Phylogenetic comparative analysis in BayesTraits*

Laura Fortunato

Santa Fe Institute
1399 Hyde Park Road
Santa Fe, NM 87501, USA

fortunato@santafe.edu

Contents

The BayesTraits package is freely available for download from `http://www.evolution.rdg.ac.uk/BayesTraits.html`. Unless otherwise specified, the information in these notes is based on Pagel and Meade (2005, 2006), Pagel et al. (2004), and on the BayesTraits manual (Pagel and Meade n.d.).

*Version v0.1, last revised Monday 31st May, 2010.

1 Introduction

The BayesTraits package includes:

- BayesMultistate, a method for the analysis of one or more traits that adopt two or more discrete states; its main application is the estimation of ancestral states of the trait(s) on a tree and the testing of hypothesis about the reconstructions;
- BayesDiscrete, a method for the analysis of two binary traits (i.e. two traits that adopt two states each); its main application is the testing of hypotheses of co-evolution between the traits;
- BayesContinuous, a method to do all the above (and more!) with continuously varying traits.

Each method can be implemented in either maximum likelihood (ML) or Markov chain Monte Carlo (MCMC) mode. Because we have limited time, we will be focusing on the simplest case of one binary trait and its analysis in BayesMultistate; the general concepts are easily extended to more complicated scenarios. In any case, for specific information on capabilities of the programmes not covered here, I highly recommend that you carefully study the BayesTraits manual and the papers describing the methods, which are posted on the wiki.

I use different coloured boxes for exercises (marked **EXE**), notes (marked **!!!**), and tips (marked **TIP**).

2 Getting started

This section aims to teach you a few basics, like setting up and running analyses, and to help you gain familiarity with the manual.

2.1 Set-up and file description

You should have downloaded a folder containing the appropriate version of BayesTraits from `http://www.evolution.reading.ac.uk/BayesTraits.html`, and folder bayesTraitsPractical from the wiki. Next, create a copy of the programme in the BayesTraits folder and move it to the bayesTraitsPractical folder. In general, to save some typing you will need a copy of the programme in the folder containing the tree and data files.

We will be using as an example the primate data distributed with BayesTraits; more information on these data is in the BayesTraits manual and in Pagel and Meade (2006). For all analyses we will use the files `primates.trees` and `mating.txt` included in the bayesTraitsPractical folder.

The tree file `primates.trees` is a posterior probability sample of 500 phylogenetic trees for 60 primate taxa. Trees are present in the sample in proportion to their posterior probability, which is

the probability of the tree conditional on the data and model of molecular evolution used in the tree-building analysis, and can be interpreted as the probability that the tree is correct (Huelsenbeck et al. 2001). The consensus tree in Figure 1 summarizes the 500 trees in the sample. For example, the great ape species included in the tree-building analysis (*Pan paniscus*, *P. troglodytes*, *Homo sapiens*, *Gorilla gorilla*, *Pongo pygmaeus*, and *P. pygmaeus abelii*) share an ancestor in 99% of the trees in the sample (indicated by the arrow); the probability that they are a "monophyletic" group is thus 0.99, given the data and model of molecular evolution used in the tree-building analysis. The degree of phylogenetic uncertainty at several of the nodes on the consensus tree emphasizes the importance of incorporating this uncertainty in the comparative analysis, by using a tree sample instead of a single "best" tree.

The data file `mating.txt` provides comparative data for the trait "mating system" for the 60 primate taxa, coded as "single-male" (state 0) or "multi-male" (state 1).

2.2 Running BayesTraits

On a Mac, launch a Terminal window from Applications>Utilities; on a Windows machine, launch a Command Prompt window from Start>Programs>Accessories. Navigate to the bayesTraitsPractical folder from within the Terminal/Command Prompt window — contact an instructor if you need help with this.

The syntax to start BayesTraits and to get it to read the files is (Mac):

`./BayesTraits` {⟨*tree file name*⟩} {⟨*data file name*⟩}

or (Windows):

`BayesTraits` {⟨*tree file name*⟩} {⟨*data file name*⟩}

where `./` is Unix for "in the current directory", ⟨*tree file name*⟩ is a plain text file in NEXUS format containing one (or more) rooted, bifurcating tree(s) with branch lengths, and ⟨*data file name*⟩ is a plain text file listing the comparative data. Roughly, this can be translated as: "in the current directory, execute BayesTraits, reading files ⟨*tree file name*⟩ and ⟨*data file name*⟩". For example, at the command prompt type (Mac):

```
./BayesTraits primates.trees mating.txt
```

or (Windows):

```
BayesTraits primates.trees mating.txt
```

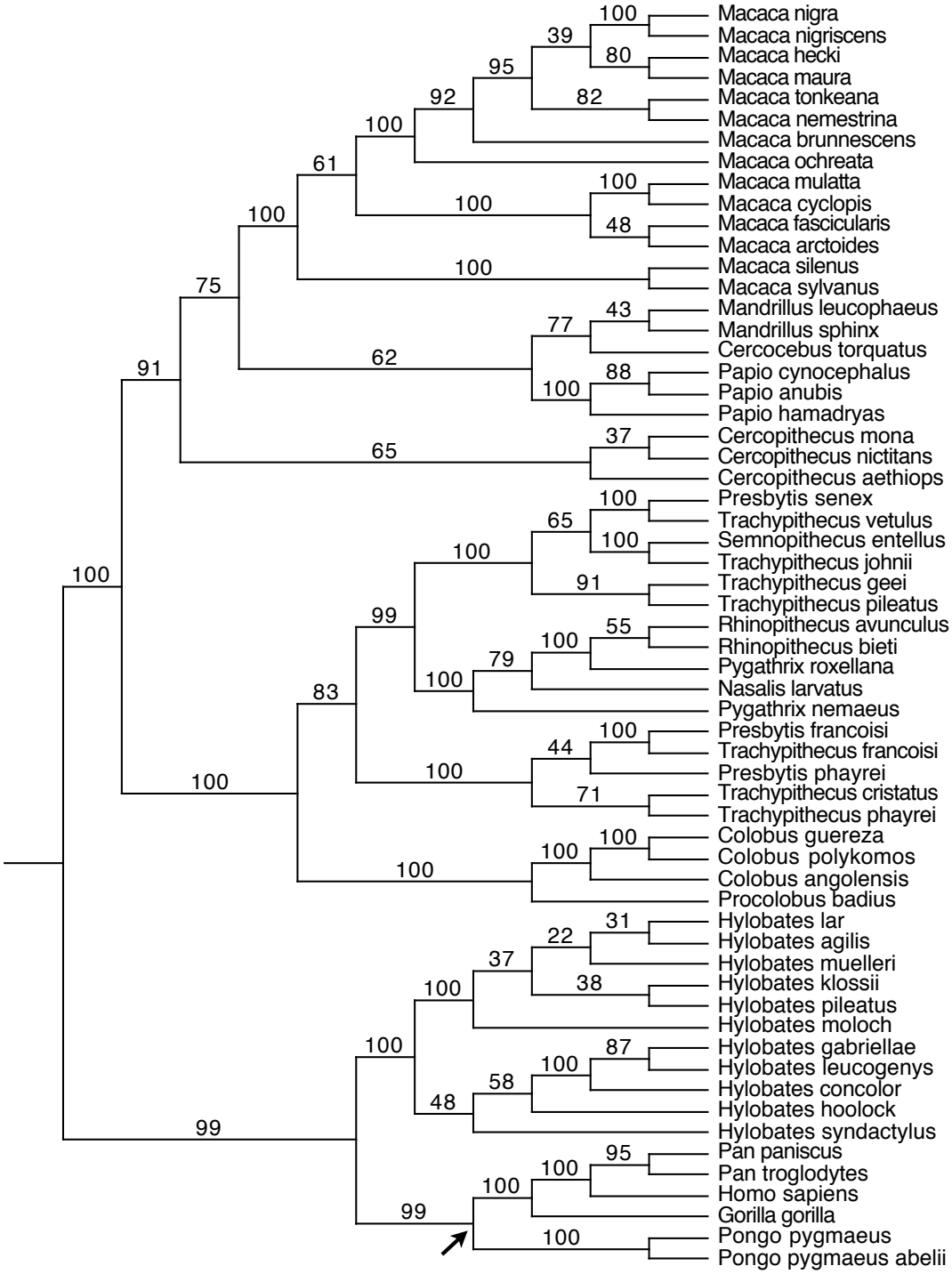

Figure 1: Majority-rule consensus tree of Pagel and Meade's (2006) sample of 500 trees for 60 primate species. The tree includes nodes present in $> 50\%$ of trees in the sample, plus other compatible groupings; the value above each node is the node's posterior probability as a percentage. The arrow marks the ancestor of the great apes.

TIP As you become more familiar with BayesTraits, consider running analyses in batch mode (as described in the manual) rather than from the command line interface. This will save a lot of typing!

After you hit ↵, BayesTraits prints the following to screen:

```
Rand Seed 1274733997
Please Select the model of evolution to use.
1) MultiState.
2) Discrete: Independent
3) Discrete: Depend
4) Continuous: Random Walk (Model A)
5) Continuous: Directional (Model B)
```

The first line, `Rand Seed`, provides a random number from which the analysis is "seeded", so obviously it will likely be different from the value shown here. What follows is a list of the different methods implemented in BayesTraits; you choose the desired method by typing the corresponding number. For example, type 1 and hit ↵. The following is printed to screen:

```
Please Select the analsis method to use.
1) Maximum Likelihood.
2) MCMC
```

This lists the two modes implemented by BayesTraits, ML and MCMC. As before, you select the desired mode by typing the corresponding number. For example, type 2 and hit ↵. BayesTraits prints the following to screen:

```
Options:
Model:                          Multistates
Tree File Name:                 primates.trees
Data File Name:                 mating.txt
Log File Name:                  mating.txt.log.txt
Summary:                        False
Analysis Type:                  MCMC
Sample Period:                  100
Iterations:                     5050000
Burn in:                        50000
Rate Dev:                       2.000000
No of Rates:                    2
Base frequency (PI's)           None
```

```
Character Symbols                0,1
Using a covarion model:          False
Restrictions:
    q01                          None
    q10                          None
Prior Information:
    Prior Categories:            100
    q01                          uniform 0.00 100.00
    q10                          uniform 0.00 100.00
Tree Information
     Trees:                      500
     Taxa:                       60
     Sites:                      1
     States:                     2
```

The first 11 lines warn you about branch length errors, and are not shown here — you can ignore these for present purposes. Under the heading `Options` is a summary of the information relating to the analysis, such as the names of the tree, data, and output files, and the default settings. You can get BayesTraits to print this information at any time before the analysis is run with command `info` (`in` for short).

!!! By default, BayesTraits generates the name of the output file by appending the suffix `.log.txt` to the name of the data file. If a file with this name already exists in the current directory (e.g. from a previous run), it will be overwritten *without warning* at the start of a new run. So, when setting up a new analysis remember to change the name of the output file to something more meaningful (e.g. by adding the run number). This is done with command `logfile` (`lf` for short) followed by the desired name.

We will change some of the default settings in Section 2.3. At this point we can start the analysis — this is done with command `run` (`ru` for short). The run will end when the programme reaches the specified number of iterations (5050000 in this case); alternatively, you can stop it at any time with Ctrl + C .

TIP A full list of the commands available in BayesTraits, with information on their use, is in the appendix at the end of the manual. Before starting a run, you can get BayesTraits to print out a list of command names (long and short forms) with command `help` (`he` for short).

EXE A little practice in running BayesTraits, and finding your way through the manual.

1. Set up a run using BayesMultistate in MCMC. Change the name of the output file to `matingRun01.txt`, then ensure that the change has been made, and start the analysis. Stop the run, and ensure that the output file has been saved in the current directory.
2. *[Optional.]* Open the output file in a text editor or spreadsheet programme (recommended — contact an instructor if you need help with this) and, with the help of the manual, work out what the different column headings represent.

2.3 Modifying default settings

This section aims to teach you how to modify the default settings for some of the MCMC chain specifications. These specifications determine whether the chain samples parameter space adequately and ultimately converges to the posterior probability distribution of states in the model of trait evolution.

Crucially, the validity of any inference you draw from your analyses depends on convergence of the chain to its stationary distribution, that is, to the posterior probability distribution of the parameters of interest to the comparative question; in turn, this depends on the ability of the chain to wander through "state space" effectively. Assessment of convergence is briefly described in Section 2.4.

!!! Because of time constraints, we will use the MCMC chain specifications provided in the BayesTraits manual for the primate example. In real life, however, you will need to determine the chain specifications for your analyses through several preliminary runs, both in ML and MCMC mode. You can find more information on the "art" of determining MCMC chain specifications in the manual.

Some of the default settings that you may consider modifying are discussed below; additional settings are described in the manual.

Rate deviation At each step in an MCMC chain BayesTraits changes the values of the rate parameters by an amount specified by the rate deviation parameter, set with command `ratedev` (`rd` for short) followed by the desired value. The mean acceptance rate should be between 20 and 40% of the proposed changes: at lower rates the chain may fail to explore state space effectively, while at higher rates it would accept nearly all proposed states, such that successive states would be highly correlated.

TIP You will need to determine an appropriate value of the rate deviation parameter by running a series of preliminary runs in which you adjust this parameter iteratively, until the mean acceptance rate falls within the 20 to 40% interval. If all other settings are left unchanged, larger values of the rate deviation will produce lower acceptance rates, and vice-versa. Use a standard statistics package to compute the mean acceptance rate from the last column of the output file.

Sampling Some degree of auto-correlation between successive states is inevitable; therefore, the chain is "thinned" with command `sample` (`sa` for short) followed by the desired sampling frequency, which sets the interval for sampling the states visited by the chain. Wide intervals ensure the near-independence of successive states sampled by the chain.

TIP After you have determined all of the chain specifications, use a standard statistics package to ensure that the auto-correlation coefficient between successive states is an arbitrarily small number.

Burn-in Because the chain is started from a random state, it will usually take some time to find regions of state space with high posterior probabilities. Convergence of the chain to the posterior probability distribution, that is, to its stationary distribution, is assessed by plotting the $\log_e$(likelihood) values of the sampled states against iteration (Section 2.4). In a typical run the $\log_e$(likelihood) values increase steadily and then fluctuate randomly up and down around a stable value. States sampled during the initial climbing phase, known as "burn-in period", are discarded; in BayesTraits the length of the burn-in is set with command `burnin` (`bi` for short) followed by the desired number of burn-in iterations. The iterations belonging to the burn-in phase are not included in the output.

TIP You will need to determine the appropriate length for the burn-in by running a series of preliminary runs in which you adjust its value iteratively. Use a standard statistics package to plot the $\log_e$(likelihood) values of the sampled states against iteration.

Length of the chain Whether the chain does reach convergence largely depends on the length of the walk (Section 2.4), which is set with command `iterations` (`it` for short) followed by the desired number of iterations.

!!! `it` specifies the *total* number of iterations, including any iterations belonging to the burn-in phase (specified with `bi`). This means that if you want a chain to run for x iterations post-burn-in, with a burn-in of y iterations, you will need to set `it z`, where `z` $= x + y$. For example, if you want a chain to run for 5000000 iterations post-burn-in, with a burn-in of 50000, you will need to set `it 5050000`.

Priors The posterior probability of a parameter value is a quantity proportional to the product of its likelihood and prior probability. At present, BayesTraits implements three prior probability distributions on the rate parameters: uniform, exponential, and gamma. A *uniform* prior, specified by a range, is used if all values of the rate parameters are believed to be equally likely. An *exponential* prior, specified by a mean, is used if small values of the rate parameters are believed to be more likely than large ones. A *gamma* prior, specified by a mean and variance, is used if the distribution of parameter values is believed to be right skewed, with small or small to intermediate values of the rate parameters more likely than large ones. Selection of prior probability distributions represents an issue for concern in the Bayesian approach, because it involves some degree of arbitrariness. One option is to use uniform ("uninformative") priors; unless the signal in the data is particularly strong, however, this strategy may result in the chains not visiting state space effectively. "Informative", i.e. non-uniform, priors are used in this case. Having specified the shape of the prior distribution (exponential or gamma), you can remain agnostic about the values of its parameters (i.e. about the value of the mean for the exponential and of the mean and variance for the gamma), by making BayesTraits estimate them from the data. In this case you specify a "hyperprior", itself a uniform distribution specified by a range; BayesTraits draws values at random from the hyperprior and uses them to seed the parameters of the desired prior distribution. Which command(s) you use to specify the priors depends on a number of factors, for example on whether you are using BayesTraits in MCMC or reversible jump (RJ) MCMC mode; on whether or not you are using a hyperprior; and on whether you want to use the same prior for all rates, or different priors for different rates. All the available commands are listed in the appendix to the manual.

TIP If you decide to use non-uniform priors, run several preliminary analyses with different combinations of prior settings (e.g. different parameters while keeping the shape constant, and vice-versa). If results vary substantially, then you should (be able to) justify your choice of priors, for example based on evidence from previous studies.

EXE A little practice in modifying default settings and interpreting output files.

1. Set up a run using BayesMultistate in MCMC mode. Set the chain to run for $10^7 = 10000000$ iterations, sampling every $10^3 = 1000$, with an *additional* burn-in of $10^5 = 100000$, and rate deviation set to 8. Use an exponential prior on the rate parameters, with mean seeded from a uniform RJ hyperprior on the interval 0–30. Name the output file `matingRun02.txt`, then ensure that all the desired changes have been made, and start the analysis. Stop the run, and ensure that the output file has been saved in the current directory. *[Hint: use the command* `rjhp` *for the prior.]*
2. From the number of iterations, the sampling period, and the length of the burn-in, determine how many points would have been sampled by the chain if the analysis had run to completion. Check your result against the completed runs in the bayesTraitsPractical>outputFiles folder.
3. Open the output file in a text editor or spreadsheet programme (recommended — contact an instructor if you need help with this) and, with the help of the manual, work out what the different column headings represent. *[Optional.]* Identify and explain any differences with the column headings in the output file from the previous run (file `matingRun01.txt`).

2.4 Assessing convergence

By the very nature of the MCMC process, it is not possible to know in advance how long it takes for the chain to reach convergence — hence the reference to the Monte Carlo casino in the name of the procedure (Felsenstein 2004, p. 292). Relatedly, the chain may fail to explore state space effectively, producing a sample that does not reflect the posterior probability distribution even after very long runs. For example, a chain may be "stuck" in one region of state space for a large number of iterations after the burn-in phase, failing to visit regions with higher posterior probabilities. For this reason, you will need to run a number of separate chains, each started from a different random state; if the chains produce comparable samples, they likely converged to the same region in state space, i.e. to their stationary distribution — in this case, it is safe to assume that the distribution of states in the samples approximates the posterior probability distribution.

You can assess convergence to the posterior probability distribution by comparing the samples returned by the separate chains, through visual inspection of (i) time-series plots of $\log_e$(likelihood) values [i.e. plots of the $\log_e$(likelihood) values of the sampled states against iteration; for an example, see file matingRun02.xls in the bayesTraitsPractical>outputFiles folder], (ii) the average deviation of parameter estimates across runs, and (iii) the posterior probability distributions of model categories (in RJ-MCMC mode only).

3 Ancestral state reconstruction

This section aims to demonstrate the different techniques available in BayesTraits for ancestral state estimation (Section 3.1) and for testing hypotheses about the reconstructions (Section 3.2).

3.1 Estimation at internal nodes and MRCAs

Given a sample of trees, you can use BayesMultistate to estimate the posterior probability distributions of the different states for a trait (i) at internal nodes on the consensus tree summarizing the tree sample, or (ii) at "most recent common ancestor" (MRCA) nodes. Say for example that you wanted to reconstruct ancestral states for the trait "mating system" for the node corresponding to the ancestor of the great ape species (Section 2.1). Since these species form a monophyletic group in only in 99% of trees in the sample, a node corresponding to their ancestor does not exist in the remaining 1% of trees. This means that no such reconstruction is possible for this 1% of trees. In practice, if the MCMC chain lands on one of these trees during an analysis reconstructing this node, BayesTraits prints out `--` in the corresponding columns in the output. This indicates that it cannot provide estimates of the posterior probabilities of the two states at the node, since the node does not exist on that tree.

An alternative approach is to reconstruct the node representing the MRCA of the taxa of interest. By definition a node representing the MRCA of a set of taxa exists in all trees in a tree sample: in some it includes only the taxa of interest, while in others it includes additional taxa besides the taxa of interest. Thus, the node corresponding to the MRCA of the great apes includes only the great ape species in 99% of trees in the sample, while in the remaining 1% of trees it includes additional taxa besides the great ape species.

Now a bit of theory, again for the simplest case of a binary trait taking states 0 and 1. The means of the posterior probability distributions of the two states at a given node, denoted $p(0|\text{node})$ and $p(1|\text{node})$, are multiplied by the posterior probability of the node itself, denoted $p(\text{node})$, to produce the combined probabilities of the two states at the node, denoted $p(0)$ and $p(1)$; $p(0|\text{node}) + p(1|\text{node}) = 1$, thus $p(0) + p(1) = p(\text{node})$. This means that if reconstruction of the node itself is uncertain, i.e. if $p(\text{node}) < 1$, the value of $p(\text{node})$ sets an upper limit to the confidence that can be placed in the ancestral state reconstructions for the node. Of course, this will only affect reconstructions at internal nodes: since, by definition, a node representing the MRCA of a set of taxa exists in all trees in a tree sample, MRCAs have $p(\text{node}) = 1$, such that $p(0|\text{node}) = p(0)$ and $p(1|\text{node}) = p(1)$.

TIP As a rule of thumb, place confidence in reconstructions with *combined* probabilities ≥ 0.70.

Internal node and MRCA reconstructions are added with commands `addnode` (`addn` for short) and `addmrca` (`mrca` for short), respectively. The syntax is:

`addnode` {⟨*node name*⟩} {⟨*taxa list*⟩}

and

`addmrca` {⟨*node name*⟩} {⟨*taxa list*⟩}

where ⟨*node name*⟩ is an arbitrary designation for the internal node/MRCA and ⟨*taxa list*⟩ is a list of the taxa of interest (separated by blanks).

TIP The translation matrix at the beginning of the tree file specifies numbers corresponding to the taxa names. To save some typing when adding internal node/MRCA reconstructions, in ⟨*taxa list*⟩ you can use these numbers instead of the taxa names.

EXE Here you get to practise the two techniques for estimating ancestral states.

1. Set up a run using BayesMultistate in MCMC mode, with the same settings used for the previous run. Add internal node and MRCA reconstructions for the great ape species (listed in Section 2.1), designated respectively `GAR` and `GAM`. Name the output file `matingRun03.txt`, then ensure that all the desired changes have been made, and start the analysis. Stop the run, and ensure that the output file has been saved in the current directory.
2. Open the output file in a text editor or spreadsheet programme and, with the help of the manual, work out what the different column headings represent, focusing on the columns reporting node and MRCA estimates.
3. From the output file in the bayesTraitsPractical>outputFiles folder, use a standard statistics programme to work out the mean of the posterior probability distributions of the two states for the internal node and for the MRCA. Use these values to work out the combined probabilities of the two states for both nodes. Identify and explain the difference between the two reconstructions, if any. Discuss how this difference would be affected if the node corresponding to the ancestor of the great ape species had $p(\text{node}) = 0.65$ instead of $p(\text{node}) = 0.99$.

3.2 Fossilization

In addition to reconstructing ancestral states at internal nodes and MRCAs, you may want to assess explicitly the relative "fit" of the different states of the trait at nodes of interest. For a each node, this involves running one set of analyses with the node fixed ("fossilized") to state 0 and one with the node fixed to state 1. The posterior probability distributions of $\log_e$(likelihood) values sampled by the chains reflect how well a given fossil state fits the node; a measure called the "Bayes factor", which is used to compare posterior probability distributions (Kass and Raftery 1995; Raftery 1996), provides an indication of the strength of the evidence in favour of one state over the other at the node. The Bayes factor for state 0 over state 1 is denoted B_{01}. $2\log_e(B_{01})$ is approximated as twice

the difference between $\log_e[H(\text{likelihood})]$ for a chain fixed on state 0 and $\log_e[H(\text{likelihood})]$ for a chain fixed on state 1, where $\log_e[H(\text{likelihood})]$ is the natural logarithm of the harmonic mean of the likelihood values. In theory, values of $2\log_e(B_{01}) > 0$ represent evidence for state 0 and values of $2\log_e(B_{01}) < 0$ evidence for state 1. Specifically, the evidence for a given state is "weak" for $0 < |2\log_e(B_{01})| < 2$, "positive" for $2 < |2\log_e(B_{01})| < 5$, "strong" for $5 < |2\log_e(B_{01})| < 10$, "very strong" for $|2\log_e(B_{01})| > 10$ (Raftery 1996, p. 165). In practice, however, harmonic means of likelihood values may vary across runs: they are expected to converge to the same value if the chains are run to infinity. Consequently, one conservative approach is to disregard any evidence for either state given by $|2\log_e(B_{01})| < 2$.

TIP As the MCMC chain runs, BayesTraits prints out the natural logarithm of the running harmonic mean of the likelihood values for the post-burn-in phase; this is the third column of the output files. $2\log_e(B_{01})$ is obtained as twice the difference between the last entry in this column for a chain fixed on state 0 and the last entry in this column for a chain fixed on state 1.

Fossilizations are added with command `fossil` (`fo` for short). The syntax is:

`fossil` {⟨*node name*⟩} {⟨*fossil state*⟩} {⟨*taxa list*⟩}

where ⟨*node name*⟩ is an arbitrary designation for the node you are fossilizing, ⟨*fossil state*⟩ is the state you are fossilizing the node to, and ⟨*taxa list*⟩ is a list of the taxa of interest (taxa names or the corresponding numbers, separated by blanks).

EXE Here you get to practise BayesTraits's fossilization technique.

1. Set up a run using BayesMultistate in MCMC mode, with the same settings used for the previous run. Fossilize to state 0 the node corresponding to the ancestor of the great ape species (listed in Section 2.1), designated `GA0`. Name the output file `matingRun04.txt`, then ensure that all the desired changes have been made, and start the analysis. Stop the run, and ensure that the output file has been saved in the current directory.
2. Repeat the previous step, but this time fossilize the node to state 1, designated `GA1`, and name the output file `matingRun05.txt`.
3. Open the output files in a text editor or spreadsheet programme and, with the help of the manual, work out what the different column headings represent, focusing on the columns reporting the fossilizations.
4. From the output files in the bayesTraitsPractical>outputFiles folder, calculate $2\log_e(B_{01})$, and determine the strength of the evidence in favour of one of the two states, if any. Compare this result with results from the ancestral state reconstruction analyses.

References

Felsenstein, J. (2004). *Inferring phylogenies*. Sinauer Associates, Inc. Sunderland, MA.

Huelsenbeck, J. P., Ronquist, F., Nielsen, R., and Bollback, J. P. (2001). Bayesian inference of phylogeny and its impact on evolutionary biology. *Science*, 294(5550):2310–2314.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Pagel, M. and Meade, A. (2005). Bayesian estimation of correlated evolution across cultures: a case study of marriage systems and wealth transfer at marriage. In Mace, R., Holden, C. J., and Shennan, S., editors, *The evolution of cultural diversity: a phylogenetic approach*, chapter 13, pages 235–256. UCL Press, London.

Pagel, M. and Meade, A. (2006). Bayesian analysis of correlated evolution of discrete characters by reversible-jump Markov chain Monte Carlo. *The American Naturalist*, 167(6):808–825.

Pagel, M. and Meade, A. (n.d.). BayesTraits *manual (version 1.0)*. School of Biological Sciences, University of Reading. Retrieved September 1, 2008, from `http://www.evolution.reading.ac.uk/BayesTraits.html`.

Pagel, M., Meade, A., and Barker, D. (2004). Bayesian estimation of ancestral character states on phylogenies. *Systematic Biology*, 53(5):673–684.

Raftery, A. E. (1996). Hypothesis testing and model selection. In Gilks, W. R., Richardson, S., and Spiegelhalter, D. J., editors, *Markov chain Monte Carlo in practice*, chapter 10, pages 163–187. Chapman & Hall/CRC, London.