

mrc.py - Evaluate the compatibility lengths of supertrees and gene trees

Manual Version 1.0 (March 5, 2010)

Anne Kupczok

Center for Integrative Bioinformatics Vienna,
Max F. Perutz Laboratories,
Dr. Bohr-Gasse 9/6, A-1030 Vienna, Austria.
email: `anne.kupczok @ mfpl.ac.at`

1 Legal Stuff

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

2 Introduction

`mrc.py` is a computer program to evaluate the compatibility lengths given supertrees and gene trees and to extract the supertrees with the minimal lengths. It does not do any tree search under the compatibility criterion.

Matrix Representation with Compatibility (MRC) was first suggested by Rodrigo (1996). The optimizing function of MRC is maximizing the number of gene tree splits that can be arranged in a tree without conflict. Thereby the *compatibility length* (CL) of a supertree is the number of gene tree splits *not* in the supertree. `mrc.py` computes the CLs of all input supertrees given a set of gene trees (in newick format or a matrix representation) and outputs the subset of supertrees having the minimal length.

This implementation was used for the computations in Kupczok (2009).

3 Installation

The command-line program is freely available from <http://www.cibiv.at/software/mrc/>. It is written in python and should run on every computer with python version 2.4 or newer versions of python 2.x. It will not run with python 3.x. Python can be downloaded from <http://www.python.org/>.

First unzip `mrc-1-0.zip`, then change into the directory `MRC` and type `python mrc.py [options]` to run the program.

4 Command-line options

Run `python mrc.py -h` to print out a short description of available options:

Usage: `mrc.py [options]`

Options:

`-h, --help` show this help message and exit

Input options:

`-s SOURCE, --source=SOURCE`

Filename for source trees in newick format

`-t TAXA, --taxa=TAXA`

Taxafilename: each taxon one line, default no file,
read out of gene trees (ignored if smatrix given)

`-z SMATRIX, --smatrix=SMATRIX`

Source trees as matrix representation in nexus format
(ignored if source trees given)

`-r STREE, --tree=STREE`

File with supertrees in newick format

Output options:

`-o HEADER, --head=HEADER`

Header for output (default mrc)

`-v, --time`

Verbose mode with printing times to stdout (default)

4.1 Examples

```
python mrc.py -s examples/trees.nck -r examples/trees_6 -o trees1 -t examples/taxa
python mrc.py -z examples/matrep.nex -r examples/trees_6 -o trees2
```

To test whether the program runs on your system you can test these commands and compare whether the resulting files `trees1.eval`, `trees1.tree`, `trees2.eval` and `trees2.tree` equal those in the `example`-directory.

4.2 Input options

`-s SOURCE, --source=SOURCE` The filename of a list of source (gene) trees in newick format. The gene trees may contain missing data but the union of their taxon sets needs to be equal to the taxon set of the supertrees.

`-t TAXA, --taxa=TAXA` This option is only considered if a list of gene trees is given with `-s`. If the taxa are not given, they are read from the gene trees.

`-z SMATRIX, --smatrix=SMATRIX` The source trees in the binary matrix representation. For this analysis a PAUP input file can be used, see example in `examples/matrep.nex`. Note that this paup file can be run using PAUP* (Swofford, 2002) with `paup -n matrep.nex`, this corresponds to the MRP analysis. If the PAUP block contains weights, these are considered by `mrc.py` as well. Note that the weights are rescaled such that the weights sum up to the original alignment length. Thus the compatibility length are returned as rational numbers. They can be rescaled to integers by multiplying them with s/l where s is the sum of weights and l is the alignment length. Missing taxa may be coded with `-`.

`-r STREE, --tree=STREE` List of supertrees in newick format.

4.3 Output options

- o HEADER, --head=HEADER Header for output file. Two files are generated by `mrc.py`: `header.eval` and `header.tree`. The `eval`-file lists the parsimony length for each given supertree in the same order. The `tree`-file contains the subset of the supertrees with the minimal length.
- v, --time This option suppresses the output of the elapsed time after each step in the computation.

References

- Anne Kupczok. Consequences of different null models on the tree shape bias of supertree methods. submitted to *Syst. Biol.*, 2009.
- Allen G. Rodrigo. On combining cladograms. *Taxon*, 45:267–274, 1996.
- David L. Swofford. *PAUP*: Phylogenetic Analysis Using Parsimony (*and Other Methods). Version 4*. Sinauer Associates, Sunderland, Massachusetts, 2002.