DupLoss-2 User Manual

https://compbio.engr.uconn.edu/software/duploss-2/

Description

DupLoss-2 is a program for phylogenomic species tree inference using gene tree parsimony. It takes as input a collection of gene trees (rooted or unrooted) and seeks a species tree that best reconciles the input gene trees under a gene duplication and loss reconciliation model. DupLoss-2 can lead to significant improvements in species tree reconstruction accuracy compared to other existing methods on phylogenomic datasets where gene duplication and loss are the primary drivers of gene family evolution. Notably, DupLoss-2 is substantially more accurate than iGTP-DupLoss, the previous version of this method/software.

DupLoss-2 is easy to use and scalable to whole-genome datasets with thousands of gene trees from hundreds of taxa. Methodological details and experimental results appear in the paper cited below.

> *DupLoss-2: Improved Phylogenomic Species Tree Inference under Gene Duplication and Loss*
> Rachel Parsons and Mukul S. Bansal
> Under review

DupLoss-2 is freely available open-source under GNU-GPL v3 from the following URL: https://github.com/Bansal-CompBioLab/DupLoss-2

Using DupLoss-2

Input format. DupLoss-2 takes as input a single file consisting of all input gene trees in Newick format. Each gene tree must be terminated by a semicolon and must be binary (i.e., fully resolved). Multifurcations in the gene trees will trigger a warning message and all multifurcations will be arbitrarily resolved. Leaf names in the gene trees must correspond to the species from which the gene was sampled. A gene trees can contain any number (zero, one, or more) of homologous genes from the same species, and can be either rooted or unrooted. The rooting of a gene tree can be specified by the prefix [&U] or [&R] for unrooted or rooted, respectively. A gene tree with no prefix is assumed to be rooted. Gene trees specified as unrooted must still be provided in rooted Newick format. For example, an input file consisting of three input gene trees (two rooted and one unrooted) may look like the following:

```
(((speciesA, speciesB), speciesC), speciesA);
(((speciesB, speciesB), speciesC), speciesD);
[&U](((speciesA, speciesB), speciesC), speciesD);
```

Species labels can only use alphanumeric characters and underscores. Lables with non-alphanumeric characters (e.g., spaces, hyphens, etc.) have to be encapsulated in apostrophes or quotation marks. For example: speciesA, "species A" or 'species-A'.

Gene tree weighting (optional): The gene trees can optionally be weighted. If a gene tree is weighted, its reconciliation cost is multiplied by its weight. Thus, gene trees with higher weight will have greater impact on the tree search. The weight must be a positive number (floating point) between 0 and 1 and can be specified using the prefix [&WEIGHT=<value>]. By default, each gene tree has a weight of 1. For example, an input file might consist of two weighted and one unweighted gene tree, as shown below.

```
[&WEIGHT=0.5](((speciesA, speciesB), speciesC), speciesA);
[&WEIGHT=0.8](((speciesB, speciesB), speciesC), speciesD);
(((speciesA, speciesB), speciesC), speciesD);
```

Three sample input files are provided in the testData directory of the software.

Executing DupLoss-2: Given a collection of rooted and/or unrooted, binary gene trees, the default behavior of DupLoss-2 is to first construct an initial species tree using a quick, step-wise leaf addition heuristic. This initial species tree serves as a starting point for the SPR-based local search heuristic that follows. The output consists of the best species tree found during the search, written in Newick format, along with some additional information. Multiple runs of DupLoss-2 on the same input can result in different species trees, depending on the search path taken by the heuristic. Thus, we recommend executing DupLoss-2 multiple times (we suggest 10 times) on the same dataset and selecting the species tree(s) with lowest total reconciliation cost. The MultiRunScript.py Python script, described later, can help with automating such multiple executions. DupLoss-2 also provides options for starting the local search heuristic using a user-provided initial species tree, and for imposing constraints on the topology of the species tree; these options are described below.

The only required parameter to run DupLoss-2 is an input file containing gene trees as described above. E.g.,

```
./DupLoss-2.linux -i inputFile.newick
```

```
./DupLoss-2.linux -i testData/vertebrates.newick
```

An output file can be specified using the “-o” command line argument. E.g.

```
./DupLoss-2.linux -i inputFile.newick -o SpeciesTree.output
```

If an output file is not specified, the output species tree, along with processing output and additional information, is written to standard output. If an output file is specified, processing information is still written to standard output but the species tree, along with additional information such as the total weighted reconciliation cost, are written to the output file.

Providing a starting species tree (optional): DupLoss-2 can optionally be provided with a user-given initial species tree to start the local search. This can be done by using the command line argument "--generator 0". In this case, the first tree in the input will be interpreted as the starting species tree (and all remaining trees as gene trees). This user-provided starting species tree must be in Newick format and must be rooted and binary. If a starting species tree is provided, DupLoss-2 will skip the initial species tree generation step and directly apply the SPR-based local search heuristic. E.g.,

```
./DupLoss-2.linux --generator 0 -i inputFileWithSpeciesTree.newick
```

Specifying clade constraints (optional): DupLoss-2 allows users to optionally specify constraints on the species tree topology. Specifically, users can specify one or more disjoint clades and the final species tree computed by DupLoss-2 is guaranteed to contain those clades. This has to be done differently depending on whether the user provides a starting species tree (--generator 0 option) or not. If a starting species tree is provided, constraints can be specified by attaching [&CONSTRAINT] to the root node of the clade to be constrained in the species tree. E.g.,

```
(((cat,turtle),dog)[&CONSTRAINT],(rice,arabidopsis));
```

In this example, cat, turtle, and dog will always form a clade. However, note the final species tree need not maintain the specific relationships within this clade. Thus, the resulting final species tree might be (((cat,dog),turtle),(rice,arabidopsis));

If DupLoss-2 is run in its default mode (i.e., not starting species tree provided), then a separate file containing the constraints can be specified using the “--constraints” command line argument. This constraints file can contain multiple clade constraints, one per line, where each clade constraint is a comma separated list of species terminated with a semicolon. In the resulting species tree, the species in a constraint will form a clade. Multiple constraints can be specified, but any single species can be part of at most one constraint. In other words, the constraints cannot be nested. E.g.,

```
./DupLoss-2.linux -i inputFile.newick --constraints constraints.txt
```

Where the constraints.txt file may contain the following two clade constraints:

```
cat, dog, turtle, human;
rice, arabidopsis;
```

Full list of command line options:

-i, --input	Input file
-o, --output	Output file
--genetrees	Output input gene trees and their reconciliation costs after the species tree
-g, --generator 0|1	Options for species tree to use to initialize the local search heuristic 0 - use user-provided species tree as initial species tree 1 - use leaf adding heuristic to build initial species tree [default]
--constraints	A file containing groupings of species for generator 1.
--limit	Limit species tree to leaf set of gene tree when computing losses. Not recommended.
-q, --quiet	No processing output.
--seed <integer>	Set a user defined random number generator seed.
-v, --version	Output the version number.
-h, --help	Output brief help message and example commands.

Test data and example commands

We provide three sample test files to help illustrate the proper input format for DupLoss-2. The file vertebrates.newick consists of several rooted gene trees. The file vertebrates.unrooted.newick consists of several unrooted gene trees. And the file vertebrates.weighted.newick consists of several weighted gene trees.

Example commands: DupLoss-2 can be applied to these test datasets using the following sample commands. These commands assume that the testData directory is located in the directory containing the DupLoss-2 executable.

```
./DupLoss-2.linux -i testData/vertebrates.newick

./DupLoss-2.linux --seed 123 -i testData/vertebrates.newick

./DupLoss-2.linux --quiet -i testData/vertebrates.newick
```

```
./DupLoss-2.linux -i testData/vertebrates.newick -o SpeciesTree_1.txt

./DupLoss-2.linux -i testData/vertebrates.unrooted.newick

./DupLoss-2.linux -i testData/vertebrates.weighted.newick
```

Using the MultiRunScript.py script

The MultiRunScript.py script makes it easy to execute multiple runs of DupLoss-2 on the same input dataset. The script also guarantees that each run of DupLoss-2 uses a different random seed. The script can be invoked as follows:

python MultiRunScript.py <num_executions> <Input file name> <Output file name prefix> <optional random seed>

Note the three required command line arguments (number of executions, input file name, and output file name prefix) and one optional command line argument (user specified random seed). The script assumes that the DupLoss-2 executable is present in the same directory as the script, and named either DupLoss-2.linux or DupLoss-2.mac depending on the operating system. For example, this script may be used to analyze one of the test datasets as follows:

```
 python MultiRunScript.py 10 testData/vertebrates.newick SpeciesTree
```

Executing the above will result in 10 output files named, SpeciesTree_1, SpeciesTree_2,… SpeciesTree_10, containing the species trees found by the 10 executions of DupLoss-2.

Users familiar with Python programing may be able to further adapt the script for their exact needs.

Notes and recommendations

Due to the randomized nature of the search heuristic, we recommend executing multiple runs (say 10) of DupLoss-2 on the dataset being analyzed, with each run utilizing a different random seed. Once multiple species trees have been estimated, users can either compute a strict consensus of all the estimated species trees, or of only the species trees with lowest (weighted) reconciliation cost.

If a user specified random seed is provided, either to MultiRunScript.py or directly to DupLoss-2, then that analysis can be exactly recreated by using the same seed again. If a user specified random seed is not provided, then a random seed is internally generated based on the system time.

By default, DupLoss-2 generates processing output, such as the status of the search, current reconciliation cost, etc., and displays it on the standard output. If needed, users can disable such processing output by using the “--quiet” command line option. The script MultiRunScript.py uses this quiet option when executing multiple runs of DupLoss-2.

If desired, DupLoss-2 can output the reconciliation cost of each input gene tree with the estimated species tree. This may be useful, for example, to identify gene trees have very high reconciliation costs and to exclude them from the analysis. Users can use the “--genetrees” command line option to tell DupLoss-2 to output such individual gene tree reconciliation costs after the species tree.

Finally, DupLoss-2 provides an option (--limit) to first restrict the species tree to only the leaf set of each individual gene tree when evaluating that species tree and computing its total reconciliation cost. Such a reconciliation cost was used in iGTP-DupLoss, the previous version of DupLoss-2, and was intended to account for incomplete gene sampling. However, we have since found that restricting the species tree in this way can negatively impact the accuracy of the computed species tree. In fact, in our experiments, DupLoss-2 shows a 30% reduction in species tree reconstruction error compared to iGTP-DupLoss-2, even in the presence of incomplete gene sampling. We therefore caution against utilizing this option. Further details appear in the DupLoss-2 manuscript cited on the first page of this user manual.

Contact information

Please email Mukul Bansal (mukul.bansal@uconn.edu) if you have any questions about using DupLoss-2, suggestions, or concerns.