Scoring Function

Jake Mortimer

November 23, 2019

1 Description

My scoring function will have two rules that differentiate it from the simple, single letter cost function:

- The cost of indels starts at the cost of the particular single character score, however, for each additional indel we increase a divisor by some exponential factor based on the number of indels in a sequence. We perform an integer division of the cost of this string of indels by the exponential divisor; this means that for an incredibly long string of indels the cost would be very low. This makes sense biologically because it would mean that some large section of the DNA sequence may have been omitted, and would have a lower cost.

- The scoring function will also implement multi-character substitutions. The allowed substitutions will be determined by scanning the sequences for mismatch combinations (e.g ABC and CBA) of length 3 and recording the number of occurrences. We then sort this list by the number of occurrences and then take the top n mismatches. This suggests that the genome has undergone some adaptation over time as this mismatch occurs numerous times. The score given to these sequences are given by the formula $S(s_1, s_2) = \sum_{n=1}^{lengthofseqs} max(scoreFunc(s_1[n], s_1[n]), scoreFunc(s_2[n], s_2[n]))$ where scoreFunc is the function which dictates the score given to a match of the first argument with the second. This basically means that the score for the mismatched sequence is the sum of the maximum of matches between the character in the first sequence and itself or the character in the second sequence.

2 Examples

1. s_1 =“ABCABCABCBBBBBBBBBBBBBBBB”
   s_2 =“CBACBAABCAAAAAAAAAAAAA”

   Leads to the alignment ⇓

s_1 =“ABCABCABCBBBBBBBBBBBBBBBB————————————————
”

s_2 =“CBACBAABC————————————————————AAAAAAAAAAAAA”

2. s_1 =“AABABAAABAAABB”
   s_2 =“CCCACACCCCCCCCCC”

   Leads to the alignment $\Downarrow$

   s_1 =“AAB——ABAAABA–AAB”
   s_2 =“CCCACA——CCC–CCCC”