Grundlagen der Bioinformatik

SoSe 2022

Tutor: Theresa/ Mathias

1	2	3	4	$\sum$

Marina Dittschar & Clarissa Auckenthaler

Blatt 5

(Abgabe am 02.06.2022)

Theoretical Assignments

Task 1: Parsimony score and MSA (4)

We used the following MSA for the 4 taxa:

$$A \longrightarrow a_1 = GA$$
$$B \longrightarrow a_2 = AT$$
$$C \longrightarrow a_3 = CC$$
$$D \longrightarrow a_4 = TC$$

We inserted the sequences into the three given trees and replaced the taxa { A, B, C, D} there with each sequence. Then we calculated the parsimony score for each configuration with four sequences to proof the fit of our selected MSA (Figure 1).
As we can see in the Figure 1 all given rooted trees have the same parsimony score if we replace the taxa with the sequences from above.

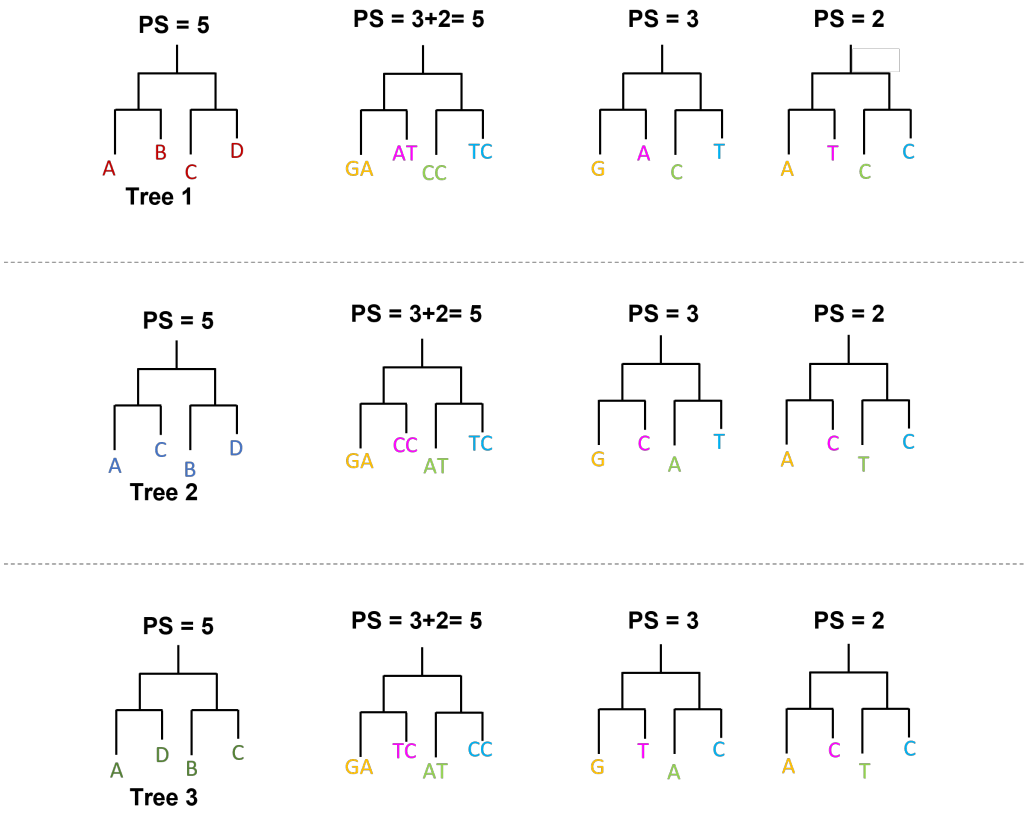

Figure 1: The leftmost trees display the given rooted trees with taxa A,B,C and D. The second tree in each line displays the leftmost tree structure, but with each sequence inserted into the taxa. In the third and fourth tree on each line, the sequences are then split up into characters. You can see the parsimony score for every tree above it.

Task 2: Step-wise addition heuristic to compute a maximum parsimony tree (6)

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

$$a_1 : TTC$$
$$a_2 : CGC$$
$$a_3 : CAC$$
$$a_4 : TCC$$
$$a_5 : GTC$$

We start with the tree of the first three sequences of which there is only one figuration (Figure 2).

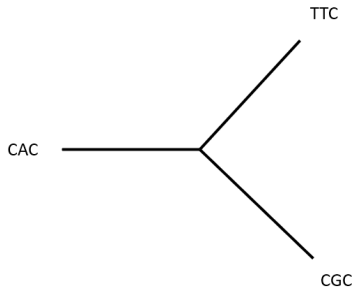

Figure 2: Only possible configuration for a three-leafed tree

We then start to examine the parsimony scores for the different configurations that occur by adding the fourth sequence ("TCC") to the base tree. The obtained parsimony scores were 4 for the best configuration (see Figure 3) and 5 for the other two configurations (see Figure 4 and 5). Therefore, we chose the tree with the lowest parsimony score, which was the first tree. We then took this tree and added the sequence ("GTC") in each possible configuration to it. This yielded a parsimony score of 5 for every option, always having a parsimony score of +2 on one sub-tree and a parsimony score of +3 on another, while the third tree always had a parsimony score of 0 (see Figures 6, 7, 8 and 9). The optimal tree is therefore an arbitrary choice between all options with parsimony score 5. We chose the first tree as our solution (see Figure 10).

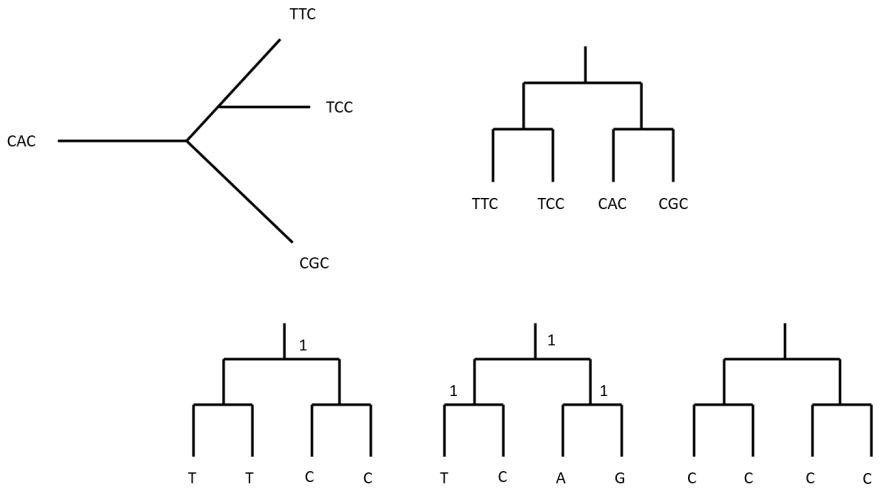

Figure 3: First possible tree configuration with four sequences. Top: Whole Tree, left unrooted, right rooted. Bottom: Sequences split into characters and annotated locations where the parsimony score increases by one

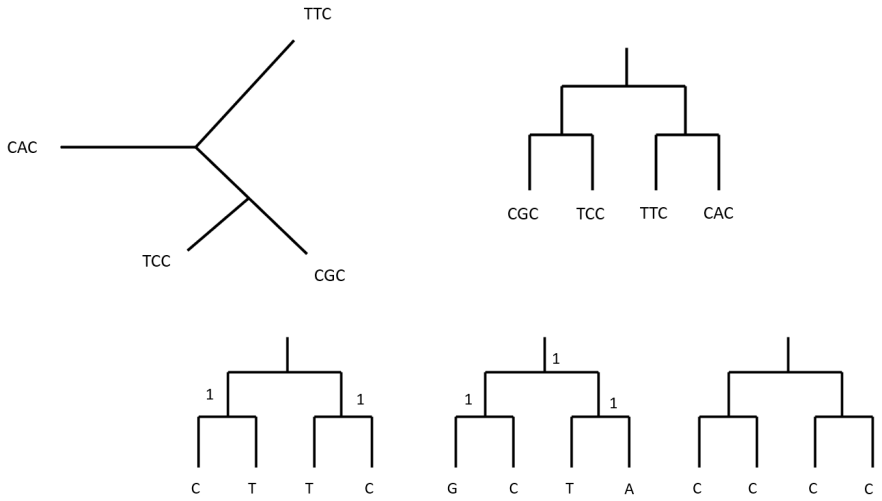

Figure 4: Second possible tree configuration with four sequences. Top: Whole Tree, left unrooted, right rooted. Bottom: Sequences split into characters and annotated locations where the parsimony score increases by one

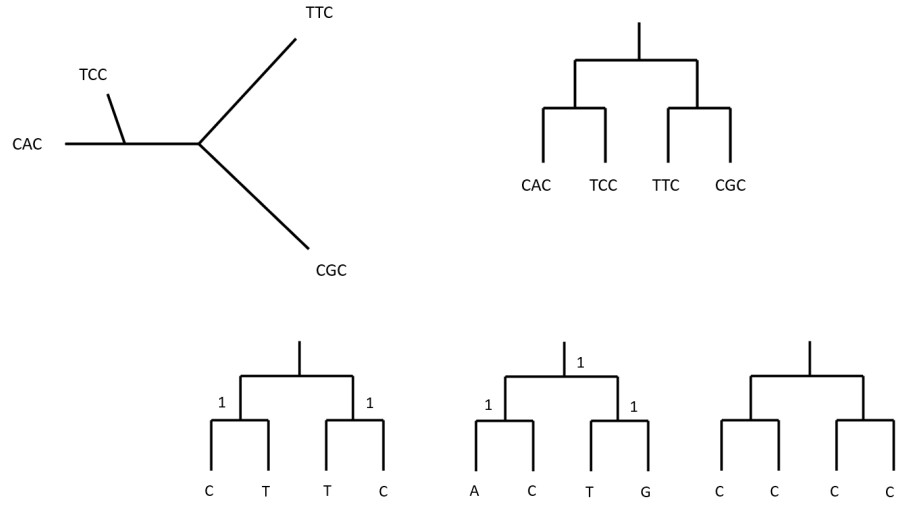

Figure 5: Third possible tree configuration with four sequences. Top: Whole Tree, left unrooted, right rooted. Bottom: Sequences split into characters and annotated locations where the parsimony score increases by one

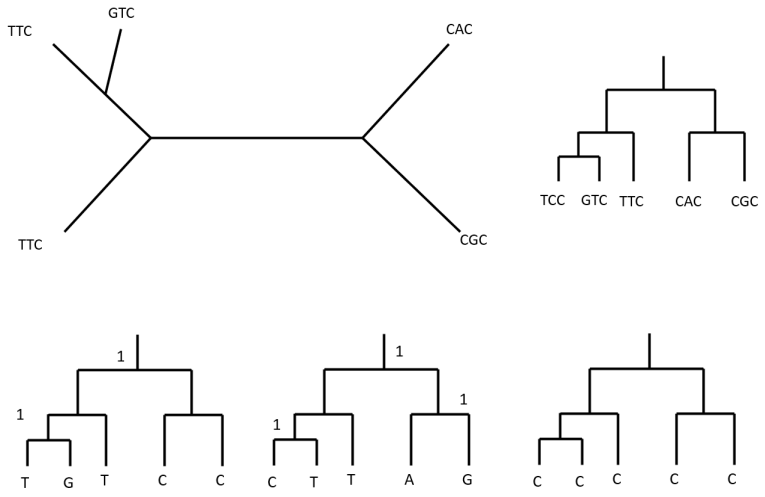

Figure 6: First possible tree configuration with five sequences. Top: Whole Tree, left unrooted, right rooted. Bottom: Sequences split into characters and annotated locations where the parsimony score increases by one

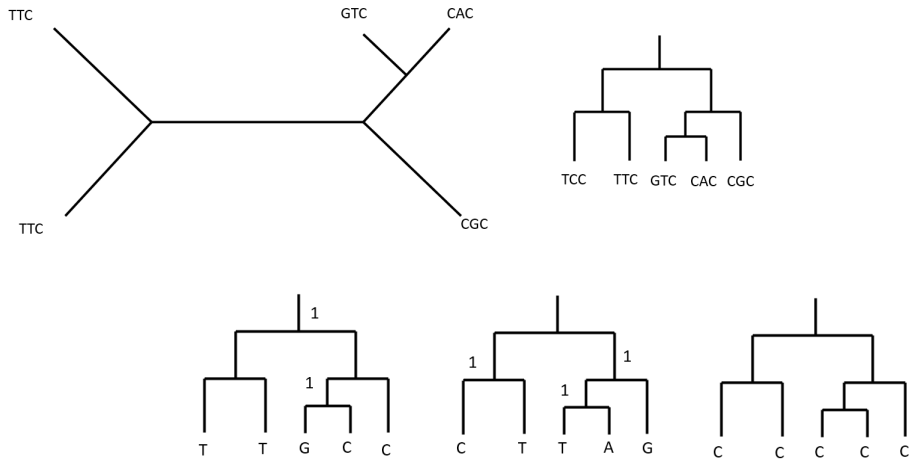

Figure 7: Second possible tree configuration with five sequences. Top: Whole Tree, left unrooted, right rooted. Bottom: Sequences split into characters and annotated locations where the parsimony score increases by one

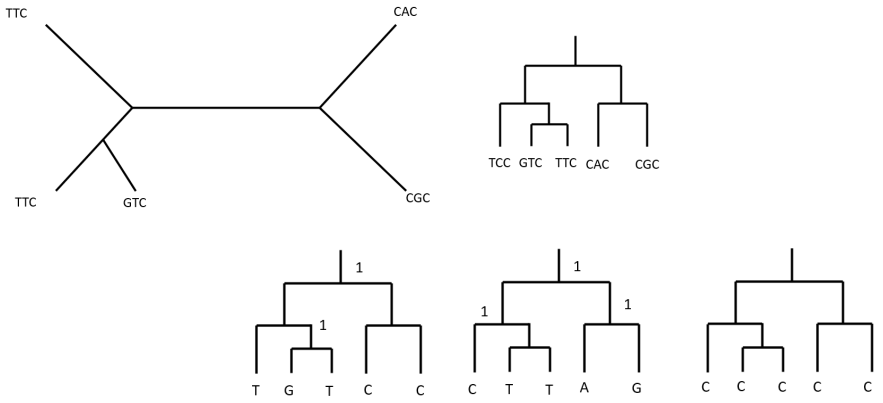

Figure 8: Third possible tree configuration with five sequences. Top: Whole Tree, left unrooted, right rooted. Bottom: Sequences split into characters and annotated locations where the parsimony score increases by one

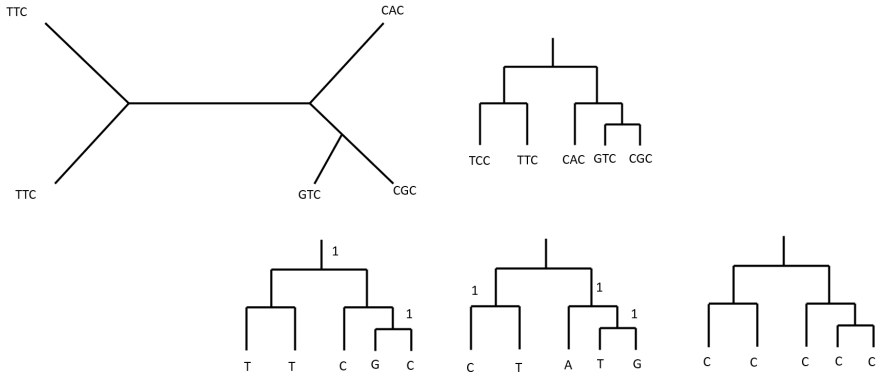

Figure 9: Fourth tree configuration with five sequences. Top: Whole Tree, left unrooted, right rooted. Bottom: Sequences split into characters and annotated locations where the parsimony score increases by one

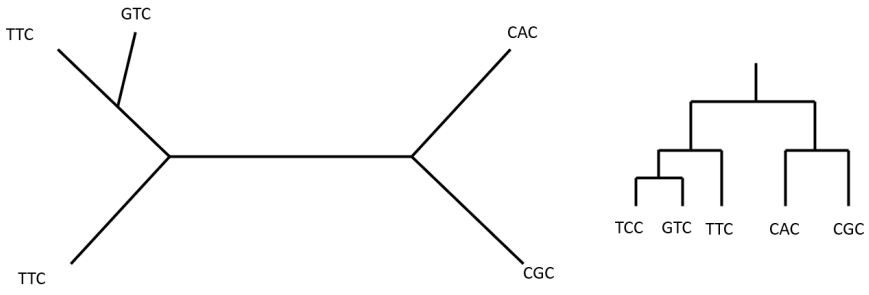

Figure 10: One solution for the optimal final tree with all five sequences. Left: Unrooted. Right: Rooted

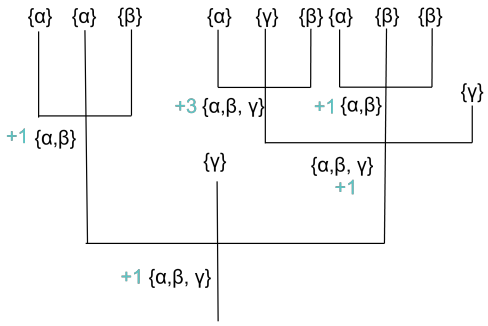

Figure 11: Fitch's "forward-pass" parsimony algorithm to ternary phylogenetic trees, in this case with 11 taxa

.

Task 3: Fitch's algorithm adaptation to ternary trees (4)

We started to solve this exercise by first evaluating the Parsimony score using the "backward-pass" of the Fitch algorithm. As you can see laid out in Figure 12, the Parsimony score had to add up to 7. We then evaluated the different nodes. As a letter assignment function for internal nodes, we took the union of all three child-nodes if the child-nodes did not have a common element. As the node with the leaves $\{\alpha\}\{\alpha\}\{\beta\}$ and $\{\alpha\}\{\beta\}\{\beta\}$ had the same proportion of equivalent and different elements between sets, they had to have the same score. Since the root node and the third child node of the root node had the same input sets ($\{\alpha\beta\gamma\}\{\alpha\beta\}\{\gamma\}$), they also had to have the same score. We thought that a node with three different letters in three different sets with no common elements would have a higher parsimony score than a node where two child-nodes would have the same element, because arguably more evolutionary events need to happen in order to produce these child-nodes. We therefore designed a formula that would assign as a parsimony score the number of nodes *minus* the highest number of sets that have an intersection over any letter. Therefore, a parsimony score of +3 was assigned if all leaves were different (because there is no intersection between child-nodes). If among all three child-nodes there is one combination of child-nodes that has no intersection, a parsimony score of +1 is assigned because the maximum number of overlapping sets over any letter is two. In the two nodes where the three child-nodes are $\{\alpha\beta\gamma\}\{\alpha\beta\}\{\gamma\}$, there is one relationship between the sets without overlap ($\{\alpha\beta\}\{\gamma\}$), therefore the maximum number of overlapping sets over any element is 2, and the parsimony score for this node is +1. You can see the end-result for this tree with these rules in Figure 11.

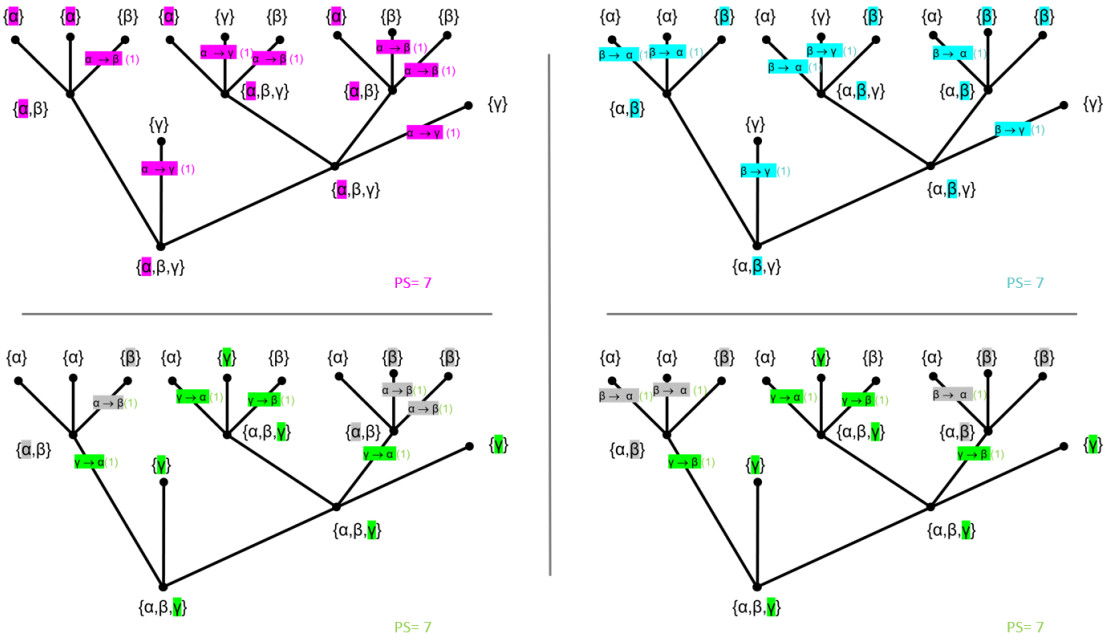

Figure 12: Proof of Fitch "backward-pass" getting the same PS-scores for each starting letter $\{\alpha, \beta, \gamma\}$ Upper left starts with α, upper right with β the both lower left start with γ.

The pseudocode is as follows:

Input: A ternary phylogenetic tree T, a state c(w) for each leaf (v)) w of T

Output: The parsimony score P S(T, c) for T and c

Set P S(T, c) = 0

***Initialisation**: for all leaf nodes v ϵ T set F(v) = c(v)*

for each node** v ϵ T ≠ leaf, in bottom-up order **do

*for the three children w1, w2 and w3 of v **do***

if** F(w1) ∩ F(w2) ∩ F(w3) ≠ ∅ **then

Set F(v) = F(w1) ∩ F(w2) ∩ F(w3)

PS(T,c) = PS(T,c)

else

Set F(v) = F(w1) ∪ F(w2) ∪ F(w2)

PS(T,c) = PS(T,c) + sum(w1 + w2 + w3) - max(sum(w) that share any c(w))

***return** score PS(T,c)*

Practical Assignments

Task 4: Cophenetic correlation coefficient (6)

For Task 4 we used Python 3.8.8, and the following libraries: **getopt, sys, csv, pandas, numpy**

Enter the following code in the command line to run our code:

```
python dittschar_auckenthaler_assignment5.py -a "distances_original.dist" -b "distances_tree1.dist" -c "distances_tree2.dist"
```

If you want to change the files you can do this, by changing the arguments in the command line:

- -a original_distance: (input file)
- -b first_tree_distance: (input file)
- -c second_tree_distance: (input file)