Integrative Transcriptomics
Prof. K. Nieselt,
Institute for Bioinformatics and Medical Informatics Tübingen
Prof. S. Nahnsen,
Institute for Bioinformatics and Medical Informatics Tübingen

Lecture: Grundlagen der Bioinformatik SoSe 2022

Assignment 3 (20 points)

Hand out: Thursday, May 12
Hand in due: Thursday, May 19, 18:00
Direct inquiries via the ILIAS forum or to your respective tutor at:
Mathias Witte Paz: iizwi01@uni-tuebingen.de
theresa-anisja.harbig@uni-tuebingen.de
meret.haeusler@student.uni-tuebingen.de
jules.kreuer@student.uni-tuebingen.de
simon.heumos@qbic.uni-tuebingen.de

Theoretical Assignments

1. **An exact multiple sequence alignment** (2P)

   For the computation of an optimal multiple sequence alignment (MSA) of n sequences, you need an n-dimensional matrix (i.e. an n-dimensional hypercube). A resulting alignment corresponds to a path through the hypercube. For the following MSA of 3 nucleotide sequences

```
Sequence X: AATG
Sequence Y: A-TG
Sequence Z: --TG
```

   the path is: $(0,0,0),(1,1,0),(2,1,0),(3,2,1),(4,3,2)$. Provide the coordinates of the cells (x,y,z) from the 3D-hypercube that were taken into account in the computation of the values of the cells $(3,2,1)$ and $(4,3,2)$.

2. **MSA: How small is small?** (2P)

   Earth started its life from the solar nebula about 4.54 billion $(4.54 \cdot 10^9)$ years ago. Assume one highly intelligent creature had already back then built a supercomputer and had with the birth of the Earth started to multiply align r sequences, each of length $L = 50$. Assume this computer needed 10^{-12} seconds per pairwise alignment, and 10^{-6} seconds for 4 sequences. Using the simplified time complexity of $O(2^r L^r)$ of the dynamic programming algorithm based on the sum of pairs score, the supercomputer has computed an MSA of these sequences. Compute how many sequences r this supercomputer would have been able to align until today.

Practical Assignments

3. Profile alignment (8P)

The idea of this task is to compute a multiple sequence alignment using the *pair-guided alignment* approach as explained in the lecture (Chap. 5, Slide 43 or page 62 of the script). The four nucleotide sequences to align are found in the file *to_msa.fasta*. To solve this task, solve the following points:

(a) Extend your program from the previous assignment in a way such that it can read four sequences from the input FASTA file.

(b) Next, compute for each sequence combination a pairwise alignment. If you were not able to complete the task of the previous assignment, you may use the Needleman-Wunsch provided as a template[1]. **Note**: You need to modify the Needleman-Wunsch function provided in the template in order to also return the optimal alignment score and to match the scores as defined in the lecture.

Take the sequence pair that has the maximal alignment score, call this A^*_{max}. We then call the alignment of the remaining two sequences A^*_{rest}.

(c) From the four possible combinations $A(i, j)$ with $i \in A^*_{max}$ and $j \in A^*_{rest}$, take those sequences that have the maximal alignment score to get the aligned pair A_{cross}, i.e. $A_{cross} = A(i, j)$ with $\arg\max_{i,j} S(i, j)$ and $i \in A_{max}, j \in A_{rest}$.

(d) Use the alignment A_{cross} to combine the profiles A^*_{max} and A^*_{rest} by inserting gaps in the corresponding positions.

(e) Print the final MSA into the console.

Apply your program to the sequences in the file *to_msa.fasta* with the following parameters:

$$s(a, b) = -2 \text{ if } a \neq b \text{ and } s(a, a) = +3 \text{ and } d = 4$$

[1] Template taken from: `https://gist.github.com/slowkow/06c6dba9180d013dfd82bec217d22eb5`

4. **Distance matrix calculation using Feng-Doolittle distances for MSA** (8P)

In this task we ask you to implement the `Feng-Doolittle distance` function as defined in eq. (5.2) of the lecture notes and to compute a distance matrix for r sequences. For this:

(a) Compute $S_{obs}(X, Y)$. You may use the given template or your implementation of the last assignment to compute the optimal global alignment score of two sequences X, Y.

(b) Compute $S_{id}(X, Y)$, the average alignment score of the two sequences aligned with itself. Again, either use your own NW-code or use the template code for this step.

(c) Implement a function that reads in two aligned sequences, the scoring values and the linear gap penalty d. From the alignment compute the number of gaps, $N(g)$. Then compute the random score $S_{\texttt{rand}}(a, b)$ according to the formula:

$$S_{\texttt{rand}}(X, Y) = \frac{1}{L} \sum_{a \in X} \sum_{b \in Y} s(a, b) N_X(a) N_Y(b) - N(g)d,$$

where L = length of the aligned sequences, $N_X(*), N_Y(*)$ = number of times residue * appears in sequence X, Y, $s(a, b)$ = score given by the scoring values, $N(g)$ = number of gaps in the optimal alignment of X and Y, d = gap penalty.

(d) Compute the distance matrix $D = d[i, j]$, where $d[i, j]$ = Feng-Doolittle distance for sequence i, j for all $\binom{r}{2}$ combinations of the sequences of the file *to_msa_feng_doolittle.fasta*. Consider the following scoring parameters:

$$s(a, b) = -2 \text{ if } a \neq b \text{ and } s(a, a) = +3 \text{ and } d = 4$$

(e) Print the distance matrix in the console and export it to a file.

Note that for both practical tasks, you can also hand-in code for only some parts of the task, in order to achieve some points. However, make sure that these parts can be executed.

Please read the questions carefully. If there are any questions, you may ask them during the tutorial session or in the forum of ILIAS. You will usually get an answer in time, but late e-mails (e.g. the evening of the hand-in) might not be answered in time. Please upload all your solutions to ILIAS. Don't forget to put your names on every sheet **and** in your source code files. Please pack both your source code as well as the theoretical part into one single archive file and give it a name using this scheme: `<name1>_<name2>_<Assignment>_<#>.zip`. The program should run without any modification needed.