Similarity-preserving Metrics for Amino-acid Sequences
(*Poster abstract*)

Igor Fischer
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen, Sand 1, 72076 Tübingen, Germany
fischer@informatik.uni-tuebingen.de

Sequence alignments and sequence similarity scores derived from them are the most common tools for comparing amino acid and DNA sequences. Different scoring schemes, from simple +6/-1 to PAM and BLOSUM scoring matrices have been devised for highlighting particular biological or evolutionary properties of the sequences to compare. However, most methods of classical, as well as of non-parametrical statistics, including a number of neural network approaches, rely on another measure of similarity between data: the distance measure. It would be therefore of an advantage, if we could derive a distance between two sequences from their similarity score.

Intuitively, it is clear that these two measures are somehow related: the higher the similarity between sequences, the lower the distance between them should be. But, contrary to similarity score, which can be defined in a fairly arbitrary (although not always meaningful) manner, there are three elementary requirements for a distance measure.

A Distance measure on a set D is a function $d : D \times D \rightarrow$ **R** which is equal to zero iff both arguments are equal, symmetric and satisfies the Cauchy-Schwarz-Bunyakovskii inequality (the triangle inequality). As a consequence, such a function is always positive-semidefinite. Distances are most commonly defined on vector spaces, but are not limited to them. Any space on which a function with above properties is defined is called a metric space.

A number of metrics for strings have been proposed, many of them not really being metrics, for failing on one or more of the requirements. A simple and computationally very effective ”distance” measure for sequences is the feature distance [1]. A feature is a short substring, usually referred to as N-gram, N being its length. The feature distance is then computed as the number of features the two strings differ in. It must be noted that this measure is not a distance, for two different strings can have zero distance. For example, strings, AABA and ABAA contain the same bigrams, so with $N = 2$ the ”distance” between them is zero.

Another very common distance measure for strings is the Levenshtein distance [2]. It measures the minimum effort needed to transform one string into another, using basic edit operations: replacement, insertion and deletion of a symbol. Generally, each of these operations has a cost assigned to it, in which case the distance function is usually referred to as weighted Levenshtein distance. This includes very common cases, where replacements of different symbols appear with different probabilities and are therefore assigned different costs, specific for each symbol-to-symbol transformation, as well as above mentioned case, where insertions and deletions are more expensive than replacements.

Although (weighted) Levenshtein distance for any two strings can be computed directly, in cases where there are already devised scoring schemes - like in computational molecular biology - it is desirable to compute a distance that is consistent with the similarity score of the strings. By consistent distance we mean a function, which assigns lower distance value to more similar strings. This can be achieved by appropriate weighting of edit operation costs.

A simple method for computing “distance” from similarity score for proteins was applied by Agrafiotis [3]. His approach can be summarized as follows: For a set of proteins to compare, compute the similarity score for each pair of them. Scale the similarity to the range [0, 1] and define the distance as $d = 1 - ss$, ss being the scaled similarity score. Beside practical drawbacks, like high storage requirements and non-aplicability in on-line algorithms, the main problem with this measure is that it is generally not metric. If not all the proteins have the same similarity score with themselves, scaling leads to values different than 1 and, consequently, to distances different from zero for identical proteins, thus violating one of the requirements for a metric. And it is not clear that the triangle inequality is satisfied, either.

Setubal and Meidanis [4] propose a more mathematically founded method for computing distance from similarity score and vice versa, albeit applicable only if the similarity score of each symbol with itself is the same for all symbols. Unfortunately, this condition is not satisfied for scoring matrices used in computational molecular biology, like PAM or BLOSUM, where diagonal elements - determining similarity of amino acids with themselves - have different values.

Another method for computing distance from similarity score, relying on analogy with vector spaces and inner product is proposed here. Recall that a distance in a vector space can be computed over the norm, which, in turn, is computed over the inner product:

$$d(\mathbf{a},\mathbf{b}) = \|\mathbf{a}-\mathbf{b}\| = \sqrt{\langle \mathbf{a}-\mathbf{b},\mathbf{a}-\mathbf{b}\rangle} = \sqrt{\langle \mathbf{a},\mathbf{a}\rangle + \langle \mathbf{b},\mathbf{b}\rangle - 2\langle \mathbf{a},\mathbf{b}\rangle} \tag{1}$$

By analogy, the distance for strings s and t can be defined over their similarity score. For analogy with the scalar product, the similarity score is denoted $\langle s|t\rangle$:

$$d(s,t) = (\langle s|s\rangle + \langle t|t\rangle - 2\langle s|t\rangle)^{\frac{1}{n}} \tag{2}$$

Of course, for $n = 2$ the analogy is perfect. Defined this way, this function is metric if the similarity scheme obeys some simple, common-sense, sufficient rules:

1. Similarity of a symbol with itself is always positive.
2. Every symbol is more similar to itself than to any other symbol.
3. The similarity function is symmetrical.
4. Space is less similar to all non-spaces than any other symbol.
5. Similarity score for two spaces equals 0.
6. The triangle inequality is satisfied for all single symbols.

Using $n = 1$, the rule (6) reduces to:

$$\langle\alpha|\alpha\rangle - \langle\alpha|\beta\rangle + \langle\beta|\beta\rangle - \langle\beta|\gamma\rangle \geq \langle\alpha|\alpha\rangle - \langle\alpha|\gamma\rangle \tag{3}$$

which means nothing more than that the sum of drops in similarity (i.e. the price we pay in terms of similarity score) when replacing a symbol α with β, and then replacing β with γ must be at least equal as the price for replacing α directly with γ. This condition is satisfied for BLOSUM62 but not for PAM matrices. Using $n = 2$ (i.e. taking the square root), PAMs can be used, too.

Having a distance measure, algorithms like Sammon mapping, nearest-neighbor classifier, k-means, self-organizing maps and many others can be applied to amino-acid sequences.

References

[1] Teuvo Kohonen, “Median strings,” *Pattern Recognition Letters*, vol. 3, pp. 309–313, 1985.

[2] L. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Soviet Physics–Doklady*, vol. 10, no. 7, pp. 707–710, 1966.

[3] Dimitris K. Agrafiotis, “A new method for analyzing protein sequence relationships based on Sammon maps,” *Protein Science*, vol. 6, no. 2, pp. 287–293, June 1997.

[4] J. C. Setubal and J. Meidanis, *Intorduction to Computational Molecular Biology*, PWS Publishing Company, Boston, 1987.