# RF+ software version 0.2

## Description

RF+ is a prototype program for computing RF(+) distances between phylogenetic trees. RF(+) distance is designed to more meaningfully compute the Robinson-Foulds distance between two trees that only have a partially overlapping leaf set. The traditional approach for computing Robinson-Foulds distance between two trees that only have a partially overlapping leaf set is to first restrict the two trees to their shared leaf set and then compute their Robinson-Foulds distance. We refer to distances computed in this way as RF(-) distances. In contrast, the RF(+) distance between two arbitrary trees is computed by first optimally completing each tree on the union of the leaf sets of both trees so as to minimize the Robinson-Foulds distance between them, and then reporting the Robinson-Foulds distance between the two completed trees.

The current prototype implements the algorithms described in the manuscript cited below and can (i) compute the RF(+) distance between a pair of trees where the leaf set of one of the trees is a subset of the leaf set of the other, (ii) compute the Extraneous-Clade-Free-RF(+) (EF-RF(+)) distance between two trees with arbitrary leaf sets, and (iii) compute the RF(+) distance between two trees with arbitrary leaf sets. All trees must be rooted and binary. If the two given trees share no leaves in common, then an optimal completion is trivial and the EF-RF(+) call will return a tree which merely joins the two trees together with a new root node, even though this is not extraneous clade free.

We refer the reader to the following paper for further details on RF(+) and EF-RF(+) distance.

“[Linear-Time Algorithms for some Phylogenetic Tree Completion Problems under Robinson-Foulds Distance](https://compbio.engr.uconn.edu/wp-content/uploads/sites/2447/2018/08/RF_recombcg2018.pdf)”, Mukul S. Bansal, *RECOMB Comparative Genomics Conference (RECOMB-CG) 2018*; LNCS 11183: 209-226.

## Implementation details and requirements

RF+ is implemented in Python and requires version 3.0 or greater. The implementation also assumes that ETE 3 toolkit is already installed. ETE toolkit is available freely from etetoolkit.org

We point out that while the algorithms presented in the manuscript cited above have linear, O(n), time complexity, this current implementation of RF+ has O(n log n) time complexity since it implements a slightly suboptimal algorithm for Least Common Ancestor (LCA) computation.

RF+ is freely available open source under GNU GPL.

## Usage

RF+ takes as input two or more trees and it compares the first tree with every other tree in the input file. All input trees must be in newick format with only leaf node labels, no edge lengths, and must be in a single input file with each tree appearing on a separate line. By default, the program outputs only the optimal completions. These completions are output in pairs, for every possible pair of trees (considered in order of their appearance in the input file). Note that if the first tree already contains all leaves present in the other tree then only the other tree is completed and the first tree is output as-is. The input file is specified using the “-i” option. An output file (optional) can be specific using the “-o” option. For example,

python3 RF+.py -i input.newick -o output.txt

will write the RF(+) completions, grouped together by pair of input trees where each tree is on its own line, into the specified output.txt file. The “-r” option can be used to output the RF(-), EF-RF(+) and RF(+) distances between every pair of trees. For example,

python3 RF+.py -i input.newick -o output.txt -r

will compute every pairwise RF(-), EF-RF(+) and RF(+) distance from the specified input file and write them all in the specified output file. The “-ext” option can be used to output *only* EF-RF(+) *completions* to either be written to an output file or printed. The “-ext” option is unnecessary if the “-r” option is used. For example,

python3 RF+.py -i input.newick -ext

will print every pair of optimal EF-RF(+) completions. The “-u” option can be used to compute unrooted EF-RF(+) and/or RF(+) distances. For example,

python3 RF+.py -i input.newick -o output.txt -u -ext

will write every pair of optimal EF-RF(+) completions from the specified input file of unrooted trees into the specified output file. Lastly, the “-a” option can be used to write certain pieces of information for all pairs of trees, such as the leaf set intersection size and runtimes, into a specified csv file. For example,

python3 RF+.py -i input.newick -a csvfile.csv

will compute every pairwise RF(+) distance and store collected information into the csvfile.csv file provided.

## Example dataset

As an example, we provide the binarized marsupials, legumes and placental mammals datasets used in the paper cited above. This dataset consists of a full tree on 272 species followed by 10 trees that each have a small subset of the leaf set of the full tree. Sample command to run RF+ on this dataset:

python3 RF+.py -i MarsupSources2.newick -o outputfile.txt -r