include database searching such as BLAST (Altschul et al., 1990) and FASTA (Pearson, 1990), pairwise and multiple alignment [e.g., Clustal (Thompson et al., 1994)], phylogenetic analysis of proteins (Hein, 1990), homology modeling to three-dimensional (3D) structures (e.g., MODELLER3 [Sanchez and Sali, 1997]), and sequence neighboring (e.g., Entrez [Benson et al., 1999]). The results of alignment-based analyses may be starting points for many laboratory procedures in molecular biology and biochemistry, and with the rapid expansion of sequence databanks, strategies based on sequence comparison have become the norm. Although numerous scoring functions have been proposed, the chapter focuses primarily on those currently used by popular alignment algorithms.

II. SCORING BASICS

The alignment of a pair of protein sequences is best visualized by a “dot plot,” where the residue positions of one sequence make up the horizontal axis and those of the other the vertical axis. In a basic dot plot, a point is marked on the grid defined by the axes when the same amino acid occurs in the corresponding sequence positions. More complex dot plots will also mark conservative replacements in some manner (Sonnhammer and Durbin, 1995) (Fig. 1). Regions of high amino acid similarity between the sequences show up on a dot plot as diagonal streaks representing aligned segment pairs. These are assigned a score, and if the score is large enough, they are called high-scoring segment pairs (HSPs). Four HSPs are indicated in Fig. 1. When two HSPs are offset but near one another on a dot plot, the eye wants to connect them across the intervening region of low similarity. Alignment programs use numerical scoring functions to decide if aligned segment pairs are significantly similar, and whether to connect them. The gold standard alignment programs use dynamic programming (Needleman and Wunsch, 1970; Smith and Waterman, 1981), which considers all the possible alignments on a dot plot and so is guaranteed to find the optimal one for a given scoring function. The total alignment score is the sum of the scores of the HSPs included in the final alignment, minus penalties charged for connecting them across regions of low similarity.

Dynamic programming algorithms are useful for finding the optimal alignment of a pair of protein sequences, but may be too slow to practically compare a query sequence with all of the sequences in a database. Approximate methods that usually find the optimal alignment are based on local alignments. Strictly local or block alignment methods align
only regions of high similarity and overlook intervening regions; this approach is used by the original FASTP (Lipman and Pearson, 1985) and BLAST (Altschul et al., 1990) algorithms. Their final alignment consists of unconnected HSPs. Other algorithms start by finding local alignments, but then connect them by inserting null characters in one or the other sequence; each run of null characters is called a gap [FASTA (Pearson, 1990), WU-BLAST (Altschul and Gish, 1996), Gapped BLAST (Altschul et al., 1997)]. These fast methods will usually obtain the alignment with the highest possible score. In doing so, they approximate the slower Smith–Waterman algorithm (Smith and Waterman, 1981), which is guaranteed to find the highest scoring alignment. Another alignment approach is global, in which the sequences are aligned along their entire lengths by inserting gaps (Needleman and Wunsch, 1970). It is usually used only for aligning pairs of sequences known to be related and not for database searching. Both local and global alignments require substitution scores to compare pairs of residues, and the gapped approaches require gap scores to compare a residue with a null character.

More generally, ungapped local alignment approaches can be thought of as using gap scores of minus infinity.

Formally, 400 (20 × 20) substitution scores are needed, one for each possible pair of the 20 amino acids, and for gapped alignments, up to
40 (20 × 2) additional gap scores are needed. This 20 × 20 array of scores is called a “substitution,” “replacement,” or “exchange” matrix. These terms imply that the scores are interpreted in evolutionary terms, and they are sometimes thought of as the propensity of one amino acid to replace another during the evolution of a sequence. Likewise, gap scores are sometimes viewed as the propensity for insertions or deletions to occur in a sequence over time. Indeed, a high-scoring alignment of two sequences is sometimes said to imply homology, but strictly speaking, it just indicates similarity within the framework of the scoring system used to compute it.

In practice, only 210 substitution scores are usually specified because the order of the amino acid pairs is ignored (lower half of Fig. 2). For gapped alignments, the same gap score is usually used for a null character aligned with any amino acid, although the first null character in a gap is often given a different score than subsequent null characters in the same gap. So for the gapped alignment of two protein sequences, most algorithms require a scoring function with 212 scores. The computation of scoring functions is generally based on a theoretical model. Some scoring systems model protein sequence evolution, others model conserved regions, and still others model 3D structural similarities.

| C | S | T | P | A | G | N | D | E | Q | R | K | M | I | L | V | F | Y | W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 2 | 2 | 0 | 1 | 1 | 0 | 1 | 1 | 3 | 4 | C |
| -3 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | S |
| C | 13 | -3 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | I |
| S | -1 | 5 | -2 | 1 | 0 | 1 | 0 | 1 | 1 | 2 | 0 | 1 | 0 | 2 | 1 | 0 | 0 | -1 | T |
| T | -1 | 2 | 5 | 3 | 0 | 1 | 2 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | -1 | A |
| F | -4 | -1 | 10 | -1 | 0 | 1 | 2 | 1 | 1 | 2 | 1 | -1 | 0 | 0 | 1 | -1 | -1 | -1 | G |
| A | -1 | 1 | 0 | -1 | 5 | -3 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | N |
| G | -3 | 0 | -2 | -2 | 0 | 8 | -3 | 1 | 1 | 2 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | D |
| N | -2 | 1 | 0 | -2 | -1 | 0 | 7 | -2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | -1 | -1 | E |
| D | -4 | 0 | -1 | -1 | -2 | -1 | 2 | 8 | -4 | 0 | 1 | 0 | -1 | 1 | 0 | 1 | 1 | -1 | -2 | O |
| E | -3 | -1 | -1 | -1 | -3 | 0 | 2 | 6 | -4 | 1 | 1 | 0 | 2 | 1 | 2 | 1 | 0 | 2 | H |
| Q | -3 | 0 | -1 | -1 | -2 | 0 | 0 | 2 | 7 | -2 | 0 | 0 | 0 | 2 | 1 | 1 | 0 | -1 | R |
| H | -3 | -1 | -2 | -2 | -2 | 1 | -1 | 0 | 1 | 0 | 1 | 0 | -3 | 1 | 1 | 1 | 1 | 1 | K |
| R | -4 | -1 | -3 | -3 | -2 | -3 | -1 | -2 | 0 | 1 | 0 | 7 | -3 | 0 | 0 | 1 | 2 | 0 | 0 | M |
| K | -3 | 0 | -1 | -1 | -1 | -2 | 0 | 1 | 2 | 3 | 6 | -1 | -1 | 1 | 0 | 1 | I |
| M | -2 | -2 | -1 | -1 | -3 | -2 | -2 | -2 | 0 | -1 | -2 | 7 | -1 | 1 | 1 | 1 | 1 | L |
| L | -3 | -1 | -3 | -4 | -3 | -4 | -3 | -4 | -4 | -3 | 2 | 5 | -2 | 1 | 0 | 0 | 1 | V |
| V | -2 | -3 | -1 | -4 | -4 | -3 | -3 | -4 | -3 | -3 | 3 | 2 | 5 | -1 | 1 | 1 | 3 | F |
| F | -2 | -3 | -2 | -4 | -4 | -4 | -5 | -4 | -4 | -4 | -4 | -3 | 1 | 4 | 1 | 5 | 0 | 2 | Y |
| Y | -3 | -2 | -3 | -2 | -3 | -2 | -3 | -2 | -2 | -1 | -2 | -1 | 0 | -1 | -1 | -1 | 4 | 8 |
| W | -5 | -4 | -3 | -3 | -4 | -3 | -4 | -5 | -3 | -1 | -3 | -3 | -1 | -3 | -2 | -3 | 1 | 2 | 15 |

**Fig. 2.** The BLOSUM 50 substitution matrix (Henikoff and Henikoff, 1992) (lower) and the difference matrix (upper) obtained by subtracting it from the Gonnet matrix (Gonnet et al., 1992) position by position.
III. Theory

Substitution scores can be understood in terms of local alignment theory, which yields a recipe for computing them (Altschul, 1991). We assume that a pair of aligned amino acids $a_i$ from the first sequence and $a_j$ from the second is assigned the substitution score $s_{ij}$. The probability that $a_i$ appears randomly is $p_i$, also called its background frequency, which may be thought of as the abundance of the $i$th amino acid in the protein database being searched. The maximal segment pair (MSP) is the aligned pair of equal length segments, one from each sequence, with the greatest aggregate score, which is computed as the sum of the $s_{ij}$s at each position. One can think of the MSP as the strongest diagonal in a dot plot, where extending it farther will not increase its score. To obtain an MSP, the expected or average substitution score must be negative; otherwise longer alignments would always receive higher scores, and the MSP would just be the longest possible alignment of two sequences. These MSP scores follow an extreme value distribution, which is completely specified by location and scale parameters, analogous to the mean and standard deviation of a normal distribution. The Karlin–Altschul theory of local alignment statistics (Karlin and Altschul, 1990) yields the result that the scale parameter $\lambda$ is calculated as the unique solution of the equation

$$\sum_{i=1}^{20} \sum_{j=1}^{20} p_i p_j e^{s_{ij}} = 1$$

(1)

Among MSPs from the comparisons of two random sequences using the scores $s_{ij}$, residues $a_i$ and $a_j$ will be aligned with target frequency

$$q_{ij} = p_i p_j e^{s_{ij}}$$

(2)

The target frequencies characterize the type of alignments favored by the substitution scores and can be thought of as the frequencies of residue pairs in correct alignments. So, given any set of substitution scores $s_{ij}$ and background frequencies $p_i$, the scale parameter $\lambda$ and set of target frequencies $q_{ij}$ can be deduced. Conversely, given a set of target frequencies and background frequencies, substitution scores can be computed:

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

(3)

$\lambda$ can be selected as any convenient scaling factor because multiplying all scores by a constant does not affect the MSP. When $\lambda$ is set equal to
The scores are scaled in “nat” units; when it is set to ln 2, they are scaled in “bit” units; when it is set to ln 2/2, in half-bits, etc. These scores are called log-odds scores, because they are computed as the log of an odds ratio, which is the ratio of target to background frequencies. A positive log-odds score indicates that the exchange is more likely to occur in a target alignment than in a chance alignment, whereas a negative score indicates the opposite, so log-odds scores measure similarity.

With the scores scaled in bit units, the relative entropy of the target and background frequencies is computed as

\[ H = \sum_{i=1}^{20} \sum_{j=1}^{20} q_i s_{ij} \]  

interpreted as the average amount of information available per aligned position to distinguish the alignment from chance as modeled by the background frequencies. Higher \( H \) implies that the target and background distributions can be more easily distinguished and that shorter alignments with the target distribution can be detected. Similarly, lower \( H \) implies longer alignments are needed. The key to constructing effective substitution scores is to estimate good target and background frequencies for the alignment task.

## IV. Models for Substitution Scores

### A. MDM (PAM) Model

The use of substitution scores was pioneered more than 30 years ago by Dayhoff and coworkers with the mutation data matrix (MDM) model (Dayhoff and Eck, 1968; Dayhoff, 1978). They realized that sequence pairs could be accurately aligned using log-odds scores as described previously, and that target frequencies could be collected by examining amino acid replacements in sequences so similar that they can be aligned based on shared identities. Dayhoff et al. (1968) conservatively aligned sets of nearly identical sequences (no more than 15% different), which were known to be homologous, and inferred a common ancestral sequence, then tallied the amino acid exchanges that had apparently occurred between the ancestor and the present-day sequences. To construct scoring systems useful for more distant relationships, they introduced the concept of an accepted point mutation (PAM), a basic unit of evolution defined as the mutation of a single amino acid in a sequence.
such that the new amino acid may be accommodated in the structure and function of the protein. They then estimated the target frequency for each amino acid pair corresponding to an overall rate of one accepted mutation per 100 positions (1 PAM unit). Next, they applied a Markovian model in which each mutational event is assumed to be independent of previous events; this allowed extrapolation to higher PAMs in order to represent the compounding of successive mutations. So the popular PAM 250 matrix is constructed by multiplying PAM 1 by itself 250 times and represents 250 accepted point mutations.

The series of matrices of substitution scores based on the MDM model at increasing PAMs can be used to score alignments at increasing distances (George et al., 1990). Longer alignments are required to detect more distant relationships, which are expected to have weaker sequence similarity. For example, PAM 250 has relative entropy of about 0.36 bits per aligned residue, whereas that of PAM 120 is 0.98 (Altschul, 1991). The relative entropy indicates how much information per position is optimally available, and, for a given alignment, the optimal score can be attained only by using the appropriate matrix. To decide on the appropriate matrix in advance, one must take into account how many bits of information must be obtained from a correct alignment to distinguish it from the highest scoring chance alignments. About 16 bits are needed for a significant pairwise alignment between two protein sequences of average length, and so 32 bits (16 + \log_2{64,000} = 32) are needed to distinguish a target alignment from chance for a database of 64,000 sequences, etc. PAM 250 should be most effective when weaker correct alignments range from about 50 to 130 amino acids long, whereas PAM 120 should be most effective for shorter, stronger alignments, from about 20 to 50 amino acids long (Altschul, 1991).

Although Dayhoff's method has been criticized (Wilbur, 1985; George et al., 1990; Benner et al., 1994), the basic notion of obtaining target frequencies from alignments was seminal and is incorporated into all currently popular substitution matrices. The MDM series made excellent use of the limited sequence data and computational tools available at the time and was the standard for nearly all protein alignment applications for decades. In 1992, the MDM model was modified by Jones and coworkers (Jones et al., 1992). Instead of attempting to infer common ancestral sequences for sets of homologous proteins, they counted substitutions between pairs of highly similar present day sequences to automatically calculate PAM 1 from an entire protein database.

A potential source of inaccuracy in the MDM model comes from the extrapolation of alignments of closely related sequences to model distant relationships. For example, closely related sequences are dominated by
amino acid exchanges that require only a single nucleotide exchange. In general, the genetic code places constraints on the occurrence of mutations, and this can potentially bias target frequencies (Wilbur, 1985). In 1992, Gonnet and coworkers (Gonnet et al., 1992) addressed this problem by basing the extrapolation on mutation rates estimated from more distant relationships. Alignments were obtained from all-versus-all pairwise matching of a protein sequence database for calculation of target and background frequencies (Benner et al., 1994).

B. BLOSUM Model

A feature of the MDM model is that accepted mutations will be most frequent in positions that are the least subject to selective constraints, and these are the positions that are least likely to be conserved over long evolutionary periods. If there are systematic differences in the rate of accepted mutations between conserved and nonconserved regions of proteins, extrapolations from PAM 1 will be inaccurate in precisely those conserved regions that database searching detects. Although extrapolation from more distant relationships (Gonnet et al., 1992) will allow inclusion of mutations occurring in slowly evolving regions, there are more mutational events at diverged positions, and these will account for the bulk of the substitution data used to obtain target frequencies. Moreover, the process of extrapolation that is used to model distant evolution has the potential of magnifying biases and inaccuracies in the initial dataset.

To address these problems, the authors introduced the Blocks substitution matrix (BLOSUM) series of substitution scores in 1992 (Henikoff and Henikoff, 1992). The idea was to mimic what happens in a local alignment; target frequencies were obtained by directly counting amino acid pairs from multiple alignments of the conserved regions of distantly related sequences, known as blocks, without any extrapolation. Substitution scores were then calculated from target frequencies. To reduce the contribution of closely related sequences to pair counts, sequence segments were clumped within blocks based on percentage identity, and the contributions of segments within a cluster were averaged. For example, BLOSUM 62 is the log-odds matrix derived from pair counts between clusters of sequence segments that are less than 62% identical. This procedure resulted in a series of log-odds matrices parameterized by the cluster percentage. Matrices with a higher cluster percentage have higher relative entropy, corresponding to MDM matrices of lower PAM: BLOSUM 45 has about the same relative entropy as PAM 250,
whereas BLOSUM 62 and PAM 160 have relative entropy of about 0.70 bits (Fig. 3).

Although the BLOSUM model is not explicitly evolutionary, the clumping procedure, which is required to construct a series of matrices, may be interpreted in evolutionary terms. The restriction of counts to conserved regions makes these matrices especially appropriate for database searching, where distant relationships are detected primarily because conserved positions are shared.

C. Structure-Based Models

Suitable target frequencies representing distant evolutionary relationships can also be obtained from alignments based on superposition of amino acid residues in structural alignments. Because protein sequence alignment reflects underlying structural alignment, the target frequencies obtained from sequence alignments and structural superpositions should be similar. Indeed, a log-odds matrix based on structural alignment data, also published in 1992 (Overington et al., 1992) provided scores that were roughly comparable to matrices in the PAM and BLOSUM series in types of substitutions that are preferred (Johnson and Overington, 1993). Correlation analysis reveals that all popular alignment-based matrices favor substitutions that maintain hydrophobicity and amino acid volume over other types of amino acid properties, such as secondary structure (Tomii and Kanehisa, 1996).

![Fig. 3. Relative entropies for BLOSUM clustering percentages and Dayhoff PAM units.](image-url)
D. Other Models

For phylogenetic reconstruction, simplifications inherent in current models may reduce accuracy. For example, the MDM model infers mutation rates from a common ancestor without consideration of events that may have occurred in the intervening period. To address this problem, Koshi and Goldstein (1995) developed a different evolutionary model. Instead of starting with alignments of pairs of present-day homologs, deducing a common ancestral sequence, and counting substitutions between the ancestor and present-day sequences, they started with 120 more complete, but still very simple phylogenetic trees drawn from alignments of multiple present-day homologs. Then they considered all possible mutational paths from the root of each tree to the present-day sequences, summing over all possible substitutions at each node, and finally computed an optimal substitution matrix for the given evolutionary trees and sequences. To do this, they used a Bayesian formalism to find the most probable substitution scores that can account for the observed alignments and the trees. This method produces a single set of substitution scores, which they expect will be useful for the reconstruction of ancestral sequences.

Substitution matrices have also been modeled on amino acid or protein structural properties rather than on alignment data (Grantham, 1974; Miyata et al., 1979; Rao, 1987). However, the relationship between scores and target frequencies means that any imaginable set of scores has an implicit set of target frequencies, and obtaining these from the alignment data itself is the most direct approach (Altschul, 1991).

V. Special Substitution Scores

The MDM, BLOSUM, and structure-based substitution scores discussed so far are general purpose and are intended to compare any two protein sequences. A number of other substitution matrices have been constructed for special tasks. The structure-based target frequencies discussed previously were subdivided into environment-specific tables, and substitution matrices were constructed with the expectation that they would improve alignment to sequences with known structures (Overington et al., 1992). Indeed several other groups have published environment-specific or secondary structure-specific substitution matrices based on 3D structural parameters (Luthy et al., 1991; Miyazawa and Jernigan, 1993; Gracy et al., 1994; Koshi and Goldstein, 1995). An interesting application of these matrices is secondary structure prediction. Mehta and coworkers (1995) obtained excellent predictions of...
helix, turn, and coil regions of multiple alignments by comparing regional alignment scores using secondary structure-specific matrices.

Other specialized substitution matrices have been modeled on transmembrane regions, which appear to be under different selective constraints that the globular proteins that contribute the most to target frequencies of general substitution matrices. For example, Jones and coworkers (Jones et al., 1994) used aligned transmembrane regions to construct a transmembrane PAM matrix that was quite dissimilar from general purpose matrices. Because so many protein families of current interest, such as the G-protein-coupled receptors, are highly diverged transmembrane proteins, special transmembrane protein matrices are potentially valuable. However, their superiority to general purpose matrices has not been demonstrated by comprehensive performance evaluations.

VI. Gap Scores

A. Aligning Residues with Null Characters

Local alignments can be based purely on substitution scores; for example, a dot plot represents HSPs as diagonals (Fig. 1). Where the end of one diagonal is near the beginning of another diagonal, the eye can easily connect them. In Fig. 1, the intervals between HSPs 2 and 3 and 3 and 4 appear easy to bridge, but bridging the interval between 1 and 2 seems more difficult. An algorithm that finds an optimal alignment requires explicit gap scores to decide whether and how to make a connection. The total alignment score is the sum of the substitution scores for aligned pairs of amino acids plus the sum of the gap scores for residues in one sequence aligned with null characters in the other. Gap scores are scaled to the substitution scores so that the inclusion of a gap penalizes the total alignment score. An entry for the null character could be added to a substitution matrix, resulting in a 21 × 21 matrix, but this is not usually done in practice because estimating the scores depends on having reliable gapped alignments. Structural superpositions or phylogenetic trees can provide these alignments, and some matrices may include these entries (Overington et al., 1992; Koshi and Goldstein, 1995). However, the practical value of residue-versus-null scores has not been established, and popular sequence alignment programs do not explicitly use them.

Early gap scores charged the same fixed penalty for any residue in either sequence aligned with a null character in the other (Needleman
and Wunsch, 1970; Sankoff, 1972). However, more accurate alignments were obtained by using schemes based on the realization that a single evolutionary event might result in a gap of almost any length. The most popular of these schemes are affine gap scores, which charge a stiff gap-opening penalty for the first null character in a gap and a lesser gap-extension penalty for subsequent null characters in the same gap (Fitch and Smith, 1983). The most widely used formula for gap extension is one in which the penalty is proportional to gap length. Note that this formula has no theoretical rationale, but is easy to implement. Based on curve fitting to their alignment data, Gonnet and coworkers (1992) have argued for a nonlinear gap penalty; however, their formula has not been adopted by popular alignment-based programs. In ClustalW, several modifications have been introduced to encourage gaps in likely loop regions by making gap scores position-specific (Thompson et al. 1994).

The most popular database searching programs have switched from strictly local ungapped alignments (FASTP [Lipman and Pearson, 1985]) and BLASTP [(Altschul et al., 1990)] to local gapped alignments (FASTA [Pearson, 1990] and Gapped BLAST [Altschul et al., 1997]) using affine gap scores. FASTA matches gap scores with substitution scores based on extensive empirical testing (Pearson, 1995), whereas Gapped BLAST gap scores are based on simulations (Altschul and Gish, 1996). Extensions to local alignment theory are needed to accommodate gap scores because it only applies to the MSP (highest scoring segment pair) in the dot plot, and these extensions have not yet been completely worked out. However, empirical evidence from several studies suggests that the distribution of optimal scores from gapped alignments still follows an extreme value distribution as long as gap penalties are severe enough (Mott, 1992; Vingron and Waterman, 1994; Altschul et al., 1997). Insufficiently severe gap scores relative to substitution scores can lead to optimal alignment scores that do not follow this distribution but simply increase with alignment length. Recently, Mott and Tribe (1999) have applied a simplified model to investigate the theoretical distribution of gapped local alignment scores. Encouragingly, their simulations suggest that the most widely used substitution and gap score combinations provide alignment scores that conform to the principles of local alignment theory, confirming empirical evidence.

**B. Leaving Unalignable Regions Unaligned**

A generalized model for affine gap scores recognizes that some regions of pairs of similar sequences should be left unaligned (Altschul, 1998).
Although this basic idea underlies motif-based multiple alignment methodology (Posfai et al., 1989; Henikoff and Henikoff, 1999), it has only recently been applied to pairwise alignment methods (Alexandrov and Luthy, 1998; Altschul, 1998). To understand how generalized affine gap costs work, imagine that there are two HSPs separated by an unalignable region of equal length in both sequences. This might be the case for independent insertions of loop regions into homologous genes. Local alignment algorithms might be unable to connect up the two truly alignable regions because of the many negative scores for the intervening aligned amino acid pairs, thereby missing a true positive alignment in a database search. However, it may be possible to connect up the two HSPs by applying instead a weak fixed penalty for intervening unalignable regions. The resulting alignment, with aligned homologous regions separated by an unaligned nonhomologous region, has the potential of being more realistic than one obtained with affine gap scores. This gap model has not yet been incorporated in popular local alignment programs.

In the example (Figs. 1 and 4), the C-terminal domain of human breast cancer type 1 (BRCA1) protein is aligned with human p53 binding (53BP1) protein (Koonin et al., 1996). In the highest scoring alignment using Gapped BLAST with BLOSUM 62 and default affine gap scores, short regions of high similarity are separated by a region that seems arbitrarily aligned, which did not provide enough information to raise it above the twilight zone in a database search. As a result, this interesting similarity was missed using conventional pairwise alignment programs. Altschul (1998) then showed that sufficient information is obtained when these sequences are aligned using generalized affine gap scores with the same BLOSUM 62 matrix. A major difference between the alignment obtained with affine gap scores (Fig. 4B-D) and that with generalized affine gap scores (Fig. 4A) are two regions between HSPs 1 and 2 that have multiple gaps when scored with affine gap scores, but are left unaligned with generalized affine gap scores.

By leaving highly uncertain regions unaligned, generalized affine gap scores provide a more informative alignment display. Pairwise alignments are typically shown with all positions aligned, even though some regions are essentially unalignable. Ultimately, better evidence is likely to show these alignments to be largely incorrect. Thus, researchers who would consider a small overinterpretation of experimental data to be a serious breach routinely publish heavily overinterpreted alignments and draw conclusions from them. This problem is mitigated by the use of alignment displays that show unalignable regions in lower case (as in Fig. 4A) or not at all. As a practical matter, regions of alignment uncertainty can
be identified in the pairwise context because they are sensitive to changes in the scoring function. To illustrate this point, consider the example chosen by Altschul for generalized affine gap scores (Fig. 1, 4A). Three popular matrices with optimized gap scores give identical HSP alignments, but there is not a single position that is aligned the same for the

Fig. 4. Alignment of two sequences using four different scoring functions. (A) From Altschul (1998) and (B–D) from Smith–Waterman alignments using SSEARCH (Pearson, 1996). Asterisks with numbers above the alignments correspond to the four HSPs indicated in Fig. 1.
49–53 amino acid (aa) region between HSPs 1 and 2 (Fig. 4B-D). Thus, alignment of this region cannot distinguish between divergence from a common ancestral segment and nonhomologous replacement of an ancestral segment in either lineage.

C. Global Alignments

Although the discussion so far has concerned local alignment methods, many alignment applications are global, in that they assume the proteins being aligned are homologous from one end to the other. It is well known that this model is unrealistic for a large fraction of distant protein relationships, especially for modular proteins; thus database searching programs such as BLAST, FASTA, and SSEARCH utilize local alignment algorithms. Global alignment algorithms are unsuitable where only a portion of the sequences are homologous. These algorithms force misalignments outside of homologous regions to the ends of proteins. However, in applications in which these issues are not of practical concern, global alignments may be more sensitive to distant relationships. For example, Clustal (Thompson et al., 1994) is a popular global alignment program that achieves high sensitivity in part because it will traverse dissimilar regions to connect similar ones. To convert a local alignment program into a global program, one can simply add a constant value to the substitution matrix such that the minimum score is zero. As a consequence, alignments will be forced to proceed to the end of one of the sequences in both directions. This procedure solves the problem of traversing unalignable regions, although it does so by often reporting a questionable alignment. An alternative to using positive matrices to obtain a global alignment is to adjust gap scores. A strictly local alignment can be converted into a global one by reducing the gap extension penalty. This allows an alignment program to connect separated HSPs by increasing the likelihood that a dissimilar region in between will be traversed, which can increase sensitivity to distant relationships where avoidance of false-positive result is not an issue.

VII. Evaluating Scoring Functions

A. Based on Database Searching Performance

The proof of the pudding is in the eating, and so it goes for scoring functions given the importance of alignment-based methods in biology. Replacing one scoring function by another is a simple matter, and most
Web search sites allow the user to choose among several. Their easy interchangeability means that substitution and gap score combinations are readily compared in performance. Over the past several years, results of numerous comprehensive tests have appeared. Some of these evaluation studies have confined themselves to log-odds matrices made from alignment data, whereas others have included matrices that are based on amino acid properties.

For comprehensive evaluation of scoring functions, numerous database searches or alignments are performed, where the correct outcomes are assumed to be known. To assess searching performance, the separation between true positives and true negatives is measured. It is important to realize that different applications may perform best with different scoring functions, and so searching and alignment programs include scoring system recommendations and defaults (Pearson, 1995; Altschul et al., 1997). Although it is usually an easy matter to try multiple scoring functions in database searches, one should be aware that each change is likely to pull up different high scoring false-positive scores, and as a result, multiple searches increase the chance of a false-positive result being mistaken for a true-positive one.

We introduced the BLOSUM series of substitution matrices (Henikoff and Henikoff, 1992) with comprehensive searching results demonstrating much better overall performance when used with BLASTP and FASTA compared with matrices based on the MDM model, including those of Jones et al. (1992) and Gonnet et al. (1992). BLOSUM 62, the best performer with BLASTP, was adopted as the default substitution matrix for the BLAST family of programs in 1993. Presumably, better performance derives from the fact that BLOSUM matrices are modeled on conserved regions, which are the same regions detected in database searches. Performance differences increased as matrices decreased in relative entropy (higher PAM, lower BLOSUM), suggesting that the extrapolations inherent in the MDM model may have magnified inaccuracies in the target frequencies (Henikoff and Henikoff, 1993). This inference was supported by the finding that a structure-based matrix with target frequencies derived directly from alignments also performed better than all extrapolated matrices (Overington et al., 1992). Dayhoff’s 1978 PAM series was outperformed at all relative entropy levels by the updated 1992 series (Jones et al., 1992), indicating that increased amount of alignment data improved performance. The updated PAM matrices therefore are recommended for applications in which an evolutionary model is needed.

A limitation of the original BLOSUM tests is that they did not assess the effect of gap scores. However, Pearson (1995) has comprehensively
evaluated substitution matrices over a range of gap scores. Although his performance results were similar to ours for BLASTP using a different test set, optimization of substitution and gap score combinations mostly neutralized performance differences between BLOSUM matrices and extrapolated matrices. For example, best overall performance with the Smith–Waterman algorithm was obtained using either BLOSUM 45 or the Gonnet matrix paired with appropriate gap scores. Some differences were seen among substitution and gap score combinations depending on the statistics used to order search scores, and as a result, Pearson found somewhat different optimal score combinations for different versions of FASTA and the Smith–Waterman algorithm.

B. Based on Alignment Accuracy

The preceding scoring evaluation studies were based on database searching performance. What made this possible were large compilations of protein families, such as PROSITE (Bairoch and Boeckmann, 1992) and PIR (Protein Information Resource) superfamilies (Wu et al., 1996) in which the large majority of true family members are identified with confidence. Detection of true-positive identifications and exclusion of false-positive ones in a similarity search can then be accurately assessed. A different task is alignment of a pair of related sequences, where more alignment accuracy is desirable. For this, matrix evaluation studies have compared sequence alignments of homologs to 3D superpositions of the same proteins, and the structural alignment is deemed the correct one. The measurement of alignment accuracy against an accepted standard emphasizes a different aspect of scoring function performance from that emphasized in searching. In searches, scores for alignment of conserved regions are the ones that matter most, because the challenge is to avoid false-positive hits resulting from the large number of alignments scored. However, when aligning two diverged protein sequences, it is the diverged regions that are most challenging to align and that likely account for performance differences. Another difference is that evaluation studies based on alignment, as opposed to database searching, utilize global alignment algorithms, and so typically transform substitution scores to all positive numbers. As a result, a matrix that better tolerates uncertain alignments may outperform a more fastidious matrix by allowing high-scoring segments to be connected through intervening regions for which no alignment is meaningful.

Johnson and Overington (1993) measured alignment accuracy and found good performance for structure-based, BLOSUM, and Gonnet matrices. As in other studies, log-odds matrices performed the best.
However, their test set was limited to the relatively small number of families available with multiple structures, and performance differences appear to be rather minor. A subsequent study by Vogt and coworkers (Vogt et al., 1995) utilized a more diverse set of homologous structures and reached essentially the same conclusions. In both studies, a range of gap scores was explored to find the set that gave the best performance for the substitution scores being tested. It is revealing that slight changes in gap penalties affected matrix performance. For instance, the best matrix in these tests of Vogt et al. was a positive Gonnet matrix with gap penalties of 6 for opening and 0.8 for extension, but it fell below sixth place when the extension penalty was reduced to 0.6. A positive BLOSUM 50 performed nearly as well as Gonnet (6, 0.8), but with quite different gap penalties (9.5, 0.6). These results emphasize the importance of applying gap scores that are optimized for the particular substitution scores.

C. Based on Fold Recognition

Abagyan and Batalov (1997) compared substitution matrices in the context of this question: “Do aligned sequences share the same fold?” They performed an all-versus-all search of a database consisting of 2819 sequences with known structures, then ranked matrices based on how many true-positive sequences scored higher than the first false-positive sequence. In this case, structural superposition provided the standard for distinguishing true-positive sequences, which share a common fold, from false-positive sequences, which do not. They found that making substitution scores positive, as was done for the alignment accuracy tests described previously, led to poorer performance. A likely explanation is that good performance in aligning two sequences is achieved by tolerating misalignment, but in a database search, such tolerance will lead to an unacceptable level of background hits. By using so many sequences in this cross comparison, Abagyan and Batalov were able to sensitively detect performance differences among the substitution and gap penalty combinations that they tested. The Gonnet matrix and BLOSUM 50 with rather different gap penalties detected the most true positive alignments: the Gonnet matrix was more sensitive, detecting 75.3% of significant alignments in the twilight zone, compared to 72.0% for BLOSUM 50. However, the Gonnet matrix also reported alignments to 90.9% of unrelated sequences, compared to 56.2% for BLOSUM 50. Thus it seems that top performance in fold recognition is accompanied by a tolerance for misalignment.
Substraction of BLOSUM 50 from the Gonnet matrix reveals a preponderance of positive off-diagonal values (Fig. 2, Upper); thus low-scoring regions will tend to score higher with the Gonnet matrix than with BLOSUM 50. This could account for differences seen in Fig. 4B-D, where gaps were inserted into the central 49-53 aa region using both BLOSUM matrices, whereas an ungapped alignment was returned using the Gonnet matrix. These differences may reflect differences in derivation of the matrices: BLOSUM target frequencies were obtained from ungapped conserved regions of proteins, whereas the Gonnet matrix was obtained from pairwise global alignments. In each case, an iterative procedure was used to optimize the matrix to the alignment model. As a result, BLOSUM matrices are optimized for finding conserved regions and so may be more selective, and the Gonnet matrix is optimized for finding global alignments and so may be more sensitive.

VIII. Position-Specific Scores

Substitution and gap scores are tailored to the problem of aligning pairs of protein sequences. However, once a pair is aligned, the alignment itself becomes available for deriving scores characteristic of those sequences. In fact, the most widely used multiple alignment programs, such as Clustal, proceed in just this way, first aligning pairs of the most closely related sequences, then using the pairwise alignment scores to align to another close sequence, and so on. In other words, a sequence can be generalized to a multiple sequence alignment which can then be compared with another sequence in the same manner as a single sequence. Rather than being composed of single residues at each position, the multiple alignment counterpart of a sequence can have as many as 20 different amino acids plus a null at each position. It is convenient to represent each position as a vector of scores, and the ordered set of such vectors constitutes a position-specific scoring matrix (PSSM), also called a profile (Gribskov et al., 1987) or hidden Markov model (Krogh et al., 1994).

In a sense, a substitution matrix is the poor man’s PSSM because it applies the same 20 residue scores to every position in an alignment. A PSSM is designed to provide scores for aligning a multiple alignment with a sequence in the same way that substitution and gap scores are designed to provide scores for aligning two sequences. Figure 5 illustrates the relationship between a substitution matrix and a PSSM with a plot of scores along a sequence (MYOD CHICK) that contains two conserved blocks (labeled A and B) representing a helix–loop–helix motif. In this “embedded” PSSM (Henikoff and Henikoff, 1997), scores for the blocks
Fig. 5. Graphical representation of an embedded PSSM representing the helix–loop–helix DNA-binding motif and flanking sequences. Adapted from Henikoff and Henikoff (1997) with permission. The location of the motif is shown as two boxes in (A). (B) PSSM columns from two different positions (corresponding to K137 and K143) in the B motif (arrows in A) and from the three lysines (K90, K95 and K95) in MYOD_CHICK flanking sequence (arrowheads in A).

are computed from the multiple alignment, and scores outside of the blocks are taken from a substitution matrix (BLOSUM 55) corresponding to the MYOD_CHICK sequence (position numbers below). Each aligned position is associated with a vector of 20 scores (represented by triangles in Fig. 5A). Thus, a lysine in any flanking position is scored the same as any other flanking lysine (arrowheads). In contrast, the invariant lysine at position 137 (K137) has a different score vector from that of K143, which is less strongly conserved (Fig. 5B and arrows in Fig. 5A). Notice that the PSSM scores are distinctive in having many strongly negative scores, resulting in stronger penalties for residues that are unlikely to be found at these positions in an alignment. Because the large majority of sequences will align with some of these low scoring residues, the PSSM increases specificity (excludes more false-positive hits) in a database search better than a substitution matrix.

The specific information present in a PSSM, even when it is made from only two or three homologous sequences, particularly favors the similarity between the sequences aligned to compute it and can improve searching sensitivity dramatically. This is because constraints on each position in a multiple alignment become better defined as more exam-
examples of the protein family are added to it. A single sequence provides only a single example residue for each position, but multiple sequences provide multiple examples, giving additional evidence about the range of allowable residues found there and their degree of conservation. There are many different ways to compute a PSSM from a multiple alignment. Issues that must be dealt with include how to handle redundant evidence from nearly identical sequences [use sequence weights to penalize redundancy (Altschul et al., 1989; Sibbald and Argos, 1990; Gerstein et al., 1994; Henikoff and Henikoff, 1994; Thompson et al., 1994; Eddy et al., 1995)] and how to treat unobserved residues [add fractional imaginary occurrences modeled on what is expected based on the examples observed (Claverie, 1994; Tatusov et al., 1994; Bailey and Gribskov, 1996; Henikoff and Henikoff, 1996; Sjolander et al., 1996)].

Searching sequence databases with an embedded PSSM query (Henikoff and Henikoff, 1997) has been elegantly applied in the popular position-specific iterated BLAST (PSI-BLAST) program (Altschul et al., 1997). After one round of Gapped BLAST searching using conventional substitution and gap scores, all hits above a threshold are aligned with the query sequence and a PSSM column is computed from each aligned position. There will typically be segments of the query sequence with many aligned hits and other segments with few. Subsequent rounds of searching are performed by scoring alignments of the PSSM from the preceding round with the database, adding new hits to the multiple alignment, and recalculating the PSSM. PSI-BLAST is remarkable for its sophisticated implementation, sensitivity, speed, and ease of use. As a result, biologists with no special expertise can readily detect distant and profound sequence similarities that were previously confidently detected only by experts.

PSI-BLAST becomes more powerful as databanks expand because more alignable hits become available to refine its PSSMs. An alternative iterative approach that also takes advantage of expanding databanks is transitive pairwise searching (Pearson, 1996; Neuwald et al., 1997; Grundy, 1998). Here, hits from the initial search are themselves used to query the database, and hits in common among these multiple searches are considered to be true-positive results. In phylogenetic terms, this procedure involves starting at a single leaf of a tree, finding leaves connected by nearby nodes, then using these leaves to find leaves from more distant nodes. This procedure differs from PSSM-based approaches, in which multiple alignment information is an approximation of the root node of a phylogenetic tree. PSSM and transitive searching strategies are complementary and may be used in combination (Neuwald et al., 1997).
IX. Using Multiple Substitution Matrices

A. Multiple Searches

Substitution scores may be interpreted as quantifying the evolutionary distance between two sequences. Therefore, using the same scores over the entire length of a sequence would seem to assume that it has evolved everywhere at the same rate. Also, when searching the sequence against a large number of others, the evolutionary distances of any similarities are not known in advance, so using any one set of scores is a compromise. One response to these issues is simply to do the search multiple times with different scoring functions. Altschul (1993) showed how to sensibly select a set of matrices to cover all evolutionary distances and also worked out the consequent statistics. Tests of this strategy showed that, when used with BLAST, the combined searches did indeed identify more known homologs (Henikoff and Henikoff, 1993). However, after an appropriate statistical correction was made for performing multiple searches, the results were no more significant than searching with a single good matrix. Therefore, doing multiple searches would be most advisable when there is reason to suspect that the default matrix is not efficient for true-positive alignments involving the query.

B. Subdividing the Query

One problem when searching a large database with a query that contains some compositional bias, such as runs of prolines, is the propensity of any scoring function to match sequences in the database with a similar bias. To combat this problem, compositionally biased regions of the query are commonly excised algorithmically using filters such as SEG (Wootton and Federhen, 1993) and XNU (Claverie and States, 1993). These filters simply prevent some regions of the query from being scored at all.

Conversely, filters may be used to select regions to be scored. Compositional bias has been used to recognize regions of biological interest, such as coiled-coil and transmembrane segments (Wootton, J. C. and Sonnhammer, E. M., 1999 personal communication). These filters might be used to scrutinize the query sequence for particular regions before scoring, and then a scoring function customized for those regions could be applied.

C. Dispensing with Gap Scores

The generalized affine gap model allows for both gaps and unaligned regions. Alternatively, gap scores can be dispensed with completely, in
which case aligned regions are simply separated by unaligned regions. This idea is decades old (Sankoff, 1972), but was only recently reconsidered in the pairwise alignment context (Zhu et al., 1998). The Bayesian block aligner bypasses the requirement to specify particular substitution and gap scores. Instead, a series of substitution matrices and the maximum number of unaligned regions are specified. The substitution matrices are used by the program to generate candidate aligned segments, and the algorithm returns the most probable aligned segments separated by unaligned regions. To date, this algorithm has been implemented only for aligning pairs of sequences and not for database searching.

X. Conclusions

During the past decade, the ability to detect and analyze distant protein sequence relationships has improved substantially. Remarkably, detection ability has improved despite the exponential growth of sequence databanks. This improvement cannot be attributed to better pairwise alignment programs, because the Smith–Waterman algorithm introduced in 1981 remains the gold standard. Rather, improved scoring functions and statistics account for improved searching performance. In addition, scoring function improvements have benefited other widely used alignment-based methods, including multiple alignments, phylogenetic reconstructions, and homology modeling. Current alignment methods use substitution scores introduced in 1992, which have subsequently been optimized by matching them with gap scores that perform best in various applications. It may be that current scoring functions are as good as they can be for their intended applications given that general purpose scoring functions are inherently compromises. However, gap scores in current use are still crude and are expected to improve as the theory underlying them is better understood. The efficient exploitation of more and better multiple alignment information means that we will more often be using PSSMs and less often applying substitution matrices, and PSI-BLAST exemplifies this trend. With the easy accessibility of powerful alignment-based tools via the World Wide Web, future advances should become available almost immediately to biologists.

Acknowledgments

Our work is supported by a grant from NIH. Preparation of the manuscript was aided by stimulating discussions during a visit to NCBI. We thank Paul Talbert and Elizabeth A. Greene for comments.
REFERENCES

Dayhoff, M. O., and Eck, R. V. (1968). ‘‘Atlas of protein sequence and structure.’’ Vol. 3 National Biomedical Research Foundation, Silver Spring, Maryland (p. 33).
AMINO ACID SUBSTITUTION MATRICES