TreeFix-TP Manual

Version 1.2.1, Compiled November 24, 2020.

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 (3.5 or greater)
- C compiler (gcc)
- SWIG (1.3.29 or greater)
- Numpy (1.5.1 or greater)
- Scipy (0.7.1 or greater)
- Dendropy (Optional for `ttp-parse-log`)
- 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.

Formatting Note 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].`** See `examples/test_TP.fasta` for an example.

2 Usage

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 are in the same directory with the same root name, with different extensions. The default extensions are ".fasta" and ".tree" for the multiple sequence alignment and maximum likelihood phylogeny, 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. The default statistical test (Shimodaira-Hasegawa) and cost calculation (Fitch's Algorithm) can be substituted with user defined Stat and Cost models.

```
$ treefix-tp --help
Usage: treefix-tp [options] <gene tree> ...

TreeFix-TP is a phylogenetic program for improving viral phylogenetic tree
reconstructions using a test statistic for likelihood equivalence and a
transmission aware cost function. See http://github.com/samsledje/TreeFix-TP
for details.

Options:
  Input/Output:
    -A <alignment file extension>, --alignext=<alignment file extension>
                        alignment file extension (default: ".fasta")
    -o <old tree file extension>, --oldext=<old tree file extension>
                        old tree file extension (default: ".tree")
    -n <new tree file extension>, --newext=<new tree file extension>
                        new tree file extension (default: ".treefix.tree")
    -r, --reroot        set to reroot the input tree

  Likelihood Model:
    -m <module for likelihood calculations>, --module=<module for likelihood
    calculations>
                        module for likelihood calculations (default:
                        "treefix_tp.models.raxmlmodel.RAxMLModel")
    -e <extra arguments to module>, --extra=<extra arguments to module>
                        extra arguments to pass to program

  Likelihood Test:
    -t <test statistic>, --test=<test statistic>
                        test statistic for likelihood equivalence (default:
                        "SH")
    --alpha=<alpha>     alpha threshold (default: 0.05)
    -p <alpha>, --pval=<alpha>
                        same as --alpha

  Transmission Cost Model:
    -M <module for transmission cost calculation>, --smodule=<module for
    transmission cost calculation>
                        module for transmission cost calculation (default:
                        "treefix_tp.models.fitchmodel.FitchModel")
```

```
  Search Options:
    -b <# bootstraps>, --boot=<# bootstraps>
                        number of bootstraps to perform (default: 1)
    -x <seed>, --seed=<seed>
                        seed value for random generator
    --niter=<# iterations>
                        number of iterations (default: 100)
    --nquickiter=<# quick iterations>
                        number of subproposals (default: 50)
    --freconroot=<fraction reconroot>
                        fraction of search proposals to reconroot (default:
                        0.05)
    --maxtime=<maximum runtime>
                        maximum runtime (per tree) in seconds

  Information:
    --version           show program's version number and exit
    -h, --help          show this help message and exit
    -V <verbosity level>, --verbose=<verbosity level>
                        verbosity level (0=quiet, 1=low, 2=medium, 3=high)
    -l <log file>, --log=<log file>
                        log filename.  Use '-' to display on stdout.

  Debug:
    --debug=<debug mode>
                        debug mode (octal: 0=normal, 1=skips likelihood test,
                        2=skips cost filtering on pool, 4=computes likelihood
                        for all trees in pool)
```

2.1 Additional Tools

ttp-parse-log Get summary statistics and consensus trees from the phylogenies considered by TreeFix-TP (requires DendroPy).

```
$ ttp-parse-log --help
Usage: ttp-parse-log [options] <log file>

Options:
  -h, --help            show this help message and exit
  --out=OUT_PATH        Path for output
  --near=NEAR_PERCENT   Trees within <--near>% of the optimal cost will be
                        captured
  --true=TRUE_TREE_PATH
                        Can provide a true tree to compare multiple optimal
                        trees with
  --include_near        Include nearby optimal trees in summary statistics
  --separate_trees      Create two output files separating trees and
                        statistics
```

ttp-check-cost Compare the transmission cost of the input phylogeny and the TreeFix-TP optimal phylogeny.

```
$ ttp-check-cost --help
usage: python ttp-check-cost.py [old tree file] [new tree file]
```

ttp-check-likelihood Compare the sequence support of the input phylogeny with the TreeFix-TP optimal phylogeny.

```
$ ttp-check-likelihood --help
usage: python ttp-check-likelihood.py [alignment] [old tree file] [new tree file]
```

3 Example

```
$ cd examples/
$ # Show Files
$ ls
test.sh    test_TP.fasta    test_TP.network    test_TP.raxml    test_TP.true_tree

$ # Run TreeFix-TP
$ treefix-tp -A .fasta -o .raxml -n .treefix -V1 -l test_TP.log test_TP.raxml

$ # Show Consensus Trees and compare with true tree
$ ttp-parse-log test_TP.log --true test_TP.true_tree

$ # Check Sequence Support
$ ttp-check-likelihood test_TP.raxml test_TP.treefix

$ # Check Cost Decrease
$ ttp-check-cost test_TP.fasta test_TP.raxml test_TP.treefix
```

Attribution

 If you use TreeFix-TP please cite TreeFix-TP: Phylogenetic Error-Correction for Infectious Disease Transmission Network Inference. Samuel Sledzieski, Chengchen Zhang, Ion Mandoiu, and Mukul S. Bansal. Pacific Symposium on Biocomputing (PSB) 2021: Proceedings, pages 119-130.