# Bioinformatics - Home assignment

Arno Deceuninck

June 10, 2023

## 1 Local and global alignment with n sequences

### 1.1 Creating alignment matrix

When aligning $n$ sequences, the algorithm creates an alignment matrix with $n$ dimensions and a backtrack matrix of the same size. Each of the dimensions correspond to one of the sequences. The length of each dimension is the length of the corresponding sequence. $m$ is the length of the longest sequence.

The aalignment matrix will contain the scores for each position of the matrix, which is the maximum of the scores from the neigbours with some values added to it. The backtrack matrix will contain the neighbours from which a step was taken to the current position to get to given position.

The scores get created based on the other neighbouring scores, for which all indices where either the same or one lower. To fill the matrix, all different positions of the alignment matrix are iterated over ($\mathcal{O}(m^n)$ matrix positions). Since the scores depends on the scores of the previous neigbours, this was done in the order of the sum of the indices, since this assured the score for the required preceeding neighbours was already calculated and could be used for generating the given score.

The score was calculated for each position. For the starting position, the score was set to 0. For all other positions, the score was generated over all preceeding neighbours of which the index of the neighbour was either the same dimension as the current position or one les. There was an iteration over all those preceeding neighbours to calculate the score from that neighbour to the current position ($\mathcal{O}(2^n)$ neighbours, since for each dimension the neigbour is either at the same index or one lower).

The score was calculated for each of those neighbours. The score was given by the score of the neighbour, plus a score based on each pair of sequences. For this, we had to iterate across all different pairs of sequences ($\mathcal{O}(n(n-1) = \mathcal{O}(n^2)$). For each of those pairs, the score was calculated based on whether the position increased in both of those dimensions (match), only one of those dimensions (gap) or both of the dimensions (two gaps) ($\mathcal{O}(1)$). The score corresponding to the action was summed to the total score of given position (on top of the score of the neighbour and the pairwise scores of the other pairs).

The maximum score of all neighbours was taken as score for given position. When doing local alignments instead of global alignment, $max(0, score)$ was taken. The score is then filled in the alignment matrix and the neighbour that caused this score was filled in the backtrack matrix.

The total efficiency for creating the alignment matrix (and backtrack matrix) by iterating over all matrix positions, for each position over the preceeding neighbours and for each neighbour iterating over all pairs and generating a score is $\mathcal{O}(m^n 2^n n^2)$.

### 1.2 Creating alignments

Once the alignment matrix is created, the sequences can be aligned by following a path through the backtrack matrix. For global alignments, this path starts at the position with the highest sum of indices and ends at the position with the lowest sum of indices (all 0). For local alignments, this path starts at the position of the maximum score in the matrix and ends once a score of 0 is seen on the path. This length of this path is $\mathcal{O}(mn)$.

For each position in this path, each sequence either outputs either a character of the corresponding sequence or a gap ($\mathcal{O}(\backslash)$).

This results in an efficiency of $O(mn^2)$, which is lower than the creation of the alignment matrix, so the total efficiency of creating the alignment matrix and aligning the sequences is $\mathcal{O}(m^n 2^n n^2 + mn^2) = \mathcal{O}(m^n 2^n n^2)$.

For the total space complexiy, the alignment and backtrack matrix needs to be stored, all neighbour positions (since we are iterating over them and calculated them beforehand instead of on the fly) each iteration, the pairs are also calculated beforehand and not on the fly, so need to be stored. However, the matrices are larger than the others, so the space complexity is $\mathcal{O}(m^n)$.

## 2 Application on biological problem

When globally aligning the sequences using the default parameters, this alignmnt is obtained:

```
unknown_J_region_1: GYSSASKIIFGSGTRLSIRP
unknown_J_region_2: NTE.AF...FGQGTRLTVV.
unknown_J_region_3: NYG.YT...FGSGTRLTVV.
```

There were two TCB J protein sequences and one TCA J protein sequences. Region 2 and 3 are more similar to each other, so this are the TCB J sequences and region 1 is the TCA J sequence.

When aligning the sequences locally usign the default parameters, except for mismatch, which is set to -4, the following alignment is obtained:

```
unknown_J_region_1: FGSGTRL
unknown_J_region_2: FGQGTRL
unknown_J_region_3: FGSGTRL
```

The conserved regbion in the three chains is FG[SQ]GTRL.