

Routines for GLS estimation

Version 1.0

Edward Susko

Department of Mathematics and Statistics, Dalhousie University

Introduction

The main programs implement the methods described in Susko (2003) which utilize the results of Susko (2011); please cite these references when using the software. Four programs are included: `glsphyl`, `wlsnphyl` give the GLS and WLS routines for testing while `glsphylest` and `wlsphylest` give the GLS and WLS routines for estimation.

Installation

The main programs are compiled from C and Fortran 77 source code. Implementation of non-negativity constraints on edge-lengths is through the NNLS routine of Lawson and Hanson (1974). The programs also rely upon the Fortran 77 LAPACK libraries (Anderson et al. 1999).

Precompiled binaries are available for Windows. To install the programs on other systems

1. Download and unpack the software:

```
$ tar xzf glsv1.0.tar.gz
```

This will create a directory `glsv1.0` that contains the source code.

2. Change directories to `glsv1.0` and create the main program files `glsphyl`, `wlsnphyl`, `glsphylest` and `wlsphylest` with the `make` command.

```
$ cd glsv1.0
```

```
$ make
```

Alternatively, if a version of the LAPACK libraries exists on the system, copy `withlapack.makefile` to `Makefile`, open a text editor with the newly created `Makefile` and insert the usual compile flags you would use to compile with the LAPACK libraries immediately after “`LAPACKL=`” on a single line.

3. To test the programs, issue the command

```
$ ./glsphyl F84n1000f1234tt2.ct1 > F84n1000f1234tt2.out
```

This should create a file `F84n1000f1234tt2.out` comparable to the output in the running example below. To test the other programs, replace `glsphyl` with `wlsnphyl`, `glsphylest` or `wlsphylest`

The main program files can then be copied to a location in your `PATH` and the source code directory can be removed.

The source code has been compiled and tested using `gcc` and `gfortran` (versions 4.4.0) on a CentOS Linux distribution (release 5.3). While the program has not been tested on another platform, it should compile under any Linux distribution as well as Mac OS X, assuming a fortran compiler is installed.

Input

The programs can be run at the command line with the command

```
$ glsphylprogram controlfile
```

where `glsphylprogram` is one of `glsphyl`, `wlsnphyl`, `glsphylest` or `wlsphylest`. The `glsphyl` and `wlsnphyl` programs are used to test whether trees in a specified tree file can be included in a confidence region of trees. The `glsphylest` and `wlsphylest` programs are used for estimation alone.

All input to the routine is through a main control file, `controlfile`. The control file is similar in format to the control files used by the programs `baseml` and `codeml` in the PAML package (Yang 1997, 2007). For instance, the `model` variable specifies the substitution model and gives a subset of the models available in PAML, with the same numbering scheme. As a running example, consider the test file `F84n1000f1234tt2.ctl`:

```
* Example control file
seqfile = F84n1000f1234tt2 * sequence file

treefile = six-taxon.trees * treefile
ntrees = 105 * number of trees in treefile

model = 3 * F84 model
ttratio = 2 * transition/transversion ratio
nneg = 1 * use nonnegativity constraints
pattwt = single * single set of pattern weights from NJ tree
```

As with PAML control files, blank lines are allowed and all text following a `*` till the end of a line is treated as a comment. The word on the left of an equal sign gives a control variable and the word on the right gives the value of that variable. Spaces are required on both side of an equal sign. The order of variables is unimportant. The control variables are as follows. All variables not indicated as optional are required.

treefile and **ntrees**: **treefile** is the name of a file containing tree(s). The treatment of this file and of the variable **ntrees** differs depending upon whether testing programs (**glsphyl** and **wlsnphyl**) or estimation programs (**glsphylest** and **wlsphylest**) are used. A treefile is required for the testing programs but not for the estimation programs.

For testing programs, the file should include all of the topologies that are to be tested and **ntrees** should indicate the number of trees in the file. For instance, the example control file had

```
treefile = six-taxon.trees * treefile
ntrees = 105 * number of trees in treefile
```

and the file **six-taxon.trees** contained text giving 105 topologies in Newick format:

```
(3,(2,5),(1,(0,4)));
((2,5),(3,1),(0,4));
...
(1,(5,0),(3,(2,4)));
```

Edge-lengths can be included in tree specification but are ignored.

For the estimation routines, the first tree in the file, is taken as the starting tree for optimization. If no tree is provided, the NJ tree is used as a starting tree.

For estimation routines, the variable **ntrees** indicates the maximum number of trees that should be output. The default value is 1, in which case the tree giving the smallest WLS or GLS statistic value is output. If **ntrees** is set to -1, all of the trees encountered during tree searching are output, ranked from smallest to largest WLS or GLS statistic value. In the example file, **ntrees** was 105, so that at most 105 of the trees encountered during searching would be output.

The tree should conform to the Newick standard. A discussion of this standard as implemented in PHYLIP is given at

<http://evolution.genetics.washington.edu/phylip/newicktree.html>

and a more formal description is available at

http://evolution.genetics.washington.edu/phylip/newick_doc.html

Allowable features of the Newick standard that will likely create difficulties are:

1. Quoted labels.

2. Nested use of the characters '[' and/or ']' in comments. The characters '[' and ']' can only be used to delimit comments and cannot be used within comments.
3. Long leaf labels. A limit of 10 non-null characters is allowed for leaf names.
4. Underscores are not converted to blanks.

seqfile: This is the name of the file containing the sequence alignment. The file should conform to PHYLIP standards for input with 10 character long names padded by blanks. The names should match the names used in the input treefile. Input can be either interleaved or sequential with one caveat: The lines 2 through $m + 2$, where m is the number of taxa, must contain the name of taxa followed by sequence data. For instance the start of a sequence file might be

```
6 3414
Homsa      ANLLLLIVPI LI...
Phovi      INIISLIIFI LL...
...
```

but not

```
6 3414
Homsa
ANLLLLIVPI LI...
Phovi
INIISLIIFI LL...
...
```

which would be allowed under the sequential format by PHYLIP.

nchar: An optional integer indicating that the model was for nucleotide data (**nchar=4**) or amino acid data (**nchar=20**). The default value is 4.

model: An integer code for the substitution model. For nucleotide data (**nchar=4**), the models currently implemented are

<u>model</u>	<u>Model</u>
0	JC
2	F81
3	F84
4	HKY
7	GTR

and for amino acid data (**nchar=20**) the models currently implemented are

<code>model</code>	Model
0	Poisson
1	Proportional
2	Empirical
3	Empirical+F
8	REVaa

The documentation for the PAML package gives a good description of the models listed and can fit all of them.

The GTR and REVaa models refer to the most general time-reversible models in the nucleotide and amino acid case, respectively. The Poisson and Proportional models are the analogues of the JC and F81 models for amino acid data. The Poisson and Proportional models have substitution probabilities

$$P_{ij}(t) = \begin{cases} \pi_j + (1 - \pi_j) \exp[-\mu t] & \text{if } i = j \\ \pi_j - \pi_j \exp[-\mu t] & \text{otherwise} \end{cases}$$

where $\mu = [\sum \pi_i(1 - \pi_i)]^{-1}$ and π_j gives the stationary of the j th amino acid. In the Poisson model, the frequencies are all 1/20

When `model=2` or `3` an empirical model is fit. The model is specified by the variable `aaRatefile`. When `model=2`, the stationary frequencies are the stationary frequencies of the specified empirical model.

When `model=3`, the stationary frequencies are determined as the frequencies of the character states in the sequence data in `seqfile`. In this case, the empirical model is used to specify the exchangeabilities. The exchangeability of amino acid i and j is defined as

$$S_{ij} = Q_{ij}/\pi_j$$

where, for the specified empirical model, Q_{ij} is the rate of substitution from i to j and π_j is the stationary frequency j . When `model=3`, the rate of substitution from i to j is

$$\tilde{Q}_{ij} = S_{ij}\tilde{\pi}_j$$

where $\tilde{\pi}_j$ is the frequency of j in the alignment.

aaRatefile: Only required for empirical amino acid models (`model=2` or `3` and `nchar=20`). The name of the empirical model to fit. The models currently implemented are

<code>dayhoff.dat</code>	Dayhoff or PAM	Dayhoff et al. (1978)
<code>jones.dat</code>	JTT	Jones et al. (1992)
<code>wag.dat</code>	WAG	Whelan and Goldman (2001)
<code>mtREV24.dat</code>	mtREV	Adachi and Hasegawa (1996)
<code>lg.dat</code>	LG	Le and Gascuel (2008)

The naming scheme was chosen to be consistent with PAML. However, `aaRatefile` is not actually the name of file, rather it identifies a model.

Qfile: Only required for the general time reversible model, GTR or REVaa (`model=7`, `nchar=4` or `model=8`, `nchar=20`). The name of a file containing the entries of the rate matrix separated by blanks.

For nucleotides the file should contain the entries

$$\begin{array}{cccc} Q_{AA} & Q_{AC} & Q_{AG} & Q_{AT} \\ Q_{CA} & Q_{CC} & Q_{CG} & Q_{CT} \\ Q_{GA} & Q_{GC} & Q_{GG} & Q_{GT} \\ Q_{TA} & Q_{TC} & Q_{TG} & Q_{TT} \end{array}$$

Here the (3,2) entry Q_{GC} gives the rate of substitution from G to C. The Q_{ii} satisfy that $Q_{ii} = -\sum_{j \neq i} Q_{ij}$. Note that this ordering differs from the T, C, A and G ordering of PAML.

Amino acids are ordered alphabetically:

alanine, arginine, asparagine, aspartic, cysteine,
glutamine, glutamic, glycine, histidine, isoleucine,
leucine, lysine, methionine, phenylalanine, proline,
serine, threonine, tryptophan, tyrosine, valine

which is the same ordering used by most phylogenetic packages including PAML, PHYLIP and TREE-PUZZLE.

kappa or **ttratio**: One of these is required for the F84 and HKY models (`model=3` or `4` and `nchar=4`). A real number giving the κ parameter for the model. The F84 model has a rate matrix proportional to

$$\begin{bmatrix} \cdot & \pi_C & (1 + \kappa/\pi_R)\pi_G & \pi_T \\ \pi_A & \cdot & \pi_G & (1 + \kappa/\pi_Y)\pi_T \\ (1 + \kappa/\pi_R)\pi_A & \pi_C & \cdot & \pi_T \\ \pi_A & (1 + \kappa/\pi_Y)\pi_C & \pi_G & \cdot \end{bmatrix}$$

where $\pi_R = \pi_A + \pi_G$ and $\pi_Y = \pi_C + \pi_T$. The HKY model has a rate matrix proportional to

$$\begin{bmatrix} \cdot & \pi_C & \kappa\pi_G & \pi_T \\ \pi_A & \cdot & \pi_G & \kappa\pi_T \\ \kappa\pi_A & \pi_C & \cdot & \pi_T \\ \pi_A & \kappa\pi_C & \pi_G & \cdot \end{bmatrix}$$

The transition-transversion ratio (**ttratio**) is related to the κ parameter in the F84 model through

$$R = \kappa \times \frac{\pi_A\pi_G/\pi_R + \pi_C\pi_T/\pi_Y}{\pi_R\pi_Y} + \frac{\pi_A\pi_G + \pi_C\pi_T}{\pi_R\pi_Y}$$

where $\pi_R = \pi_A + \pi_G$ and $\pi_Y = \pi_C + \pi_T$. For the HKY model the relationship is

$$R = \kappa \times \frac{\pi_A \pi_G + \pi_C \pi_T}{\pi_R \pi_Y}$$

alpha: Only required if a gamma rates-across-sites model (Yang 1994) is desired. A real value giving the shape parameter of the gamma distribution.

ncatG: Optionally, an integer giving the number of categories to use in a discrete gamma approximation. Only used if **alpha** is positive. The default is 0. If **alpha** is positive and **ncatG** is positive, a discrete gamma model is fit. If **alpha** is positive and **ncatG** is 0, a continuous gamma model is fit. The discrete gamma approximation used is the same as the default of PAML 4.2; the representative rate is the conditional mean for the class.

nrate: Optionally used to specify the number of rate classes in an arbitrary discrete rates-across-sites model. The variable is not used if **alpha** is set. If **alpha** is not set the default is to fit an equal rates model.

ratefile: Only used if **alpha** is not set and **nrate** > 1. The name of a file with rates and weights for the arbitrary discrete rates-across-sites model. Each line should give a rate followed by a corresponding weight. For example, the file

```
0.00026 0.25
0.02369 0.25
0.33291 0.25
3.64313 0.25
```

indicates a rates-across-sites model with four rates where, for instance, there is a 25% chance that a site will have a rate multiplier 3.64313. This consequently allows for implementations of the discrete gamma rates-across-sites models that give different discrete rates or weights than PAML to the same shape parameter.

nneg: Optionally 0 or 1. The default is 1. If **nneg** is 1, WLS or GLS estimation imposes non-negativity constraints on estimated edge-lengths, otherwise no constraints are imposed.

pattwt: Optionally **observed**, **single** or **multiple**. The default value is **multiple**.

As discussed in greater detail in Susko (2011), covariance matrix estimates are of the form

$$\sum_x p_x \mathbf{a}_x \mathbf{a}_x^T \quad (1)$$

where the sum is a sum over patterns x for all of the taxa in the sequence file. Estimates differ depending on the choice of p_x . With **pattwt** set to **observed**, the observed pattern frequencies are used. With **pattwt** set to **single**, the p_x are pattern probabilities on a NJ

tree with least squares edge lengths. With `pattwt` set to `multiple`, the p_x and consequently covariance matrix estimates are topology-dependent. The p_x give the pattern probabilities for the tree, with LS edge lengths, that the WLS or GLS test statistic is being calculated for. Performance tends to be better for `multiple` by comparison with `single` and `observed` but is much more computationally intensive, particularly for amino acid data. `single` is more computationally intensive than `observed` but tends to perform better, particularly for GLS estimation.

`pvalue`: Optionally 0 or 1. The default is 1. If `pvalue` is 0, p-values are not computed. This option is only recognized within the `wlsnphyl` routine. It can be used to speed computation if estimation alone is of interest.

Output

The output differs depending upon whether testing programs (`glsphyl` and `wlsnphyl`) or estimation programs (`glsphylest` and `wlsnphyl`) are used. Output trees are output according to their GLS or WLS statistics, from smallest to largest. For estimation routines, for each tree, test statistics, p-values, and estimated trees with estimated edge-lengths are output. For testing routines, in addition, the index of where the tree was ordered in the input treefile is output.

For `glsphyl` and `glsphylest`, p-values are obtained using the chi-squared distributions described in Susko (2003). For `wlsnphyl` p-values are obtained using the normal simulation methodology described in Susko (2011). For `wlspylest`, however, “p-values” are determined using a chi-squared approximation that effectively treats covariances between distances as 0. There is evidence that these values tend to be larger than should be and thus they are still informative: small p-values from `wlspylest` will likely be even smaller when calculated with `wlsnphyl`. The reason for not including proper p-values with `wlspylest` is to avoid unnecessary computation for some of the poor trees that might be encountered during tree-searching. Any number of trees from `wlspylest` can be used as input for `wlsnphyl` to obtain more appropriate p-values.

Considering the running example, we obtain

```
$ glsphyl F84n1000f1234tt2.ct1
3.051197 0.802394 (2:0.11727,(0:0.09065,1:0.10248):0.12237,...); 63
45.395996 0.000000 (2:0.13062,(3:0.11965...); 25
45.395996 0.000000 (4:0.07876,((0:0.09059,1:0.10237):0.12555,...,5:0.20552); 51
...
155.786872 0.000000 (2:0.14527,(1:0.10524,...); 4
```

which indicates that the best tree was the 63rd tree in the input treefile had a GLS test statistic value of 3.05, a p-value of 0.80. The next two best trees had very small p-values and were tied due to a zero-length internal edge that made them topologically equivalent. The 105th ranked tree was

the 4th tree in the treefile and had a test statistic of 155.79. the 95% confidence region in this case consists of a single tree. Different output resulted from `wlsnphyl` but the format was the same.

As an example of an estimation routine, consider the output

```
$ wlsphylest F84n1000f1234tt2.ct1
0.561460 0.997008 (3:0.10894,(2:0.11661,(0:0.09156,1:0.10156):0.12288...);
35.181551 0.000004 (4:0.12920,(2:0.11621,(0:0.09134,...);
...
248.310710 0.000000 (3:0.17753,(1:0.15346,2:0.18932):0.00265,...);
```

Because of the form of the control file, starting from the first tree in the treefile, the routine performed a SPR search for a better tree and recursed this process until no better tree could be found within one SPR of the current best tree. It stopped its search after evaluating WLS statistics for 36 trees and ended up estimating a tree that had a WLS test statistic value of 0.56. Because `ntrees` was set to 105 in the control file, which ended up being larger than the 36 that were considered, all 36 trees were output.

Limitations

Very few reasons for error are output. Analyses should not include identical sequences and very similar sequences can create difficulties.

References

- Adachi, J. and Hasegawa, M. (1996). Model of amino acid substitution in proteins encoded by mitochondrial DNA. *J. of Mol. Evol.* 42:459–468.
- Dayhoff, M.O., Schwartz, R.M. and Orcutt, B.C. (1978). A model of evolutionary change in proteins. pp 345–352, *Atlas of protein sequence and structure*. Vol. 5, Suppl. 3. National Biomedical Research Foundation. Washington D.C.
- Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A. and Sorensen, D. (1999). *LAPACK Users' Guide* Society for Industrial and Applied Mathematics, Philadelphia, PA.
- Felsenstein, J. (2004). *PHYLIP Phylogeny Inference Package* (version 3.6). Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.
- Felsenstein, J. (1989). *PHYLIP - Phylogeny Inference Package* (version 3.2). *Cladistics* 5: 164-166.
- Jones, D.T., Taylor, W.R., and Thornton, J.M. (1992). The rapid generation of mutation data matrices from protein sequences. *CABIOS* 8:275–282.

- Kosiol, C. and Goldman, N. (2005). Different versions of the Dayhoff rate matrix. *Mol. Biol. Evol.* 22:193–199.
- Lawson, C.L. and R.J. Hanson. (1974). Solving least squares problems. Prentice-Hall, NJ.
- Le, S.Q. and Gascuel, O. (2008). An improved general amino acid replacement matrix. *Mol. Biol. Evol.* 25:1307–1320.
- Susko, E. (2011). Improved least squares topology testing and estimation. *Syst. Biol.* 60:668–675.
- Susko, E. (2003). Confidence regions and hypothesis tests for topologies using generalized least squares. *Mol. Biol. Evol.* 20:862–868.
- Whelan, S. and Goldman, N. (2001). A general empirical model of protein evolution derived from multiple protein families using a maximum likelihood approach. *Mol. Biol. Evol.* 18:691–691.
- Yang, Z. (2007). PAML 4: a program for phylogenetic analysis by maximum likelihood. *Mol. Biol. Evol.* 24:1586–1591.
- Yang, Z. (1997). PAML: a program for phylogenetic analysis by maximum likelihood. *Comput. Appl. Biosci.* 13:555–556.
- Yang, Z. (1994). Maximum likelihood phylogenetic estimation from DNA sequences with variable rates across sites: approximate methods. *J. Mol. Evol.* 39:306–314.