BadiRate

Estimating Family Turnover Rates by Likelihood-Based Methods

Version 1.35

Pablo Librado
Filipe G. Vieira
Julio Rozas

Departament de Genètica

UNIVERSITAT DE BARCELONA

http://www.ub.es/softevol/badirate

Last updated: November 4, 2011

1 Overview

It is generally accepted that gene and genome duplications are a major evolutionary mechanism for generating functional innovation [1,2]. The increasing availability of closely related genome sequences allows an accurate analysis of gene family evolution [3,4,5]. Such studies have shown that most families are highly dynamic and evolve under a birth-and-death (BD) process [6]. Indeed, the comprehensive analysis of gene gains and losses can provide helpful insight into the role of natural selection and adaptation in shaping gene family size variation. In this context, analyses of fast gene family expansions and contractions are of exceptional interest [7].

Even with complete genome information from recently diverged species, reliable estimation of BD rates is not an easy task. Currently, gene gain and loss events can be estimated either from the number of gene family members in the extant species of a phylogeny [8], or via gene tree/species tree reconciliation [9,10]. The latter methods, however, have important limitations [11] such as their dependence on correct gene trees and the true species tree, as well as incomplete lineage sorting problems. Although there have been some improvements implemented to minimize the gene-tree uncertainty by taking into account clade support values, branch lengths or synteny information [12], gene tree/species tree reconciliation is not well suited to conduct statistical hypothesis testing, and as such, it is not very useful for identifying rapid gene family expansions or contractions.

The stochastic birth-and-death models have a different number of applications in biology [13], such as the population growth analysis, the phylogenomic inference and the estimation of the gene family turnover rates [14,15]. For instance, the stochastic birth-and-death model (BDM) [16] implemented in the programs CAFE [17] and BEGFE [18] can circumvent some of the above-mentioned limitations. It ,has, however, some known drawbacks. First, it assumes equal birth-and-death rates, an assumption that may not hold. Second, because duplications from zero ancestral genes are not possible (zero is an absorbing state in the probabilistic BDM), CAFE cannot handle gene families without a single element in the phylogenetic root. These assumptions can therefore bias the estimates of both the number of members in internal phylogenetic nodes as well as the birth-and-death rate.

Two recently developed computer programs, GLOOME [19] and Count [20], overcome some of these difficulties by implementing stochastic population models that relax the assumption of equal birth-and-death rates, and where zero is not an absorbing state. Nevertheless, GLOOME can only model presence/absence of family members instead of size changes, whereas Count assumes independent turnover rates for all lineages, which precludes testing biological relevant hypothesis such as linage-specific accelerations.

Here, we describe BadiRate, a new software tool to estimate family turnover rates, as well as the number of elements in internal phylogenetic nodes, by maximum likelihood (ML), maximum a posteriori (MAP) and parsimony. The software implements two stochastic population models within a phylogenetic context, the Gain-and-Death (GD) and the Birth-Death-and-Innovation (BDI, also known as Birth-Death-and-Immigration), which allows modeling families of diverse functional elements, such as microRNAs, *cis*-acting regulatory elements or coding-protein genes. Additionally, these models provide the appropriate statistical framework for hypothesis testing, such as family expansions or contractions in specific lineages.

Authors

Pablo Librado	plibrado@ub.edu
Filipe G. Vieira	fgarret@ub.edu
Julio Rozas	jrozas@ub.edu

BadiRate references

Librado P, Vieira FG, Rozas J
BadiRate: Estimating Family Turnover Rates by Likelihood-Based Methods (submitted)

BadiRate Web Site

http://www.ub.es/softevol/badirate

2 Installation

BadiRate is distributed as a multiplatform Perl script (**BadiRate.pl**). Hence, it does not need to be installed or compiled. Nevertheless, it requires to have installed the following resources:

Perl programming language (version 5.10 or higher)
gcc compiler
nmake or dmake program (only for Windows)

If you have these programming tools already installed, skip steps 1, 2 and 3 of the following instructions:

1) Perl programming language

Windows

We strongly recommend you to download and install the Strawberry Perl programming language from:
http://strawberryperl.com/
The Strawberry Perl package includes MinGW (a gcc compiler) and the dmake tools.

Alternatively, you can also install ActivePerl from:
http://www.activestate.com/activeperl
but you will need to install a gcc compiler and a make tool (nmake or dmake) manually. Moreover, depending on your Perl version you might need to install a specific gcc compiler or make tool.

Unix-based platforms

You can download and follow installation instructions from http://www.perl.org/get.html.

2) gcc compiler

MacOS X

Download the last version of Xcode from http://connect.apple.com/ (you will need to create an Apple Developer Connection account).

3) Decompress BadiRate package

Windows

You can use WinZip, WinACE or any other similar tool. It is very important to decompress it on a folder without blank spaces in its name.

Unix-based platforms

To decompress the BadiRate package, type:

```
tar xzf badirate-version.tar.gz
```

The BadiRate package contains the following folders:

> **lib**: contains all library dependencies (BioPerl and CPAN modules) required to run BadiRate. It should be placed on the same folder as the **BadiRate.pl** file.
>
> **examples**: includes data file examples: Newick format (***.nwk**), family size (***.tsv**), control (***.ctl**) and output (***.out**) files.

3 BadiRate Models and Methods

We have used a three-word code to designate the methods implemented in `BadiRate`:

TurnoverRates-BranchModel-EstimationProcedure

TurnoverRates (see **`–rmodel`** option)
`BadiRate` allows estimating the following gene (or other functional elements) family turnover rates:

- BDI: Birth (β), Death (d) and Innovation (i) rates
- LI: Lambda (λ) and Innovation (i) rates
- GD; Gain (γ) and Death (δ) rates
- BD: Birth (β) and Death (d) rates
- L: Lambda (λ) rates

BranchModel (see **–`bmodel`** option)
Badi`Rate` can model gene family evolution assuming that phylogenetic branches (lineages) might have different turnover rates (see the **`-bmodel`** option). There are three major branch models:

- Free Rates (FR) model: Each branch has its own turnover rates.
- Global Rates (GR) model: All branches have the same turnover rates.
- Branch-specific Rates (BR) model: Any other branch model specifying different turnover rates in particular branches.

EstimationProcedure (see **`–ep`** option)
Denotes the statistical method used to estimate the turnover rates. The methods are the following:

- ML: Maximum Likelihood.
- MAP: Maximum A Posteriori.
- CML: Counting the gain/loss events from the number of members in internal nodes inferred by ML.
- CMAP: Counting the gain/loss events from the number of members in internal nodes inferred by MAP.
- CWP: Counting the gain/loss events from the number of members in internal nodes inferred by the Wagner parsimony algorithm.
- CSP: Counting the gain/loss events from the number of members in internal nodes inferred by a modification of the Sankoff algorithm (this algorithm take into account both the branch length and the uncertainty in the number of family members in the internal nodes).

Under the counting-based methods (CML, CMAP, CWP and CSP), the inferred number of events will be used to estimate the turnover rates as in equation 1 and 2 of Vieira et al. (2007). It is worth noting that these counting methods likely underestimate the family turnover rates since some gain and loss events can mask each other (and therefore they are not considered). For these methods, it is very convenient to separate the gene family members into its component orthologous groups (or subfamilies) to obtain better turnover estimates (see **`–family`** option).

Some Examples

BDI-GR-ML. Estimation of the β, δ and ι rates under the BDI stochastic model, and assuming that all branches have the same turnover rates (GR). The turnover rates are estimated by ML. This is one of the methods/models utilized in the computer simulations described in the Librado et al. (2011) paper.

LI-FR-ML. Estimation of the λ and ι rates under the LI stochastic model, and assuming that each branch has its specific turnover rates (FR). The turnover rates are estimated by ML.

GD-GR-ML. Estimation of the γ and δ rates under the GD stochastic model, and assuming that all branches have the same turnover rates (GR). The turnover rates are estimated by ML.

BDI-FR-CML. Estimation of the β, δ and ι rates under the BDI stochastic model, and assuming that each phylogenetic lineage has its specific turnover rates. The BDI rates are estimated by counting the gain and loss events from the number of family members in the internal nodes (estimated by ML), and applying the equation 1 and 2 of Vieira et al. (2007). This is the method used in Vieira and Rozas (2011) to estimate the OBP and CSP birth and death rates.

BDI-FR-CWP. Gene family analysis under parsimony. The β, δ and ι rates are calculated by counting gain and loss events from the number of family members in the internal nodes (inferred by the Wagner parsimony algorithm) and applying the equation 1 and 2 of Vieira et al. (2007).

4 Input Data Files

BadiRate requires two input data files (in plain text):

- The **species phylogenetic tree** (ultrametric, rooted tree in Newick format).
- The **size of each family (or subfamily) in extant species** (tabulated data).

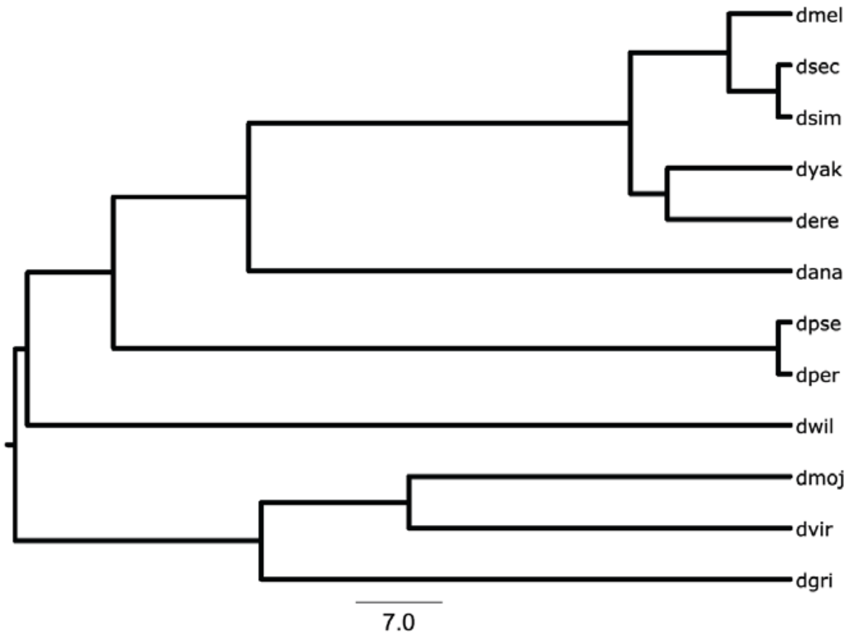
Newick tree file (*.nwk)

All files in **examples/*.nwk** are examples of the Newick format. To know about Newick syntaxis, please read:
http://evolution.genetics.washington.edu/phylip/newicktree.html

For instance, the file **examples/droso.12sp.tamura.nwk** contains the following information:

```
((((((dmel:5,(dsec:1,dsim:1):4):8,(dyak:10,dere:10):3):31,dana:44):11,(dpse:1,dper:1):54):7,dw
il:62):1,((dmoj:31,dvir:31):12,dgri:43):20);
```

that can be visualized as a phylogenetic tree using a tree editor program -such as FigTree [21], MEGA [22] or TreeView [23]:



Family size file (*.tsv)

All files in **examples/*.tsv** are examples of this format. These files contain tabulated data including a header (first line) with the label "FAM_ID" (or any other word) and the species ids. Subsequent rows specify the name and size (number of members) for each family -or subfamily- in the extant species (one family or subfamily per row). Note that species ids should correspond with those indicated in the Newick file.

BadiRate allows estimating family turnover rates, and ancestral family sizes, for a specific set of gene families including:

1) A set (or subset) of gene families

The file `examples/4FAMs.12sp.tsv` contains data from four gene families in the 12 sequenced *Drosophila* species

```
FAM_ID     dmel   dsim   dsec   dyak   dere   dana   dpse   dper   dwil   dmoj   dgri   dvir
OBPs       52     52     51     55     50     50     45     45     62     43     46     41
WD-40      42     34     38     37     37     40     34     54     37     40     55     41
MCarr      33     31     33     39     34     43     28     39     36     32     37     36
Lambik     17     19     17     12     16     13     15     12     15     15     14     16
```

where `OBPs` is name of the first gene family, and 52 is the number of `OBPs` members *in D. melanogaster.*

2) A single gene family

The file `examples/obp_all.12sp.tsv` contains data from a single gene family (OBP gene family)

```
FAM_ID     dmel   dsim   dsec   dyak   dere   dana   dpse   dper   dwil   dmoj   dgri   dvir
OBPs       52     52     51     55     50     50     45     45     62     43     46     41
```

3) A single gene family with separate information of the component subfamilies

The file `examples/obp_sub.12sp.tsv` contains information from the 56 subfamilies of the OBP gene family

```
OG         dmel   dsim   dsec   dyak   dere   dana   dpse   dper   dwil   dmoj   dvir   dgri
OG_0       1      1      1      2      1      1      1      1      1      1      1      1
OG_1       1      1      1      1      1      1      1      1      1      1      1      1
OG_2       2      2      2      2      2      2      2      2      2      2      2      2
OG_3       1      1      1      1      1      1      1      1      2      1      1      1
…
OG_55      1      1      1      1      1      1      1      1      1      1      1      1
```

Prior File (*.prior)

All files in `examples/*.prior` are examples of this format. The file should be a tabulated text file with the following fields:

<branch_class>   <parameter>   <prior distribution>   <distribution parameter>

For instance, the file `examples/low_innovation.prior`:

```
0  innovation       gamma     0.0
```

defines a Gamma prior with a shape value of 0.0 for the innovation rate of the branch class 0 (see **`bmodel`** option).

5 Running BadiRate

To run `BadiRate.pl`, type on the terminal:

```
perl BadiRate.pl [options|control_file]
```

where **`[options|control_file]`** indicates that you can use some of the `BadiRate` options (listed below), or just a control file (i.e. a file specifying all needed `BadiRate` options):

Option	Description
-anc	Report family sizes at internal nodes, and the minimum number of gains/losses in each lineage
-bmodel	Run global, free or branch specific models
-family	Family turnover analysis using information of the component orthologous groups (subfamilies)
-ep	Define the estimation procedure
-help	Display this help
-n_max_int	Modify the maximum number of family members in the internal phylogenetic nodes
-out	Set the output file
-outlier	Report families that likely have not evolved under the estimated stochastic process
-print_ids	Display nodes ids in a Newick format
-priorfile	Set the prior file
-rmodel	Family turnover rates to be estimated
-root_dist	Estimation method for the root *a priori* distribution
-seed	Specify the seed of the pseudo-random number generator
-sizefile	Family Size File
-start_val	Starting values for the likelihood methods
-treefile	Phylogenetic tree in Newick format
-unobs	Correct the likelihood for families absent in all extant species
-version	Report the BadiRate version

Options for the running command

The basic command to run `BadiRate` is:

```
perl BadiRate.pl -treefile NEWICK_FILE -sizefile FAMILY_SIZE_FILE
[add_options] -out OUTPUT_FILE
```

where **`BadiRate.pl`** is the name of the program, **`NEWICK_FILE`** is the file containing the phylogenetic tree in Newick format, **`FAMILY_SIZE_FILE`** is the tabulated data file with the number of family (or subfamily) members in extant taxa, and **`OUTPUT_FILE`** is the output data file. In **`[add_options]`**, you can specify a number of additional options (see below).

List of Options:

`anc:` This option reports the number of members at internal phylogenetic nodes for each family, and the total minimum number of gains/losses per lineage; the latter number is calculated as the difference between the number of genes in the ancestral and derived phylogenetic nodes. If a particular data set contains more than one gene family (or subfamily), the number of events will be calculated separately for each one, and then summed across all encompassing gene families (or subfamilies).

Running Example (model: BDI-GR-ML):

```
perl BadiRate.pl –treefile examples/droso.6sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv –anc
```

- **bmodel:** Use this option to run BadiRate under different branch specific models. By default BadiRate assumes that all branches have the same turnover rates (GR). For instance, to run BadiRate under the FR branch model, type:

Running Example (model: BDI-FR-ML):

```
perl BadiRate.pl –treefile examples/droso.12sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv -bmodel FR
```

To apply a Branch-specific Rates (BR) model, run BadiRate twice; the first time to display branch **ids** (see below), and the second one to specify the phylogenetic branches evolving under particular turnover rates. For example, to choose a model with three different turnover rates (in the *D. willistoni* lineage, in the *D. sechellia* and *D. simulans* lineages, and in the rest of branches), we should first run:

```
perl BadiRate.pl –treefile examples/droso.12sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv –print_ids
```

BadiRate will report the following output:

```
((((((dmel_1:5,(dsec_2:1,dsim_3:1)4:4)5:8,(dyak_6:10,dere_7:10)8:3)9:31,dana_10:44)11:11,(dpse
_12:1,dper_13:1)14:54)15:7,dwil_16:62)17:1,((dmoj_18:31,dvir_19:31)20:12,dgri_21:43)22:20)23;
```

The above Newick information (see the "print_ids" option) reports the **id** for each phylogenetic branch. In particular, the **ids** for the *D. willistoni*, *D. sechellia* and *D. simulans* lineages are **17->16**, **4->2** and **4->3**, respectively. We can then run BadiRate to estimate family turnover rates as:

Running Example (model: BDI-BR-ML):

```
perl BadiRate.pl –treefile examples/droso.12sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv -bmodel "17->16_4->2:4->3"
```

where branches joined with a colon '**:**' (**4->2** and **4->3)** have the same family turnover rates, and those joined with a '_' have different family turnover rates (**17->16**). The rest of the branches (not specified on the command) are assumed to belong to another branch class. Note that the branch information should be between quotes (**"**).

- **ep:** Use this option to define the procedure to estimate the turnover rates specified in the **-rmodel** option. By default, BadiRate uses the ML approach.(see the "BadiRate Models and Methods" section)

Running Example (model: BDI-GR-ML):

```
perl BadiRate.pl –treefile examples/droso.6sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv –ep ML
```

Running Example (model: BDI-FR-ML):

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -ep ML -bmodel FR
```

Running Example (model: BDI-GR-CML):

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_sub.12sp.tsv -ep CML -family
```

Running Example (model: BDI-GR-CSP):

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_sub.12sp.tsv -ep CSP -family
```

- **family:** Use this option to estimate a gene family turnover rates using information of the component subfamilies (see "Input Data Files", "BadiRate Models and Methods" and "Examples" sections). First, BadiRate will estimate,the number of family members in the internal phylogenetic nodes. Then, from this information, BadiRate will calculate family turnover rates as in [12].

Running Example (model: BDI-GR-CML):

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_sub.12sp.tsv -ep CML -family -unobs
```

Running Example (model: BDI-GR-CWP):

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_sub.12sp.tsv -ep CWP -family
```

- **help:** Report this help, with the BadiRate options.

Running Example:

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -help
```

- **n_max_int:** Use this option to modify the maximum number (upper bound) of family members in the internal phylogenetic nodes (*max_int*). The upper bound is defined as:

$$max\_int = 2max\_ext + n\_max\_int,$$

where *max_ext* is the maximum number of family members in the external phylogenetic nodes, and *n_max_int* is a constant value (by default is 10). The user might change *n_max_int* to 50 as follows:

Running Example:

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -n_max_int 50
```

- **out:** Set the output file (by default, displayed on the screen).

Running Example:

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -out output_example.bd
```

- **outlier:** Given the estimated evolutionary family rates, and the number of family members in internal nodes, `BadiRate` will also report the families that likely have not evolved under the estimated turnover rates.

Running Example:

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -outlier
```

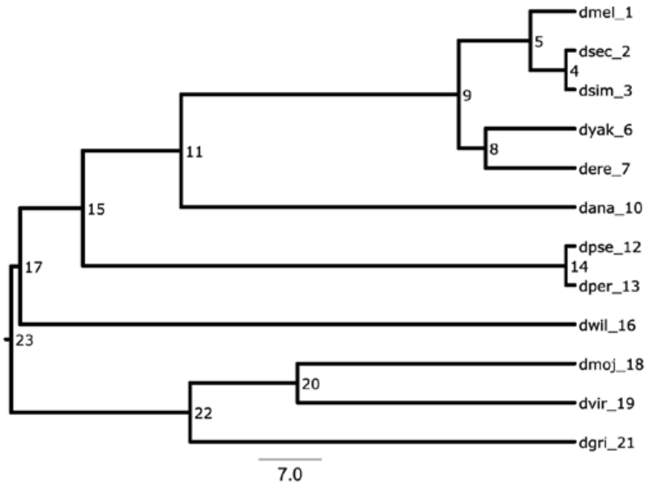
- **print_ids:** Use this option to display the node ids in a Newick format (`BadiRate` will report all results using these **ids**).

Running Example:

```
perl BadiRate.pl -treefile examples/droso.12sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -print_ids
```

will report the following output:

```
##NODES-INTERNAL_ID ASSOCIATION
((((((dmel_1:5,(dsec_2:1,dsim_3:1)4:4)5:8,(dyak_6:10,dere_7:10)8:3)9:31,dana_10:44)11:11,(dpse
_12:1,dper_13:1)14:54)15:7,dwil_16:62)17:1,((dmoj_18:31,dvir_19:31)20:12,dgri_21:43)22:20)23;
```



The node **id** for `dmel` (*D. melanogaster*) is 1, whereas the **id** for its most recent ancestor is 5. Likewise, the **id** of the branch leading from node 5 to *D. melanogaster* is denoted as 5->1.

- **priorfile:** Use this option to specify the file with the *a priori* probability distribution for the family turnover rates (the prior should be a gamma distribution).

Running Example (model: BDI-GR-MAP):

```
perl BadiRate.pl -treefile examples/droso.12sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -ep MAP -priorfile
examples/low_innovation.prior
```

- **rmodel** Specify the family turnover rates to be estimated. By default, `BadiRate` will estimate BDI rates.

Running Example (model: LI-GR-CSP):

```
perl BadiRate.pl –treefile examples/droso.6sp.tamura.nwk –sizefile
examples/obp_sub.12sp.tsv –ep CSP –rmodel LI -family
```

Running Example (model: GD-GR-ML):

```
perl BadiRate.pl –treefile examples/droso.6sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv –rmodel GD
```

Running Example (model: L-GR-ML):

```
perl BadiRate.pl –treefile examples/droso.6sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv –rmodel L
```

Running Example (model: BD-GR-ML):

```
perl BadiRate.pl –treefile examples/droso.6sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv –rmodel BD
```

root_dist: `BadiRate` models the *a priori* family size distribution for the phylogenetic root node using a Poisson or a Negative Binomial distribution.

To estimate the Poisson parameter by parsimony (one parameter value for each family), use the **-root_dist 0** value.

Running Example (model: BDI-GR-ML):

```
perl BadiRate.pl –treefile examples/droso.6sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv –root_dist 0
```

To co-estimate the Poisson parameter by ML (the same parameter value for all the families), use the **-root_dist 1** value (default value):

Running Example (model: BDI-GR-ML):

```
perl BadiRate.pl –treefile examples/droso.6sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv –root_dist 1
```

To co-estimate the Poisson parameter by ML (one parameter value for each family), use the **-root_dist 2** value:

Running Example (model: BDI-GR-ML):

```
perl BadiRate.pl –treefile examples/droso.6sp.tamura.nwk –sizefile
examples/obp_all.12sp.tsv –root_dist 2
```

To co-estimate the Negative Binomial parameters by ML (the same parameter value for all the families), use the **-root_dist 3** value:

Running Example (model: BDI-GR-ML):

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -root_dist 3
```

Finally, to co-estimate the Negative Binomial parameters by ML (one parameter value for each family), use the **`-root_dist 4`** value:

`Running Example (model: BDI-GR-ML):`

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -root_dist 4
```

- **`seed:`** Set the seed of the pseudo-random number generator.

`Running Example:`

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -seed 1238912789
```

- **`sizefile:`** Set the file (*.tsv) with information of the number of members for each family (or subfamily) in extant taxa.

`Running Example:`

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv
```

- **`start_val:`** Use this option to change the starting values for the likelihood methods. By default, BadiRate uses as starting values the family turnover rates and the number of family members in the internal phylogenetic nodes inferred by parsimony. To use different initial values (useful to avoid local optima), use the **`-start_val 1`** option value.

`Running Example:`

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -start_val 1
```

- **`treefile:`** Set the file with the phylogenetic tree of the studied species in Newick format.

`Running Example:`

```
perl BadiRate.pl -treefile examples/droso.12sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv
```

- **`unobs:`** Use this option to correct the likelihood for the families absent in all extant taxa (see the Supplementary Material of the `BadiRate` publication).

`Running Example:`

```
perl BadiRate.pl -treefile examples/droso.12sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -unobs
```

- **version:** Report the BadiRate version.

Running Example:

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -version
```

Basic running commands

Use the following commands to run the examples on the "BadiRate Models and Methods" section

BDI-GR-ML

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv
```

LI-FR-ML

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -rmodel LI -bmodel FR
```

GD-GR-ML

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -rmodel GD
```

BDI-FR-CML

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_sub.12sp.tsv -unobs -bmodel FR -ep CML -family
```

BDI-FR-CWP

```
perl BadiRate.pl -treefile examples/droso.6sp.tamura.nwk -sizefile
examples/obp_sub.12sp.tsv -bmodel FR -ep CWP -family
```

Note that, in the last two examples, we are analyzing the **examples/obp_sub.12sp.tsv** *sizefile*, which contains the orthologous groups comprising the OBP gene family; *i.e.* the **-family** option must be enabled.

Options using a Control File

BadiRate can also be run through a control file; i.e., a file specifying all needed BadiRate options:

Running Example (model: GD-BR-ML):

```
perl BadiRate.pl CONTROL_FILE
```

where **CONTROL_FILE** is a file that includes all needed BadiRate options. For instance, the example control file **examples/miRNA_6rates.ctl** contains:

```
sizefile   = examples/miRNA.og.tsv
treefile   = examples/droso.12sp.tamura.nwk
seed       = 213231
print_ids  = 0
bmodel     = 17->16_23->17:17->15:15->11:11->9:9->5:5->1
anc        = 1
n_max_int  = 10
rmodel     = GD
outlier    = 1
priorfile  = 0
ep         = ML
version    = 0
help       = 0
unobs      = 1
root_dist  = 1
start_val  = 0
out        = STDOUT
```

where an option value of 0 (for instance, `print_ids = 0`) indicates that the option is disabled and *vice versa*. For instance, `priorfile = 0` and `anc = 1` indicates that no prior file will be used and that the `anc` option is activated, respectively. To use a prior file the path should be specified.

This control file can be run as:

```
perl BadiRate.pl examples/miRNA_6rates.ctl
```

Using the above control file example, `BadiRate` will report, from `examples/miRNA.og.tsv` data, using the `examples/droso.12sp.tamura.nwk` tree, and by ML (`ep = ML`):

(i) the family turnover rates
(ii) the gene families that have not likely evolved under the estimated stochastic Gain-and-Death process (`rmodel = GD` and `outlier = 1`)
(iii) the number of ancestral miRNA members (`anc = 1`)

`BadiRate` will use a BR model with three different classes of GD rates: one for the branch `17->16`, another for the branches `23->17` , `17->15` , `15->11` , `11->9` , `9->5` , `5->1`, and the third one for the other branches of the phylogeny. Please, note that the value for the **`bmodel`** is not written between quotes in the control file. The output results will be displayed on the screen (`out = STDOUT`).

The equivalent running command-line is:

```
perl BadiRate.pl –treefile examples/droso.12sp.tamura.nwk –sizefile
examples/miRNA.og.tsv –rmodel GD –bmodel “17->16_23->17:17->15:15-
>11:11->9:9->5:5->1” –outlier
```

6 BadiRate Units and Output

Units

The family turnover rates are expressed in the following units:

Turnover Rate	Units
`Gain (γ)`	Gains (births plus innovations) per million of years
`Innovation (ι)`	Innovations per million of years
`Birth (β)`	Births per gene and per million of years
`Death (δ)`	Deaths per gene and per million of years
`Lambda (λ)`	Births (or deaths) per gene and per million of years

Note that β, δ and λ are density dependent rates; the higher the number of members of a family in internal nodes, the higher the probability of a birth (duplication) or a death (deletion, pseudogenization) event.

Output

The output of `BadiRate` includes the following three outcomes:

1) The node ids

```
##NODES-INTERNAL_ID ASSOCIATION
((((((dmel_1:5,(dsec_2:1,dsim_3:1)4:4)5:8,(dyak_6:10,dere_7:10)8:3)9:31,dana_10:44)11:11,(dpse_
12:1,dper_13:1)14:54)15:7,dwil_16:62)17:1,((dmoj_18:31,dvir_19:31)20:12,dgri_21:43)22:20)23;
```

2) The user input with the specified options

```
INPUT
      Execution date: 2011-09-19 11:24:02
      Version: 1.35.00
      root_dist = 1
      sizefile = examples/miRNA.og.tsv
      treefile = examples/droso.12sp.tamura.nwk
      priorfile =
      n_max_int = 10
      outlier = 1
      seed = 213231
      unobs = 1
      rmodel = GD
      ep = ML
      help = 0
      out = STDOUT
      anc = 1
      version = 0
      print_ids = 0
      bmodel=
             4->2   2
             5->1   1
             9->5   1
             17->15 1
             23->22 2
             17->16 0
             4->3   2
             20->18 2
             15->14 2
             23->17 1
             9->8   2
             15->11 1
             11->10 2
             14->13 2
             8->6   2
             14->12 2
             22->21 2
             11->9  1
             5->4   2
             20->19 2
             22->20 2
             8->7   2
      start_val = 0
      family = 0
END INPUT
```

3) Estimates of the family turnover rates, the ancestral family sizes, the minimum number of gains and losses per branch and, the outlier families

```
OUTPUT

      ##Family Turnover Rates
             #Likelihood: -748.906403572024
             #Branch_Group Gain   Death  Loss(approx from death)
             0      0.0012324     0.0003985     0.0003985
             1      0.0027364     0.0000000     0.0000000
             2      0.0005815     0.0011940     0.0011954

      ##Ancestral Family Size
             #FamilyAncestral Family Size Tree
             bantam
      ((((((dmel_1:5,(dsec_1:1,dsim_1:1)1:4)1:8,(dyak_1:10,dere_1:10)1:3)1:31,dana_1:44)1:11,(dp
se_1:1,dper_1:1)1:54)1:7,dwil_1:62)1:1,((dmoj_1:31,dvir_1:31)1:12,dgri_1:43)1:20)1;
             let-7
      ((((((dmel_1:5,(dsec_1:1,dsim_1:1)1:4)1:8,(dyak_1:10,dere_1:10)1:3)1:31,dana_1:44)1:11,(dp
se_1:1,dper_2:1)1:54)1:7,dwil_1:62)1:1,((dmoj_1:31,dvir_1:31)1:12,dgri_2:43)1:20)1;
             mir-1


               …
       Total Ancestral Size
       ((((((dmel_152:5,(dsec_149:1,dsim_143:1)150:4)150:8,(dyak_148:10,dere_142:10)142:3)142:3
1,dana_115:44)115:11,(dpse_106:1,dper_107:1)105:54)106:7,dwil_123:62)104:1,((dmoj_103:31,dvir_1
01:31)103:12,dgri_112:43)103:20)104;
      ##Minimum number of gains and losses per branch
             #BranchGains  Losses
             23->17 0      0
             17->15 2      0
             15->11 9      0
             11->9  27     0
             9->5   8      0
             5->1   2      0
             5->4   1      1
             4->2   1      2
             4->3   2      9
             9->8   0      0
             8->6   9      3
             8->7   4      4
             11->10 0      0
             15->14 2      3
             14->12 1      0
             14->13 5      3
             17->16 22     3
             23->22 0      1
             22->20 0      0
             20->18 1      1
             20->19 2      4
             22->21 11     2

      ##Outlier Families per Branch
             #Family       Branch P-valueFDR_P-value
             mir-964       17->16 0.0000000     0.0000000
             mir-10        8->6   0.0000164     0.0137339
             mir-125       14->13 0.0000002     0.0001881
             mir-959       17->16 0.0000000     0.0000309

      ##Execution time (seconds): 85
END OUTPUT
```

In the GD model, `BadiRate` also calculates the approximate loss rate (losses per million of years) from the death rate (deaths per gene and per million of years). Therefore, the gain and loss rates can be directly compared (both are expressed in the same units).

Likelihood Convergence Problems

1) Complex models or small data sets (i.e, those having too many parameters relative to the number of observations), may pose some convergence problems (it might yield several local maxima). If the Amoeba hill-climbing algorithm does not reach the convergence criterion after 150000 iterations, `BadiRate` will print the following message:

> **"Amoeba excedeed the maximum number of iterations. Check convergence (see -start_val option)!"**

However, some local maxima might also satisfy the convergence criterion. To examine such cases you can:

1.1) Run `BadiRate` using different starting values (see `–start_val` option). For instance, running:

```
perl BadiRate.pl -sizefile examples/obp_all.12sp.tsv -treefile
examples/droso.12sp.tamura.nwk -start_val 1 -seed 1568465 –root_dist 0
```

reports the following results:

```
        ##Family Turnover Rates
                #Likelihood: -38.0770296976169
                #Branch_Group   Birth   Death   Innovation
                0       0.0001973       4.1116335       202.8365705

        ##Execution time (seconds): 143
```

To examine whether these estimates represent a local maxima, we should run `BadiRate` using other starting_values (see the `-seed` and the `–start_val` options).

```
perl BadiRate.pl -sizefile examples/obp_all.12sp.tsv -treefile
examples/droso.12sp.tamura.nwk -start_val 0 -seed 1 –root_dist 0
```

```
##Family Turnover Rates
                #Likelihood: -32.5489511730966
                #Branch_Group   Birth   Death   Innovation
                0       0.0059622       0.0045715       0.0000000

        ##Execution time (seconds): 33
```

The better (higher) likelihood of the second run indicates that the first run represent a local maxima.

1.2) Reduce the number of parameters. For example, you can use a BD instead of the BDI model.

```
perl BadiRate.pl -sizefile examples/obp_all.12sp.tsv -treefile
examples/droso.12sp.tamura.nwk -start_val 0 -seed 9877 –root_dist 0 –
rmodel BD
```

```
       ##Family Turnover Rates
                #Likelihood: -32.5489511330279
                #Branch_Group   Birth   Death
                0       0.0059608       0.0045709
```

```
        ##Execution time (seconds): 30
```

1.3) Incorporate (if possible) more families to the analysis

```
perl BadiRate.pl -sizefile examples/4FAMs.12sp.tsv -treefile
examples/droso.12sp.tamura.nwk -start_val 0 –root_dist 0
```

```
        ##Family Turnover Rates
                #Likelihood: -182.349197899147
                #Branch_Group   Birth   Death
                0       0.0599468       0.0604083

        ##Execution time (seconds): 48
```

2) Conversely, simplest models might have problems to fit the data. For example, running:

```
perl BadiRate.pl -sizefile examples/sdic_all.6sp.tsv -treefile
examples/droso.6sp.tamura.nwk –rmodel BD –root_dist 0
```

reports the following warning:

```
        ##Family Turnover Rates
                #Likelihood: -inf
                #WARN: Try using a more complex model, or changing the starting values. See
the -rmodel, -bmodel and -start_val options

        ##Execution time (seconds): 0
```

The above data set includes the *Sdic* gene family, a *D. melanogaster* family originated by a gene fusion. Since such density-independent acquisitions can not be modelled using a BD model, we might use a BDI or a GD instead (see the "Introduction" section).

Other common instances of this warning might occur:

- If we model family expansions/contractions using a L or a LI model (*i.e.* assuming equal birth and death rates).
- If we use a GR branch model (see `–bmodel` option) to analyze phylogenetic lineages with heterogeneous turnover rates
- …

3) Lastly, if the starting values are too far from the actual ML values, the Amoeba algorithm might have problems to estimate the parameters. You can solve this problem by running `BadiRate` with different starting values (see `–start_val` option).

7 Examples

In this section, we include three examples:

1) The commands used to analyze miRNA data [24]
2) The analysis of the OBP family turnover in 12 *Drosophila* species.
3) The analysis of the OBP family turnover in 18 *Hexapoda* species (a subset of the species used in Vieira, F.G. and Rozas, J. [5]).

The examples in this section were run in a Windows XP machine with one CPU processor "x86 family 15 Model 6 Stepping 4 GenuineIntel ~3391 MHz".

1) miRNA

The basic run of `BadiRate` is:

Running Example (BDI-GR-ML):

```
perl BadiRate.pl –treefile examples/droso.12sp.tamura.nwk –sizefile
examples/miRNA.og.tsv
```

However, since a significant proportion of miRNA are originated by innovation [24], we add **-rmodel GD** option to analyze the data under a GD (Gain-Death) stochastic model. Moreover, we also corrected the likelihood for the miRNA families absent in all species (-**unobs** option).

Running Example (GD-GR-ML):

```
perl BadiRate.pl –treefile examples/droso.12sp.tamura.nwk –sizefile
examples/miRNA.og.tsv –rmodel GD -unobs
```

To test whether *D. willistoni* has distinct miRNA turnover rates, we re-run twice the program specifying the following Branch-specific Rates (BR) models (see `BadiRate` paper):

A) GD rates can vary in the *D. willistoni* branch (17->16 in that case), and in all the branches leading to *D. melanogaster*.

Running Example (GD-BR-ML):

```
perl BadiRate.pl –treefile examples/droso.12sp.tamura.nwk -
sizefile examples/miRNA.og.tsv –rmodel GD –bmodel "17->16_23-
>17:17->15:15->11:11->9:9->5:5->1" -unobs
```

B) GD rates can only vary in all the branches leading to *D. melanogaster*

Running Example:

```
perl BadiRate.pl –treefile examples/droso.12sp.tamura.nwk –
sizefile examples/miRNA.og.tsv –rmodel GD –bmodel "23->17:17-
>15:15->11:11->9:9->5:5->1" -unobs
```

The likelihood values, the number of parameters and the AIC values (Akaike Information Criterion) for the different branch models are:

	ln (Likelihood)	**Parameters**	**AIC**
GD-BR-ML (A)	-748.9064	6	1509.8128
GD-BR-ML (B)	-755.1959	4	1518.3918
GD-GR-ML	-783.3624	2	1570.7248

AIC values are computed as:

$$AIC = 2K - 2\ln(L)$$

where K is the number of parameters, and ln(L) is the natural logarithm of the likelihood value.

Finally, to estimate the ancestral miRNA family sizes and the minimum number of gains and losses per branch, we should add the **`-anc`** option and run `BadiRate` under the six rate model:

`Running Example (GD-BR-ML (A)):`

```
perl BadiRate.pl -treefile examples/droso.12sp.tamura.nwk -sizefile
examples/miRNA.og.tsv -rmodel GD -bmodel “17->16_23->17:17->15:15-
>11:11->9:9->5:5->1” -anc
```

Alternatively, we can also use the `-anc` under a FR branch model.

`Running Example (GD-FR-ML):`

```
perl BadiRate.pl -treefile examples/droso.12sp.tamura.nwk -sizefile
examples/miRNA.og.tsv -rmodel GD -bmodel FR -anc
```

Since the data set includes information of 153 miRNA families, `BadiRate` will report the total minimum number of gain and loss events by summing the gain and loss events inferred in each family.

2) OBPs in *Drosophila*

Here we have included some analysis of the OBP gene family in the 12 *Drosophila* species. For instance, to estimate the family turnover rates under the CML method (a method similar to that conducted in Vieira et al 2007), we might run:

`1. Running Example (BDI-GR-CML):`

```
perl BadiRate.pl -treefile examples/droso.12sp.tamura.nwk -sizefile
examples/obp_sub.12sp.tsv -unobs -family -ep CML
```

```
##Family Turnover Rates
       #Branch_Group Birth  Death  Innovation
       0      0.0022566     0.0009904     0.0000000
       …
##Execution time (seconds): 23
```

Note that we use the **`-family`** option to estimate the turnover rates from the **`examples/obp_sub.12sp.tsv`** *sizefile*, which includes the 56 orthologous groups defined for the OBP gene family. Using this command, `BadiRate`:

1. Infers the number of family members in the internal nodes by ML (assuming the same BDI turnover rates for all the orthologous groups and lineages).
2. Calculates the minimum number of gene gains and losses from the inferred (step 1) number of family members in internal nodes.

3. From the number of members in internal nodes (step 1) and the minimum number of gene gains and losses (step 2), computes the gene family turnover rates as in Vieira, et al. (2007).

BadiRate does not return any likelihood value in the counting-based methods (CML, CSP, CWP and CMAP). It is worth noting that these methods can underestimate the family turnover rates even if the input data includes the gene family separated into its component orthologous groups (or subfamilies).

To estimate the family turnover rates by full-likelihood methods (ML or MAP), we should run:

2. Running Example (BDI-GR-ML):

```
perl BadiRate.pl -treefile examples/droso.12sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -root_dist 0
```

```
 OUTPUT

       ##Family Turnover Rates
              #Likelihood: -32.5489513641113
              #Branch_Group Birth  Death  Innovation
              0      0.0059594     0.0045692     0.0000000

       ##Execution time (seconds): 48
END OUTPUT
```

The estimate provided by CAFE (v2.2) is λ = 0.005147.

We can also run BadiRate under the L model (*i.e.* assuming equal birth and death rates) as follows:

3. Running Example (L-GR-ML):

```
perl BadiRate.pl -treefile examples/droso.12sp.tamura.nwk -sizefile
examples/obp_all.12sp.tsv -root_dist 0 -rmodel L
```

```
OUTPUT

        ##Family Turnover Rates
                #Likelihood: -32.7083090017046
                #Branch_Group   Lambda
                0       0.0054998

        ##Execution time (seconds): 11
END OUTPUT
```

Obviously, the execution time is lower for simpler models. For instance, estimations under the L model take 11 seconds (*i.e.* assuming equal birth and death and null innovation rates saves 37 seconds of the execution time), whereas CAFE (v2.2) takes 20 seconds.

3) OBPs in *Hexapoda*

Here we include the commands used to analyze the turnover rates and the number of OBP genes in the internal nodes of the hexapoda phylogeny [5]. The `BadiRate` command to reproduce the analysis of Vieira, F.G. and Rozas, J. [5] and estimate the turnover rates using the 86 ortholog groups included in the file **`examples/obp_sub.hexapoda.tsv`** is:

1. Running Example (BDI-GR-CML):

```
perl BadiRate.pl -treefile examples/hexapoda.nwk -sizefile
examples/obp_sub.hexapoda.tsv -ep CML -anc -family -unobs
```

In particular, running the above command reports the following output (**examples/obp_sub.hexapoda.out**):

```
        ##Family Turnover Rates
                #Branch_Group   Birth   Death   Innovation
                0       0.0082081       0.0005562       0.0000000
…
##Execution time (seconds): 86
```

It is also possible to run this command applying specific branch models. In that cases, `BadiRate` will estimate the number of family members in internal nodes (step 1) using the turnover rates specified in each branch. For instance, to run `BadiRate` under a free rates (FR) model you should add the **`-bmodel FR`** option:

Running Example (BDI-FR-CML):

```
perl BadiRate.pl -treefile examples/hexapoda.nwk -sizefile
examples/obp_sub.hexapoda.tsv -bmodel FR -ep CML -anc -family -unobs
```

Using this command, `BadiRate` infers the number of family members in internal nodes by ML assuming independent family turnover rates in each phylogenetic lineage.

```
##Family Turnover Rates
                #Branch_Group   Birth   Death   Innovation
                0     0.0000000      0.0000000      0.0000000
                1     0.0071429      0.0000000      0.0000000
                2     0.0125000      0.0000000      0.0000000
                3     0.0700000      0.0100000      0.0000000
                4     0.0080214      0.0000000      0.0000000
                5     0.1000000      0.0000000      0.0000000
                6     0.0000000      0.0000000      0.0000000
                7     0.0082645      0.0000000      0.0000000
                8     0.0026882      0.0000000      0.0000000
                9     0.0000000      0.0000000      0.0000000
                10    0.0000000      0.0000000      0.0000000
                11    0.0000000      0.0048077      0.0000000
                12    0.0000000      0.0000000      0.0000000
                13    0.0392157      0.0000000      0.0000000
                14    0.0000000      0.0000000      0.0000000
                15    0.0057692      0.0000000      0.0000000
                16    0.0019231      0.0057692      0.0000000
                17    0.0023674      0.0014205      0.0000000
                18    0.0004209      0.0000000      0.0000000
                19    0.0000000      0.0000000      0.0000000
                20    0.0000000      0.0000000      0.0000000
                21    0.0065982      0.0000000      0.0000000
                22    0.0012500      0.0000000      0.0000000
                23    0.0000000      0.0000000      0.0000000
                24    0.0023603      0.0007868      0.0000000
                25    0.0000000      0.0000000      0.0000000
                26    0.0034033      0.0005672      0.0000000
                27    0.0167500      0.0000000      0.0000000
                28    0.0138462      0.0000000      0.0000000
                29    0.0183036      0.0000000      0.0000000
                30    0.0080952      0.0014286      0.0000000
                31    0.0095238      0.0000000      0.0000000
                32    0.0000000      0.0023669      0.0000000
                33    0.0026627      0.0011834      0.0000000
                Global_Rates 0.0086856      0.0006896      0.0000000
…
##Execution time (seconds): 23509
END OUTPUT
```

The estimates of the turnover rates (β = 0.0082081, δ = 0.0005562 and ι = 0.0000000) under the free rates branch model (BDI-FR-CML) are very similar to that estimated under the global rates (BDI-GR-CML) branch model (β = 0.0086856, δ = 0.0006896 and ι = 0.0000000).

The next command allows estimating the OBP turnover rates by a full-likelihood method (**`-ep`** ML).

2. Running Example (BDI-GR-ML):

```
perl BadiRate.pl –treefile examples/hexapoda.nwk –sizefile
examples/obp_all.12sp.tsv –root_dist 0 –ep ML
```

```
##Family Turnover Rates
       #Likelihood: -62.9060466514088
       #Branch_Group   Birth   Death   Innovation
        0       0.0110473       0.0090283       0.0000000
##Execution time (seconds): 140
```

The estimates of the family turnover rates are higher using full-likelihood methods, which probably reflects the underestimation done by the counting-based methods. Even so, to estimate the minimum number of events (`-anc`), inferred by counting approaches (regardless of the value of the `–ep` option), it is very convenient to use a data file with containing information of its orthologous groups.

8 References

1. Innan H, Kondrashov F (2010) The evolution of gene duplications: classifying and distinguishing between models. Nat Rev Genet 11: 97-108.

2. Ohno S (1970) Evolution by Gene Duplication. Berlin: Springer.

3. Sanchez-Gracia A, Vieira FG, Rozas J (2009) Molecular evolution of the major chemosensory gene families in insects. Heredity 103: 208-216.

4. Hahn MW, Han MV, Han SG (2007) Gene family evolution across 12 Drosophila genomes. PLoS Genet 3: e197.

5. Vieira FG, Rozas J (2011) Comparative Genomics of the Odorant-Binding and Chemosensory Protein Gene Families across the Arthropoda: Origin and Evolutionary History of the Chemosensory System. Genome Biol Evol 3: 476-490.

6. Nei M, Rooney AP (2005) Concerted and birth-and-death evolution of multigene families. Annu Rev Genet 39: 121-152.

7. Demuth JP, Hahn MW (2009) The life and death of gene families. Bioessays 31: 29-39.

8. Iwasaki W, Takagi T (2007) Reconstruction of highly heterogeneous gene-content evolution across the three domains of life. Bioinformatics 23: i230-239.

9. Arvestad L, Berglund AC, Lagergren J, Sennblad B (2003) Bayesian gene/species tree reconciliation and orthology analysis using MCMC. Bioinformatics 19 Suppl 1: i7-15.

10. Vernot B, Stolzer M, Goldman A, Durand D (2007) Reconciliation with non-binary species trees. Comput Syst Bioinformatics Conf 6: 441-452.

11. Hahn MW (2007) Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution. Genome Biol 8: R141.

12. Vieira FG, Sanchez-Gracia A, Rozas J (2007) Comparative genomic analysis of the odorant-binding protein family in 12 Drosophila genomes: purifying selection and birth-and-death evolution. Genome Biol 8: R235.

13. Novozhilov AS, Karev GP, Koonin EV (2006) Biological applications of the theory of birth-and-death processes. Brief Bioinform 7: 70-85.

14. Gu X, Zhang H (2004) Genome phylogenetic analysis based on extended gene contents. Mol Biol Evol 21: 1401-1408.

15. Spencer M, Susko E, Roger AJ (2006) Modelling prokaryote gene content. Evol Bioinform Online 2: 157-178.

16. Hahn MW, De Bie T, Stajich JE, Nguyen C, Cristianini N (2005) Estimating the tempo and mode of gene family evolution from comparative genomic data. Genome Res 15: 1153-1160.

17. De Bie T, Cristianini N, Demuth JP, Hahn MW (2006) CAFE: a computational tool for the study of gene family evolution. Bioinformatics 22: 1269-1271.

18. Liu L, Yu L, Kalavacharla V, Liu Z (2011) A Bayesian model for gene family evolution. BMC Bioinformatics 12: 426.

19. Cohen O, Ashkenazy H, Belinky F, Huchon D, Pupko T (2010) GLOOME: gain loss mapping engine. Bioinformatics 26: 2914-2915.

20. Csuros M (2010) Count: evolutionary analysis of phylogenetic profiles with parsimony and likelihood. Bioinformatics 26: 1910-1912.

21. Rambaut A (2006) FigTree.

22. Tamura K, Dudley J, Nei M, Kumar S (2007) MEGA4: Molecular Evolutionary Genetics Analysis (MEGA) software version 4.0. Mol Biol Evol 24: 1596-1599.

23. Page RD (1996) TreeView: an application to display phylogenetic trees on personal computers. Comput Appl Biosci 12: 357-358.

24. Nozawa M, Miura S, Nei M (2010) Origins and evolution of microRNA genes in Drosophila species. Genome Biol Evol 2: 180-189.