CSCI 4118/6105 Lab 1 (Part 2):

Comparison of Evolutionary trees (rooted binary trees)

Winter 2022

Objective

1. Compare evolutionary trees using Subtree Prune and Regraft distance and the associated maximum agreement forests.
2. Calculate exact and approximate Subtree Prune and Regraft (rSPR) distances and the associated maximum agreement forests (MAFs) between pairs of rooted binary trees with and without cluster reduction.
3. The objective of this lab is not to evaluate the thorough understanding of rSPR, rather the ability to experiment and report meaningful insights.

   Extension for Graduate students:
4. Deep dive into details and impact of cluster reduction.

Pre-requisites

1. Basic understanding of Tree rearrangements (Subtree pruning and regrafting).
2. Knowledge to use basic git commands to clone, pull, push, commit and merge.
3. Java programming language: While it's not necessary to have proficiency in Java, the ability to read and understand any programming language is crucial.

Pre-read/Terminologies

1. Graph Theory: Tree, Rooted tree, Forest, agreement forest, rSPR (rooted subtree prune and regraft distance), and cluster reduction (for computing rSPR).
2. Evolutionary/Phylogenetic trees, Lateral gene transfer (LGT).
3. P, NP, NP-Hard, NP-Complete.
4. Algorithm Design Paradigm: Divide and Conquer.

Resources

- For tree pair example and answers submission: https://git.cs.dal.ca/courses/2022-winter/csci-4118-6105/lab1/???? where ???? is your CSID
- rSPR Software: https://github.com/cwhidden/rspr/

Getting Started

Scenario: In evolutionary trees, each leaf typically represents a present-day species, and each interior vertex corresponds to a hypothetical (extinct) ancestor, while the edges indicate the relationship between distinct taxa. The task is to quantify the dissimilarity between two phylogenies.

Note:

- Undergraduate students are encouraged to attempt additional questions meant for graduate students and get bonus points.
- The rSPR software should be cloned in a different folder, not the same folder as the lab1 repository.

1. Clone the repository: *git clone https://github.com/cwhidden/rspr.git*
2. Build the project: *make* and run test cases: *make test*
    a. Mac M1 users, remove the "-march=native" flag from "CFLAGS" in the rspr/MakeFile.
3. Start with playing with the rSPR software with the examples available in rspr/test_trees with different options. Example usage: *./rspr < test_trees/trees2.txt*

Example: ./rspr < test_trees/trees2.txt

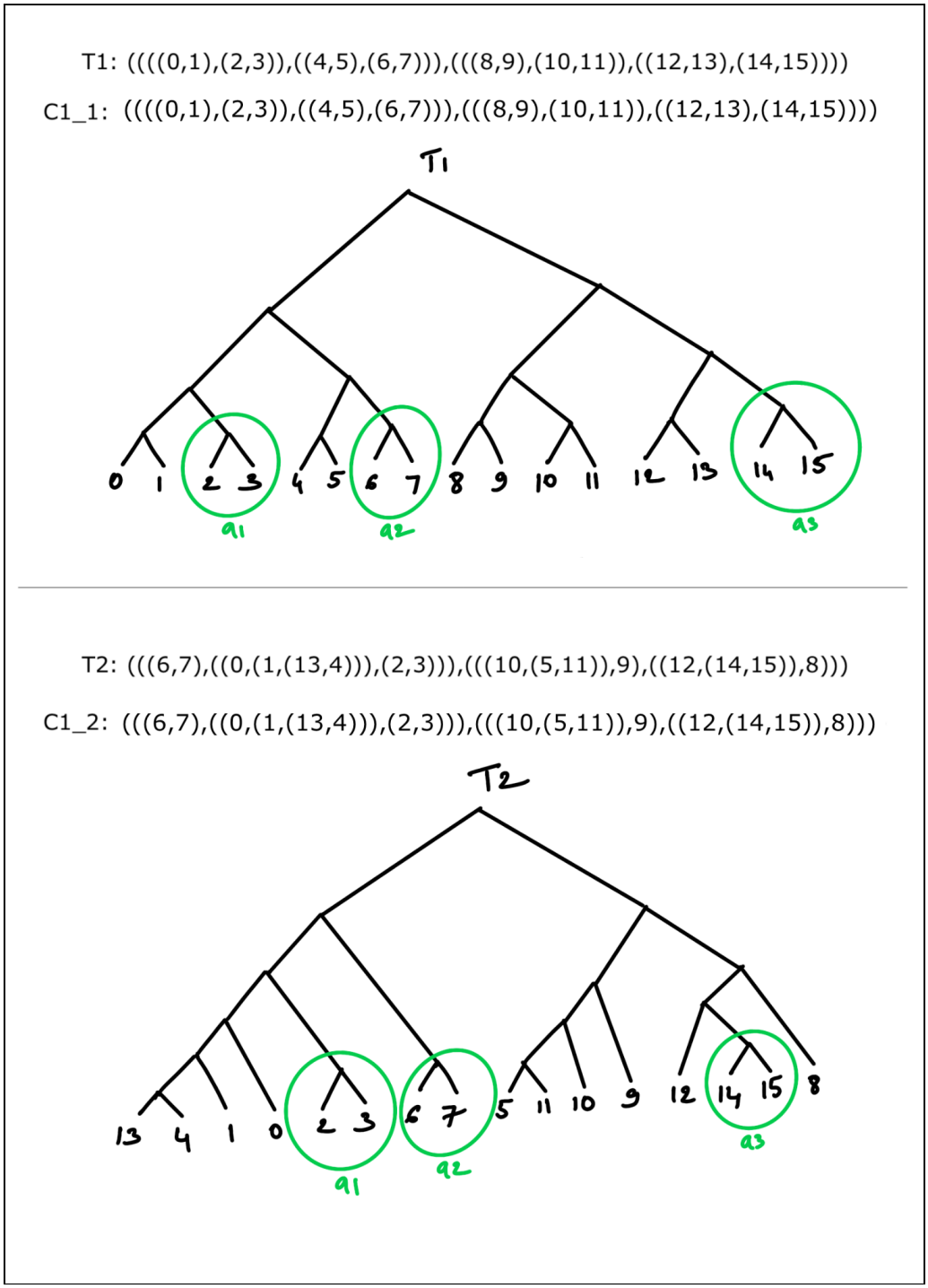

Figure 1: Visualization of a pair of rooted binary trees

Procedure and Questions

1. Report the impact of cluster reductions and optimization(s):
    a. Run rSPR on big_test:
        i. With Clustering (Default): *./rspr < test_trees/big_test*
        ii. Without Clustering: *./rspr -bb < test_trees/big_test*
    b. Now, compare the same with other optimizations:
        i. O(2.42^k n): Options: *-cob -cab -sc*
    c. Note the number of clusters and the SPR distance of the cluster(s).
        i. How many clusters did rSPR break the trees into?
        ii. What was the largest distance in any one cluster?
    d. Lastly, compare the performance for real and randomly generated tree pairs (For random trees, refer to the "dataset" section below) with and without clustering for different optimizations (*-cob -cab -sc*).
        i. How many clusters are found?
        ii. How is this different from the real big_test trees?

    Extension for Graduate students:
2. Extend Q1; To correlate the performance of cluster reductions and optimization(s) with datasets having more clusters vs. fewer or no clusters (At least five tree pairs). Construct the datasets and report your findings.

Dataset

1. Find the folder *lab/rspr/examples* in the repository.
2. Example usage: *./rspr < /path/to/folder/lab/rspr/examples/tree_100_1*
3. Now, try with other examples: tree_100_X (For all the tree pair examples in the folder). For each of these random tree examples, a random tree with 100 leaves was generated. Then X random SPR moves were applied.

Note: When running rSPR without clustering or certain optimization algorithms, you may need to kill the program if it takes too long (ctrl-C).

Submission

Note: For submission - git add, commit and push the answers to the lab1/???? repository and do not commit anything to rspr. Make sure to verify the submission in the GitLab web interface.

1. Answer the questions in *"questions_part_2.txt"*
2. Paste all the input-output pairs in *"output_part_2.txt"*, with headers for each input/output pair and use it as a reference for answering the questions in *"questions_part_2.txt"*

Grading

Task	5 Points (x1)	3 Point	0 Points
1	Question #1 Thoroughly tested and reported findings in "*questions_part_2.txt*" and "*output_part_2.txt*".	Question #1 tested partially and reported findings in "*questions_part_2.txt*" and "*output_part_2.txt*".	No evidence of testing.

Extension for Graduate students:

Task	5 Points (x1)	3 Point	0 Points
2	Question #2 thoroughly tested and reported findings in "*questions_part_2.txt*" and "*output_part_2.txt*".	Question #2 tested partially and reported findings in "*questions_part_2.txt*" and "*output_part_2.txt*".	No evidence of testing.

Grading Summary (Part 1 & 2)

Grading for Graduate Students:

Part	**Task 1**	**Task 2**	**Task 3**	**Grade**
Part 1	3	2	5	10
Part 2	5	5	-	10
Total				20

Grading for Under-graduate Students:

Part	**Task 1**	**Task 2**	**Grade**
Part 1	3	2	5
Part 2	5	-	5
Total			10

However, under-graduate students can get an additional 2 points (+1%) for attempting graduate questions (1 point for each part).