Chapter 3

Amino Acid Substitution Matrices

In prior lectures, we introduced Markov models of nucleotide substitution. We derived expressions for the probability that nucleotide x will change to nucleotide y after elapsed time t. Further, we used the model to account for multiple substitutions, by estimating the number of actual substitutions that occurred, given the number of observed mismatches.

Here, we focus on Markov models of amino acid replacement and their use in deriving amino acid substitution matrices. An amino acid substitution matrix assigns a score to a pair of aligned amino acids, x and y. A good substitution matrix should have the following properties:

- *Biophysical properties of residues:* Amino acids differ in size and charge. Some are acidic, some are basic, some have aromatic side chains. Generally, replacement of an amino acid with another amino acid with similar properties is less likely to break the protein or cause dramatic changes in function than replacement with an amino acid with different properties. A substitution matrix should reflect this.

- *Evolutionary divergence:* The observation of identical or functionally conservative amino acids at the same site is more surprising in highly diverged protein families than in families characterized by little sequence divergence. The best results are obtained using a substitution matrix based on the statistics of amino acid replacements typical of the degree of evolutionary divergence of the proteins under consideration. Therefore, a family of matrices that is parameterized by sequence divergence is desired.

- *Multiple substitutions:* The score associated with an amino acid pair, x and y, should reflect the probability of observing x aligned with y, taking into account the possibility of multiple replacements at the same site.

There are two commonly used families of amino acid substitution matrices that have these properties, the PAM matrices (Dayhoff *et al.*, 1978) and the BLOSUM matrices

(Henikoff and Henikoff, 1992.) There are two commonly used families of amino acid substitution matrices that have these properties, the PAM matrices (Dayhoff *et al.*, 1978) and the BLOSUM matrices (Henikoff and Henikoff, 1992.) Both substitution matrix families are parameterized by sequence divergence. The PAM matrices are based on a formal Markov model of sequence evolution. The BLOSUM matrices use an *ad hoc* approach. Both families were derived according to the following general approach, although the details of each step differ between the two methods.

1. Use a set of "trusted" multiple sequence alignments (ungapped) to infer model parameters.
2. Count observed amino acid pairs in the trusted alignments, correcting for sample bias.
3. Estimate substitution frequencies from amino acid pair counts.
4. Construct a log odds scoring matrix from substitution frequencies.

3.1 A log likelihood ratio framework for scoring alignments

Before introducing the PAM and BLOSUM matrices, we briefly introduce the log likelihood framework in which these matrices were developed. Suppose α^{κ} is an ungapped alignment of sequences σ and τ of length n. We can assign a similarity score to α^{κ} under the assumption of positional independence by adding the similarities of the symbols in each position in the alignment,

$$\mathcal{S} = \sum_{i=1}^{n} p(\sigma[i], \tau[i]), \tag{3.1}$$

where $p(x, y)$ is a quantitative measure of the similarity of x and y. Recall that earlier in the semester, we used a simple scoring scheme with a single match score, $p(x, x) = M$ and a single match score, $p(x, y) = m$, where $x, y \in \sum$ and $x \neq y$. Since all matches (respectively, mismatches) have the same score, under this scoring scheme, if α^{κ} has $\hat{m}$ mismatches and $(n - \hat{m})$ matches, then

$$\mathcal{S} = \hat{m} \cdot m + (n - \hat{m}) \cdot M.$$

This scoring scheme has limitations, especially for amino acids. First, if M and m are chosen arbitrarily, then alignment scores have no intuitive meaning in an absolute sense. For example, if I tell you that a given alignment has a score of 14, you know that it is better than some other alignment of the same sequences that has a score of 12, but you have no way of assessing whether the alignment is inherently good or bad.

Second, this scoring scheme does not take the evolutionary divergence of σ and τ into account. If we are testing the hypothesis that σ and τ are related and have changed very little since they diverged from their common ancestor, then we might interpret any mismatch as evidence against this hypothesis. If we are testing the hypothesis that σ and τ are related and have changed a great deal since their divergence, then we might interpret mismatches

that represent conservative replacements (i.e., the juxtaposition of x and y, where x and y have similar biochemical properties) as evidence that supports this hypothesis. In order to capture these nuances, we require a scoring method that is parametized by evolutionary divergence.

Third, the current scoring scheme treats all replacements in the same way. Since all mismatches are assigned the same score, it cannot reflect differences in the biochemical similarity.

One way of assessing whether an alignment is good in an absolute sense is to ask whether $\alpha^\kappa(\sigma, \tau)$ reflects more similarity than we expect to see by chance. Let H_0 be the hypothesis that σ and τ are unrelated sequences. The alternate hypothesis, H_A, is that σ and τ are related sequences with a given amount of evolutionary divergence. We can assess whether $\alpha^\kappa(\sigma, \tau)$ reflects more than chance similarity by calculating the ratio of the probabilities of the alignment under H_A and H_0. This *likelihood ratio* is

$$\mathcal{LR}(\alpha^\kappa) = \frac{p(\alpha^\kappa|H_A)}{p(\alpha^\kappa|H_0)} \tag{3.2}$$

will be greater than 1 if the alignment of σ and τ is more similar than expected by chance. If the ratio is much greater than 1, then we have strong evidence that the sequences share common ancestry. Under the assumption of positional independence,

$$\mathcal{LR}(\alpha^\kappa) = \prod_{i=1}^{n} \frac{p(\alpha^\kappa[i]|H_A)}{p(\alpha^\kappa[i]|H_0)}, \tag{3.3}$$

where $\alpha^\kappa[i]$ is the alignment of $\sigma[i]$ and $\tau[i]$. Note that since $\log(x)$ increases monotonically with x, the alignment that maximizes $\mathcal{LR}(\alpha^\kappa)$, also maximizes $\log \mathcal{LR}(\alpha^\kappa)$. Thus, $\log \mathcal{LR}(\alpha^\kappa)$ can also be used to assess the extent to which $\alpha^\kappa(\sigma, \tau)$ represents more than chance similarity. Taking the log of both sides of Equation 3.3 yields

$$\begin{aligned} \log \mathcal{LR}(\alpha^\kappa) &= \log \prod_{i=1}^{n} \frac{p(\alpha^\kappa[i]|H_A)}{p(\alpha^\kappa[i]|H_0)} \\ &= \sum_{i=1}^{N} \log \frac{p(\alpha^\kappa[i]|H_A)}{p(\alpha^\kappa[i]|H_0)}. \end{aligned}$$

The right hand side of this equation looks very similar to the right hand side of Equation 3.1, suggesting that we can use the log likelihood ratios to define a scoring scheme. If we define the similarity score of x aligned with y to be

$$p(x, y) = \log \frac{p(\begin{smallmatrix} x \\ y \end{smallmatrix}|H_A)}{p(\begin{smallmatrix} x \\ y \end{smallmatrix}|H_0)},$$

then the score of an alignment is equivalent to the log of the ratio of its probabilities under the alternate and null hypotheses:

$$\mathcal{S} = \log \mathcal{LR}(\alpha^{\kappa}).$$

Thus, we obtain a scoring scheme that can be interpreted in an absolute, as well as a relative context.

The log transformation has several advantages. We obtain a measure of the similarity that can be calculated by a sum instead of a product, avoiding the inconvenience of dealing with very small fractions. If $\log \mathcal{LR}(\alpha^{\kappa}) = 0$, then σ and τ are no more similar than expected by chance; $\log \mathcal{LR}(\alpha^{\kappa}) > 0$ indicates a highly significant similarity. Moreover, if the alternate hypothesis can be expressed by a model of similarity that accounts for the amount of evolutionary change that has occurred, then this log likelihood approach also provides a scoring scheme that is parameterized by evolutionary divergence.

3.2 PAM matrices

The PAM matrices were developed by Margaret Dayhoff and her colleagues in 1978. A PAM is a unit of evolutionary distance. The term "PAM" means "percent accepted mutation." We say the divergence between two sequences is N PAMs, if, on average, N amino acid replacements per 100 residues (including multiple substitutions at the same site) occurred since their separation.

The Dayhoff matrices are parameterized by PAM distance. Dayhoff used the following strategy to obtain amino acid substitution matrices that are parameterized by evolutionary distance:

- Construct a Markov chain to model amino acid substitution at a single site i. This chain has twenty states, one for each possible amino acid at that site. If the chain is in state x at time t, we say that we see amino acid x at site i at time t. Note that this model assumes site independence.

- For this Markov chain, we derive the PAM-1 transition probability, $P_{xy}^{(1)}$, from closely related alignments that are assumed to contain no multiple substitutions. $P_{xy}^{(1)}$ is the probability that amino acid x will be replaced by amino acid y in sequences separated by 1 PAM of evolutionary distance.

- The PAM-N transition probability, $P_{xy}^{(N)}$, is obtained by extrapolating from the PAM-1 transition probability. This is the probability that x will be replaced with y after N time steps. We can also think of $P_{xy}^{(N)}$ as the probability of observing amino acid x aligned with amino acid y in one time step in sequences that are N PAM units apart.

Dayhoff's implementation of the general approach given above is as follows:

1. As training data, Dayhoff *et al* used a set of ungapped, global multiple sequence alignments of 71 groups of closely related sequences. Within each group, the sequence identity was 85% or greater. The rationale is that sequences with at least 85% identity will contain no site that has sustained more than one mutation.

2. Observed amino acid pair frequencies were tabulated from the 71 multiple alignments. Sample bias was corrected by counting the minimum number of changes required to fit the data to a tree, according to a parsimony model. The counts were averaged over all most parsimonious trees. For each tree, T, we calculate A_{xy}^T by counting the number of edges connecting x and y, for $x \neq y$. Note that $A_{xy}^T = A_{yx}^T$, since every edge connecting x with y also connects y with x. We define A_{xx}^T to be twice the number of edges connecting x and x. This is because the edges connecting two dissimilar residues are also counted twice, once in the xy direction and once in the yx direction. The overall counts are obtained by averaging over all trees:

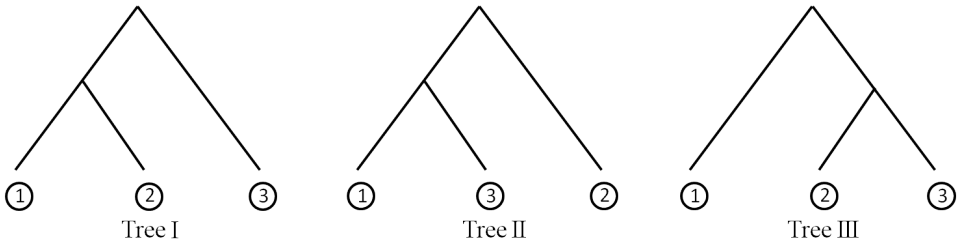
$$A_{xy} = \frac{1}{n_T} \sum_T A_{xy}^T,$$

where n_T is the number of trees with an optimal parsimony score.

To see how this works in practice, suppose we have an alignment of three sequences, each of which has two amino acids:

```
1:  VV
2:  II
3:  VI
```

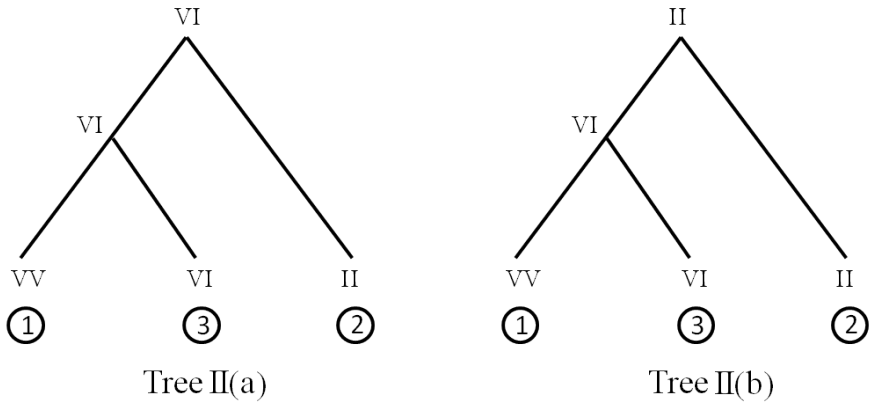
There are three rooted tree topologies with three leaves



Note that although all three trees appear to have the same shape, the leaf labels differ, corresponding to different evolutionary hypotheses. For example, Tree I corresponds to the hypothesis that sequence (1) is more closely related to sequence (2) than to sequence (3), while Tree II says that sequences (1) and (3) are most closely related.

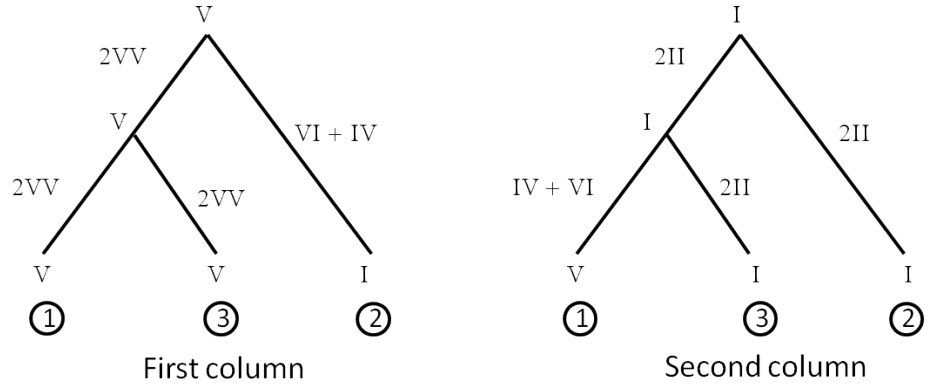
In order to determine A_{xy}^T, the count for tree T, each leaf is annotated with the corresponding present-day sequences (*i.e. VV, II*, and *VI*). The sequences on internal nodes are unknown, since they correspond to ancestral sequences. Dayhoff inferred the sequences on the internal nodes according to the *parsimony criterion*, which states that the best hypothesis is the hypothesis that requires the fewest mutations to explain the data. In other words, she assigned sequences to the internal nodes of each tree in such a way that the total number of changes along branches of the tree is minimal. In general, there can be more than one way to assign sequences to internal nodes that minimizes the total change. In our example, there are two possible assignments of ancestral sequences for both Tree II and Tree III. Tree I has a unique set of internal labels.

The two possible internal node labelings of Tree II are shown here:



Since there is no way of knowing which set of inferred ancestral sequences is correct, we must consider all possibilities. When the internal labels are taken into account in this example, there are five most parsimonious trees, one for Tree I and two each for Trees II and III. In the Dayhoff framework, we estimate the pair counts on each of the five trees and take the average.

The figure below illustrates how state changes are counted for Tree II(a) in our example.



We need to determine the number of pairs of each type in each column and then sum over all columns. The number of `VV`, `VI`, `IV`, and `II` pairs are shown on the left for the first column and on the right for the second column of the alignment. Combining these counts, we obtain

$$A_{VV} = 6, \qquad A_{VI} = 2, \qquad A_{IV} = 2, \qquad A_{II} = 6$$

for Tree II(a).

3. The transition matrix $P_{xy}^{(1)}$ is derived from the counts, A_{xy}, obtained in step 2, as follows:

$$P_{xy}^{(1)} = m_x \frac{A_{xy}}{\sum_{h \neq x} A_{xh}}, \quad x \neq y \tag{3.4}$$

$$P_{xx}^{(1)} = 1 - m_x \tag{3.5}$$

Here, m_x is the "mutability" of amino acid x and is defined to be

$$m_x = \frac{1}{L p_x z} \sum_{l \neq x} A_{xl}, \tag{3.6}$$

where p_x is the background frequency of x, L is the length of the alignment, and z is a scaling that guarantees that the transition matrix will correspond to exactly 1 PAM. We select the scaling factor, z, so that

$$\sum_{x=1}^{20} (p_x m_x) = \frac{1}{100}. \tag{3.7}$$

This scaling factor is required because although the training alignments are sufficiently conserved to contain no multiple substitutions, but the frequency of replacements in each alignment may not be exactly one in a hundred.

We obtain an expression for the scaling factor, z, by substituting the right hand side of equation (3.6) for m_x in equation (3.7) and solving for z. This yields

$$z = \frac{100}{L} \sum_{x=1}^{20} \sum_{l \neq x} A_{xl}. \tag{3.8}$$

We now replace the z in equation (3.6) with the right hand side of equation (3.8) to obtain the mutability of x,

$$m_x = \frac{0.01}{p_x} \frac{\sum_{l \neq x} A_{xl}}{\sum_h \sum_{l \neq h} A_{hl}}.$$

Substituting the expression for m_x into the right hand side of equation (3.4), we obtain the PAM1 transition probability

$$P_{xy}^{(1)} = \frac{0.01}{p_j} \frac{A_{xy}}{\sum_h \sum_{l \neq h} A_{hl}}.$$

Note that $P_{xy}^{(1)}$ in equation (3.4) is consistent with the definition of a Markov chain: the rows of the transition matrix sum to 1 and it is history independent. This Markov chain is finite, aperiodic and irreducible ("connected"). Therefore, it has a stationary distribution.

We now derive the PAM-2 transition matrix. Note that the residue at site i can change from x to y in two time steps via several state paths: $x \to x \to y$, $x \to y \to y$, or $x \to l \to y$, where l is a third amino acid, not equal to x or y. Recall that the probability of changing from x to y in two time steps is

$$P_{xy}^{(2)} = \sum_l P_{xl}^{(1)} P_{ly}^{(1)}$$

$P^{(2)}$ can be derived by squaring the matrix $P^{(1)}$ by matrix multiplication. This is the transition probability of a second order Markov chain that models amino acid replacements that occur in two time steps. Similarly, we can use matrix multiplication to derive the PAM-N transition matrix for any $N \geq 2$ as follows:

$$P^{(N)} = \left(P^{(1)}\right)^N.$$

4. We obtain a log odds scoring matrix from the transition probability matrix as follows. Let $q_{xy}^{(N)} = p_x P_{xy}^{(N)}$ be the probability that we see amino acid x aligned with amino acid y at a given position in an alignment of sequences with N PAMs of divergence; i.e., that amino acid x has been replaced by amino acid y after N PAMs of mutational change. Then, we define the PAM-N scoring matrix to be

$$S^N[x,y] = \lambda \log \frac{q_{xy}^{(N)}}{p_x p_y} \tag{3.9}$$

$$= \lambda \log \frac{P_{xy}^{(N)}}{p_y}, \tag{3.10}$$

where λ is a constant chosen to scale the matrix to a convenient range. Typically $\lambda = 10$ and the entries of S^n are rounded to the nearest integer. Note that equation (3.10) is a log odds ratio, where $q_{xy}^{(N)}$ is the probability of seeing x and y aligned under the alternate hypothesis that x and y share common ancestry with divergence N and $p_x p_y$ is the probability that x and y are aligned by chance.

It is easy to verify that the PAM-N transition matrix is not symmetric; that is, $P_{xy}^{(N)} \neq P_{yx}^{(N)}$. This makes sense since replacing amino acid x with amino acid y may have different consequences than replacing y with x. In contrast, the substitution matrix *is* symmetric; that is, $S^N[x,y] = S^N[y,x]$. This makes sense because in an alignment, we cannot determine direction of evolution, so we assign the same score when x is aligned with y, and when y is aligned with x.

3.3 BLOSUM Matrices

The BLOSUM (BLOck SUbstitution Matrices) matrices were derived by Steven and Jorja Henikoff in 1992 [1]. They were based on a much larger data set than the PAM matrices, and used conserved local alignments or "blocks," rather than global alignments of very closely related sequences. The *"trusted" alignments* used to construct the BLOSUM matrices consisted of roughly 2000 blocks of conserved regions representing 500+ groups of proteins.

Here, we discuss the procedure for constructing a substitution matrix in the BLOSUM framework from a single aligned block. In reality, the BLOSUM matrices were constructed from many blocks. See Ewens and Grant, Section 6.5.2, for a detailed treatment of the BLOSUM matrices, including a discussion of how pair frequencies from multiple blocks are combined. Their treatment includes a worked example with more than one block. Note that their notation is somewhat different from the notation we use in class.

[1] *Amino acid substitution matrices from protein blocks*, PNAS, 1992 Nov 15;89(22):10915-9

BLOSUM matrix construction uses clustering rather than an explicit evolutionary model, to account for different degrees of sequence divergence. Clustering with different values of N, ranging from 45% to 90%, produces a parameterized set of matrices representing different degrees of sequence divergence. In order to construct a BLOSUM-N matrix, the sequences in each block are first grouped into clusters, such that the percent identity of any pair of sequences from different clusters is less than N. Next, for every pair of clusters, amino acids pairs consisting of one amino acid from each cluster are tabulated. Pairs of amino acids within the same cluster are ignored. Amino acid pair counts are normalized by cluster size so that all clusters contribute equally to the pair statistics.

The clustering step in BLOSUM matrix construction has two purposes: parameterizing evolutionary divergence and accounting for sample bias. First, since only amino acids pairs sampled from two different clusters are tabulated, the data used to construct the matrix consists of amino acid pairs observed in sequences with a particular divergence (i.e., sequences that are less than N identical). Second, to control for sample bias, the contribution of each residue in a cluster is normalized by the number of sequences in that cluster. As a result, each cluster contributes the same amount of information to the estimation of amino acid pair frequencies, even though clusters may contain different numbers of sequences.

The specific procedure for BLOSUM matrix construction is as follows:

Partitioning sequences into clusters with N% identity: The clustering step takes as input a block of k sequences of length L (no gaps) and generates C non-overlapping clusters. The ith cluster, C_i, has k_i sequences of length L, where $k = \sum k_i$. The sequences in the block are partitioned in such a way that every sequence in a cluster is at least N% identical to at least one other sequence in the cluster.

One way to obtain such a clustering is to represent the block as a weighted graph, where the nodes correspond to sequences. The nodes for each pair of sequences are connected by an edge that is weighted by their percent identity. To obtain clusters with an N% identity threshold, all edges with weights lower than N% are removed, resulting in one or more connected components. Each connected component corresponds to a cluster. If N is greater than the greatest edge weight, then each cluster will contain a single sequence. If N is smaller than the lowest edge weight, then all sequences will be in a single cluster. If this happens, it is not possible to construct a BLOSUM matrix for this value of N.

Amino acid pair counts: Following the clustering step, the *observed* frequency of amino acid x aligned with amino acid y is calculated as follows. For each pair of clusters, C_i and C_j, we determine the number of x, y and y, x pairs, where x and y are in the same column, but in different clusters. Let $N_l(C_i, x)$ be the number of times that residue x appears in the l^{th} column of cluster C_i. Then, the total number of pairs in column l involving an x in

one cluster and a y in the other cluster is

$$N_l(C_i, x) \cdot N_l(C_j, y) + N_l(C_i, y) \cdot N_l(C_j, x).$$

However, each of the clusters contributes only one count per column, so we must down weight the number of pairs by the product of the size of the clusters. Suppose clusters C_i and C_j contain k_i and k_j sequences, respectively. Then, the contribution of column l in clusters C_i and C_j to the pair count for x and y is

$$\frac{N_l(C_i, x) \cdot N_l(C_j, y) + N_l(C_i, y) \cdot N_l(C_j, x)}{k_i \cdot k_j}.$$

To obtain the total x, y pair count from this block, we sum over all pairs of clusters and over all columns, yielding

$$A_{xy}^N = \sum_{i=1}^{C} \sum_{j=i+1}^{C} \sum_{l=1}^{L} \frac{N_l(C_i, x) \cdot N_l(C_j, y) + N_l(C_i, y) \cdot N_l(C_j, x)}{k_i \cdot k_j}, \tag{3.11}$$

where $x \neq y$. We use the superscript N to indicate that these are pair counts for a BLOSUM-N matrix, where N is the threshold used in the clustering. When $x = y$, the pairs are only counted in one direction:

$$A_{xx}^N = \sum_{i=1}^{C} \sum_{j>i} \sum_{l=1}^{L} \frac{N_l(C_i, x) \cdot N_l(C_j, x)}{k_i \cdot k_j} \tag{3.12}$$

Estimating substitution frequencies: The frequencies of amino acid pairs are derived from the pair counts by normalizing by the total number of possible pairs; that is, by the product of the number of sites in the block and the number of pairs of clusters:

$$q_{xy}^N = \frac{A_{xy}^N}{L \cdot \binom{C}{2}}.$$

Estimating the expected pair frequencies: The expected frequency of x aligned with y is the product of the *background* probabilities of observing x and y independently. In PAM matrix construction, the background frequency of an amino acid is assumed to be the frequency of that amino acid in typical proteins, for example, as tabulated by Robinson and Robinson[2]. In contrast, in BLOSUM matrix construction, the expected frequencies are estimated from the BLOCK data and adjusted for the current value of N.

[2] *Distribution of glutamine and asparagine residues and their near neighbors in peptides and proteins*, PNAS, 1991 Oct;88:8880-4

In order to get the *expected* frequency of x aligned with y, we first estimate the frequencies of the individual residues in the current block, again using the clusters to correct for sample bias. As above, the counts from each cluster are "discounted" by a factor of $1/k_i$, and then normalized by the total number of elements, $L \cdot C$, to obtain the amino acid background frequency:

$$p_x = \frac{1}{L \cdot C} \sum_{i=1}^{C} \sum_{l=1}^{L} \frac{N_l(C_i, x)}{k_i}.$$

The expected pair frequencies are then obtained from the products of the background frequencies:

$$\begin{aligned} E_{xy} &= p_x p_y + p_y p_x \\ E_{xx} &= p_x^2. \end{aligned}$$

Finally, the BLOSUM-N *log odds scoring matrix* is calculated from the ratios of the observed and expected frequencies:

$$S^N[x, y] = 2 \log_2 \frac{q_{xy}^N}{E_{xy}}.$$

3.4 Comparing PAM and BLOSUM Matrices

We began this endeavor with the goal of deriving substitution matrices that are parameterized by evolutionary divergence. In other words, a given alignment should be scored with a matrix with scores that are appropriate for the evolutionary divergence of the sequences being compared. In addition, these scores should implicitly account for multiple substitutions per site, consistent with the typical evolutionary divergence associated with each matrix in the family. A further goal is that the matrices should reflect the biophysical properties of amino acids. The scores for amino acid pairs with similar biophysical properties (i.e., conservative replacements) should be greater than scores for amino acid pairs with divergent biophysical properties (i.e., non-conservative or radical replacements).

The PAM and BLOSUM matrices were both constructed in an explicit log-odds framework, with entries of the form

$$S^N[x, y] = c \; log_2 \frac{q_{xy}^N}{p_x p_y},$$

where the numerator, q_{xy}^N, is the frequency of the amino acid pair (x, y) in alignments of related sequences with divergence N and the denominator, $p_x p_y$, is the frequency with

	PAM	BLOSUM
Evolutionary model	Explicit evolutionary model	None
Data	Full length MSAs	Conserved blocks
Bias correction	Trees	Clustering
Multiple substitutions	Markov model: $P^N = (P^1)^N$	Implicitly represented in data
Evolutionary distance	Markov model: $P^N = (P^1)^N$	Clustering
Matrices	Transition & log odds scoring matrices	Log odds scoring matrix only
Parameter N	Distance increases with N	Distance decreases with N
Biophysical properties	Derived indirectly from data	Derived indirectly from data

Table 3.1: Properties of the PAM and BLOSUM matrices.

which the pair (x, y) will occur if amino acids are sampled according to their background frequencies. The constant c is a scaling factor chosen for convenience. Multiplying every entry in the matrix by a constant changes the value of the entries in an absolute sense, but does not change the ratio between any two entries of the matrix. As a result, the constant does not change the extent to which one amino acid pair is preferred over another. Scaling a matrix with a constant, c, can be used to obtain scores in a convenient range, *e.g.* between 1 and 20.

Although the PAM and BLOSUM matrices have the general log-odds framework in common, they differ in many aspects of their construction, as summarized in Table 3.1. In both cases, the frequencies of amino acid pairs, q_{xy}^N, were estimated from amino acid pair counts in *"trusted"* alignments, but these trusted alignments are different in nature. In contrast to the PAM alignments, the BLOSUM matrices are based on locally conserved regions (ungapped blocks) in multiple alignments of sequences that were not highly conserved along their entire length. The PAM matrices were constructed from full length alignments of closely related sequences with at least 85% identity. These sequences are assumed to contain no site at which more than one substitution has occurred. The trusted alignments used to construct the BLOSUM matrices consisted of roughly 2000 blocks of conserved regions representing 500+ groups of proteins. In other words, some protein families contribute more than one block.

Both matrix families are parameterized by sequence divergence, but this is achieved using very different methods. The PAM matrices are based on a Markov chain that models amino acid replacement explicitly. The use of a Markov model allowed Dayhoff and her colleagues to address several challenges in matrix construction. A PAM-1 transition matrix is constructed from amino acid pair counts obtained from the trusted alignments. The effect of sample bias on these pair frequencies was mitigated by counting changes

on the branches of maximum parsimony trees. Dayhoff accounted for both evolutionary divergence and multiple substitutions by deriving higher order Markov chains from the PAM-1 transition matrix. With PAM matrices, the divergence parameter increases with evolutionary divergence. A rough equivalence between PAMs and percent identity can be determined through simulations, as shown in Table 3.2.

The BLOSUM matrices have no underlying mathematical model. In BLOSUM matrix construction, clustering is used to address sample bias and to obtain different degrees of divergence. Sequences with at least $N\%$ identity are placed in the same cluster. Amino acid pairs are only counted across clusters, not within clusters. In contrast to the PAM matrices, the BLOSUM divergence parameter *decreases* as evolutionary divergence increases. BLOSUM matrices can also be roughly calibrated by percent identity using empirical methods, providing an approximate mapping between the PAM divergence scale and the BLOSUM divergence scale (Table 3.2).

Sequence identity	PAM	BLOSUM
83%	20	-
-	30	-
63%	60	-
-	70	-
43%	100	90
38%	120	80
30%	160	60
25%	200	50
20%	250	45

Table 3.2: Correspondance between percent identity and the divergence of PAM and BLOSUM matrices.

Neither matrix family explicitly considers biophysical properties. The PAM and BLOSUM matrices are constructed from aligned sequences that are conserved because the amino acids in each column are under selective constraints. Nevertheless, the matrices favor amino acid pairs that share biochemical properties. Inspection of the BLOSUM62 matrix, for example, shows that alignments of residues in the same biochemical group tend to have positive log odds scores. These residues are more likely to be observed together in alignments of related sequences than by chance. Residues from different biochemical groups tend to have negative scores. These residues are less likely to be observed together in related sequences than in chance alignments. A score of zero means that this pair of residues is equally likely in related and chance alignments.