TreeFixVP

Mukul Bansal
Department of Computer Science
University of Connecticut

Samuel Sledzieski
Department of Computer Science
University of Connecticut

February 14, 2018

1 Introduction

TreeFixVP is a program for reconstructing highly accurate viral phylogenetic trees. TreeFixVP incorporates host information with sequence data to reconstruct the tree using a mixed maximum likelihood / maximum parsimony framework. Given a maximum likelihood phylogeny and a multiple sequence alignment, TreeFixVP searches among multiple topologies which are supported equally by sequence data (using the Shimodaiara-Hasegawa test statistic). The selected topology is the one that minimizes then umber of necessary transmissions.

Requirements

- Python (2.5.4 or greater)
- C compiler (gcc)
- SWIG (1.3.29 or greater)
- Numpy (1.5.1 or greater)
- Scipy (0.7.1 or greater)

- Additionally, Python modules are required for computing the p-value for likelihood equivalence.

Likelihood TreeFixVP uses the Shimodaira-Hasegawa (SH) test statistic with RAxML site-wise likelihoods to compute p-values for each candidate tree.

Parsimony TreeFixVP scores each statistically equivalent candidate tree using Fitch's algorithm. Given host labels at the leaf nodes, the Fitch module computes a score equivalent to the minimum necessary number of transmissions needed to label the internal nodes.

2 Tutorial

Input TreefixVP requires a seed tree, generally a maximum likelihood tree, and a multiple sequence alignment

2.1 Usage

treefixVP -[options] <treefile>

Options TreeFixVP assumes that the multiple sequence alignment and seed tree have the same root name, with different extensions. The default extensions are ".align" and ".tree", but other extensions can be specified. The output tree file will have the same root name, and will have either the default extension ".treefix.tree", or a user specificed extension.

- -A <alignment file extension>, --alignext=<alignment file extension>
- -o <old tree file extension>, --oldext=<old tree file extension>
- -n <new tree file extension>, --newext=<new tree file extension>

The default likelihood model uses RAxML site-wise likelihoods, but a different user-defined module can be substituted. The default likelihood test is the SH test, but again a user-defined test can be used.

- -m <module for likelihood calculations>, --module=<module for likelihood calculations>
- -e <extra arguments to module>, --extra=<extra arguments to module>
- -t <test statistic>, --test=<test statistic>
- --alpha=<alpha>→ alpha threshold (default: 0.05)
- -p <alpha>, --pval=<alpha>

The default module for transmission cost uses Fitch's algorithm. A user defined cost model can be substitued.

- -M <module for transmission cost calculation>, --smodule=<module for transmission cost calculation>

A number of options can be specified to guide the search conducted by TreeFixVP.

- -x <seed>, --seed=<seed>→ seed value for random generator
- --niter=<number of iterations>→ number of iterations (default: 1000)
- --nquickiter=<number of quick iterations>→ number of subproposals (default: 100)

Finally, there are a handful of informational options.

- --version → show program's version number and exit
- -h, --help → show the help message and exit
- -V <verbosity level>, --verbose=<verbosity level>
- -l <log file>, --log=<log file>→ log filename. Use '-' to display on stdout.

3 Test Results

TreeFixVP was tested on 8 sample trees and multiple sequence alignments. Below is a list of the 8 datasets. The column labeled "Old Score" is the Fitch score of the original tree, and the column labeled "New Score" is the Fitch score of the tree reconstructed by TreeFixVP. Also included are the number of leaves for the given tree, and the time of the run.

Name	Leaves	Old Score	New Score	Runtime (minutes)
AA	118	7	5	48
AB	48	2	1	28
AC	86	3	3	32
AD	76	2	1	67
AE	29	3	2	15
AH	53	2	1	31
AI	163	71	51	71
AJ	49	3	3	21

3.1 Example

Find a sample dataset that we can release and walk through an example here

4 Changes from TreeFixDTL

4.1 treefixVP

- Renamed treefixDTL executable
- Removed most options in parser, only need alignment extension, old extension, new extension
- Changed default smodule to FitchModel
- Call treefixVP rather than treefix

4.2 fitchmodel.py

- Copy of rangerdtlmodel.py

- Changed compute_cost() to call fitch.linux executable
- Removed stree, smap from FitchModel.optimize_model
- Removed call to CostModel.optimize_model()

4.3 treefix_for_VP

- Renamed treefix to treefix_for_VP - this avoids path collision with treefix and treefixDTL
- Removed smap and stree as required arguments
- Changed to common.check_req_options(species=False)
- Removed stree and gene2species from check_input_tree()
- Removed stree and gene2species from search_landscape()
- Removed check_congruent_tree() function, no longer applies
- Removed 'if flag: return mintree'
- Removed reading species tree and species map from main()
- Removed stree and gene2species from calls to optimize_model(), check_input_tree(), and search_landscape() in main

4.4 General

- Removed all ranger executables
- Removed treefix_compute
- Removed rangerdtl model from models module
- Modified setup files to install the proper executables
- Updated treefixVP.py with version and software info
- Updated INSTALL.txt, CHANGES.txt, and README.txt
- Added README.pdf