Scoring conservation through diversity of observed residues

Andrew C.R. Martin

28th October 2022
Version 2 — 29th October 2022

The problem is that we want to know how much diversity (or conservation) there is at a given position in a sequence from a huge set of data where most entries have the native amino acid. A simple conservation score (e.g. from `scorecons`) doesn't work very well because the vast majority of the sequences being considered have the native residue and, therefore, it is always scored as highly conserved. I have already modified my implementation of `scorecons` to allow this count to be capped, or for logs to be taken of the numbers of observations — both of these help considerably.

An alternative is just to look at the observed residues that are present and not consider the counts. I have already calculated the number of mutated residues possible for each native residue assuming a single base change (and also assuming that the native residue might be encoded by any one of the appropriate codons).

In a very simple way we could then calculate a conservation score as:

$$C = 1 - D \tag{1}$$

where D is the observed diversity, calculated as:

$$D = \frac{o}{N_n} \tag{2}$$

where o is the number of unique amino acids observed at a given position (**excluding** the native) and $N_n = |\mathbf{M}_n|$ is the number of unique mutated amino acids in $\mathbf{M}_n$ which is the set of possible mutant residues, starting from native amino acid n (also excluding the native). This would give a value of 1 if all possible amino acids are seen and a value of 0 if no mutations are seen.

However this does not consider similarity between the amino acids. What we need is for conservative mutations to contribute less to the diversity.

We can use normalized PAM scores[1] (where the diagonal of the matrix is equal and maximal and set to 1) to score the similarity.

Consequently we replace Equation 2 by:

$$D = \frac{s_o}{N_n} \tag{3}$$

[1]The method for doing this is described in the Valdar `scorecons` paper, but it is used by my implementation of `scorecons` and is present in the BiopLib data directory as `pet91.mat`, though the diagonal here is 10, so all values would need to be divided by 10.

where s_o is defined as:

$$s_o = \sum_{m \in \mathbf{M}_o} (1 - P(n, m)) \tag{4}$$

where n is the native residue, m is a mutant residue, $\mathbf{M}_o$ is the set of observed mutated residues for n and $P(n, m)$ is the normalized PAM score for a mutation between n and m. Given that $P(n, n)$ is, by definition, 1.0, this means that, if all observed residues were native, then s would be 0.0, D would be 0.0 and C would be 1.0. (In practice, since we only look at the *mutant* residues and ignore the native residue, s would, in any case be 0.0.)

However, Equation 3 needs to be normalized with respect to the native residue since the divisor (N_n) is simply the number of possible residues and doesn't take into account the range of similarities between the native residue and the ones to which it can be mutated (i.e. the minimum normalized PAM score for mutants of this residue).

Consequently we replace Equation 3 by:

$$D = \frac{s_o}{t_p} \tag{5}$$

where t_p is defined as:

$$t_p = N_n \times \max_{m \in \mathbf{M}_n} (1 - P(n, m)) \tag{6}$$

Combining Equations 1 and 5, and writing them out in full:

$$C = 1 - \frac{\sum_{m \in \mathbf{M}_o} (1 - P(n, m))}{N_n \times \max_{m \in \mathbf{M}_n} (1 - P(n, m))} \tag{7}$$

where:

- C is the calculated conservation score
- n is the native mutant amino acid
- m is a mutant amino acid
- $\mathbf{M}_o$ is the set of distinct observed mutant amino acids
- $P(n, m)$ is the normalized PAM matrix score for mutation n to m
- N_n is $|\mathbf{M}_n|$, the number of mutant amino acids possible with native residue n
- $\mathbf{M}_n$ is the set of possible mutant amino acids for native residue n