Some brief background on scoring matrices
Eliot Bush

Why scoring matrices?

When we score an alignment we need to generate a score for every possible alignment column we may see. The easiest way to do this is with a simple match-mismatch scoring system. Such a system gives the same score to all mismatches, regardless of which amino acids are involved. For example, in this alignment C-A and G-A would get the same score.

M**C**Q**G**P
M**A**Q**A**P

The match-mismatch system has the virtue of being simple. However for most real applications we use more complicated systems involving scoring matrices. A scoring matrix specifies the score to be given for every possible combination of amino acids (or nucleotides).

Fig. 1. Part of the PAM250 protein scoring matrix (showing 8 of 20 amino acids).

	A	R	N	D	C	Q	E	G
A	2	-2	0	0	-2	0	0	1
R	-2	6	0	-1	-4	1	-1	-3
N	0	0	2	2	-4	1	1	0
D	0	-1	2	4	-5	2	3	1
C	-2	-4	-4	-5	12	-5	-5	-3
Q	0	1	1	2	-5	4	2	-1
E	0	-1	1	3	-5	2	4	0
G	1	-3	0	1	-3	-1	0	5

The scoring matrix approach is more complicated, but is worth using because it allows us to make substantially better alignments. Why is this?

The goal of making an alignment is to identify which amino-acids (or nucleotides) are homologous with each other. Scoring matrices are useful because certain substitutions are more likely than others. In proteins some amino acids are more similar to each other chemically (e.g. alanine and glycine which both have non-polar side chains). For this reason they are more likely to substitute for each other than amino acids with different properties (e.g. alanine and cysteine, an amino acid with a polar side chain). This difference is the result of natural selection. Because alanine and glycine are similar chemically, they can play relatively similar roles in a protein. Thus when one substitutes for the other, this will have a less deleterious effect on the protein's three dimensional structure and function. On the other hand when a very different amino acid substitutes, such as cysteine for alanine, this can have a large negative impact on protein function. Substitutions with a large negative impact tend to be selected against, and as a result, such substitutions are less common in evolution.

Having information about which amino acids are more or less likely to substitute for each other is useful in constructing an alignment because it helps us identify which residues are homologous. Take the amino acids mentioned above as an example. If we are looking at an alignment and we see an alanine and a cysteine paired, their relative lack of similarity makes us like the hypothesis of homology a little bit less than if we saw an alanine and a glycine paired. The score we give the alignment corresponds to how attractive the hypothesis of homology is given the data. So when we see an alanine and a cysteine, we should give this a more negative score than if we see an alanine and glycine. Scoring matrices are designed to reflect this principle.

Making scoring matrices

It would be nice if we could construct scoring matrices from first principles, based on our knowledge of the chemistry of different amino acids. Unfortunately it turns out that our knowledge of amino acid and protein chemistry is insufficient to do this. As a result, all practical amino acid scoring matrices are empirical. That means that they are constructed based on the pattern of substitutions we see in known homologous sequences. The earliest matrices of this type were made by Margaret Dayhoff, one of the grandparents of molecular evolution. In this exercise we will construct this type of a matrix, also known as a PAM matrix.

In the description to follow we'll use the nomenclature of Jones and Pevsner (2004). The process of making a scoring matrix begins with a set of trusted alignments. These come from closely related proteins which differ by about 1% from each other. Being this closely related, it is possible to align them by hand or with a match-mismatch system and to be confident that they are correct. Then in these aligned sequences we can count the number of times different amino acids replace each other. For example let $f(i,j)$ represent the frequency at which we see amino acids i and j aligned with each other in our alignments. (That is the count of times they are aligned divided by the total number of alignment columns). If $f(i)$ and $f(j)$ are the frequency of amino acids i and j in the sequences, then we can define the entry in the PAM1 matrix for amino acids i and j to be the following:

$$\delta(i,j)=\log\frac{f(i,j)}{f(i)*f(j)}/\lambda$$

Here, λ is a scaling constant which is included for computational convenience. And the expression $f(i)*f(j)$ represents the frequency that amino acids i and j would be expected to align by chance. You can think of $\delta(i,j)$ as a log odds. It is the log odds of homology given that we observe a particular alignment column. $f(i,j)$ Is the probability of seeing i and j aligned in truly homologous alignments, and $f(i)*f(j)$ is the probability of seeing them aligned in non-homologous alignments.

This formula tells us how to make a PAM1 matrix. This matrix is suitable for aligning sequences which are very similar to each other. However it turns out that for more distantly related sequences the relative proportions of different kinds of substitutions change. This means that to align more distantly related sequences we need to use different matrices. Such matrices (such as the PAM250 matrix shown above) are constructed in the following way. Let us define $g(i,j)$ as follows:

$$g(i,j)=\frac{f(i,j)}{f(i)}$$

We can look at this as telling us the probability that amino acid i mutates to amino acid j in the amount of time that separates our trusted alignments (one PAM unit). If we matrix multiply **g** against itself n times, then the i,j entry of the resulting matrix will tell us the probability of i mutating to j in n PAM units. An entry in the PAM n matrix would then be the following:

$$\delta(i,j)=\log\frac{g^n_{i,j}}{f(j)}/\lambda$$

To make the PAM250 matrix above, g was multiplied against itself 250 times. The result is a scoring matrix which is suitable for aligning protein sequences which are highly diverged from each other.

References

NC Jones and PA Pevzner. *An introduction to bioinformatics algorithms*. 2004. MIT Press.