

CPR/SE 491 Weekly Report MAY 15-10 Week 6

Advisors: Ruchi Chaudhary

Client: Ruchi Chaudhary

Members (roles): Ben -- Team leader

Cole -- Team Key Concept Holder

Ian -- Webmaster

Paul -- communication leader

Project Title: Gene Tree Improvement Tool / TreeFix

Weekly Summary

With the results of last week, we decided that this week we would be most effective by splitting our team into two groups: one group to define a pseudo-code skeleton for the MulRF cost evaluation model, and the other to identify code in the existing TreeFix program that should be reused in implementing our cost evaluation model.

Paul and Cole looked through the Treefix code to see if certain tree checking methods (such as if an input tree is binary or rooted) could be used in our MulRF algorithm as well, specifically the `is_rooted` method in `treelib`, and the `reconcile` function in the `phylo` class. The findings were then reported back to the group to take inventory of what functions needed to be changed for our new algorithm. Meanwhile, Ben and Ian met up to tackle designing the pseudo-code skeleton. During this time, they recognized that a “general” cost evaluation model (G-CEM) exists in the TreeFix package structure. We are attempting to prove with just enough rigor that we can simply tweak this general structure in order to complete our goal with project. The implications of successfully proving that the model can be extended are that (1) we might easily extend the scope of our project, because (2) it should vastly decrease the amount of work we actually need to perform.

A workflow diagram was created for the duploss algorithm, this will give us bit of a structure to work off for creating the MulRF algorithm. This is being done because we feel that if the structures are similar then the implementation will be dramatically easier. This will give us our starting point to begin the architecture design for the algorithm and begin the design documentation.

A github repository was also set up this week for all team members to have the same working copy of code. The plan for the future for is for all members to comment sections of code when they analyze it.

Meeting notes:

10/10 Group Meeting with Advisors

Duration: 45 min

Members Present: Cole, Ian, Paul

Purpose and Goals: Discuss our findings with Ruchi about the feasibility of simply borrowing and tweaking code from the G-CEM, in order to implement MulRF.

We discussed our finding from the week, and also talked about the general flow of Treefix in order to figure out potential problems with retooling it for MulRF. Ruchi mentioned that the `raxml` algorithm, which takes in dna sequences to turn into graphs, might only output trees in binary format, which would create problems for refactoring. Because of this, Paul will look into the `raxml` algorithm to determine if this is the case.

Pending issues

1. Getting Ruchi’s final opinion of the implications of G-CEM.

Plans for next week

1. Ben and Ian: Start developing MulRF algorithm
2. Ben and Ian: Start creating the design documentation
3. Cole and Paul: finish up going through the `phylo` package, and start digging through the `treelib` library class to identify code for uses in MulRF implementation

4. Paul: look through the raxml algorithm to determine if it outputs only binary trees.

Individual Contributions(this week)

Ben Streit(4 hrs)- Created a workflow graph for the duploss algorithm, met with ian and discussed our possible structure

Ian Ray (5.5 hr)- (1) Met with Ben to tackle algorithm design. Attempted to use PyCallGraph to structurally analyze TreeFix. Attempt failed because of program's inability to identify package dependencies. (2) Made the discovery in `__init__.py` of G-CEM and realized its potential implications. (3) Extended Time tables from 7 weeks to 14 weeks.

Paul Leichty (2.5 hr) analyzed the `is_rooted`, `is_binary`, and `reconcile` functions in Treefix to determine if any sections can be reused and retooled for the MulRF algorithm. Also set up a github repository for the group, and committed all code so the team can collectively work on the project.

Cole Poffenberger (2 hr) analyzed several methods in `phylo.lib` with paul, and documented what changes would have to be made in order to implement the MulRF algorithm.

Total contributions for the project

Ben Streit(15.75 hrs)

Ian Ray(18.5 hrs)

Paul Leichty (13 hr)

Cole Poffenberger(12.75 hr)