dist_est: Estimation of Rates-Across-Sites Distributions in Phylogenetic Subsititution Models Version 1.0

Edward Susko

Department of Mathematics and Statistics, Dalhousie University

Introduction

The program `dist_est` implements some of the methods described in Susko et al. (2003); please cite this reference when using the software. The program `dist_est` obtains what may be considered non-parametric estimates (referred to as the discrete estimate (DE) throughout) and discrete gamma estimates (DGPE) as described in Susko et al. (2003). Both routines are intended for large numbers of rate categories. The routine also provides 4 different types of rate estimates. These are the conditional and mean estimates described in Susko et al. (2003) and are calculated for both the DE and DGPE rates-across-sites distributions.

Installation

The program `dist_est` is compiled from C and Fortran 77 source code. It utilizes the DCDFLIB library of Brown, Lovato and Russell, which is included in `dist_est.tar.gz` Not included in `dist_est.tar.gz` are the routines `VE11` and `ID05` which can be downloaded, free of charge, from the Harwell Subroutine Library Archive (HSL Archive) web site

`http://hsl.rl.ac.uk/archive/hslarchive.html`

The downloaded routines should be the double precision versions.
To install `dist_est`

1. Download and unpack the software:

   ```
   $ tar zxf dist_est.tar.gz
   ```

   This will create a directory `dist_est` that contains the source code and a test input file.

2. Change directories to `dist_est` and create the main program file `dist_est` with the `make` command.

   ```
   $ cd dist_est
   $ make
   ```

This should create the program file `dist_est` which can be copied to a location in your `PATH`. To test the program, still in the `dist_est` directory, issue the commands

```
$ ./dist_est hsp70.ctl
```

The output should be comparable to the output in `hsp70.out` in the same directory.

The source code has been compiled and tested using gcc and gfortran (versions 4.2.3) on an Ubuntu Linux distribution (Release 8.04). While the program has not been tested on another platform, it should compile under any Linux distribution as well as Mac OS X, assuming they have a fortran compiler installed.

Rate Distribution Estimation

The routine `dist_est` calculates both the DE rate-across-sites distribution and the α value for the corresponding DGPE rate distribution. It is called at the command line with

```
$ dist_est controlfile
```

```
$ dist_est hsp70.ctl
DE log-likelihood -9.9459903780e+03
5.4365310054e-01 -9.9566466337e+03 -6.9495999366e-06
DGPE log-likelihood -9.9566466337e+03 (alpha=0.54365)
```

Here -9.9459903780e+03 is the log likelihood for the DE rate distribution. Output after this line relates to the DGPE rate estimate. The next line gives the α estimate, log likelihood and derivative of the log-likelihood for the DGPE estimate. The derivative should be close to 0.0 if the algorithm has converged.

The rates and weights for the DE rate distribution are output to the file `DE.dat`. For instance in the example, the first few lines of this file were

```
0.0000000000000000e+00 9.2538948254778958e-02
1.0000000000000001e-01 2.3335981671868214e-01
2.0000000000000001e-01 0.0000000000000000e+00
2.9999999999999999e-01 1.8597685460683272e-01
4.0000000000000002e-01 8.7872132973107392e-02
...
```

The control file, `hsp70.ctl` specified that there were to be 101 rates in the rate distribution with an upper bound of 10.0 (see the Input section below). Thus the rates, which are given in the first column of the file, are 0.0, 0.1, 0.2, ... The second column gives the weights, or the estimates of the probabilities that a site will have these rates. For instance, the estimate of the probability that a site will have rate 0.1 is 0.23.

Rate Estimation

The routine `dist_est` also calculates rate estimates for the DGPE or DE rate estimates. The file `rate_est.dat` contains these. The ith row of the file gives the estimates for the ith site in the alignment. Each row contains four different rate estimates for the same rate. These are ordered as

DE conditional mode DE conditional mean DGPE conditional mode DGPE conditional mean

In the example, the first few lines of this file were

```
1.200000 0.918332 0.700000 0.846984
0.300000 0.287151 0.200000 0.364639
0.100000 0.152346 0.000000 0.118395
0.100000 0.203509 0.100000 0.246088
```

Input

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

```
$ dist_est controlfile
```

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, `hsp70.ctl`

```
treefile =  hsp70.fitchml.tre  * treefile
seqfile = hsp70.dat            * sequence data

nchar = 20             * amino acid data
model =  3             * empirical + F
aaRatefile = jones.dat * JTT substitution model

nrate = 101            * number of rates
ub = 10.0              * upper bound for rates
```

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:` The name of a file containing the tree of interest. This might, for instance, be the ML tree. Packages such as PHYLIP (Felsenstein, 1989, 2004), TREE-PUZZLE (Schmidt et al. 2002) and PAML (Yang 1997, 2007) can be used to obtain these.

The tree should conform to the Newick standard. The programs in PHYLIP (Felsenstein, 1989, 2004), TREE-PUZZLE (Schmidt et al. 2002) and PAML (Yang 1997, 2007), which can be used to obtain ML estimates of edge-lengths for the models described here, will output trees in this format. 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      INIISLIIPI LL...
...
```

but not

```
6 3414
Homsa
ANLLLLIVPI LI...
Phovi
INIISLIIPI 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

`model`	Model
0	JC
2	F81
3	F84
4	HKY
7	GTR

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

`model`	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 jth 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

`dayhoff.dat`	Dayhoff or PAM	Dayhoff et al. (1978)
`jones.dat`	JTT	Jones et al. (1992)
`wag.dat`	WAG	Whelan and Goldman (2001)
`mtREV24.dat`	mtREV	Adachi and Hasegawa (1996)
`lg.dat`	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

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

`nrate`: Optionally used to specify the number of rates to use in fitting the DE and DGPE rate distributions. The default is 101.

`ub`: Optionally used to specify an upper bound for the rates. The default it 10.0.

The `nrate` rates are equally spaced from 0 to `ub`. In the example, these were explicitly set to default values:

```
nrate = 101             * number of rates
ub = 10.0               * upper bound for rates
```

So that the 101 rates are 0, 0.1, 0.2, ...

`DEest`: An optional integer indicating whether DE estimation is required (`DEest = 1`) or not (`DEest = 0`). The default is 1.

Limitations

Very few reasons for error are output. Convergence criteria for DE estimation are not indicated. Because of the nature of the problem (a convex programming problem), the algorithm should converge to the global optimum.

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.

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.

Le, S.Q. and Gascuel, O. (2008). An improved general amino acid replacement matrix. Mol. Biol. Evol. 25:1307–1320.

Schmidt, H. A., Strimmer, K., Vingron, M. and von Haesler, A. (2002). TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing. Bioinformatics 18:502–504.

Susko, E., Field, C., Blouin, C. and Roger, A. (2003). Estimation of Rates-Across-Sites Distributions in Phylogenetic Substitution Models. Syst. Biol. 52:594–603.

Whelan, S. ad Goldman, N. (2001). A general empirical mdoel 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.