Where did the BLOSUM62 alignment score matrix come from?

Sean R Eddy

Many sequence alignment programs use the BLOSUM62 score matrix to score pairs of aligned residues. Where did BLOSUM62 come from?

Back in the good old days, so many things were easier to understand. I once disassembled the engine of my 1972 MG just to see how it worked, but now I won't touch the squirrel's nest of technology that's inside my modern Honda Civic. Likewise, in the early days of sequence comparison, alignment scores were straightforward stuff that anybody could tweak. The first sequence comparisons just assigned −1 per mismatch and −1 per insertion/deletion, and if you didn't like that, you could make up whatever scores you thought gave you better-looking alignments. Those days are gone. Look inside a modern amino acid score matrix, and you'll see a squirrel's nest of 400 numbers. These highly tuned matrices, which go by industrialized acronyms like BLOSUM62 and PAM250, no longer seem to have any user serviceable parts inside. Blame probability theory.

Alignment scores are log-odds scores

What we want to know is whether two sequences are homologous (evolutionarily related) or not, so we want an alignment score that reflects that. Theory says that if you want to compare two hypotheses, a good score is a log-odds score: the logarithm of the ratio of the likelihoods of your two hypotheses. If we assume that each aligned residue pair is statistically independent of the others (biologically dubious, but mathematically convenient), the align-

Sean R. Eddy is at Howard Hughes Medical Institute & Department of Genetics, Washington University School of Medicine, 4444 Forest Park Blvd., Box 8510, Saint Louis, Missouri 63108, USA.
e-mail: eddy@genetics.wustl.edu

ment score is the sum of individual log-odds scores for each aligned residue pair. Those individual scores make up a 20 × 20 score matrix. The equation for calculating a score $s(a,b)$ for aligning two residues a and b is:

$$s(a,b) = \frac{1}{\lambda} \log \frac{p_{ab}}{f_a f_b}$$

The numerator (p_{ab}) is the likelihood of the hypothesis we want to test: that these two residues are correlated because they're

---

> The definition of 'conservative substitution' in a score matrix is purely statistical. It has nothing directly to do with amino acid structure or biochemistry.

---

homologous. Thus, p_{ab} are the target frequencies: the probability that we expect to observe residues a and b aligned in homologous sequence alignments. The denominator ($f_a f_b$) is the likelihood of a null hypothesis: that these two residues are uncorrelated and unrelated, occurring independently. Thus, f_a and f_b are background frequencies: the probabilities that we expect to observe amino acids a and b on average in any protein sequence. λ is a scaling factor. It is usually set to something that lets us round off all the terms in the score matrix to sensible integers.

If we expect to find a and b aligned together in homologous sequences more often than we expect them to occur by

chance ($p_{ab} > f_a f_b$), then the odds ratio is greater than one and the score is positive. Operationally, we say that positive scores mean conservative substitutions, and negative scores indicate nonconservative substitutions. This definition of 'conservative substitution' in a score matrix is purely statistical. It has nothing directly to do with amino acid structure or biochemistry.

This explains some details in BLOSUM62 that may seem counterintuitive at first glance. For instance, tryptophan (W/W) pairs score +11, while leucine (L/L) pairs only score +4; why shouldn't all identitites get the same score? The rarer the amino acid is, the more surprising it would be to see two of them align together by chance. In the homologous alignment data that BLOSUM62 was trained on, leucine/leucine (L/L) pairs were in fact more common than tryptophan/tryptophan (W/W) pairs ($p_{LL} = 0.0371, p_{WW} = 0.0065$), but tryptophan is a much rarer amino acid ($f_L = 0.099$, $f_W = 0.013$). Run those numbers (with BLOSUM62's original $\lambda = 0.347$) and you get +3.8 for L/L and +10.5 for W/W, which were rounded to +4 and +11.

Another example is that BLOSUM62 awards a +1 to an apparently nonconservative alignment of a positively charged glutamic acid, but a seemingly more innocuous alignment of an alanine to a leucine gets penalized −1. A/L pairs are indeed slightly more frequent in homologous alignments than K/E pairs ($p_{AL} = 0.0044$, $p_{KE} = 0.0041$ in the BLOSUM62 training data), but A and L are more common amino acids ($p_A = 0.074$, $p_L = 0.099$, $p_K = 0.058$, $p_E = 0.054$). With $\lambda = 0.347$, this gives a score of −1.47 for A/L (rounded to −1) and 0.76 for K/E (rounded to +1).

Where did *those* numbers come from?

So much for the scores. But we've just pushed the question to a different level. Where did we get the target frequencies p_{ab}?

The target frequencies are the probability we expect to see *a,b* aligned in homologous alignments. Thus, the basic idea is to take lot of known, trusted pairwise alignments similar to what we expect our next alignment to look like, and count the frequency at which each residue pair occurs.

The more information we have about the two sequences we're aligning, the better we'll be able to estimate what their target frequencies should be. For example, if we know that we're aligning the sequences of two integral membrane proteins, our target frequencies would be biased toward hydrophobicity. There are endless ways of slicing sequence alignment databases and estimating new score matrices specialized for certain organisms or certain types of sequences. A cottage industry of bioinformatics toils in this happy realm. For a general purpose matrix like BLOSUM62, though, we can't really use sequence- or species-specific sources of information. One source of information remains crucial: evolutionary distance. The target frequencies depend very strongly on the evolutionary distance between the two sequences. If the two sequences diverged recently, the target frequencies should be peaked on identical residues. The more divergent the relationship we're looking for, the flatter the target frequencies need to be. All modern amino acid score matrices are therefore estimated from frequencies observed in trusted alignment data, using some procedure to make a series of related matrices that are appropriate for different expected divergences.

The procedure that Steve and Jorja Henikoff used to estimate the BLOSUM matrices was straightforward[1]. The Henikoffs took a big database of trusted alignments (their BLOCKS database), and (in effect) only counted pairwise sequence alignments related by less than some threshold percentage identity. A threshold of 62% identity or less resulted in the target frequencies for the BLOSUM62 matrix. An 80% threshold gave the more highly conserved target frequencies of the BLOSUM80 matrix, and a 45% threshold gave the more divergent BLOSUM45 matrix. Empirically, the BLOSUM matrices have performed very well. BLOSUM62 has become a *de facto* standard for many protein alignment programs.

DIY score matrices

We can even make up the values if we state some assumptions, which is especially practical for smaller, simpler 4 × 4 DNA score matrices. Say we want to make a DNA scoring matrix optimized for finding 88% identity alignments. Let's assume that all mismatches are equiprobable, and the composition of both alignments and background sequences is uniform at 25% for each nucleotide. Then, our values are 0.22 for the four identities and 0.01 for each of the 12 types of mismatch, and our background frequencies $f_a, f_b = 0.25$ for all *a,b*.

What's the difference between making up our target frequencies and calculating scores, versus just making up scores?

Plug those into the log-odds equation, and we get (if $\lambda = 1$) +1.26 for a match and −1.83 for a mismatch. Scale up a bit with $\lambda = 0.25$ and round off, and *voilà*, we have a new scoring system of +4/−7.

What's the difference between making up our target frequencies and calculating scores, versus just making up scores? When we make up our p_{ab} values, we're directly describing what we expect homologous alignments to look like (here, simply 88% identity), and the resulting score matrix is optimal for detecting alignments that match our target frequencies. If instead, we make up an arbitrary score matrix, we're blindly looking for a scheme that works well.

Even arbitrary scores imply target alignment frequencies

Remarkably, even if we do make up arbitrary scores, they still imply target frequencies. It's useful to know what these implicit target frequencies are, so we know what sort of alignments the score matrix will optimally detect. The proof that arbitrary scores still imply optimal target frequencies is subtle (an important statistical result from Sam Karlin and Steve Altschul[2,3]), but the arithmetic is straightforward.

Rearrangement of the log-odds equation gives us $p_{ab} = f_a f_b e^{\lambda s_{ab}}$; the problem is the unknown λ. The sum of all the p_{ab} values must be 1, by definition, because they're probabilities. So, set

$$\sum_{a,b} f_a f_b e^{\lambda s_{ab}} = 1$$

and solve for a nonzero λ. Such a λ exists so long as the score matrix has two key properties: it must have at least one positive score, and the expected score for random sequence alignments must be negative. Most score matrices have these properties because the same properties are necessary to make local sequence alignment algorithms like BLAST and Smith/Waterman work[3]. (Both conditions are met by definition for matrices derived as log-odds scores, except for the useless case of $p_{ab} = f_a f_b$ for all *a,b*.)

For instance, both FASTA and WU-BLASTN use an arbitrary +5/−4 scoring system for matches/mismatches in DNA alignments, whereas NCBI BLASTN uses a +1/−2 scoring system. Is there a big difference? Probably hard to tell just from looking at those scores. If you run the calculation, you find that these two scoring systems are almost polar opposites. NCBI BLASTN's +2/−1 system is optimal for detecting homologous DNA alignments that are 95% identical—almost perfect matches. FASTA and WU-BLASTN's +5/−4 system is optimal for detecting homologous DNA alignments that are only 65% identical—at the edge of the 'twilight zone' for gapped alignment methods' ability to recognize homologous DNA alignments.

Note: Supplementary information is available on the Nature Biotechnology website.

1. Henikoff, J.G. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA* **89**, 10915–10919 (1992).
2. Karlin. S. & Altschul, S.F. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc. Natl. Acad. Sci. USA* **87**, 2264–2268 (1990).
3. Altschul, S.F. Amino acid substitution matrices from an information theoretic perspective. *J. Mol. Biol.* **219**, 555–565 (1991).

Further study

You can download an ANSI C program for calculating the implicit target frequencies p_{ab} of a score matrix (see **Supplementary Notes**). The BLOSUM62 score matrix and its background frequencies are included as an example. The code also contains two basic methods of solving for roots of equations like the one for λ: the bisection method, and the Newton/Raphson method.

nature biotechnology

Wondering how some other mathematical technique really works? Send suggestions for future primers to askthegeek@natureny.com.