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 2022

Assignment 5 (20 points)

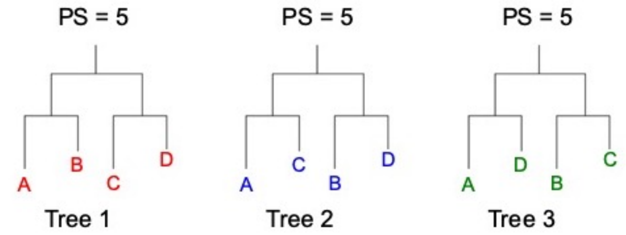
Hand out: Thursday, May 26
Hand in due: Thursday, June 2, 18:00
Direct inquiries via the ILIAS forum or to your respective tutor at:
Mathias Witte Paz: iizwi01@uni-tuebingen.de
theresa-anisja.harbig@uni-tuebingen.de
meret.haeusler@student.uni-tuebingen.de
jules.kreuer@student.uni-tuebingen.de
simon.heumos@qbic.uni-tuebingen.de

Reminder: If have not done it yet, you should hand-in also Task 3 from Assignment 4 (UPGMA and NJ by hand).

Theoretical Assignments

1. **Parsimony score and MSA** (4P)

   For the following three phylogenetic trees on 4 taxa A, B, C, D with respective parsimony scores, set up a minimal multiple sequence alignment (with respect to the number of columns) $\mathbf{A}^*$ for each tree that result in the respective parsimony score $PS(T, \mathbf{A}^*)$:



2. **Step-wise addition heuristic to compute a maximum parsimony tree** (6P)

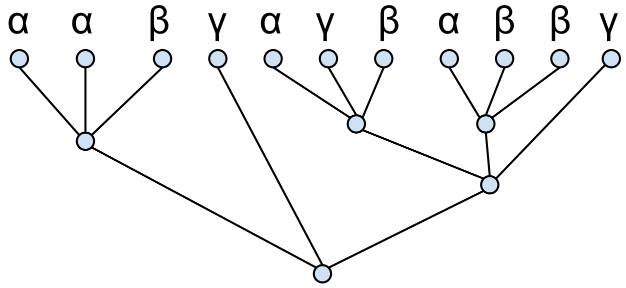
   Apply the step-wise addition heuristic as introduced in the lecture / script (p. 105-106) to the following MSA on 5 taxa:

$$\begin{array}{llll} a_1: & T & T & C \\ a_2: & C & G & C \\ a_3: & C & A & C \\ a_4: & T & C & C \\ a_5: & G & T & C \end{array}$$

   You should start with the first three sequences, then include a_4 and finally a_5. Provide all intermediate steps. Also report the final tree and its parsimony score.

3. **Fitch's algorithm adaptation to ternary trees** (4P)

   Generalize Fitch's "forward-pass" parsimony algorithm to ternary phylogenetic trees (i.e., trees whose internal nodes have degree 4). In particular pay attention to the formula that assigns sets of letters to internal vertices. Apply it to compute the parsimony score for the following tree with 11 taxa and an alignment consisting of one column (only) and character states = $\{\alpha, \beta, \gamma\}$.



Practical Assignments

To solve the following task, please download the file `material_A5.zip` from Ilias.

4. **Cophenetic correlation coefficient** (6P)

   After a phylogenetic tree has been computed for a given distance matrix using a (distance) method of choice, one often would like to compute how well the reconstructed tree reflects the input data. One possibility is the so-called cophenetic correlation coefficient (CCC). The CCC takes two distance matrices as input, one is the original distance matrix and one is the derived distance matrix from a computed tree (called patristic or cophenetic distances, see p. 97 of the lecture notes). The CCC is then computed as

$$c(D,T) = \frac{\sum_i \sum_j (D_{ij} - \overline{D})(T_{ij} - \overline{T})}{\sqrt{\sum_i \sum_j (D_{ij} - \overline{D})^2 \sum_i \sum_j (T_{ij} - \overline{T})^2}}$$

   where

   - D_{ij} are the input distances between objects i, j in D.
   - T_{ij} are the cophenetic (patristic) distances between leaves i, j in T.
   - $\overline{D}$ and $\overline{T}$ are the average distances of D and T, respectively.

   A CCC = 1 states that the computed tree perfectly reflects the input distances. A value close to 1 is a very good solution, while a value close to 0 reflects a random solution.

   (a) Implement your own method that computes the CCC of two distance matrices. You are not allowed to used any library that provides a direct computation of the CCC.

   (b) Read the original matrix 'distances_original.dist', as well as the two derived distance matrices 'distances_tree1.dist' and 'distances_tree2.dist'.

   (c) Apply your method to evaluate the computation of the derived distance matrices 'distances_tree1.dist' and 'distances_tree2.dist' with respect to the original matrix 'distances_original.dist'. Print the CCC value for each comparison to console with the name of the matrices.

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.