

0.1 PAM: the first major amino acid substitution matrices

The sequence alignments computed for a protein database search are almost always *weighted* alignments, and the choice of the *scoring matrix* (amino acid substitution matrix) used can have a large effect on the search results. It is sometimes suggested that the proper scoring matrix is the most critical technical element in a successful search of protein databases. Ideally, the scores in the matrix should reflect the biological phenomena that the alignment seeks to expose. In the case of sequence divergence due to evolutionary mutations, the numbers in the scoring matrix should *ideally* be derived from empirical observation of ancestral sequences and their present-day descendants. In the case of conserved motifs, or well defined sequence-to-structure correlations, the numbers should be derived from collections of sequences containing those motifs or exhibiting those structural features.

0.1.1 PAM units and PAM matrices

The term “PAM”, which is an acronym for “point accepted mutation” or “percent accepted mutations” has two related uses. First, it is used as a *unit* to measure of the amount of evolutionary divergence (or evolutionary distance) between two amino acid sequences. In that context, one might say that sequence S_1 is 5 PAMs diverged (or 5 PAMs distant) from sequence S_2 . Second, the term “PAM” is used to refer to certain amino acid substitution matrices (scoring matrices) whose scores have a relationship to PAM units. The methodology of PAM units and the first specific PAM matrices were developed by Margaret Dayhoff and co-workers [?, ?].

To discuss PAM units and PAM matrices, it is useful to distinguish the *ideal* PAM objects, which are defined in terms of data that is unavailable, from the *real* PAM objects that are computed using less than ideal data.

0.1.2 PAM units

Definition Ideally, two sequences S_1 and S_2 are defined as being *one PAM unit diverged* if a series of *accepted* point mutations (and no insertions and deletions) has converted S_1 to S_2 with an average of one accepted point-mutation event per one-hundred amino acids.

The term “accepted” here means a mutation that was incorporated into the protein, and passed on to its progeny. Therefore, either the mutation did not change the function of the protein, or the change in the protein was beneficial to the organism (or was at least non-lethal).

The above definition of one PAM unit sounds like “one PAM unit of divergence between S_1 and S_2 implies a one percent sequence difference

between those sequences”. It does not. The reason is that a single position can undergo more than a single mutation, so that (for example) if there have been eight point-mutations in an 800 length amino acid sequence, it could happen that the same position mutates twice. It could even happen that the position mutates back to its original amino acid. Therefore, if we align S_1 and S_2 (recalling that there have been no insertions or deletions), the expected number of positions where an amino acid difference is observed is (somewhat) less than eight.

The difference between the correct definition of a PAM unit, and the (common) mis-statement of its meaning, is slight for sequences which are only *one* PAM unit diverged. But the difference grows as the number of units does. For example, it is not true that two sequences that are 100 PAM units diverged are expected to be different in every position. In fact, even amino acid sequences that have diverged by 200 PAM units are expected to be identical in about 25% of their positions, and sequences that are 250 PAM units diverged can generally be distinguished from a pair of random sequences.

There are a number of criticisms of the entire *concept* of a PAM unit, but ignoring these larger issues, how does one determine the number of PAM units separating two sequences S_i and S_j ? In the ideal case, when one extant sequence has diverged from an ancestral sequence due only to point mutations, one simply counts the number of observed positions where the two sequences differ. Then a simple stochastic formula is applied that relates the expected number of observed differences to the expected number of point mutation events¹.

In practice, there are two problems in implementing this ideal scenario. The first is that all the sequences we know today are from *extant* organisms. We don’t know sequences where one is actually derived from the other. The second problem is that insertions and deletions do occur in protein evolution, so that one sometimes cannot be certain of the correct correspondence between sequence positions. Additional criticisms of the PAM unit concept include the fact that not all positions mutate with equal frequency.

The first problem, the lack of ancestral protein sequences, is handled by

¹This may be additionally complicated by the fact that the mutations occur at the DNA level, but the observations occur at the amino acid level. The degeneracy of the genetic code, plus the fact of “silent mutations”, means that some assumptions must also be made about codon frequencies in order to translate the number of *observed* amino acid changes into the *expected* number of true amino acid mutations. For example, TCT is one of the codons for the amino acid serine, and TAT and AGC are two codons for tyrosine. TCT mutates to TAT with a single (DNA) point mutation, while it takes three point mutations to transform to AGC. So if one observes a change of serine to tyrosine, the question of how many mutations occurred must involve details at the DNA level. There is some controversy about the correct way to connect the DNA level to the amino acid level [?].

0.1. PAM: THE FIRST MAJOR AMINO ACID SUBSTITUTION MATRICES 3

appeal to the molecular clock theory (see Section ??). The two sequences S_i and S_j have each diverged from some common ancestor S_{ij} , and the molecular clock theory implies that the expected PAM distance (number of PAM units of divergence) between S_{ij} and S_i equals the expected PAM distance between S_{ij} and S_j . So one uses half the number of differences in the alignment of S_i to S_j to calculate the PAM distance between S_{ij} and its two derived sequences S_i and S_j . For this to be correct, one needs also to assume that amino acid mutations are reversible, and equally likely in either direction.

The second problem, of insertions and deletions, is more difficult. To determine the proper correspondence between positions in the two sequences, one must unambiguously identify the true historical gaps, or at least identify large intervals in the two sequences where the correspondence is correct. That cannot always be done with certainty, especially when the sequences are diverged by a large number of PAM units.

0.1.3 PAM matrices

PAM matrices are amino acid substitution matrices (scoring matrices) that encode and summarize expected evolutionary change at the amino acid level. Each PAM matrix is designed to be used to compare pairs of sequences that are a specific number of PAM units diverged. For example, the PAM 120 matrix is ideally used to compare two sequences known to be 120 PAM units diverged. For any specific pair of amino acid characters, denoted A_i, A_j , the (i, j) entry in the PAM n matrix *reflects* the frequency that A_i is expected to replace A_j in two sequences that are n PAM units diverged.

The PAM n matrix can be obtained *ideally* (but not practically) as follows. Collect many distinct pairs of homologous sequences that are known to be n PAM units diverged (by point mutations only). Align each pair of sequences, and for each amino acid pair A_i, A_j , count the number of times that A_i aligns opposite A_j , and divide that number by the total number of pairs in all the aligned data. Let $f(i, j)$ denote the resulting frequency. Let f_i and f_j respectively be the frequencies that amino acids A_i and A_j appear in the sequences. That is, f_i is the number of times A_i appears in all the sequences, divided by the total lengths of the sequences. Then, the (i, j) entry for the ideal PAM n matrix is

$$\log \frac{f(i, j)}{f(i)f(j)}.$$

This is another example of a *log-odds* ratio. The reason for dividing $f(i, j)$ by $f(i)f(j)$ is to normalize the true (historical) replacement frequency by the replacement frequency one expects due to chance alone. The reason for the logarithm is because addition is easier than multiplication.

0.1.4 How are PAM matrices actually derived?

The above description of the PAM n matrix is the ideal case. But when insertions and deletions have occurred, the proper correspondence between positions can't be known unless the gaps are deduced correctly. In other words, one needs to align the sequences in a way that correctly recovers the history of their divergence before collecting the needed statistics to construct a PAM matrix. But, a scoring matrix is usually used to obtain the alignment. That creates a circular bind. How did Dayhoff resolve this problem?

Dayhoff took *highly similar* sequences which were each believed to have diverged from a common ancestor by only a *low* number of PAM units. “In order to minimize the occurrence of changes caused by successive accepted mutations at one site, the sequences ... were less than 15% different from one another” [?]. Because the divergence was small, the sequences in each pair were essentially the same length, and the few insertions and deletions that might have occurred were easily spotted and accounted for. So, establishing the correct correspondence between characters in those sequences was not a problem, and accumulation of the statistics followed essentially the ideal case². This solves the problem for a low number of PAM units, but not for a high number.

To construct the PAM matrix for high number of PAM units, Dayhoff (roughly) proceeded as follows. Assume that the data consists of aligned pairs of sequences that are one PAM unit diverged. From that data, calculate the frequency, denoted $M(i, j)$, that amino acid A_i mutates to amino acid A_j in one PAM unit. That is, given that a position contains amino acid A_i (no matter how frequently or infrequently it does) $M(i, j)$ is the frequency that the position mutates to A_j in one PAM unit. Let M denote the 20 by 20 matrix of these frequencies. Then matrix M^n (the result of multiplying M by itself n times) gives the probability that any particular amino acid mutates to another particular one in n PAM units. The (i, j) entry for the PAM n matrix is therefore $\log \frac{f(i)M^n(i, j)}{f(i)f(j)} = \log \frac{M^n(i, j)}{f(j)}$, where $f(i)$ and $f(j)$ are the observed frequencies of amino acids A_i and A_j . This approach assumes that the frequencies of the amino acids remain constant over time, and that the mutational processes causing replacements in an interval of one PAM unit operate the same for longer periods. These assumptions are made for practicality, and are further deviations from the ideal description of PAM matrices.

The values in the PAM matrices are usually rounded to integers, and often people add a constant (between 2 to 8) to each entry in the matrix,

²Actually, there is an additional detail in Dayhoff's method, involving the use of phylogenetic trees. We will discuss that detail in Section ?? in Part IV, after phylogenetic trees have been more fully introduced.

0.1. PAM: THE FIRST MAJOR AMINO ACID SUBSTITUTION MATRICES⁵

other than an amino acid character against a space. This has the effect of muting some of the differences between the entries, and the use of this muted matrix has been observed to be beneficial in obtaining “better” alignments in some contexts.

0.1.5 The use of the PAM matrix

Although the PAM matrices were defined and derived from aligned pairs of sequences containing few mismatches (point mutations) and few insertions and deletions, their major use is in weighted alignment with affine gap weights where there may be many mismatches and indels. The alignment of two sequences that are 250 PAM units diverged will generally not have a large number of matches, even after an optimal insertion of gaps. Moreover, when aligning two sequences, one should ideally use the particular PAM matrix corresponding to their PAM distance (the number of PAM units that the two sequences have diverged), but the PAM distance of the two sequences is not generally known. Indeed, one purpose of computing the alignment is to insert gaps to reveal ungapped segments that can be used to compute a PAM distance. Because of these two problems, one cannot know from “first principles” how effective a PAM matrix will be in finding “biologically informative” alignments.

Yet, it seems that the general experience of practitioners is that PAM matrices are very effective in finding alignments which highlight important biological phenomena. In particular, the PAM 250 matrix is reported to be very effective for aligning sequences used in evolutionary studies, and until fairly recently it had often been regarded as the “canonical” protein scoring matrix³. However the BLOSUM62 matrix is increasingly displacing PAM250 as the default matrix.

³An amusing story about PAM matrices: I attended a workshop that was designed for computer scientists, mathematicians and biologists to develop joint activities in computational biology. It was assumed that most of the non-biologists attending would have little background in computational biology, and most of the talks were introductory. However, one speaker (whose discipline I have forgotten) launched into a rather advanced talk that concerned fine points of PAM matrices. He assumed that everyone knew the basics of PAM matrices and so did not introduce them, or say anything about their function, origin or construction. Finally, about half-way into the talk, after the speaker had mentioned “the PAM matrix” perhaps thirty times, a frustrated listener (who had never previously heard of PAM matrices) raised his hand and said something like “you’ve mentioned the PAM matrix repeatedly, but you’ve never told us what it is. Just *what is* this PAM matrix you are talking about?” The speaker rapidly responded “it’s 250” and went on with his talk without hesitation.

0.2 PROSITE

PROSITE is a “dictionary of sites and patterns in proteins” [?, ?] that is linked to the protein sequence database Swiss-Prot. Some aspects of PROSITE were mentioned earlier in Sections ?? and Exercise ?? of Chapter ?. The goal of the PROSITE developer, Amos Bairoch, is to identify and represent *biologically significant* patterns in protein families (particularly those believed to be important to the function of the protein) that allow new protein sequences to be reliably assigned to the proper family. The emphasis on biologically significant patterns distinguishes PROSITE from other efforts to find good *discriminators* which only seek to reliably identify known family members while excluding known non-members. A PROSITE pattern is represented either as a signature motif (see Section ??) or as a profile (see Section ??) derived from a multiple alignment of the family members. PROSITE currently contains around one thousand patterns. Each one comes with cross references to the entries in Swiss-Prot where the pattern is found, along with known false positive and false negative matches in Swiss-Prot.

The *signature* patterns in PROSITE are written as *regular expressions* (see Section ??). For example, $G-[GN]-[SGA]-G-xR-x-[SGA]-C-x(2)-[IV]$ is a PROSITE pattern. Each capital letter specifies a specific amino acid, capital letters inside brackets indicate that any *one* of the enclosed amino acids is permitted, x indicates any amino acid is permitted, $x(2)$ indicates any *pair* of amino acids is permitted. When searching a new sequence for an occurrence of a PROSITE pattern, one can use regular expression pattern matching methods (Sections ??). If the total length of all the regular expressions is n and the length of the new sequence is m , then this approach takes $\Theta(nm)$ time. However, a PROSITE pattern only recognizes finite-length substrings. Moreover, as noted in Exercise ?? (page ??) of Chapter ?, wildcards cause no problems for methods such as *Shift-And* or *agrep*. Thus, as addressed in Exercise ?? of Chapter ?, PROSITE patterns that don't contain a large number of repetition-range specifiers can be more efficiently handled using *Shift-And* or *agrep* style methods.

PROSITE patterns represented as *profiles* are each derived from a multiple alignment of family members and are used when the family similarity is not sufficient to derive effective motif signatures. Additional ability to recognize new family members is obtained by “extrapolating” the profile, i.e., by using knowledge of amino acid substitutability to give scores for amino acids in certain positions where those amino acids have not been observed. For example, if the amino acid Leucine is seen frequently in a given position of the multiple alignment, but Isoleucine is not, Isoleucine may still be given a large score for that position because Isoleucine and Leucine have similar

chemical properties and are frequently substituted for each other.

0.3 BLOCKS and BLOSUM

BLOCKS is a database of protein motifs that is derived from the PROSITE library, and BLOSUM is an amino acid substitution matrix that is derived from BLOCKS. Both BLOCKS and BLOSUM were developed by Steven and Jorja Henikoff [?, ?, ?].

The BLOCKS database attempts to represent the most highly conserved substrings (motifs) in amino acid sequences of related proteins. At this level of description, that goal applies equally well to the PROSITE library of motifs. However, the PROSITE motifs were collected with particular attention paid to known functions or known structures of the proteins. Each motif in PROSITE is expected to have a known biological meaning. In contrast, the motifs in BLOCKS are based on conserved sequence similarity, even if no function is known for the motifs.

Definition A *block* is a short contiguous interval in a multiple alignment of amino acid sequences.

The BLOCKS database holds roughly three thousand blocks of short, highly conserved sequences derived from roughly 800 groups of related proteins. The system that builds BLOCKS, called PROTOMAT, separately processes each of the groups of related proteins in the PROSITE library. For each PROSITE group, PROTOMAT multiply aligns the sequences in that group and then searches for intervals of up to sixty positions where the aligned amino acids are highly similar in a “critical number” (at least 50%) of the sequences. The details are intentionally left vague and only the underlying ideas are described, since the methods and specific parameters used are highly heuristic and have been obtained through extensive experimentation.

After PROTOMAT finds individual blocks in the multiple alignment, it selects a subset of the blocks to form a “path”, i.e., an ordered set of non-overlapping blocks that occur in a “critical number” of sequences. A path identifies regions of high local similarity in the set of proteins without concern for the length of the gaps between its constituent blocks. Given a score for each individual block (based on its width, the level of similarity, the number of sequences it contains, etc.), one can find a path whose block scores sum to the largest number. This is exactly the *chaining problem* discussed in Section ???. The resulting path might identify an effective set of motifs to characterize that protein group. However, the criteria in [?] for evaluating a path is more complex than the sum of block scores, and the “best path” in BLOCKS is found by using brute force enumeration to examine all possible paths. The BLOCKS database holds the best path found for each PROSITE protein group.

Using BLOCKS to classify new sequences

The blocks (or paths) in the BLOCKS database are used to identify potential new members of the protein group that the block (or the path) is derived from. A *profile* (see Section ??) is created for each block, and a new sequence is aligned to each block profile. Sequences that align well to the profiles from many blocks in a single path are considered highly likely to be members of underlying protein group.

0.4 The BLOSUM substitution matrices

The BLOSUM⁴ amino acid substitution matrices are derived from the BLOCKS database [?] and are currently the main competition for the Dayhoff PAM matrices. The basic BLOSUM amino acid substitution score for a pair of amino acids i, j is roughly computed as follows. Define P as the set of all *pairs* of positions in the blocks contained in BLOCKS, such that the two positions in any pair in P are contained in the same column of the same block. For example, if there are n columns contained in all the blocks in BLOCKS, and each block contains exactly k rows, then P consists of $n \times \binom{k}{2}$ pairs of positions. Now let $n(i, j)$ be the number of pairs in P such that one position of the pair contains amino acid i and the other contains amino acid j . Intuitively, $n(i, j)$ increases if amino acids i and j tend to appear together in the same columns of the blocks, and decrease if they tend not to appear together. To normalize $n(i, j)$, define $f(i)$ and $f(j)$ as the fraction of all positions in BLOCKS occupied by amino acids i and j respectively, and define $e(i, j)$ as $|P|f(i)f(j)$. Next, define $s(i, j)$ as $\log_2 \frac{n(i, j)}{e(i, j)}$. Finally, $s(i, j)$ is multiplied by two and rounded to the nearest integer to form the i, j entry in the basic BLOSUM matrix. Intuitively, $s(i, j)$ is the (log of the) ratio of the number of times that amino acids i and j appear together in the same column, divided by the number of times one would (roughly) expect to see i, j pairs in the same column if the placement of amino acids i and j were random throughout BLOCKS. Like the values in a PAM matrix, a BLOSUM value is an example of a *log-odds* ratio.

An important additional refinement to the basic BLOSUM methodology is the de-emphasis of pairs of rows that are “too” highly similar in BLOCKS. For example, if two rows that appear in a given block are identical, then one is removed in order to reduce the influence of that pair of rows in the resulting BLOSUM scores. Extending this idea, the BLOSUM x matrix (for x generally between 50 and 80) is created after first removing one row from any pair of rows in a block that are more than x percent identical. The

⁴BLOSUM stands for *blocks substitution matrix*.

result is that any two remaining rows in a given block will be less than x percent identical.

The purpose of reducing the influence of closely related sequences is to better capture more distant, yet conserved, sequence motifs from a set of widely diverged sequences. Ideally, the sequences in the set should uniformly sample the full evolutionary range of the species being studied. But often the sequences do not provide a uniform sample, and are clustered into subsets of more highly related sequences. Iteratively removing a row of a block that is "too similar" to some remaining row is an attempt to create a block reflecting a less biased data set. The effectiveness of this approach is a purely empirical question. Current opinion is that the BLOSUM 62 matrix has been found to be particularly effective as a less biased reflection of important amino acid conservation.

How BLOSUM matrices differ from PAM matrices

The scores in Dayhoff's PAM matrices are extrapolated from data obtained from very similar sequences. In fact, as explained earlier, the sequences used were intentionally selected to be highly similar. However, "the most common task involving substitution matrices is the detection of much more distant relationships" [?], and the BLOSUM matrices were developed as a way to explicitly represent those important distant relationships. The belief is that highly conserved sequence segments from otherwise highly diverged protein sequences (of proteins in a given family) lead to substitution scores that more effectively encourage local alignment algorithms to produce alignments highlighting biologically important similarities. The general view is that the BLOSUM matrices have been successful in that goal.