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

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Lecture: Grundlagen der Bioinformatik — SoSe 2023

Assignment 3 (20 points)

Hand out: Thursday, May 11
Hand in due: Thursday, May 18, 18:00
Direct inquiries via the ILIAS forum or to your respective tutor at:
caroline.jachmann@uni-tuebingen.de
meret.haeusler@student.uni-tuebingen.de
simon.hackl@uni-tuebingen.de
carolin.schwitalla@qbic.uni-tuebingen.de

Theoretical Assignments

1. **BLAST - theoretical considerations** (2P)

   For very small query lengths (n), BLAST adjust the used word size w to $w < \frac{1}{2}n$. Justify (briefly) why this word size limit is useful for finding reliable matches.

2. **Sum of pairs score - theoretical considerations** (4P)

   Consider a position/column i in an sum of pairs (SP)-optimal multi-alignment $\mathbf{A}^*$ that is *constant*, i.e., that has the same residue in all r sequences.
   What happens when we add a new sequence? If the number r of aligned sequences is small, then we would not be too surprised if the new sequence shows a different residue at the previously constant position i. However, if the number of sequences is large, then we would expect the constant position i to remain constant, if possible.

   Prove that, unfortunately, the SP score favors the opposite behavior: The more sequences there are in an MSA, the easier it is, relatively speaking, for a differing residue to be placed in an otherwise constant column.

   Your task is to set up an equation that

   (a) computes the difference of the SP score of a conserved column and the SP score of a column where all but one residue are equal to the residues of the conserved column. Consider the number of sequences as the variable of interest.

   (b) computes the ratio of the difference of (a) (with respect to the score of the constant column).

   Discuss the behavior of this equation.

Practical Assignments

3. **Analysis of ancient DNA sequences using BLAST and ClustalOmega** (6P)

In 1979, a well-preserved specimen of the Siberian woolly mammoth (*Mammuthus primigenius*) was recovered from the Siberian permafrost. Its gene for mitochondrial cytochrome b could be sequenced, we provided you with the respective translated DNA sequence material/-CYB_MAMPR.fasta in the assignment material. Using `BLAST` and a multiple sequence alignment we want to answer the question how much the cytochrome b sequence changed since the extinction of this species and how similar is it to that from other extant species?

(a) Use the `BLAST` webserver of `UniProt` (`https://www.uniprot.org/blast/` to compare the protein sequence (as translated from the ancient DNA sequences) to the modern protein database. BLAST with the *Mammuthus sequence*, choose in the 'Restrict by taxonomy' Afrotheria and 0.0001 as E-Threshold.

(b) Select all hits based on the `BLAST` results and conduct a multiple sequence alignment that is also offered by `UniProt`. On the Results Summary page, study the Percent Identity Matrix to determine the pairwise similarity of the sequences. Which organisms are most closely related with the ancient mammoth?

(c) Can you confirm the result of (b) with your `BLAST` results? Do the pairwise `BLAST` alignments and the MSA from `ClustalOmega` differ?

(d) Compare your results with the paper *Molecular Phylogenetic Inference of the Woolly Mammoth Mammuthus primigenius, Based on Complete Sequences of Mitochondrial Cytochrome b and 12S Ribosomal RNA Genes* (included in the material folder): Can you confirm the findings of the paper?

4. **Pair-guided 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 29 or page 69 of the script). The four nucleotide sequences to align are found in the file resources/to_msa.fasta, located in the assignment material. To solve this task, solve the following points:

(a) Implement an adjusted version of the Needleman-Wunsch algorithm that aligns two sequences that can contain gap symbols. A match between two gap symbols should be scored with 0. The algorithm should return the alignment score as well as the aligned sequences. You can use your Smith-Waterman implementation from the last assignment as a starting point. If you do not manage to solve this task you can include the provided lib/Needleman-Wunsch.jar into your program.

(b) Extend your program from the previous assignment in a way such that it can read four sequences from the input FASTA file. Catch cases in which not exactly four sequences are provided with a reasonable notification.

(c) Next, compute for each sequence combination (of the four parsed input sequences) a pairwise alignment (in total 6). The sequence pair that has the maximal alignment score is considered as A^*_{max}. The alignment of the remaining two sequences is called A^*_{rest}.

(d) Pick one aligned sequence of the alignment A^*_{max} and one aligned sequence of the alignment A^*_{rest}. Align these two sequences using your adapted *Needleman-Wunsch* algorithm. Output the alignment score and aligned sequences into the console.

(e′) Complete the MSA of all four sequences by inserting the new gaps of your alignment from task (c) into the other sequences of A^*_{max} and A^*_{rest}. Print the final MSA into the console.

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

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

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, a JAR file with your compiled 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.