TreeFix-TP Manual

1 Introduction

TreeFix-TP is a program for reconstructing highly accurate transmission phylogenies, i.e., phylogenies depicting the evolutionary relationships between infectious disease strains (viral or bacterial) transmitted between different hosts. TreeFix-TP is designed for scenarios where multiple strain sequences have been sampled from each infected host, and it uses the host assignment of each sequence sample to error-correct a given maximum likelihood phylogeny of the strain sequences. Specifically, given a maximum likelihood phylogeny, the multiple sequence alignment on which the phylogeny was built, and the host assignment for each sequence, TreeFix-TP searches around the maximum likelihood phylogeny to find an alternate error-corrected phylogeny which is equally well-supported by the sequence data and minimizes the number of necessary inter-host 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   TreeFix-TP uses the Shimodaira-Hasegawa (SH) test statistic with RAxML site-wise likelihoods to compute p-values for each candidate tree.

Parsimony   TreeFix-TP 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

Usage   `treefixTP -[options] <treefile>`

Input   TreeFix-TP requires a seed tree, generally a maximum likelihood tree, and a multiple sequence alignment.

Options TreeFix-TP 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 specified 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>`: same as alpha

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

- `-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 TreeFix-TP.

- `-x <seed>, --seed=<seed>`: seed value for random generator

- `--niter=<number of iterations>`: number of iterations (default: 5000)

- `--nquickiter=<number of quick iterations>`: number of sub-proposals (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 file name. Use '-' to display on stdout.

TreeFix-TP determines the host of each sequence by parsing the name assigned at the leaf. Newick and Fasta formatted tree and sequence files should have the sequences in the form [host name]_[sequence name]. Sequences can be renamed using bin/seq_utils.py.

3   Changes from TreeFix-DTL

TreeFix-TP was built by modifying TreeFix-DTL [**?**], with the most significant change being the replaced cost module (fitch.linux rather than ranger-dtl). All changes are documented below.

3.1   treefixTP

- Based on treefixDTL script

- Removed most options in parser, only need alignment extension, old extension, new extension

- Changed default smodule to FitchModel

- Call treefix_for_TP rather than treefix

3.2   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()

3.3   treefix_for_TP

- Renamed treefix to treefix_for_TP - 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

3.4 General

- Removed all ranger executables

- Removed treefix_compute

- Removed rangerdtl model from models module

- Modified setup files to install the proper executables

- Updated treefixTP.py with version and software info

- Updated INSTALL.txt, CHANGES.txt, and README.txt

- Added README.pdf