MISFITS: Evaluating the goodness of fit between a phylogenetic model and an alignment.

Manual for Version 1.0, March 2010.

Updated: April 21, 2011.

Minh Anh Thi Nguyen[1]

email: minh.anh.nguyen(at)univie.ac.at

Steffen Klaere[2]

email: steffen.klaere(at)gmail.com

Arndt von Haeseler[1]

email: arndt.von.haeseler(at)univie.ac.at

[1] Center for Integrative Bioinformatics Vienna,
Max F. Perutz Laboratories,
Dr. Bohr-Gasse 9/6, A-1030 Vienna, AUSTRIA.

[2] Computational Evolution Group,
Department of Mathematics, The University of Auckland,
Private Bag 92019, Auckland Mail Centre,
Auckland 1142, NEW ZEALAND.

Contents

1 License Agreement

2 Introduction

`MISFITS` is a program to evaluate the goodness of fit of a model to an alignment in phylogeny reconstruction. It offers a look back at the alignment to pinpoint to site patterns that do not fit to the model and the resulting tree (thereafter referred to as the tree-model). `MISFITS` then introduces a number of extra-substitutions on the tree, in a parsimonious manner to fit these site patterns in to the tree-model. These extra-substitutions plus the evolutionary model will then fully explain the alignment. Thus, the number of extra-substitutions may be interpreted as a measure to evaluate the goodness of fit of the model to the alignment: the smaller the number, the better the fit.

3 Method

The method is presented in:

Minh Anh Thi Nguyen, Steffen Klaere and Arndt von Haeseler. MISFITS: Evaluating the goodness of fit between a phylogenetic model and an alignment. *Mol. Biol. Evol.* (2011) 28 (1): 143-152. doi: 10.1093/molbev/msq180.

A schematic workflow of the method is as follows:

Input: a phylogenetic tree and a gapless alignment (see Section 6 for more details).

1. Count the observed frequency of patterns in the alignment.
2. Compute pattern likelihoods under the model and the inferred tree.
3. Determine the set of over-represented patterns $\mathcal{D}^+$ and the set of under-represented patterns $\mathcal{D}^-$.
4. For all pairs of patterns $(p, p'), p \in \mathcal{D}^+$ and $p' \in \mathcal{D}^-$, compute the minimal number of extra-substitutions to convert p into p'.
5. Select a matching between patterns in $\mathcal{D}^+$ and $\mathcal{D}^-$ such that the total number of extra-substitutions is minimal.
6. Map the extra-substitutions on the tree.
7. Determine the significance of the number of extra-substitutions computed at step 5.

Output: List of site patterns in $\mathcal{D}^+, \mathcal{D}^+$ and number of extra-mutations (see Section 7).

4 Availability

- For step 1-6: The program `MISFITS` is written in C++ and available free of charge. The executable file currently works under Unix platform (Linux and MacOS X systems) as well as Windows system.
- For step 7: Since it depends on the simulation and tree reconstruction programs that users want to use, we provide a number of bash scripts running on Unix system to carry out this task with: `SEQ-GEN` for simulation and `PHYML` for tree reconstruction. Users may modify these scripts to use other programs instead as well as to use `MISITS` with different options.
- Refer to the program website http://www.cibiv.at/software/misfits for the source code, program manual, binary file of `MISFITS` and the bash scripts.

5 External programs required

We use the following external programs in our software package:

- For step 1-6: the `MISFITS` program requires `TREE-PUZZLE` to compute likelihood of the patterns (written in a phylip format alignment) given the tree and the model with the corresponding parameter's values. Please make sure that the executable file of `TREE-PUZZLE` is named under `puzzle`.
- For step 7: a simulator and a tree-reconstruction program are needed. If you use the bash scripts we provide, you need `SEQ-GEN` and `PHYML` packages and the executable files should be named under `seq-gen` and `PhyML_3.0` respectively.

6 Command-line options and input files

Run `misfits -h` to print a short description of available options.
Usage: `misfits -a <file> -t <file> -pz <file> OPTIONS`

Required input arguments:

`-a <file>:`	file contains alignment in Phylip format.
`-t <file>:`	file contains a tree (reconstructed from the above alignment) in Newick format.
`-pz <file>:`	file contains parameter values to run `PUZZLE` to compute likelihood of site patterns (refer to Section 13 for formating).

Options:

`-h:`	print this help message.
`-s:`	to assign extra-substitutions on branches of the tree. Default is off.
`-p <file>:`	file contains site paterns, observed frequencies and log likelihood. If not given, we use `PUZZLE` to compute the pattern likelihoods. Format: each line contains: `pattern frequency log-likelihood`

`-pztree <file>:` file contains the tree to run `PUZZLE` if it is different (e.g. topological different) from the above tree.

`-o <file>:` file contains outgroup of the tree. Format: each taxon name in one line. If not given, the root will be located at the first taxon (leaf) as appears in the tree file. Note, that the position of the root will influence the score.

`-noInit:` to NOT include observed patterns which are in the confidence region but not yet reach the upper bound into $\mathcal{D}^-$ set (default is YES).

`-mdf <number>:` degrees of freedom from the model (number of model parameters) to compute the confidence region according to the χ^2 distribution. Default is 0. Refer to Section 14 for the computation.

`-ciBr:` to include number of branches of the tree into the degree of freedom used in computing the confidence region according to the χ^2 distribution. Default is off.
When both `-mdf` and `-ciBr` are not set, confidence region will be computed according to the normal distribution. We recommend to include `-mdf` and `-ciBr` into the calculation.

`-prefix:` prefix for output file names. Default is the alignment file name.

Additional options:

`-mstep <number>:` to generate and print out 1, 2 .. `mstep`-mutation away patterns. Default is 0 (do not print).

`-add:` to output additional information (see Additional outputs for more details).

7 Output files

Basic outputs: (replace the following `*` by `[prefix].misfits`)

`*:` file summarizes the main output.

`*.cost:` file contains the number of extra-substitutions computed by `misifts`.

`*.alignMoreLess:` file contains number of sequences (`seqNum`), number of patterns in the alignment (`patNum`), alignment length (`seqLen`), number of over-represented patterns (`moreNum`), number of sites in the alignment containing over-represented patterns need to be converted into under-represented patterns (`moreSites`), number of under-represented patterns in $\mathcal{D}^-$ (`lessNum`) and number of sites in the alignment they can replace for (`lessSites`).

`*.subMorePat:` in case under-represented patterns are not enough to exchange with all the over-represented patterns, a subset of over-represented patterns will be exchanged. This file contains this subset. **Attention:** please check the existence of this file. If it exists, the number in `*.cost` is the cost to replace this subset (not the whole set) of over-represented patterns by under-represented patterns.

When `-s` is indicated:

`*.longExterBr.tree:` tree with assignment of the number of extra-substitutions to the branches using delayed transformation `DELTRAN` (long external branches).

`*.longInterBr.tree:` tree with assignment of the number of extra-substitutions to the branches using accelerated transformation `ACCTRAN` (long internal branches).

Additional outputs (when `-add` is indicated):

`*.more:` file contains over-represented patterns ($\mathcal{D}^+$). Each line presents:
`pattern:` the pattern.
`No.observed:` number of sites containing this patterns in the alignment.
`Logll:` logarithm of the likelihood of this pattern under the tree-model.
`mutStep:` number of substitutions away from a constant pattern.
`No.exchange:` number of sites containing this patterns to be replaced by under-represented patterns.

`*.more.phy:` file contains over-represented patterns in the form of an alignment in Phylip format.

`*.less:` file contains under-represented patterns ($\mathcal{D}-$) in a similar format as `*.more`.

`*.alignLess:` file contains under-represented patterns observed in the alignment (a subset of $\mathcal{D}^-$).

`*.overPos:` file presents positions of the over-represented patterns. Format: each line contains a starting position and the number of over-represented site patterns starts at this position.

`*.overPosSure:` positions of the over-represented patterns with expected number of occurrence under the tree-model is 0 (not expect to see them in the alignment at all).

`*.alignPat:` a summary of all patterns in the alignment. Format: each line contains `Pattern, No.observed, Logll, mutStep` and `No.exchange`.

`*.paired.pat:` the matching between over-represented patterns (`overPat`) and under-represented patterns (`underPat`).

When `-s` is indicated:

`*.branchLabel.tree:` the input tree with branches labelled by numbers.

`*.longInterBr.table:` a table summarizing the type of the extra-substitutions according to the Kimura 3 parameter model (1 transition and 2 transversions) on each branch using `ACCTRAN` criteria.

`*.longExterBr.table:` a table summarizing the type of the extra-substitutions according to the Kimura 3 parameter model (1 transition and 2 transversions) on each branch using `DELTRAN` criteria.

8 Installation

Please make sure that all the external programs required are installed and your operating system should be able to locate these programs. To do that, copy all the binary files of the external programs required with the appropriate names and the `MISFITS` binary into a directory and specify this directory in the global variable `PATH`. For UNIX system, you have to add `export PATH="directory_path:$PATH"` to the `.bashrc` located in your home directory (`directory_path` is the path to the directory contains the binary files).

9 Example

An example to try our program is given in `example.zip`. Run `misfits` with the input provided (`-mdf 10` is equivalent to GTR+I+Γ model):

```
misfits-1.0 -a example.phy -t example.tree -pz example.puzzle-param -mdf
10 -ciBr -o outgroup.txt -s -add
```

You may also try the bash scripts we provide to:

(1) Reconstruct the tree under GTR+I+Γ model using `PHYML` and then run `MISFITS` with the default options:
`directory_path/Unix-phyml-misfits-1.0/GTRig-phyml-misfits.sh example.phy`

(2) Carry out the parametric bootstrap with 100 bootstrap samples to evaluate the significance of the number of extra-substitutions (output is writen to `prefix.simCost`):
`directory_path/Unix-seqgen-phyml-misfits-1.0/GTRig-seqgen-phyml-misfits.sh example.phy_phyml_tree.txt example.phy_phyml_stats.txt example.phy 1 100 prefix`

Note: Before doing (2) you should delete the names of the inner nodes on the tree as produced by `phyml` in (1). The command `sed -i 's/)[0-9]*.[0-9]*:/):/g' example.phy_phyml_tree.txt` will do it. Also, you should define and export `${BINDIR}` as guided in our bash script `GTRig-seqgen-phyml-misfits.sh`.

10 Version History

- March 2010: The first version `misfits-1.0` was launched

11 Credits

The source code to compute the quantile of the χ^2 distribution were taken from the ALGLIB project, copyright (C) 1999-2009 by Sergey Bochkanov and Vladimir Bystritsky. Bui Quang Minh, Nguyen Dinh Tu and Dinh Quang Huy kindly provided several small pieces of code and discussed certain issues during the implementation.

12 Acknowledgment

Support from the Wiener Wissenschafts-, Forschungs- and Technologiefonds (WWTF) to this work is greatly appreciated. A.v.H. also acknowledges the funding from the SPP 1174

(Deep Metazoan Phylogeny) project. S.K. appreciates support from a Marsden Grant to David Bryant and the research development fund at the University of Auckland.

13 Appendix A

Converting the model to GTR is the most convenient setting in order to run `PUZZLE` to compute site likelihoods of an alignment given the tree-model (with parameter values). Please prepare the file containing parameters to run `PUZZLE` in the following format (remove the text indicated by `#`):

```
d
m
m
1
rAC #A-C rate
2
rAG #A-G rate
3
rAT #A-T rate
4
rCG #C-G rate
5
rCT #C-T rate
6
rGT #G-T rate, should be 1.0
f
fA #frequency of A, in percent, e.g. 28.1
fC #frequency of C, in percent, e.g. 27.1
fG #frequency of G, in percent, e.g. 30.5. Note, no entry for frequency of T
```

#Include Gamma-rate	#Include Invariant sites	#Include I+G
w	w	w
a	w	w
α #$\alpha - shape$	i	w
c	number #pro.invariant sites	i
ncat #no.of rate cat.		number #pro.invariant sites
		a
		α #$\alpha - shape$
		c
		ncat #no.of rate cat.

```
y #this is to prompt the setting
```

14 Appendix B: Number of degrees of freedom

Degrees of freedom = Number of free parameters in the model+Number of branches of the tree.

Number of branches of three tree $tdf = 2 * n - 3$, where n is the number of taxa. Misfits computes this number *itself* if `-ciBr` is given in the command line.

The below table gives the number of free parameters (mdf) for commonly used models:

Model	JC	F81	K80	HKY	TrNef	TrN	TPM1	TPM1uf	SYM	GTR
mdf	0	3	1	4	2	5	2	5	5	8

To include invariant sites + 1, to include gamma distribution for rate across site + 1. For example, $mdf(GTR + I + \Gamma) = 8 + 1 + 1 = 10$.

If you use R to visualize $\mathcal{D}^+$ and $\mathcal{D}^-$, κ should be computed as follows:

$$\kappa = sqrt(qchisq(1 - \alpha/(2 * \ell), df)$$

where ℓ is alignment length, $\alpha = 0.05$, $df = tdf + mdf$.