AiB Project 4

Alejandro Roca and Emil Maag

November 8, 2018

Introduction

This project is about comparing evolutionary trees constructed using the Neighbor Joining (NJ) methods, QuickNJ and RapidNJ on different datasets. We implemented an algorithm, based on Days algorithm, for computing the RF distance between two unrooted evolutionary trees over the same set of species and use this implementation in different experiments explained in further detail in the section Experiments.

Methods

The algortihm follows the general design of Days algorithm and once we identified the shared splits and the unique splits, we calculated the rf-distance the following way:

rf-dist = total numbers of intervals - 2 * number of shared intervals + the splits identified as non-intervals.

The following example shows how to use the program rfdist from the command line:

```
rfdist.py tree1 tree2
```

The followng list describes the arguments in more detail:

Arguments:

- tree1: the tree should be in newick format
- tree2: the tree should be in newick format

Evaluation test case for the program rfdist.

The correctness of the algorithm we implemented were tested by running the algorithm with to test trees from the folder testdata, which where provided as a part of the project description. The correst answer for the RF-distance between these to trees should be 8. This was indeed the same result we found when running the algorithm which indicat that we inplemented the algorithm correctly.

References to alignments and trees

References to the 8 alignments (in Stockholm-format) and trees (in Newick-format) that we have produced in Experiment 1 and 2:

- Alignments (https://github.com/alekssro/AlgBioProjects/tree/master/Project04/Data/Alignments/)
- Trees for Experiment 1 and 2 (https://github.com/alekssro/AlgBioProjects/tree/master/Project04/Data/Trees)

Experiments

Experiment 1

For each alignment method (Clustal Omega, Kalign, MAFFT, MUSCLE), you build a NJ tree using QuickTree and RapidNJ, and compute the RF-distance between each combination of these eight tree. The outcome of your experiment, is an 8x8 table showing the RF-distance between each pair of constructed trees.

Answer:

```
Experiment 1:

CLU_Q   0   230 158 228 104 228 198 246
CLU_R   230 0   258 222 252 198 284 242
KAL_Q   158 258 0   198 120 230 176 248
KAL_R   228 222 198 0   220 176 268 212
MAF_Q   104 252 120 220 0   210 178 252
MAF_R   228 198 230 176 210 0   262 210
MUS_Q   198 284 176 268 178 262 0   192
MUS_R   246 242 248 212 252 210 192 0
```

From this experiment we can see that in general, the differences between the trees built using rapidnj are bigger than the differences between the trees built with quicktree. It is also interesting that the differences between the trees depends more on the NJ method used rather than the multiple alignment algorithm.

Experiment 2

Redo the above experiment where you use 395 input sequences in patbase_aibtas_permuted.fasta. This yields another 8x8 table.

Answer:

```
All results are from the permuted sequences.

Experiment 2:

CLU_Q   0   238 138 250 112 210 158 216
CLU_R   238 0   264 210 248 202 280 246
KAL_Q   138 264 0   230 124 220 172 242
KAL_R   250 210 230 0   246 190 276 252
MAF_Q   112 248 124 246 0   190 152 230
MAF_R   210 202 220 190 190 0   236 184
MUS_Q   158 280 172 276 152 236 0   208
MUS_R   216 246 242 252 230 184 208 0
```

The results in this experiment are pretty similar to the ones seen in the experiment 1. Futhermore we can see that rf-distances is different than the non-permuted which should be the case.

Experiment 3

Compute the RF-distance between the trees produced in ‘Experiment 1’ and ‘Experiment 2’ using the same alignment and tree reconstruction method. This yields 8 distances.

Answer:

```
Experiment 3:

CLU_Q CLU_R KAL_Q KAL_R  MAF_Q MAF_R MUS_Q  MUS_R
68    160   62    200    52    68    158    216
```

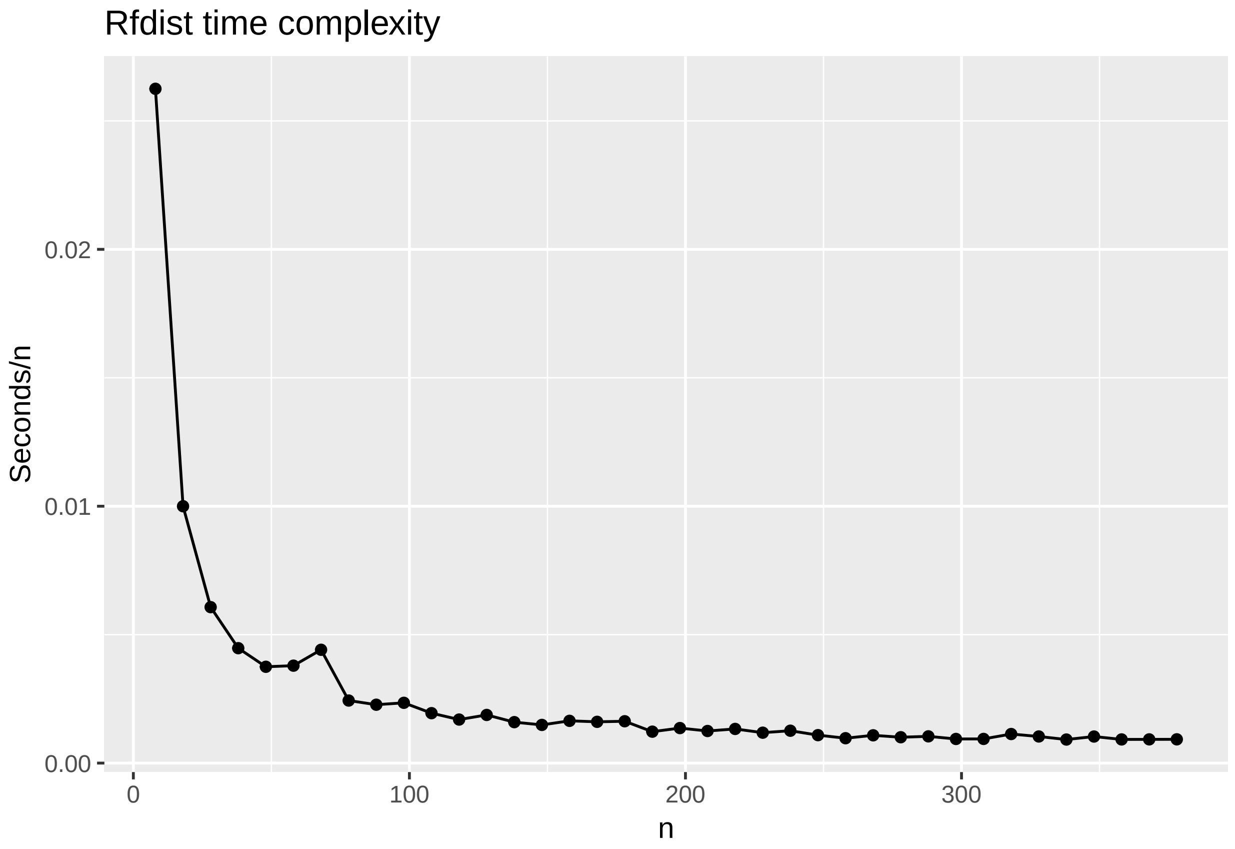
Using the same alignment and tree contruction method, permutation seems to have bigger impact in some methods than others. And generally it seems that comparing RapidsNJ trees gives a higher rf-distance then comparing Quicktrees, as observed in experiment 1.

Experiment 5

Make an experiment that shows the running time of your implementation of rfdist. You must choose test data your self. If the running time is not linear, you should explain why.

Answer:

A bash script was written that measure the time consumption for the algorithm. This bash script iterates through an increasing number (n) of sequences that are used to make two trees. The sequences corresponds to one of the sequence alignments from the patbase_aibtas.fasta datafile. In each iteration the bash scripts builds two trees and then the program rfdist is called, to calculate the rf-distance bwteeen the trees. This last step is measured in time and and the bash script returns the time corresponding to n.



Our implementation of Day's algorithm should be running in linear time O(n). The determinig factor should be how the splits are sorted. And as we used radix sort for sorting the splits this should make the algorithm run in linear time O(n). This fits with what we observe from the plot above as it seems to stabilize around n = 250.