Software for
Confidence Regions and Hypothesis Tests for Topologies using Generalized Least Squares

Edward Susko

Department of Mathematics and Statistics, Dalhousie University

Introduction

There are two main routines for the methods of Susko (2003).

1. `glsprot`: for amino acid data

2. `glsdna`: for DNA data

The routine `glsprot` is a modification of the protdist routine of the alpha PHYLIP distribution, version 3.6 and requires the same user input as the protdist routine in addition to a treefile for the trees that are being tested for inclusion in a confidence region. The same format of treefile is used as input to `glsdna`, the routine that is to be used with DNA data. This however is a standalone command line routine that requires an additional parameter file and the data file to be specified on the command line.

Unix installation

At the command prompt type

```
$ gzip -d gls_soft.tar.gz
$ tar xvf gls_soft.tar
```

This will create a directory `gls`. To create the executables type

```
$ cd gls
$ make
```

The `make` command creates the executable files `glsdna`, `glsprot`, `glsdna_eig` and `glsprot_eig`. After copying these files to a directory in your PATH, you can remove the `gls` directory and its contents.

The routine glsprot

The routine `glsprot` should be called at the command line with

```
$ glsprot ntrees treefile
```

where `treefile` gives the name of the file containing the trees that are to be tested for inclusion in a confidence region and `ntrees` is the number of trees in this file. The form of the treefile is described below. The routine is a modification of the protdist routine of the alpha PHYLIP distribution, version 3.6 and requires, in addition, the same user input as the protdist routine.

In addition to the usual output for this routine, a file called `glsprot.outfile` will be created. This will have the same format as the output for `glsdna` and is described below.

The routine glsdna

The routine `glsdna` should be called at the command line with

```
$ glsdna treefile paramfile infile
```

Here `infile` should be a standard PHYLIP format data file and `treefile` should be a treefile of the form described below. An example `paramfile` is given below

```
ttratio: 2.00
nrates: 8
rate: 0.0000 0.0017 0.0177 0.0830 0.2685 0.7163 1.7816 5.1312
prob: 0.1250 0.1250 0.1250 0.1250 0.1250 0.1250 0.1250 0.1250
ntrees: 105
```

Any additional lines will be ignored. Here

ttratio: The transition/transversion ratio. This is the same transition transversion ratio as would be input to `dnaml` in PHYLIP. It can be interpreted as the limiting ratio of transitions to transversions occurring along a tree, as the length of the tree gets large.

nrates: The number of rates for the rate distribution that will be used

rate: The rates for the rate distribution. There should be nrates entries that follow.

prob: The corresponding probabilities for the rate distribution.

ntrees: The number of trees in the treefile.

Different descriptive names can be given to the entries in the paramfile; for instance, `ttratio:` might be replaced by `transition.transversion.ratio:`. However, names are expected (the entry 2.00 would not be appropriate for a first line) and spaces are not allowed in descriptive names. The order of input should be the same as in the example paramfile.

The output is to the screen, stdout and is of the same format as the `glsprot.outfile` described below.

The treefile

The treefile required for both `glsdna` and `glsprot` should give the topologies that are to be tested for inclusion in the confidence region. Each of these trees should be in the bracketed format acceptable to the PHYLIP package. Listings of bootstrap supporting present in the output trees of some routines must be deleted. Each tree should end with a semi-colon, ;. The names used in the treefile should be consistent with the names present in the input data file.

The output

The output from the `glsprot` routine is to the file `glsprot.outfile` whereas for the `glsdna` routine it is to the screen, stdout. The input trees from the treefile are output after sorting them from the tree that gave the best (smallest) GLS value to the one that gave the worst (largest) value. Each row of output consists of

1. First column: The GLS objective value..
2. Second column: The p-value for a test of the null hypothesis that the topology for the row is the true topology.
3. Third column: The topology for the row.
4. Fourth column: The index of the topology in the input file (starting from 0 and going to ntrees-1).

Note that each "row" will require several lines if the individual input topologies required more than one line.

Limitations

The number of taxa considered should be less than 100. The individual trees in the input treefile should contain at most 240 characters (about 30 lines); this should allow for at least 60 taxa.

Very few reasons for error are output. With large numbers of taxa it is possible covariance matrices will become (almost) singular and the programs will crash. Removing closely related taxa prior to testing might provide a way to continue testing. An alternative is to use `glsprot_eig` and `glsdna_eig`.

The `glsdna_eig` and `glsprot_eig` routines

The `glsdna_eig` and `glsprot_eig` routines are additional routines not described in Susko (2003) that allow construction of confidence regions when the estimated covariance matrix V is not invertible. Both routines require an additional eigenvalue cutoff argument. For example

```
glsdna_eig treefile paramfile infile 0.0001
```

and

```
glsprot_eig ntrees treefile 1.0e-7
```

would implement the routines with eigenvalue cutoffs of 0.001 and 1.0e-7 respectively. The definitions of the eigenvalue cutoffs is given below but generally, while larger values have the advantage of being less susceptible to problems with non-invertible covariance matrices, they can be expected to be more conservative and include more topologies in a confidence region than is required. In cases that the `glsdna_eig` or `glsprot_eig` routines need to be used we recommend comparing results with a few choices of cutoffs.

In brief, generalized least squares is interpretable as weighted least squares for a set of approximately independent linear transformations of the distances. In the case that V is non-invertible, some subset of these linear transformations have an estimated variance of 0. The approach here is to ignore the linear transformations with estimated variance equal to 0 and make an adjustment to the degrees of freedom.

Linear transformations with variance close to 0 can be expected to have good discriminatory power. Consequently, although, similarly as in Susko (2003), it can be shown that with a large number of sites the probability that the true topology is contained in a $(1-\alpha) \times 100\%$ confidence region is approximately $1-\alpha$, the confidence region can be expected to be larger than it would have been if the linear transformations with variance close to 0 had been used.

In more detail, if the actual covariance matrix V were known and did not need to be estimated, it would have the eigenvector/eigenvalue decomposition

$$V = U\Lambda U^T \tag{1}$$

where U is an orthogonal matrix and Λ is a diagonal matrix with positive diagonal entries. It follows that the GLS test statistic can be expressed as

$$\sum (y_i^* - \mathbf{x}_i^{*T}\boldsymbol{\alpha})^2 \lambda_i^{-1} \tag{2}$$

where $\mathbf{y}^* = U\mathbf{y}$ and $\mathbf{x}_i^*$ is the ith row of the matrix $X^* = UX$. In fact, the λ_i are the variances of the y_i^* which are linear transformations of the original distances $\mathbf{y}$. Thus the GLS test statistic is a weighted least squares statistic for the linearly transformed distances.

In any case, since V is not known but rather estimated, it is possible that some of the $\lambda_i \approx 0$. These λ_i will give large contributions to (2) that are extremely sensitive to small changes in λ_i. For a given eigenvalue cutoff, c_e, `glsdna_eig` and `glsprot_eig` adjust for this by summing over all contributions in (2) that have $\lambda_i > c_e$. If there are n_p contributions in the resulting sum, then the p-value for the topology under consideration is the probability that a chi-squared random variable with $n_p - (2T-3)$ degrees of freedom is greater than the observed test statistic.

Susko, E. (2003). Confidence regions and hypothesis tests for topologies using generalized least squares. *Molecular Biology and Evolution*, **20**, 862–868.