DaTeR (Version 1.0)

https://compbio.engr.uconn.edu/software/dater/

Description

DaTeR (short for "Dating Trees using Relative constraints") is a program for improved dating of microbial species phylogenies using relative time constraints (e.g., obtained from high-confidence horizontal gene transfer events). Traditional phylogenetic dating approaches make use of absolute time constraints, which provide lower and/or upper bounds for one or more nodes of the underlying phylogeny, but are unable to use relative constraints that specify that some node *x* must be dated to be at least as old as some other node *y*. DaTeR takes as input a collection of chronograms sampled from the posterior using any traditional Bayesian phylogenetic dating approach (based on only absolute time calibrations), along with a set of curated relative time constraints, and minimally error-corrects each input chronogram to ensure compatibility with all available relative time constraints. It then outputs the individual error-corrected chronogram samples as well as an aggregated, final chronogram. DaTeR uses a constrained optimization framework and computes a minimal deviation from assigned node dates or branch lengths (representing time) under three appropriately designed candidate objective functions. Further technical details appear in the paper cited below.

> *DaTeR: Error-Correcting Phylogenetic Chronograms Using Relative Time Constraints*
> Abhijit Mondal, L. Thiberio Rangel, Jack Payette, Gregory P. Fournier, Mukul S. Bansal
> Under review

DaTeR is freely available from https://compbio.engr.uconn.edu/software/dater/

Dependencies

DaTeR requires Python 3 as well as the following Python libraries: scipy, dendropy, networkx, docplex, and cplex.

Note: If using the SBD and SDD objective functions with large trees (say with more than 200 leaves), users may need to install the full, unlimited version of IBM ILOG CPLEX. This full and unlimited version can be installed/used for free by faculty members/researchers/students at most academic institutions under IBM's academic initiative.

Usage

DaTeR takes as input two files: (1) An input chronograms file containing one or more sampled chronograms (dated phylogenies), one per line and in the Newick format, for the species phylogeny being dated. And (2) A constraints file listing all available relative time constraints for nodes in the species phylogeny being dated. Each internal node in the species phylogeny (i.e., in all corresponding input chronograms) must have a name/label and each edge must have a branch length (representing time). The current implementation of DaTeR requires that all chronograms be ultrametric (i.e, for any input chronogram, the root to leaf distance must be the same for all leaves in that chronogram). Each line in the constraints file specifies one relative constraint, where each constraint consists of two node

labels separated by a space. E.g., the constraint "*x y*" specifies that node *x* must be dated to be at least as old as node *y*. A sample input chronogram file and a sample constraints file are available in the software directory (see files `SampleInputTrees.newick` and `SampleConstraints.txt`, respectively).

Users must also specify an output file name and select an objective function or "model" to use for the optimization. There are three options for the model: SLRB, SBD, or SDD (as described in the associated manuscript). If a model is not specified, then the SLRB model is used by default.

DaTeR can be executed as follows:

```
python3 dater.py -i inputFile -o outputFile -c constraintsFile [-options]
```

Available command line options are listed and described below.

List of command line options

-i File containing input chronograms. File should contain one or more sampled chronograms in Newick format, one per line. This is a required parameter.

-o Output file name. This is a required parameter.

-c File containing list of relative time constraints, one per line. This is a required parameter.

-m Objective function or model to be used. Options are "SBD", "SLRB", and "SDD", with "SLRB" used by default.

-h Prints out a brief help message and exits.

Interpretation of the output

Each chronogram from the input file is error-corrected and written to the specified output file (as specified using the -o option), one per line, in Newick format. Thus, if the input chronograms file has 100 chronograms then this output file will also contain 100 error-corrected chronograms. If the input chronogram file has more than one chronogram, then a second output file containing a single *aggregated* chronogram, aggregated across all of the individual error-corrected chronograms, is also created. This aggregated chronogram represents the final (i.e., overall best estimate) chronogram output of DaTeR based on all the given input chronograms and it is written to a file whose name begins with "aggregated_" followed by the specified output file name. Note that this aggregated chronogram file is only written if the input chronogram file contains more than one chronogram.

For example, if the output file name specified using the -o options is "output.txt", then the individual error-corrected chronograms and the final aggregated chronogram will be written to the files named "output.txt" and "aggregated_output.txt", respectively.

Example input files

The software directory includes a sample input chronograms file (SampleInputTrees.newick; consisting of two chronograms for the same underlying species phylogeny) and a sample constraints file (SampleConstraints.txt; consisting of two relative time constraints. The software can be executed on this input file using the following command:

```
python3 dater.py -i SampleInputTrees.newick -o output.txt -c
SampleConstraints.txt
```

The above command would use the SLRB objective function. To use, for example, the SBD objective function instead, one would execute the following command:

```
python3 dater.py -i SampleInputTrees.newick -o output.txt -c
SampleConstraints.txt -m SBD
```

Selecting the best objective function/model

The three objective functions can result in different final chronogram estimates. Overall, our experiments suggest that SLRB may be preferable to the other objective functions in many cases since it results in the least overall percent change in branch lengths between the input and error-corrected chronograms (i.e., it rescales input branch lengths minimally). Thus, SLRB is a good ``default" objective function to use with DaTeR. However, SLRB can result in greater absolute deviation in branch lengths and node dates compared to the other two objective functions.

The SBD objective function attempts to strike a balance between absolute deviation of branch lengths and percent change of branch lengths and it often results in chronograms that are similar to those computed using SLRB. Specifically, SBD results in chronograms with greater percent change in branch lengths than SLRB but smaller absolute deviation of branch lengths and node dates. Note, also, that SLRB can be significantly slower than SBD, so SBD can be used in place of SLRB when error-correcting very large chronograms.

Finally, the SDD objective function focuses on minimizing the deviation of assigned node dates and therefore results in the chronograms with much smaller absolute deviations of branch lengths and node dates than the other two objective functions. However, SDD chronograms often show much higher percent change in branch lengths. SDD may be preferable under certain scenarios, e.g., when the goal is to minimize overall change from the input chronogram in node dates or branch lengths.

Contact Information

In case of any questions, please feel free to contact Abhijit Mondal (abhijit.mondal@uconn.edu) or Mukul Bansal (mukul.bansal@uconn.edu).