DupliPHY-ML - User Guide

D.P. Money

Contents

1 Change log

Version 1.3 Adds a birth-death-innovation-extinction model.

Version 1.2 Allows fixed branch lengths. Adds capability to deal with large trees. Adds a birth-death model with no zero state (e-mail author at daniel_money@ku.edu) for more information).

Version 1.1 Adds calculation of mean posterior rates for rates across families models.

Version 1.0 Initial release.

2 License

DupliPHY is licensed under GPL v3 (see gpl.txt for more information).

3 Citation

If you use DupliPHY please cite [1].

4 Overview

Recent large-scale studies of individuals within a population have demonstrated that there is wide-spread variation in copy number in many gene families. In addition, there is increasing evidence that the variation in gene copy number can give rise to substantial phenotypic effects. In some cases these variations have been shown to be adaptive. These observations show that a full understanding of the evolution of biological function requires an understanding of gene gain and gene loss. Accurate, robust evolutionary models of gain and loss events are, therefore, required.
We have developed weighted parsimony and maximum likelihood methods for inferring gain and loss events[1]. These methods have been tested on a range of simulated data and *Drosophila* data. We have shown that maximum likelihood and weighted parsimony have similar accuracy for reconstructing the ancestral state. For ancestral reconstruction we recommend weighted parsimony because it has similar accuracy to maximum likelihood, but is much faster.

5 Installation

DupliPHY-ML has been implemented in Java 1.6 and is distributed as an executable jar file. It is based on the GeLL library (`http://phylo.bio.ku.edu/GeLL`) and as such needs `GeLL.jar` to be in the same directory as `DupliphlyML.jar`. Both of these jar files are included in the distribution. As such there is no need for any complex installation, if java (version >= 1.6) is installed the program should run on any platform.
To check java is installed on your system type `java -version` at the command prompt. If an error message is displayed you can download and install java (version >= 1.6) from the Oracle website (`http://www.oracle.com/us/technologies/java/overview/index.html`).
Once java is installed simply download the DupliPHY jar file and follow the commands outlined in section 8.

6 Methods

DupliPHY-ML uses maximum likelihood to infer branch lengths and parameters [2], including accounting for unobservable states [3]. Standard numerical optimisation techniques were used to sequentially optimise each parameter in turn until no improvement in likelihood is found. To infer ancestral states we use the joint ancestral reconstruction method [4], where neccessary using the branch-and-bound method [5].

7 Input data

DupliPHY-ML takes as input a family file and a tree file.

7.1 Family file

The family file is a tab delimited file containing a header line and then a line per family. The header line lists the species in the analysis. Each subsequent line indicates the number of members of that family present in each species. The first column of the file is reserved for a family ID.

```
FAMILY    dana    dere    dgri    dmel    dmoj    dpse    dsim    dvir    dyak
Fam1    1    1    2    2    1    1    1    1    2
```

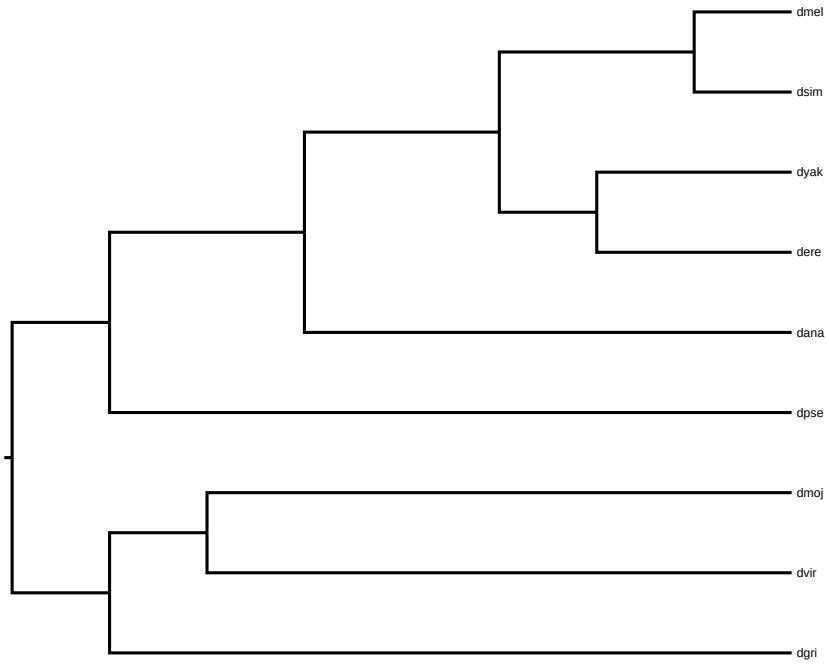

Figure 1: Phylogenetic tree of 9 *Drosophila* species.

```
Fam2    8    7    6    5    6    7    2    7    8
```

7.2 Tree file

DupliPHY-ML needs a phylogeny in order to infer the ancestral gene family sizes (Figure 1). DupliPHY-ML accepts newick formatted trees. The trees can either have labels on the internal nodes or these labels can be omitted. If no internal nodes labels are present DupliPHY will add these automatically and they will be included in the output. Any of the following trees will be accepted by DupliPHY. Branch lengths will be ignored unless the fixed branches option is chosen (see Section 8 below). If this option is chosen branch lengths must be included. **NB. All species in the phylogenetic tree must be present in the gene family file.**

```
((((( dmel:1.0, dsim:1.0) A:2.0,( dyak:2.0, dere:2.0) B:1.0) C:2.0, dana:5.0) E:2.0,
dpse:7.0) F:1.0,(( dmoj:6.0, dvir:6.0) G:1.0, dgri:7.0) H:1.0) Root:0.0;

((((( dmel,dsim),(dyak,dere)),dana),dpse),((dmoj,dvir) dgri));

(((((dmel:1.0, dsim:1.0) :1.0,(dyak:1.0, dere:1.0) :1.0) :1.0, dana:1.0) :1.0, dpse:1.0)
:1.0,((dmoj:1.0, dvir:1.0) :1.0, dgri:1.0) :1.0);
```

8 Running DupliPHY

DupliPHY is a command line tool which can be called in two different ways. Described first is the control file method of running DupliphyML. This was not avaliable in early versions of DupliphlyML but is now the preferred method as it allows the control of options not avaliable using the 'command line' method. The 'command line' method is described second although users should note this is now depreciated and only inlcuded for backwards compitability.

8.1 Control File Method

If a control file is being used then the command line is:

```
java -jar DupliphlyML.jar <control file>
```

control file The control file (described below). e.g. control.txt

The control file is a plain text file with one option per line. The option name is seperated from the option's value by a tab e.g.

```
Families    families.dat
Tree    tree.dat
Output    results
Model    BDI
```

Avaliable options are shown in Table 1. All paths can be either an absolute or relative path. The order of options is not important.

To run DupliPHY-ML on the supplied example data use:

```
java -jar control.txt
```

This will generate the results files with the prefix myResults.

8.2 Command Line Method (Depreceated)

If the 'command line' option is being used then the command line is:

```
java -jar DupliphyML.jar <Families> <Model> <Tree> <Output> <Number>
<Missing>
```

Options and there possible values are the same as for the control file. Newer options are not avaliable and this method of calling DupliphyML is only included for backwards compitability. It's use should be considered depreceated and it may not be supported in future versions.

9 Outputs

Running DupliPHY will generate three or four output files where <Output> is the input parameter:

<Output>_desc.txt A tab delimited gene family file showing the extant gene family sizes (as provided by the user) and the ancestral reconstruction at the internal nodes of the tree.

Option	Required?	Values
Families	Yes	The path of the families file.
Tree	Yes	The path of the trees file.
Output	Yes	The prefix to be used for the results files.
Model	Yes	The model to use. Options are: `Parsimony` - Use the single rate model as described in [1] `BDI` - Use the standard BDI model (also described in [1]) `BD_Nozero` - Use a birth-death model with no zero state. Please contact the author for more information. `BDIE` - Use the standard BDIE (in press) Any of these models can be used with gamma-distributed rates across families [6] by adding `+G` to the end of the appropiate model name.
Number	No	The maximum family size to be modelled. If not specified defaults to 75.
Missing	No	The path of a file that lists unobserved patterns of families, for example the case where every family has zero copies. This file is in the same format as the families file. If this option is not passed then it is assumed that all pattern are observable.
Optimizer	No	The optimization method to be used to optimize the parameters. Options are: `Golden Section` - [7] `Nelder-Mead` - [8] `Conjugate Gradient` - based on the algorithm used in BEAST [9] Defaults to `Golden Section` if not specified.
FixedBranch	No	Whether the branch lengths will be fixed to those specified in the tree file. Suggested values are `True` and `False` although any text beginning with `t` or `T` will be intpretted as true and all other text as false. Defaults to false.
Real	No	The type of real to use. Options are: `Standard` - Use the standard Java double. This is quicker but may cause underflow errors on large trees. `Small` - Use small doubles. This will allow calculations on large trees but with a significant increase in run time. Deafults to `Standard`
Ancestral	No	The type of ancestral reconstruction to perform. Options are: `Joint` - Use joint ancestral reconstuction [4, 5]. `Marginal` - Use marginal ancestral reconstruction [10]. Defaults to `Joint`
Matrix	No	Whether to output the rate matrix and stationary frequencies. Suggested values are `True` and `False` although any text beginning with `t` or `T` will be intpretted as true and all other text as false. Deafults to false.

Table 1: The avaliable DupliphyML options

<**Output**>**_tree.ph** The estimated phylogenetic tree.

<**Output**>**_params.txt** A file listing any parameters that have been estimated (either model parameters or branch lengths). Parameters are listed one per line, name first followed by it's value with the name and value being tab seperated.

<**Output**>**_rate.txt** The mean posterior rate for each family (REF needed). Families are listed one per name with their ID followed by the rate, again tab seperated. This file is only generated when a `+G` model is used.

<**Output**>**_ancprob.txt** File describing the probability of ancestral states. WRITE MORE!

<**Output**>**_matrix.txt** File containing the scaled rate matrix and the stationary frequencies. WRITE MORE!

10 Contact

DupliPHY-ML is mianted by Daniel Money who can be contacted at daniel_money@ku.edu.

Bibliography

[1] R. M. Ames, D. Money, V. P. Ghatge, S. Whelan, and S. C. Lovell, "Determining the evolutionary history of gene families," *Bioinformatics*, vol. 28, pp. 48–55, Jan. 2012.

[2] J. Felsenstein, *Inferring phylogenies*. Sinauer Associates Sunderland, Mass., USA, 2003.

[3] J. Felsenstein, "Phylogenies from restriction sites: A maximum-likelihood approach," *Evolution*, vol. 46, pp. 159–173, Feb. 1992.

[4] T. Pupko, I. Pe, R. Shamir, and D. Graur, "A fast algorithm for joint reconstruction of ancestral amino acid sequences," *Molecular Biology and Evolution*, vol. 17, pp. 890 –896, June 2000.

[5] T. Pupko, I. Pe'er, M. Hasegawa, D. Graur, and N. Friedman, "A branch-and-bound algorithm for the inference of ancestral amino-acid sequences when the replacement rate varies among sites: Application to the evolution of five gene families," *Bioinformatics*, vol. 18, no. 8, pp. 1116 –1123, 2002.

[6] Z. Yang, "Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites," *Molecular Biology and Evolution*, vol. 10, pp. 1396–1401, Nov. 1993.

[7] J. Kiefer, "Sequential minimax search for a maximum," *Proceedings of the American Mathematical Society*, vol. 4, pp. 502–506, June 1953.

[8] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, pp. 308–313, Jan. 1965.

[9] A. J. Drummond, M. A. Suchard, D. Xie, and A. Rambaut, "Bayesian phylogenetics with BEAUti and the BEAST 1.7," *Molecular Biology and Evolution*, Feb. 2012.

[10] Z. Yang, S. Kumar, and M. Nei, "A new method of inference of ancestral nucleotide and amino acid sequences," *Genetics*, vol. 141, pp. 1641–1650, Dec. 1995.