Computational Biology
Assignment I - Report

Philip Hartout

October 13, 2019

Theory questions

1. **Why are gaps more penalized than mismatches?** Insertions and deletions happen more frequently than substitutions, and silent mutations have no impact on the functional aspects of a protein, which therefore encourages penalizing mismatches less than gaps.

2. **Why does it make sense to disallow gaps at the start and end of a local alignment, considering the -2 score for a gap?** When making a local alignment, the localization of the alignment and the score are the most important factors. Adding gaps before and after the alignment makes the score and the localization both less relevant and the score lower, therefore it makes sens to disallow gaps at the start and end of the alignment.

3. **What is a potential problem that could arise when trying to locally align a very short sequence to a much longer one?** The overall probability of a match will inexorably increase, and the match can become meaningless as the shorter sequence can match multiple parts of the much longer one (e.g. when there are multiple repeats in the DNA sequence).

4. **There are $\sum_{k=0}^{n} \binom{m+k}{k}\binom{m}{m+k-n}$ possible alignments for two sequences of lengths m and n, $m \geq n$. For sequences of lengths $m = n = 100$, this amounts to 2.05×10^{75} possible alignments. Give a rough estimate — order of magnitude only — for the number of steps needed to align two sequences of this length using the Needleman-Wunsch algorithm. (Hint: consider the maximum number of iterations of the loops in the pseudocode.) For sequences of lengths $m = n = 100$, roughly how many times less than exhaustive search is this?** The computational complexity (number of steps) of the Needleman-Wunsch algorithm is in the order of $O(mn)$. Since the computational complexity of the bruteforce operation is in the order of 10^{75} and that the computational complexity of the Needleman-Wunsh algorithm for two sequences of length 100 is $100 \cdot 100 = 10^4$, there is a roughly 10^{71} less operations to perform.

5. **How could you extend the Needleman-Wunsch algorithm to k sequences, $k > 2$?** One could align sequences to a reference (or consensus) sequence and score each individual alignment. Adding an additional dimension for the score and path matrix for every additional sequence that needs to be compared could also be a way to extend the N-W algorithm for $k > 2$.