pbf: Software to correct Bayesian posterior probabilities of topologies
Version 1.0

Edward Susko
Department of Mathematics and Statistics, Dalhousie University

Introduction

The main programs, `pbf`, `infoprior` and `pbfs` implement the methods described in Susko (2015); please cite this reference when using the software.

The programs `pbf` and `infoprior` obtain the corrected posteriors using the *PBF correction* and *prior correction* as defined in Susko (2015). The program `pbfs` uses the output of `pbf` or `infoprior` to add corrected posteriors for splits either among the trees found during searching or for a user-supplied tree. An additional program, `mbliks`, parses the output files of MrBayes (Huelsenbeck and Ronquist, 2001; Ronquist and Huelsenbeck, 2003) to produce input files suitable for `pbf` and `infoprior`.

Installation

The main programs need to be compiled from C source code. To install the programs

1. Download and unpack the software:

   ```
   $ tar zxf pbfv1.0.tar.gz
   ```

   This will create a directory `pbfv1.0` that contains the source code.

2. Change directories to `pbfv1.0` and create the main program files `pbf`, `infoprior`, `pbfs` and `mbliks` with the `make` command.

   ```
   $ cd pbfv1.0
   $ make
   ```

The default installation assumes that the gcc compiler is available. To use a different compiler change the variable `CC` in `Makefile`.

mbliks: Obtaining input files for the main programs

The program `mbliks` obtains input files for the main programs `pbf` and `infoprior` from MrBayes output. It can be run at the command line with

```
$ mbliks -f prefix [-r run]
```

Here `prefix` is the prefix of the `*run*.t` and `*run*.p` files produced by MrBayes. Usually these are prefixed by input sequence file name but there are options for changing these; check the output for files of this form. The `-r` option is optional. By default MrBayes provides output for two independent MCMC runs and `mbliks` assumes this to be the case. The first run is analyzed with `-r 1` and the second run with `-r 2`. If `-r` is not specified, the results of both runs is used.

As a running example considered throughout this document, MrBayes was run with

```
$ mb < mb.in
```

where the file `mb.in` in that directory contained the following text

```
set autoclose=yes nowarn=y
exec infile
lset nst=1
prset statefreqpr=fixed(equal)
mcmc ngen=100000 samplefreq=10
sumt relburnin=no minpartfreq=0
quit
```

Here infile was a Nexus input file.

The `*run*` files output by MrBayes were

```
$ ls *run*
infile.run1.p  infile.run1.t  infile.run2.p  infile.run2.t
```

The `.t` files give the trees sampled in the MCMC sampling, at intervals according to the `samplefreq` variable in the `mb.in` file above. The `.p` files give the corresponding parameters. Calling `mbliks`

```
$ mbliks -f infile > mbliks.out
```

outputs to the file `mbliks.out` all of the log likelihoods and trees encountered in MCMC sampling:

```
-6.1284979999999996e+03 (1:0.10000,(0:0.10000,3:0.10000):0.10000,(2:0.10000,4:0.10000):0.10000);
-5.6224920000000002e+03 (1:0.10035,(0:0.10000,2:0.10000):0.10000,(3:0.10000,4:0.10095):0.04669);
-5.3458149999999996e+03 (1:0.10000,3:0.15486,(4:0.10000,(0:0.10509,2:0.32038):0.09826):0.03735);
-5.1837219999999998e+03 (1:0.09832,(3:0.15486,4:0.09749):0.03672,(0:0.12693,2:0.42947):0.15468);
....
```

By default MrBayes does not use all of the log likelihoods and trees encountered in MCMC sampling when computing posterior probabilities and means. It ignores some proportion of 'burnin' runs because it is to be expected that the starting trees will be far from those trees that are likely under the posterior distribution. These can be ignored by `pbf` and `infoprior` by removing as many of the lines at the start of the `mbliks.out` file as is considered appropriate; consult the MrBayes documentation for recommendations.

In the example the options `sumt relburnin=no minpartfreq=0` were included for testing purposes to ensure that all of the trees and log likelihoods encountered in MCMC sampling are used in obtaining posteriors and posterior edge lengths by MrBayes.

pbf: PBF-corrected posteriors

The `pbf` program obtains PBF-corrected posteriors for all of the trees encountered during MCMC sampling. It can be run at the command line with

```
$ pbf -f filename -n number_of_taxa
```

Here `number_of_taxa` is the number of taxa considered and must be present. `filename` is the name of file giving the log likelihoods and trees from MCMC sampling. Each line of the file should contain a log likelihood and the corresponding tree. If MrBayes is used, this file can be created from the MrBayes output using `mbliks`. For instance, in the running example the first few lines of the output file, `mbliks.out`, from `mbliks` was

```
-6.1284979999999996e+03 (1:0.10000,(0:0.10000,3:0.10000):0.10000,(2:0.10000,4:0.10000):0.10000);
-5.6224920000000002e+03 (1:0.10035,(0:0.10000,2:0.10000):0.10000,(3:0.10000,4:0.10095):0.04669);
-5.3458149999999996e+03 (1:0.10000,3:0.15486,(4:0.10000,(0:0.10509,2:0.32038):0.09826):0.03735);
-5.1837219999999998e+03 (1:0.09832,(3:0.15486,4:0.09749):0.03672,(0:0.12693,2:0.42947):0.15468);
....
```

Output from `pbf` is to the screen or stdout. Each line gives the posterior, PBF-corrected posterior and corresponding tree with posterior mean edge-lengths. The lines are sorted from largest to smallest PBF-corrected posterior. In the running example,

```
$ pbf -f mbliks.out -n 5
```

gave output with the first three lines

```
0.109339 0.206972 (0:0.99296,(1:0.04334,2:0.85923):0.01730,(3:0.05653,4:0.05485):0.01158);
0.098040 0.153645 ((0:0.99296,3:0.05653):0.00861,(1:0.04334,2:0.85923):0.01730,4:0.05485);
0.081142 0.149138 (0:0.99296,(3:0.05653,(1:0.04334,2:0.85923):0.01730):0.01729,4:0.05485);
```

The top ranked tree according to PBF had pbf-corrected posterior 0.206972 and posterior 0.109339.

infoprior: prior-corrected posteriors

The `infoprior` program corrects posteriors by using priors for topologies that vary across topologies. It assumes `pbf` has been run and that its output has been stored in a file. It can be run at the command line with

```
$ infoprior controlfile
```

Input

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. In the running example, the following file was used

```
treefile = pbf0.out     * pbf output file
seqfile = infilep       * sequence data file
nchar = 4               * nucleotide data
model = 0               * Jukes Cantor model
```

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 the file containing the output of `pbf`.

`seqfile`: The name of the file containing the sequence data. While MrBayes uses Nexus input files, a PHYLIP format file is required here. Prior to any analysis using `infoprior`, identical sequences should be removed as their presence will create serious numerical difficulties. It is possible that extremely similar sequences could create similar numerical difficulties. 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.

`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.

`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.

`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. This is used as an initial value for estimation of α.

`ncatG`: Only used if a discrete gamma rates-across-sites model was fit. Optionally, an integer giving the number of categories to use in the discrete approximation. The default is 4. 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.

Output

The output (to the screen or stdout) is similar to pbf. Each line gives the posterior, corrected posterior and corresponding tree with posterior mean edge-lengths. The lines are sorted from largest to smallest corrected posterior. In the running example,

```
$ infoprior controlfile
```

gave output with the first three lines

```
0.109339 0.260860 (0:0.99296,(1:0.04334,2:0.85923):0.01730,(3:0.05653,4:0.05485):0.01158);
0.098040 0.179445 ((0:0.99296,3:0.05653):0.00861,(1:0.04334,2:0.85923):0.01730,4:0.05485);
0.081142 0.169321 (0:0.99296,(3:0.05653,(1:0.04334,2:0.85923):0.01730):0.01729,4:0.05485);
```

The top ranked tree according to the prior correction is the same as for PBF and has corrected posterior 0.260860.

pbfs: PBF-corrected or prior-corrected posteriors for splits

The `pbfs` program outputs Newick format trees with corrected-posteriors for splits as labels. It takes as input either the output of `pbf` or `infoprior`. It can be run at the command line with

```
$ pbfs [-t treefile] -f output_file -n number_of_taxa
```

Here `number_of_taxa` is the number of taxa considered and must be present as must `output_file` which is the name of the file giving `pbf` or `infoprior` output. By default, when the `treefile` option is not specified, the output to the screen or stdout) is similar to pbf. Each line gives the posterior, corrected posterior and corresponding tree with posterior mean edge-lengths. The lines are sorted from largest to smallest corrected posterior as before. The change is that support values for splits are added. Considering the running example, using the output file `pbf0.out` created by `pbf`, we obtain

```
$ pbfs -f pbf0.out -n 5
0.109339 0.206972 ((3:0.05653,4:0.05485)0.377:0.01158,(1:0.04334,2:0.85923)0.510:0.01730,0:0.99296
0.098040 0.153645 (4:0.05485,(1:0.04334,2:0.85923)0.510:0.01730,(0:0.99296,3:0.05653)0.210:0.00861
0.081142 0.149138 (4:0.05485,(3:0.05653,(1:0.04334,2:0.85923)0.510:0.01730)0.212:0.01729,0:0.99296
...
```

From which we see, for instance, that in the top-ranked tree, the corrected posterior for the split 34|012 is 0.377 and for 12|034 it is 0.510.

The option `-t treefile` allows one to specify trees that corrected posteriors for splits should be added to. This can be useful when the location of hypothesized trees of interest in `pbf` or `infoprior` output files is not clear. The input trees 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.

Caveats and Warnings

Most of the testing of the routines has been with smaller numbers of taxa, so problems could occur with larger numbers of taxa, where storage requirements and numerical issues are more difficult to adjust for. In particular, a difficulty with `infoprior` is identical sequences. These lead to determinants, $|J_{jn}|^{1/2}$ that are all 0. Because of roundoff, the α_j may still be computable but will be meaningless.

Most of the testing has also been with simple models. It is possible that `mbliks` will have difficulty parsing some `*p` files corresponding to models that have not been tested.

References

Huelsenbeck J.P., Ronquist F. 2001. MRBAYES: Bayesian inference of phylogenetic trees. Bioinf. 17:754–755.

Ronquist F., Huelsenbeck J.P. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinf. 19:1572–1574.

Susko, E. (2015). Bayesian Long Branch Attraction Bias and Corrections 64: 243–255.

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.