Substitution-Cost Function

Overview of function

I have developed a substitution-cost function is designed to work with the nucleic acid notation: ACGT, however, could feasibly work with sequences built from any alphabet of letters The function operates on pairs of sequences (as opposed to pairs of letters) with any length greater than 2. All the reasoning below is fictious, by which I mean adheres to some 'made-up' biological concept of DNA sequences and their generation.

The substitution-cost function requires that a value length (>2) is passed and matches are only considered at this length.

Matching rules:

Consecutive matches will be rewarded by a multiplier, such that the longer the sequence of consecutive matches the larger the multiplier.

- The multiplier will start at x1 for a singular match and for each subsequent match encountered will increase by 0.1 (assuming the matches are consecutive)
- For example: The first match is scored normally, the second match is scored and multiplied by 1.1, the third match by 1.2 and so on until we reach the end of the match sequence.

The cost of matches themselves would simply be the cost of the summation of the scores of aligning the individual characters in each sequence.

- For example, suppose length is 3, matching AAG with AAG would score (score(A,A) + score(A,A) + score(A,G))

Mismatching rules:

The cost of a mismatch of sequence l with length n is n^2 when the sequence is of the same letters

- For example: matching GGG against AAA would incur a cost of 9.
- Evidently this would cause large costs for relatively short sequences of similar letters, hence discouraging it and perhaps lending to a gapped sequence being of lower cost.

All other mismatch costs would simply be the cost of the summation of the mismatches of the scores of aligning the individual characters in each sequence

- For example, suppose length is 3, mismatching AAG with CTT would incur a cost (mismatch(A,C) + mismatch(A,T) + mismatch(G,T))

Gaps rules:

A logarithmic gap penalty function will be incorporated. The function will produce a cost of $g(l) = a + b \ln(|l|)$ for an indel of sequence l, where:

- A is the cost to open the gap
- B is the gap extension penalty
- L the length of the gap.

We want the sequence to minimise the number of gaps, so we make the cost of opening a gap (A) large. However, once a gap is opened it is more ideal to have larger gapped sequences, so since extensions to the gap decrease logarithmically a gap will have less impact on the total score as its length increases

- For example, suppose length is 5, a is 3 and b is 2, matching AGGTT against _____ would incur a cost of 5 + 2log(5)