

## Bayesian phylogenetic analysis using MRBAYES

### THEORY

Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck

#### 7.1 Introduction

What is the probability that Sweden will win next year's world championships in ice hockey? If you're a hockey fan, you probably already have a good idea, but even if you couldn't care less about the game, a quick perusal of the world championship medalists the last 15 years (Table 7.1) would allow you to make an educated guess. Clearly, Sweden is one of only a small number of teams that compete successfully for the medals. Let's assume that all seven medalists the last 15 years have the same chance of winning, and that the probability of an outsider winning is negligible. Then the odds of Sweden winning would be 1:7 or 0.14. We can also calculate the frequency of Swedish victories in the past. Two gold medals in 15 years would give us the number 2:15 or 0.13, very close to the previous estimate. The exact probability is difficult to determine but most people would probably agree that it is likely to be in the vicinity of these estimates.

You can use this information to make sensible decisions. If somebody offered you to bet on Sweden winning the world championships at the odds 1:10, for instance, you might not be interested because the return on the bet would be close to your estimate of the probability. However, if you were offered the odds 1:100, you might be tempted to go for it, wouldn't you?

As the available information changes, you are likely to change your assessment of the probabilities. Let's assume, for instance, that the Swedish team made it to

*The Phylogenetic Handbook: a Practical Approach to Phylogenetic Analysis and Hypothesis Testing*,  
Philippe Lemey, Marco Salemi, and Anne-Mieke Vandamme (eds.) Published by Cambridge  
University Press. © Cambridge University Press 2009.

**211 Bayesian phylogenetic analysis using MRBAYES: theory**

**Table 7.1** Medalists in the ice hockey world championships 1993–2007

Year	Gold	Silver	Bronze
1993	Russia	Sweden	Czech Republic
1994	Canada	Finland	Sweden
1995	Finland	Sweden	Canada
1996	Czech Republic	Canada	United States
1997	Canada	Sweden	Czech Republic
1998	Sweden	Finland	Czech Republic
1999	Czech Republic	Finland	Sweden
2000	Czech Republic	Slovakia	Finland
2001	Czech Republic	Finland	Sweden
2002	Slovakia	Russia	Sweden
2003	Canada	Sweden	Slovakia
2004	Canada	Sweden	United States
2005	Czech Republic	Canada	Russia
2006	Sweden	Czech Republic	Finland
2007	Canada	Finland	Russia

the finals. Now you would probably consider the chance of a Swedish victory to be much higher than your initial guess, perhaps close to 0.5. If Sweden lost in the semifinals, however, the chance of a Swedish victory would be gone; the probability would be 0.

This way of reasoning about probabilities and updating them as new information becomes available is intuitively appealing to most people and it is clearly related to rational behavior. It also happens to exemplify the Bayesian approach to science. Bayesian inference is just a mathematical formalization of a decision process that most of us use without reflecting on it; it is nothing more than a probability analysis. In that sense, Bayesian inference is much simpler than classical statistical methods, which rely on sampling theory, asymptotic behavior, statistical significance, and other esoteric concepts.

The first mathematical formulation of the Bayesian approach is attributed to Thomas Bayes (c. 1702–1761), a British mathematician and Presbyterian minister. He studied logic and theology at the University of Edinburgh; as a Non-Conformist, Oxford and Cambridge were closed to him. The only scientific work he published during his lifetime was a defense of Isaac Newton’s calculus against a contemporaneous critic (*Introduction to the Doctrine of Fluxions*, published anonymously in 1736), which apparently got him elected as a Fellow of the Royal Society in 1742. However, it is his solution to a problem in so-called inverse probability that made him famous. It was published posthumously in 1764 by his friend Richard Price in the *Essay Towards Solving a Problem in the Doctrine of Chances*.

**212 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

Assume we have an urn with a large number of balls, some of which are white and some of which are black. Given that we know the proportion of white balls, what is the probability of drawing, say, five white and five black balls in ten draws? This is a problem in forward probability. Thomas Bayes solved an example of the converse of such problems. Given a particular sample of white and black balls, what can we say about the proportion of white balls in the urn? This is the type of question we need to answer in Bayesian inference.

Let's assume that the proportion of white balls in the urn is  $p$ . The probability of drawing a white ball is then  $p$  and the probability of drawing a black ball is  $1 - p$ . The probability of obtaining, say, two white balls and one black ball in three draws would be

$$\Pr(2\text{white}, 1\text{black} | p) = p \times p \times (1 - p) \times \binom{3}{2} \tag{7.1}$$

The vertical bar indicates a condition; in this case we are interested in the probability of a particular outcome given (or conditional) on a particular value of  $p$ . It is easy to forget the last factor (3 choose 2), which is the number of ways in which we can obtain the given outcome. Two white balls and one black ball can be the result of drawing the black ball in the first, second or third draw. That is, there are three ways of obtaining the outcome of interest, 3 choose 2 (or 3 choose 1 if we focus on the choice of the black ball; the result is the same). Generally, the probability of obtaining  $a$  white balls and  $b$  black balls is determined by the function

$$f(a, b | p) = p^a (1 - p)^b \binom{a + b}{a} \tag{7.2}$$

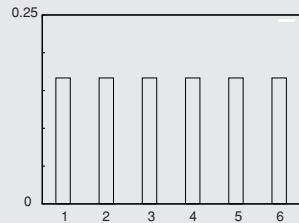
which is the **probability mass function** (Box 7.1) of the so-called binomial distribution. This is the solution to the problem in forward probability, when we know the value of  $p$ . Bayesians often, somewhat inappropriately, refer to the forward probability function as the **likelihood function**.

But given that we have a sample of  $a$  white balls and  $b$  black balls, what is the probability of a particular value of  $p$ ? This is the reverse probability problem, where we are trying to find the function  $f(p | a, b)$  instead of the function  $f(a, b | p)$ . It turns out that it is impossible to derive this function without specifying our prior beliefs about the value of  $p$ . This is done in the form of a probability distribution on the possible values of  $p$  (Box 7.1), the **prior probability distribution** or just **prior** in everyday Bayesian jargon. If there is no previous information about the value of  $p$ , we might associate all possible values with the same probability, a so-called uniform probability distribution (Box 7.1).

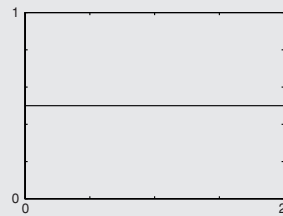
## 213 Bayesian phylogenetic analysis using MRBAYES: theory

### Box 7.1 Probability distributions

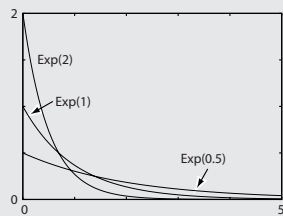
A function describing the probability of a discrete random variable is called a **probability mass function**. For instance, this is the probability mass function for throwing a dice, an example of a *discrete uniform distribution*:



For a continuous variable, the equivalent function is a **probability density function**. The value of this function is not a probability, so it can sometimes be larger than one. Probabilities are obtained by integrating the density function over a specified interval, giving the probability of obtaining a value in that interval. For instance, a *continuous uniform distribution* on the interval (0,2) has this probability density function:



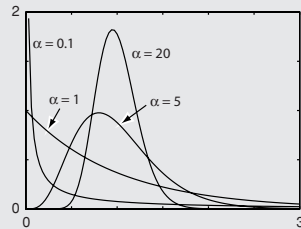
Most prior probability distributions used in Bayesian phylogenetics are uniform, exponential, gamma, beta or Dirichlet distributions. Uniform distributions are often used to express the lack of prior information for parameters that have a uniform effect on the likelihood in the absence of data. For instance, the discrete uniform distribution is typically used for the topology parameter. In contrast, the likelihood is a negative exponential function of the branch lengths, and therefore the *exponential distribution* is a better choice for a *vague* prior on branch lengths. The exponential distribution has the density function  $f(x) = \lambda e^{-\lambda x}$ , where  $\lambda$  is known as the *rate* parameter. The expectation (mean) of the exponential distribution is  $1/\lambda$ .



The *gamma* distribution has two parameters, the shape parameter  $\alpha$  and the scale parameter  $\beta$ . At small values of  $\alpha$ , the distribution is L-shaped and the variance is large;

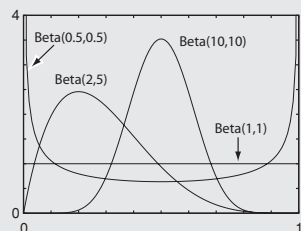
**Box 7.1 (cont.)**

at high values it is similar to a normal distribution and the variance is low. If there is considerable uncertainty concerning the shape of the prior probability distribution, the gamma may be a good choice; an example is the rate variation across sites. In these cases, the value of  $\alpha$  can be associated with a uniform or an exponential prior (also known as a *hyperprior* since it is a prior on a parameter of a prior), so that the MCMC procedure can explore different shapes of the gamma distribution and weight each according to its posterior probability. The sum of exponentially distributed variables is also a gamma distribution. Therefore, the gamma is an appropriate choice for the prior on the tree height of clock trees, which is the sum of several presumably exponentially distributed branch lengths.



The *beta* and *Dirichlet* distributions are used for parameters describing proportions of a whole, so called simplex parameters. Examples include the stationary state frequencies that appear in the instantaneous rate matrix of the substitution model. The exchangeability or rate parameters of the substitution model can also be understood as proportions of the total exchange rate (given the stationary state frequencies). Another example is the proportion of invariable and variable sites in the invariable sites model. The beta distribution, denoted  $\text{Beta}(\alpha_1, \alpha_2)$ , describes the probability on two proportions, which are associated with the weight parameters  $\alpha_1 > 0$  and  $\alpha_2 > 0$ . The Dirichlet distribution is equivalent except that there are more than two proportions and associated weight parameters.

A  $\text{Beta}(1, 1)$  distribution, also known as a flat beta, is equivalent to a uniform distribution on the interval (0,1). When  $\alpha_1 = \alpha_2 > 1$ , the distribution is symmetric and emphasizes equal proportions, the more so the higher the weights. When  $\alpha_1 = \alpha_2 < 1$ , the distribution puts more probability on extreme proportions than on equal proportions. Finally, if the weights are different, the beta is skewed towards the proportion defined by the weights; the expectation of the beta is  $\alpha/(\alpha + \beta)$  and the mode is  $(\alpha - 1)/(\alpha + \beta - 2)$  for  $\alpha > 1$  and  $\beta > 1$ .



**215 Bayesian phylogenetic analysis using MRBAYES: theory**

**Box 7.1 (cont.)**

Assume that we toss a coin to determine the probability  $p$  of obtaining heads. If we associate  $p$  and  $1 - p$  with a flat beta prior, we can show that the posterior is a beta distribution where  $\alpha_1 - 1$  is the number of heads and  $\alpha_2 - 1$  is the number of tails. Thus, the weights roughly correspond to counts. If we started with a flat Dirichlet distribution and analyzed a set of DNA sequences with the composition 40 A, 50 C, 30 G, and 60 T, we might expect a posterior for the stationary state frequencies around Dirichlet(41, 51, 31, 61) if it were not for the other parameters in the model and the blurring effect resulting from looking back in time. Wikipedia (<http://www.wikipedia.org>) is an excellent source for additional information on common statistical distributions.

Thomas Bayes realized that the probability of a particular value of  $p$ , given some sample  $(a, b)$  of white and black balls, can be obtained using the probability function

$$f(p|a, b) = \frac{f(p) f(a, b|p)}{f(a, b)} \tag{7.3}$$

This is known as Bayes' theorem or Bayes' rule. The function  $f(p|a, b)$  is called the **posterior probability distribution**, or simply the **posterior**, because it specifies the probability of all values of  $p$  after the prior has been updated with the available data.

We saw above how we can calculate  $f(a, b|p)$ , and how we can specify  $f(p)$ . How do we calculate the probability  $f(a, b)$ ? This is the unconditional probability of obtaining the outcome  $(a, b)$  so it must take all possible values of  $p$  into account. The solution is to integrate over all possible values of  $p$ , weighting each value according to its prior probability:

$$f(a, b) = \int_0^1 f(p) f(a, b|p) dp \tag{7.4}$$

We can now see that the denominator is a normalizing constant. It simply ensures that the posterior probability distribution integrates to 1, the basic requirement of a proper probability distribution.

A Bayesian problem that occupied several early workers was an analog to the following. Given a particular sample of balls, what is the probability that  $p$  is larger than a specified value? To solve it analytically, they needed to deal with complex integrals. Bayes made some progress in his *Essay*; more important contributions were made later by Laplace, who, among other things, used Bayesian reasoning and novel integration methods to show beyond any reasonable doubt that the probability of a newborn being a boy is higher than 0.5. However, the analytical complexity of most Bayesian problems remained a serious problem for a long time and it is only in the last few decades that the approach has become popular due to

**216 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

the combination of efficient numerical methods and the widespread availability of fast computers.

**7.2 Bayesian phylogenetic inference**

How does Bayesian reasoning apply to phylogenetic inference? Assume we are interested in the relationships between man, gorilla, and chimpanzee. In the standard case, we need an additional species to root the tree, and the orangutan would be appropriate here. There are three possible ways of arranging these species in a phylogenetic tree: the chimpanzee is our closest relative, the gorilla is our closest relative, or the chimpanzee and the gorilla are each other's closest relatives (Fig. 7.1).

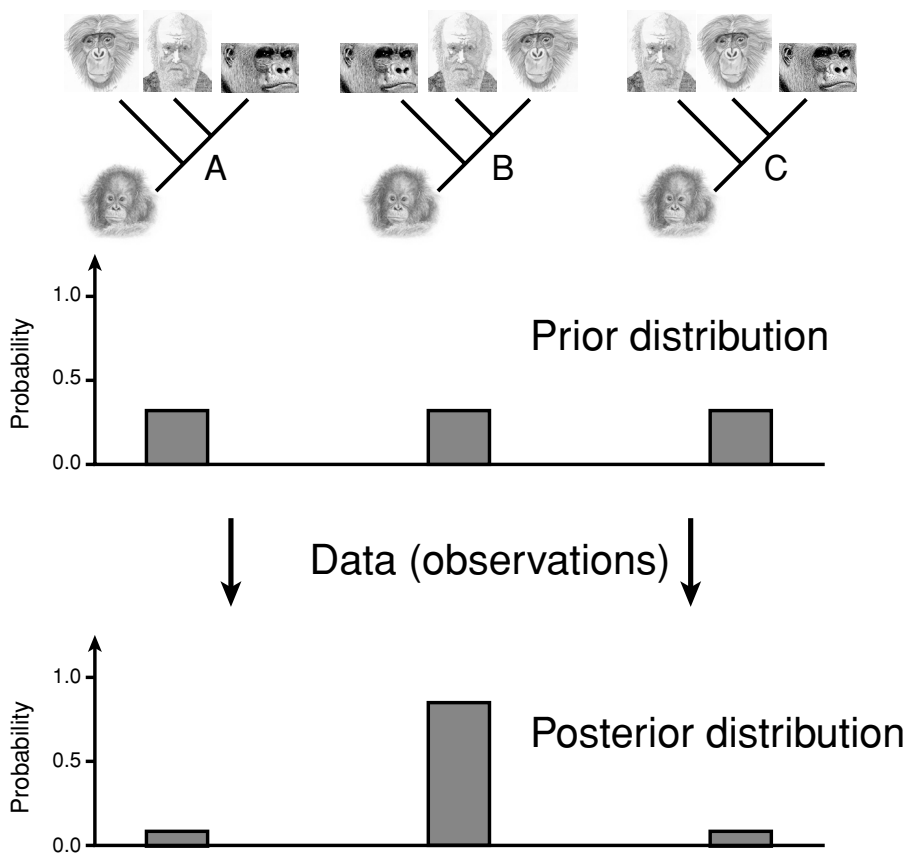


Fig. 7.1 A Bayesian phylogenetic analysis. We start the analysis by specifying our prior beliefs about the tree. In the absence of background knowledge, we might associate the same probability to each tree topology. We then collect data and use a stochastic evolutionary model and Bayes' theorem to update the prior to a posterior probability distribution. If the data are informative, most of the posterior probability will be focused on one tree (or a small subset of trees in a large tree space).

## 217 Bayesian phylogenetic analysis using MRBAYES: theory

Before the analysis starts, we need to specify our prior beliefs about the relationships. In the absence of background data, a simple solution would be to assign equal probability to the possible trees. Since there are three trees, the probability of each would be one-third. Such a prior probability distribution is known as a *vague* or *uninformative prior* because it is appropriate for the situation when we do not have any prior knowledge or do not want to build our analysis on any previous results.

To update the prior we need some data, typically in the form of a molecular sequence alignment, and a stochastic model of the process generating the data on the tree. In principle, Bayes' rule is then used to obtain the posterior probability distribution (Fig. 7.1), which is the result of the analysis. The posterior specifies the probability of each tree given the model, the prior, and the data. When the data are informative, most of the posterior probability is typically concentrated on one tree (or a small subset of trees in a large tree space).

If the analysis is performed correctly, there is nothing controversial about the posterior probabilities. Nevertheless, the interpretation of them is often subject to considerable discussion, particularly in the light of alternative models and priors.

To describe the analysis mathematically, designate the matrix of aligned sequences  $X$ . The vector of model parameters is contained in  $\theta$  (we do not distinguish in our notation between vector parameters and scalar parameters). In the ideal case, this vector would only include a topology parameter  $\tau$ , which could take on the three possible values discussed above. However, this is not sufficient to calculate the probability of the data. Minimally, we also need branch lengths on the tree; collect these in the vector  $v$ . Typically, there are also some *substitution model* parameters to be considered but, for now, let us use the Jukes Cantor substitution model (see below), which does not have any free parameters. Thus, in our case,  $\theta = (\tau, v)$ .

Bayes' theorem allows us to derive the posterior distribution as

$$f(\theta|X) = \frac{f(\theta) f(X|\theta)}{f(X)} \quad (7.5)$$

The denominator is an integral over the parameter values, which evaluates to a summation over discrete topologies and a multidimensional integration over possible branch length values:

$$f(X) = \int f(\theta) f(X|\theta) d\theta \quad (7.6)$$

$$= \sum_{\tau} \int_v f(v) f(X|\tau, v) dv \quad (7.7)$$

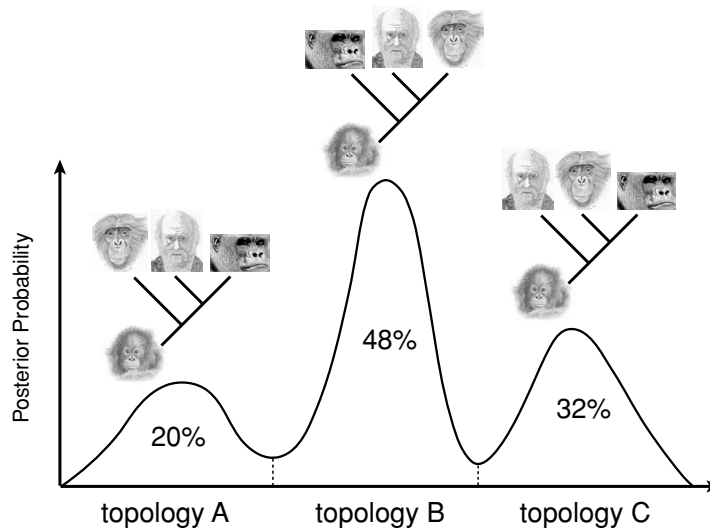


Fig. 7.2 Posterior probability distribution for our phylogenetic analysis. The x-axis is an imaginary one-dimensional representation of the parameter space. It falls into three different regions corresponding to the three different topologies. Within each region, a point along the axis corresponds to a particular set of branch lengths on that topology. It is difficult to arrange the space such that optimal branch length combinations for different topologies are close to each other. Therefore, the posterior distribution is multimodal. The area under the curve falling in each tree topology region is the posterior probability of that tree topology.

Even though our model is as simple as phylogenetic models come, it is impossible to portray its parameter space accurately in one dimension. However, imagine for a while that we could do just that. Then the parameter axis might have three distinct regions corresponding to the three different tree topologies (Fig. 7.2). Within each region, the different points on the axis would represent different branch length values. The one-dimensional parameter axis allows us to obtain a picture of the posterior probability function or surface. It would presumably have three distinct peaks, each corresponding to an optimal combination of topology and branch lengths.

To calculate the posterior probability of the topologies, we integrate out the model parameters that are not of interest, the branch lengths in our case. This corresponds to determining the area under the curve in each of the three topology regions. A Bayesian would say that we are marginalizing or deriving the *marginal probability distribution* on topologies.

Why is it called marginalizing? Imagine that we represent the parameter space in a two-dimensional table instead of along a single axis (Fig. 7.3). The columns in this table might represent different topologies and the rows different branch length values. Since the branch lengths are continuous parameters, there would actually

**219 Bayesian phylogenetic analysis using MRBAYES: theory**

		Topologies			Joint probabilities
		$\tau_A$	$\tau_B$	$\tau_C$	
Branch length vectors	$V^A$	0.10	0.07	0.12	0.29
	$V^B$	0.05	0.22	0.06	0.33
	$V^C$	0.05	0.19	0.14	0.38
		0.20	0.48	0.32	
		Marginal probabilities			

**Fig. 7.3** A two-dimensional table representation of parameter space. The columns represent different tree topologies, the rows represent different branch length bins. Each cell in the table represents the joint probability of a particular combination of branch lengths and topology. If we summarize the probabilities along the margins of the table, we get the marginal probabilities for the topologies (bottom row) and for the branch length bins (last column).

be an infinite number of rows, but imagine that we sorted the possible branch length values into discrete bins, so that we get a finite number of rows. For instance, if we considered only short and long branches, one bin would have all branches long, another would have the terminal branches long and the interior branch short, etc.

Now, assume that we can derive the posterior probability that falls in each of the cells in the table. These are *joint probabilities* because they represent the joint probability of a particular topology and a particular set of branch lengths. If we summarized all joint probabilities along one axis of the table, we would obtain the marginal probabilities for the corresponding parameter. To obtain the marginal probabilities for the topologies, for instance, we would summarize the entries in each column. It is traditional to write the sums in the margin of the table, hence the term marginal probability (Fig. 7.3).

It would also be possible to summarize the probabilities in each row of the table. This would give us the marginal probabilities for the branch length combinations (Fig. 7.3). Typically, this distribution is of no particular interest but the possibility of calculating it illustrates an important property of Bayesian inference: there is no sharp distinction between different types of model parameters. Once the posterior probability distribution is obtained, we can derive any marginal distribution of

**220 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

interest. There is no need to decide on the parameters of interest before performing the analysis.

**7.3 Markov chain Monte Carlo sampling**

In most cases, including virtually all phylogenetic problems, it is impossible to derive the posterior probability distribution analytically. Even worse, we can't even estimate it by drawing random samples from it. The reason is that most of the posterior probability is likely to be concentrated in a small part of a vast parameter space. Even with a massive sampling effort, it is highly unlikely that we would obtain enough samples from the interesting region(s) of the posterior. This argument is particularly easy to appreciate in the phylogenetic context because of the large number of tree topologies that are possible even for small numbers of taxa. Already at nine taxa, you are more likely to be hit by lightning (odds 3:100 000) than to find the best tree by picking one randomly (odds 1:135,135). At slightly more than 50 taxa, the number of topologies outnumber the number of atoms in the known universe – and this is still considered a small phylogenetic problem.

The solution is to estimate the posterior probability distribution using *Markov chain Monte Carlo sampling*, or *MCMC* for short. *Markov chains* have the property that they converge towards an equilibrium state regardless of starting point. We just need to set up a Markov chain that converges onto our posterior probability distribution, which turns out to be surprisingly easy. It can be achieved using several different methods, the most flexible of which is known as the *Metropolis algorithm*, originally described by a group of famous physicists involved in the Manhattan project (Metropolis *et al.*, 1953). Hastings (1970) later introduced a simple but important extension, and the sampler is often referred to as the *Metropolis–Hastings* method.

The central idea is to make small random changes to some current parameter values, and then accept or reject those changes according to the appropriate probabilities. We start the chain at an arbitrary point  $\theta$  in the landscape (Fig. 7.4). In the next generation of the chain, we consider a new point  $\theta^*$  drawn from a proposal distribution  $f(\theta^*|\theta)$ . We then calculate the ratio of the posterior probabilities at the two points. There are two possibilities. Either the new point is uphill, in which case we always accept it as the starting point for the next cycle in the chain, or it is downhill, in which case we accept it with a probability that is proportional to the height ratio. In reality, it is slightly more complicated because we need to take asymmetries in the proposal distribution into account as well. Formally, we accept

**221 Bayesian phylogenetic analysis using MRBAYES: theory**

**Markov chain Monte Carlo steps**

1. Start at an arbitrary point ( $\theta$ )
2. Make a small random move (to  $\theta^*$ )
3. Calculate height ratio ( $r$ ) of new state (to  $\theta^*$ ) to old state ( $\theta$ )
  - (a)  $r > 1$ : new state accepted
  - (b)  $r < 1$ : new state accepted with probability  $r$   
 if new state rejected, stay in old state
4. Go to step 2

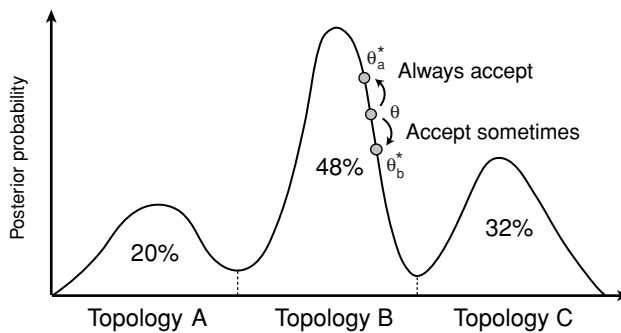


Fig. 7.4 The Markov chain Monte Carlo (MCMC) procedure is used to generate a valid sample from the posterior. One first sets up a Markov chain that has the posterior as its stationary distribution. The chain is then started at a random point and run until it converges onto this distribution. In each step (generation) of the chain, a small change is made to the current values of the model parameters (step 2). The ratio  $r$  of the posterior probability of the new and current states is then calculated. If  $r > 1$ , we are moving uphill and the move is always accepted (3a). If  $r < 1$ , we are moving downhill and accept the new state with probability  $r$  (3b).

or reject the proposed value with the probability

$$r = \min \left( 1, \frac{f(\theta^*|X)}{f(\theta|X)} \times \frac{f(\theta|\theta^*)}{f(\theta^*|\theta)} \right) \tag{7.8}$$

$$= \min \left( 1, \frac{f(\theta^*) f(X|\theta^*)/f(X)}{f(\theta) f(X|\theta)/f(X)} \times \frac{f(\theta|\theta^*)}{f(\theta^*|\theta)} \right) \tag{7.9}$$

$$= \min \left( 1, \frac{f(\theta^*)}{f(\theta)} \times \frac{f(X|\theta^*)}{f(X|\theta)} \times \frac{f(\theta|\theta^*)}{f(\theta^*|\theta)} \right) \tag{7.10}$$

The three ratios in the last equation are referred to as the *prior ratio*, the *likelihood ratio*, and the *proposal ratio* (or *Hastings ratio*), respectively. The first two ratios correspond to the ratio of the numerators in Bayes' theorem; note that the complex

**222 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

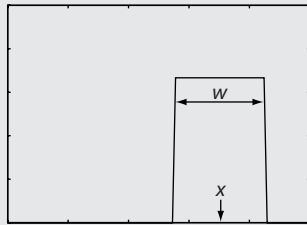
integral in the denominator of Bayes' theorem,  $f(X)$ , cancels out in the second step because it is the same for both the current and the proposed states. Because of this,  $r$  is easy to compute.

The Metropolis sampler works because the relative equilibrium frequencies of the two states  $\theta$  and  $\theta^*$  is determined by the ratio of the rates at which the chain moves back and forth between them. Equation (7.10) ensures that this ratio is the same as the ratio of their posterior probabilities. This means that, if the Markov chain is allowed to run for a sufficient number of generations, the amount of time it spends sampling a particular parameter value or parameter interval is proportional to the posterior probability of that value or interval. For instance, if the posterior probability of a topology is 0.68, then the chain should spend 68% of its time sampling that topology at *stationarity*. Similarly, if the posterior probability of a branch length being in the interval (0.02, 0.04) is 0.11, then 11% of the chain samples at stationarity should be in that interval.

For a large and parameter-rich model, a mixture of different Metropolis samplers is typically used. Each sampler targets one parameter or a set of related parameters (Box 7.2). One can either cycle through the samplers systematically or choose among them randomly according to some proposal probabilities (MRBAYES does the latter).

**Box 7.2 Proposal mechanisms**

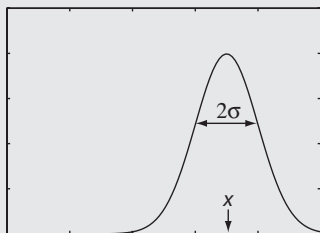
Four types of proposal mechanisms are commonly used to change continuous variables. The simplest is the *sliding window* proposal. A continuous uniform distribution of width  $w$  is centered on the current value  $x$ , and the new value  $x^*$  is drawn from this distribution. The “window” width  $w$  is a tuning parameter. A larger value of  $w$  results in more radical proposals and lower acceptance rates, while a smaller value leads to more modest changes and higher acceptance rates.



The *normal* proposal is similar to the sliding window except that it uses a normal distribution centered on the current value  $x$ . The variance  $\sigma^2$  of the normal distribution determines how drastic the new proposals are and how often they will be accepted.

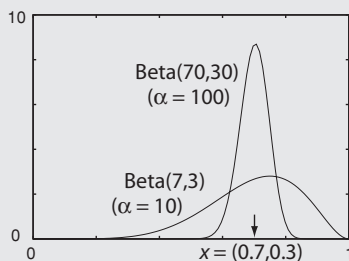
223 Bayesian phylogenetic analysis using MRBAYES: theory

Box 7.2 (cont.)

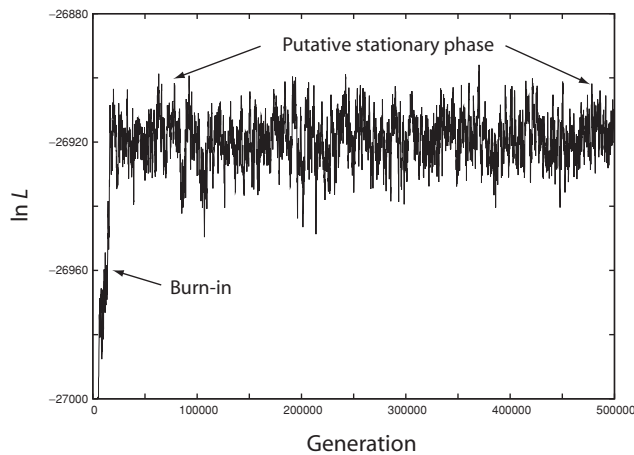


Both the sliding window and normal proposals can be problematic when the effect on the likelihood varies over the parameter range. For instance, changing a branch length from 0.01 to 0.03 is likely to have a dramatic effect on the posterior but changing it from 0.51 to 0.53 will hardly be noticeable. In such situations, the *multiplier* proposal is appropriate. It is equivalent to a sliding window with width  $\lambda$  on the log scale of the parameter. A random number  $u$  is drawn from a uniform distribution on the interval  $(-0.5, 0.5)$  and the proposed value is  $x^* = mx$ , where  $m = e^{\lambda u}$ . If the value of  $\lambda$  takes the form  $2 \ln a$ , one will pick multipliers  $m$  in the interval  $(1/a, a)$ .

The *beta* and *Dirichlet* proposals are used for simplex parameters. They pick new values from a beta or Dirichlet distribution centered on the current values of the simplex. Assume that the current values are  $(x_1, x_2)$ . We then multiply them with a value  $\alpha$ , which is a tuning parameter, and pick new values from the distribution  $\text{Beta}(\alpha x_1, \alpha x_2)$ . The higher the value of  $\alpha$ , the closer the proposed values will be to the current values.



More complex moves are needed to change topology. A common type uses stochastic branch rearrangements (see Chapter 8). For instance, the extending *subtree pruning and regrafting* (extending SPR) move chooses a subtree at random and then moves its attachment point, one branch at a time, until a random number  $u$  drawn from a uniform on  $(0, 1)$  becomes higher than a specified extension probability  $p$ . The extension probability  $p$  is a tuning parameter; the higher the value, the more drastic rearrangements will be proposed.

**224 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

**Fig. 7.5** The likelihood values typically increase very rapidly during the initial phase of the run because the starting point is far away from the regions in parameter space with high posterior probability. This initial phase of the Markov chain is known as the burn in. The burn-in samples are typically discarded because they are so heavily influenced by the starting point. As the chain converges onto the target distribution, the likelihood values tend to reach a plateau. This phase of the chain is sampled with some thinning, primarily to save disk space.

**7.4 Burn-in, mixing and convergence**

If the chain is started from a random tree and arbitrarily chosen branch lengths, chances are that the initial likelihood is low. As the chain moves towards the regions in the posterior with high probability mass, the likelihood typically increases very rapidly; in fact, it almost always changes so rapidly that it is necessary to measure it on a log scale (Fig. 7.5). This early phase of the run is known as the *burn in*, and the burn-in samples are often discarded because they are so heavily influenced by the starting point.

As the chain approaches its stationary distribution, the likelihood values tend to reach a plateau. This is the first sign that the chain may have converged onto the target distribution. Therefore, the plot of the likelihood values against the generation of the chain, known as the *trace plot* (Fig. 7.5), is important in monitoring the performance of an MCMC run. However, it is extremely important to confirm convergence using other diagnostic tools because it is not sufficient for the chain to reach the region of high probability in the posterior, it must also cover this region adequately. The speed with which the chain covers the interesting regions of the posterior is known as its *mixing behavior*. The better the mixing, the faster the chain will generate an adequate sample of the posterior.

## 225 Bayesian phylogenetic analysis using MRBAYES: theory

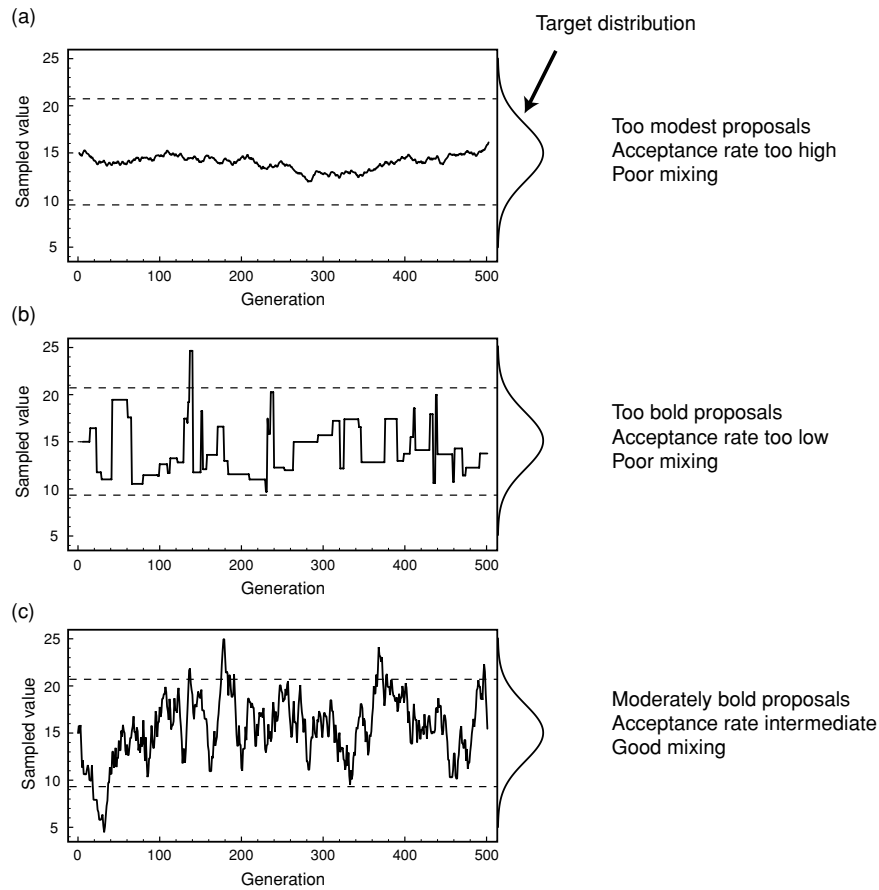


Fig. 7.6 The time it takes for a Markov chain to obtain an adequate sample of the posterior depends critically on its mixing behavior, which can be controlled to some extent by the proposal tuning parameters. If the proposed values are very close to the current ones, all proposed changes are accepted but it takes a long time for the chain to cover the posterior; mixing is poor. If the proposed values tend to be dramatically different from the current ones, most proposals are rejected and the chain will remain on the same value for a long time, again leading to poor mixing. The best mixing is obtained at intermediate values of the tuning parameters, associated with moderate acceptance rates.

The mixing behavior of a Metropolis sampler can be adjusted using its tuning parameter(s). Assume, for instance, that we are sampling from a normal distribution using a sliding window proposal (Fig. 7.6). The sliding window proposal has one tuning parameter, the width of the window. If the width is too small, then the proposed value will be very similar to the current one (Fig. 7.6a). The posterior probabilities will also be very similar, so the proposal will tend to be accepted. But each proposal will only move the chain a tiny distance in parameter space, so it will take the chain a long time to cover the entire region of interest; mixing is poor.

**226 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

A window that is too wide also results in poor mixing. Under these conditions, the proposed state is almost always very different from the current state. If we have reached a region of high posterior probability density, then the proposed state is also likely to have much lower probability than the current state. The new state will therefore often be rejected, and the chain remains in the same spot for a long time (Fig. 7.6b), resulting in poor mixing. The most efficient sampling of the target distribution is obtained at intermediate acceptance rates, associated with intermediate values of the tuning parameter (Fig. 7.6c).

Extreme acceptance rates thus indicate that sampling efficiency can be improved by adjusting proposal tuning parameters. Studies of several types of complex but unimodal posterior distributions indicate that the optimal acceptance rate is 0.44 for one-dimensional and 0.23 for multi-dimensional proposals (Roberts *et al.*, 1997; Roberts & Rosenthal, 1998, 2001). However, multimodal posteriors are likely to have even lower optimal acceptance rates. Adjusting the tuning parameter values to reach a target acceptance rate can be done manually or automatically using adaptive tuning methods (Roberts & Rosenthal, 2006). Bear in mind, however, that some samplers used in Bayesian MCMC phylogenetics have acceptance rates that will remain low, no matter how much you tweak the tuning parameters. In particular, this is true for many tree topology update mechanisms.

Convergence diagnostics help determine the quality of a sample from the posterior. There are essentially three different types of diagnostics that are currently in use: (1) examining autocorrelation times, *effective sample sizes*, and other measures of the behavior of single chains; (2) comparing samples from successive time segments of a single chain; and (3) comparing samples from different runs. The last approach is arguably the most powerful way of detecting convergence problems. The drawback is that it wastes computational power by generating several independent sets of burn-in samples that must be discarded.

In Bayesian MCMC sampling of phylogenetic problems, the tree topology is typically the most difficult parameter to sample from. Therefore, it makes sense to focus our attention on this parameter when monitoring convergence. If we start several parallel MCMC runs from different, randomly chosen trees, they will initially sample from very different regions of tree space. As they approach stationarity, however, the tree samples will become more and more similar. Thus, an intuitively appealing convergence diagnostic is to compare the variance among and within tree samples from different runs.

Perhaps the most obvious way of achieving this is to compare the frequencies of the sampled trees. However, this is not practical unless most of the posterior probability falls on a small number of trees. In large phylogenetic problems, there is often an inordinate number of trees with similar probabilities and it may be extremely difficult to estimate the probability of each accurately.

## 227 Bayesian phylogenetic analysis using MRBAYES: theory

The approach that we and others have taken to solve this problem is to focus on *split* (clade) *frequencies* instead. A split is a partition of the tips of the tree into two non-overlapping sets; each branch in a tree corresponds to exactly one such split. For instance, the split ((human, chimp),(gorilla, orangutan)) corresponds to the branch uniting the human and the chimp in a tree rooted on the orangutan. Typically, a fair number of splits are present in high frequency among the sampled trees. In a way, the dominant splits (present in, say, more than 10% of the trees) represent an efficient diagnostic summary of the tree sample as a whole. If two tree samples are similar, the split frequencies should be similar as well. To arrive at an overall measure of the similarity of two or more tree samples, we simply calculate the average standard deviation of the split frequencies. As the tree samples become more similar, this value should approach zero.

Most other parameters in phylogenetic models are continuous scalar parameters. An appropriate convergence diagnostic for these is the **Potential Scale Reduction Factor (PSRF)** originally proposed by Gelman and Rubin (1992). The PSRF compares the variance among runs with the variance within runs. If chains are started from over-dispersed starting points, the variance among runs will initially be higher than the variance within runs. As the chains converge, however, the variances will become more similar and the PSRF will approach 1.0.

### 7.5 Metropolis coupling

For some phylogenetic problems, it may be difficult or impossible to achieve convergence within a reasonable number of generations using the standard approach. Often, this seems to be due to the existence of isolated peaks in tree space (also known as tree islands) with deep valleys in-between. In these situations, individual chains may get stuck on different peaks and have difficulties moving to other peaks of similar probability mass. As a consequence, tree samples from independent runs tend to be different. A topology convergence diagnostic, such as the standard deviation of split frequencies, will indicate that there is a problem. But are there methods that can help us circumvent it?

A general technique that can improve mixing, and hence convergence, in these cases is **Metropolis Coupling**, also known as MCMCMC or (MC)<sup>3</sup> (Geyer, 1991). The idea is to introduce a series of Markov chains that sample from a *heated* posterior probability distribution (Fig. 7.7). The heating is achieved by raising the posterior probability to a power smaller than 1. The effect is to flatten out the posterior probability surface, very much like melting a landscape of wax.

Because the surface is flattened, a Markov chain will move more readily between the peaks. Of course, the heated chains have a target distribution that is different from the one we are interested in, sampled by the **cold chain**, but we can use them

228 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck

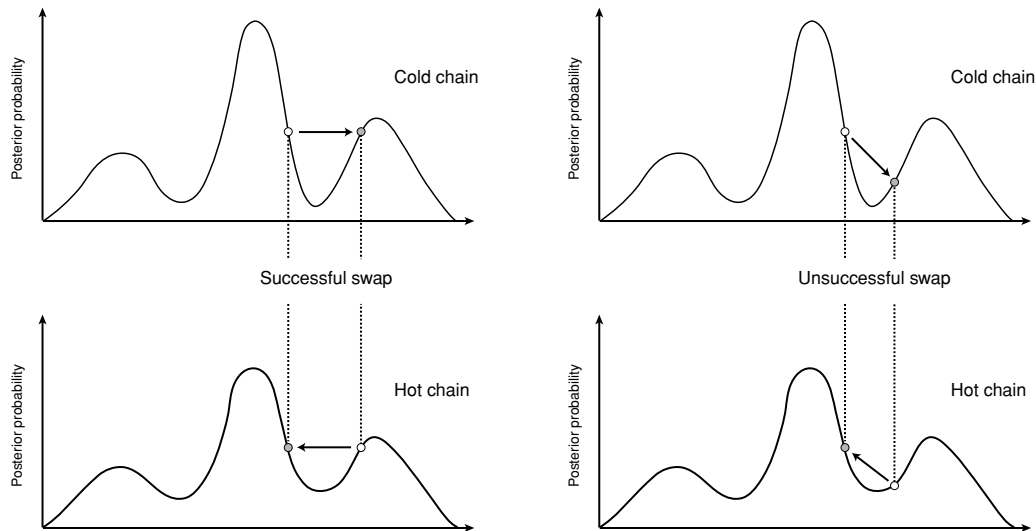


Fig. 7.7 Metropolis Coupling uses one or more *heated* chains to accelerate mixing in the so called *cold* chain sampling from the posterior distribution. The heated chains are flattened out versions of the posterior, obtained by raising the posterior probability to a power smaller than one. The heated chains can move more readily between peaks in the landscape because the valleys between peaks are shallower. At regular intervals, one attempts to swap the states between chains. If a swap is accepted, the cold chain can jump between isolated peaks in the posterior in a single step, accelerating its mixing over complex posterior distributions.

to generate proposals for the cold chain. With regular intervals, we attempt to swap the states between two randomly picked chains. If the cold chain is one of them, and the swap is accepted, the cold chain can jump considerable distances in parameter space in a single step. In the ideal case, the swap takes the cold chain from one tree island to another. At the end of the run, we simply discard all of the samples from the heated chains and keep only the samples from the cold chain.

In practice, an incremental heating scheme is often used where chain  $i$  has its posterior probability raised by the temperature factor

$$T = \frac{1}{1 + \lambda i} \tag{7.11}$$

where  $i \in \{0, 1, \dots, k\}$  for  $k$  heated chains, with  $i = 0$  for the cold chain, and  $\lambda$  is the temperature factor. The higher the value of  $\lambda$ , the larger the temperature difference between adjacent chains in the incrementally heated sequence.

If we apply too much heat, then the chains moving in the heated landscapes will walk all over the place and are less likely to be on an interesting peak when we try to swap states with the cold chain. Most of the swaps will therefore be rejected and

## 229 Bayesian phylogenetic analysis using MRBAYES: theory

the heating does not accelerate mixing in the cold chain. On the other hand, if we do not heat enough, then the chains will be very similar, and the heated chain will not mix more rapidly than the cold chain. As with the proposal tuning parameters, an intermediate value of the heating parameter  $\lambda$  works best.

### 7.6 Summarizing the results

The stationary phase of the chain is typically sampled with some thinning, for instance every 50th or 100th generation. This is done primarily to save disk space, since an MCMC run can easily generate millions of samples. Once an adequate sample is obtained, it is usually trivial to compute an estimate of the marginal posterior distribution for the parameter(s) of interest. For instance, this can take the form of a frequency histogram of the sampled values. When it is difficult to visualize this distribution or when space does not permit it, various summary statistics are used instead.

Most phylogenetic model parameters are continuous variables and their estimated posterior distribution is summarized using statistics such as the mean, the median, and the variance. Bayesian statisticians typically also give the 95% *credibility interval*, which is obtained by simply removing the lowest 2.5% and the highest 2.5% of the sampled values. The credibility interval is somewhat similar to a confidence interval but the interpretation is different. A 95% credibility interval actually contains the true value with probability 0.95 (given the model, prior, and data) unlike the confidence interval, which has a more complex interpretation.

The posterior distribution on topologies and branch lengths is more difficult to summarize efficiently. If there are few topologies with high posterior probability, one can produce a list of the best topologies and their probabilities, or simply give the topology with the maximum posterior probability. However, most posteriors contain too many topologies with reasonably high probabilities, and one is forced to use other methods.

One way to illustrate the topological variance in the posterior is to list the topologies in order of decreasing probabilities and then calculate the cumulative probabilities so that we can give the estimated number of topologies in various *credible sets*. Assume, for instance, that the five best topologies have the estimated probabilities (0.35, 0.25, 0.20, 0.15, 0.03), giving the cumulative probabilities (0.35, 0.60, 0.80, 0.95, 0.98). Then the 50% credible set has two topologies in it, the 90% and the 95% credible sets both have four trees in them, etc. We simply pass down the list and count the number of topologies we need to include before the target probability is met or superseded. When these credible sets are large, however, it is difficult to estimate their sizes precisely.

**230 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

The most common approach to summarizing topology posteriors is to give the frequencies of the most common splits, since there are much fewer splits than topologies. Furthermore, all splits occurring in at least 50% of the sampled trees are guaranteed to be compatible and can be visualized in the same tree, a *majority rule consensus tree*. However, although the split frequencies are convenient, they do have limitations. For instance, assume that the splits  $((A,B),(C,D,E))$  and  $((A,B,C),(D,E))$  were both encountered in 70% of the sampled trees. This could mean that 30% of the sampled trees contained neither split or, at the other extreme, that all sampled trees contained at least one of them. The split frequencies themselves only allow us to approximately reconstruct the underlying set of topologies.

The sampled branch lengths are even more difficult to summarize adequately. Perhaps the best way would be to display the distribution of sampled branch length values separately for each topology. However, if there are many sampled topologies, there may not be enough branch length samples for each. A reasonable approach, taken by MRBAYES, is then to pool the branch length samples that correspond to the same split. These pooled branch lengths can also be displayed on the consensus tree. However, one should bear in mind that the pooled distributions may be multimodal since the sampled values in most cases come from different topologies, and a simple summary statistic like the mean might be misleading.

A special difficulty appears with branch lengths in clock trees. Clock trees are rooted trees in which branch lengths are proportional to time units (see Chapter 11). Even if computed from a sample of clock trees, a majority rule consensus tree with mean pooled branch lengths is not necessarily itself a clock tree. This problem is easily circumvented by instead using mean pooled node depths instead of branch lengths (for Bayesian inference of clock trees, see also Chapter 18).

**7.7 An Introduction to phylogenetic models**

A phylogenetic model can be divided into two distinct parts: a tree model and a substitution model. The tree model we have discussed so far is the one most commonly used in phylogenetic inference today (sometimes referred to as the different-rates or *unrooted model*, see Chapter 11). Branch lengths are measured in amounts of expected evolutionary change per site, and we do not assume any correlation between branch lengths and time units. Under time-reversible substitution models, the likelihood is unaffected by the position of the root, that is, the tree is unrooted. For presentation purposes, unrooted trees are typically rooted between a specially designated reference sequence or group of reference sequences, the outgroup, and the rest of the sequences.

Alternatives to the standard tree model include the strict and *relaxed clock* tree models. Both of these are based on trees, whose branch lengths are strictly

**231 Bayesian phylogenetic analysis using MRBAYES: theory**

proportional to time. In strict clock models, the evolutionary rate is assumed to be constant so that the amount of evolutionary change on a branch is directly proportional to its time duration, whereas relaxed clock models include a model component that accommodates some variation in the rate of evolution across the tree. Various prior probability models can be attached to clock trees. Common examples include the uniform model, the birth-death process, and the coalescent process (for the latter two, see Chapter 18).

The substitution process is typically modeled using Markov chains of the same type used in MCMC sampling. For instance, they have the same tendency towards an equilibrium state. The different substitution models are most easily described in terms of their instantaneous rate matrices, or *Q matrices*. For instance, the general time-reversible model (GTR) is described by the rate matrix

$$Q = \begin{bmatrix} - & \pi_C r_{AC} & \pi_G r_{AG} & \pi_T r_{AT} \\ \pi_A r_{AC} & - & \pi_G r_{CG} & \pi_T r_{CT} \\ \pi_A r_{AG} & \pi_C r_{CG} & - & \pi_T r_{GT} \\ \pi_A r_{AT} & \pi_C r_{CT} & \pi_G r_{GT} & - \end{bmatrix}$$

Each row in this matrix gives the instantaneous rate of going from a particular state, and each column represents the rate of going to a particular state; the states are listed in alphabetical sequence A, C, G, T. For instance, the second entry in the first row represents the rate of going from A to C. Each rate is composed of two factors; for instance, the rate of going from A to C is a product of  $\pi_C$  and  $r_{AC}$ . The rates along the diagonal are commonly omitted since their expressions are slightly more complicated. However, they are easily calculated since the rates in each row always sum to zero. For instance, the instantaneous rate of going from A to A (first entry in the first row) is  $-\pi_C r_{AC} - \pi_G r_{AG} - \pi_T r_{AT}$ .

It turns out that, if we run this particular Markov chain for a long time, it will move towards an equilibrium, where the frequency of a state  $i$  is determined exactly by the factor  $\pi_i$  given that  $\sum \pi_i = 1$ . Thus, the first rate factor corresponds to the stationary state frequency of the receiving state. The second factor,  $r_{ij}$ , is a parameter that determines the intensity of the exchange between pairs of states, controlling for the stationary state frequencies. For instance, at equilibrium we will have  $\pi_A$  sites in state A and  $\pi_C$  sites in state C. The total instantaneous rate of going from A to C over the sequence is then  $\pi_A$  times the instantaneous rate of the transition from A to C, which is  $\pi_C r_{AC}$ , resulting in a total rate of A to C changes over the sequence of  $\pi_A \pi_C r_{AC}$ . This is the same as the total rate of the reverse changes over the sequence, which is  $\pi_C \pi_A r_{AC}$ . Thus, there is no net change of the state proportions, which is the definition of an equilibrium, and the factor  $r_{AC}$  determines how intense the exchange between A and C is compared with the exchange between other pairs of states.

**232 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

Many of the commonly used substitution models are special cases or extensions of the GTR model. For instance, the Jukes Cantor model has all rates equal, and the Felsenstein 81 (F81) model has all exchangeability parameters ( $r_{ij}$ ) equal. The *covarion* and *covariotide* models have an independent on–off switch for each site, leading to a composite instantaneous rate matrix including four smaller rate matrices: two matrices describing the switching process, one being a zero-rate matrix, and the last describing the normal substitution process in the on state.

In addition to modeling the substitution process at each site, phylogenetic models typically also accommodate rate variation across sites. The standard approach is to assume that rates vary according to a gamma distribution (Box 7.1) with mean 1. This results in a distribution with a single parameter, typically designated  $\alpha$ , describing the shape of the rate variation (see Fig. 4.8 in Chapter 4). Small values of  $\alpha$  correspond to large amounts of rate variation; as  $\alpha$  approaches infinity, the model approaches rate constancy across sites. It is computationally expensive to let the MCMC chain integrate over a continuous gamma distribution of site rates, or to numerically integrate out the gamma distribution in each step of the chain. The standard solution is to integrate out the gamma using a discrete approximation with a small number of rate categories, typically four to eight, which is a reasonable compromise. An alternative is to use MCMC sampling over discrete rate categories.

Many other models of rate variation are also possible. A commonly considered model assumes that there is a proportion of invariable sites, which do not change at all over the course of evolution. This is often combined with an assumption of gamma-distributed rate variation in the variable sites.

It is beyond the scope of this chapter to give a more detailed discussion of phylogenetic models but we present an overview of the models implemented in MRBAYES 3.2, with the command options needed to invoke them (Fig. 7.8). The MRBAYES manual provides more details and references to the different models. A simulation-based presentation of Markov substitution models is given in (Huelsenbeck & Ronquist, 2005) and further details can be found in Chapter 4 and Chapter 10.

**7.8 Bayesian model choice and model averaging**

So far, our notation has implicitly assumed that Bayes's theorem is conditioned on a particular model. To make it explicit, we could write Bayes's theorem:

$$f(\theta|X, M) = \frac{f(\theta|M) f(X|\theta, M)}{f(X|M)} \quad (7.12)$$

It is now clear that the normalizing constant,  $f(X|M)$ , is the probability of the data given the chosen model after we have integrated out all parameters. This

## 233 Bayesian phylogenetic analysis using MRBAYES: theory

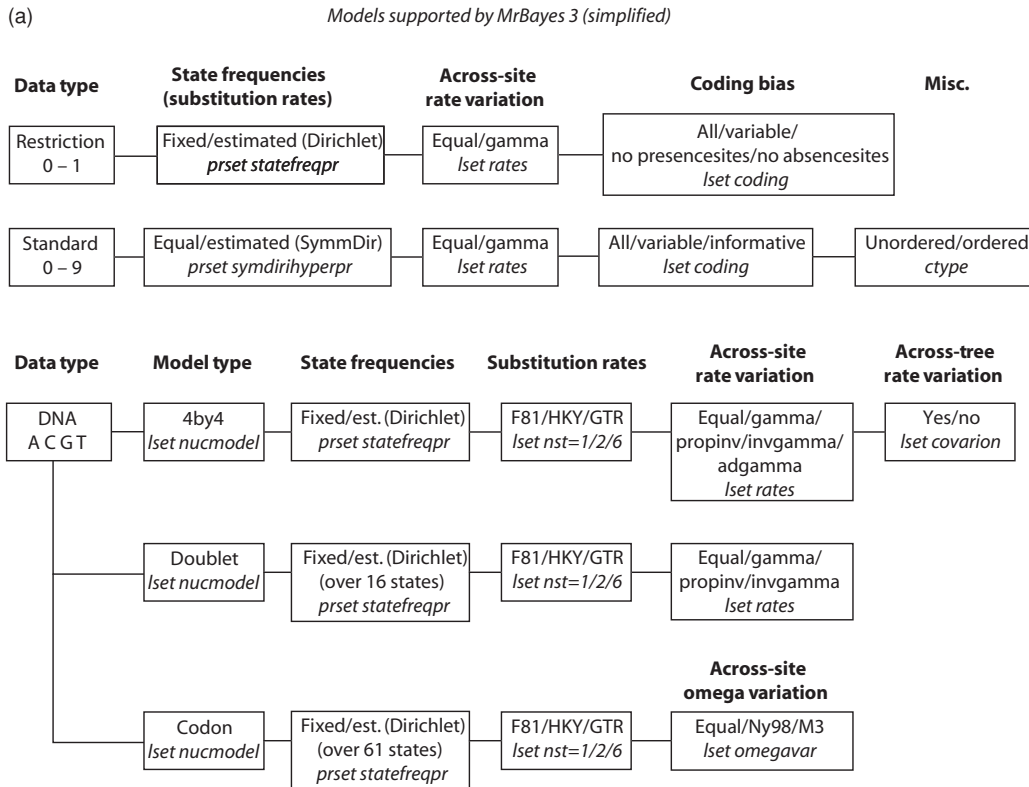


Fig. 7.8 Schematic overview of the models implemented in MRBAYES 3. Each box gives the available settings in normal font and then the program commands and command options needed to invoke those settings in italics.

quantity, known as the model likelihood, is used for Bayesian model comparison. Assume we are choosing between two models,  $M_0$  and  $M_1$ , and that we assign them the prior probabilities  $f(M_0)$  and  $f(M_1)$ . We could then calculate the ratio of their posterior probabilities (the posterior odds) as

$$\frac{f(M_0|X)}{f(M_1|X)} = \frac{f(M_0) f(X|M_0)}{f(M_1) f(X|M_1)} = \frac{f(M_0)}{f(M_1)} \times \frac{f(X|M_0)}{f(X|M_1)} \quad (7.13)$$

Thus, the posterior odds is obtained as the prior odds,  $f(M_0)/f(M_1)$ , times a factor known as the **Bayes factor**,  $B_{01} = f(X|M_0)/f(X|M_1)$ , which is the ratio of the model likelihoods. Rather than trying to specify the prior model odds, it is common to focus entirely on the Bayes factor. One way to understand the Bayes factor is that it determines how much the prior model odds are changed by the data when calculating the posterior odds. The Bayes factor is also the same as

**234 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

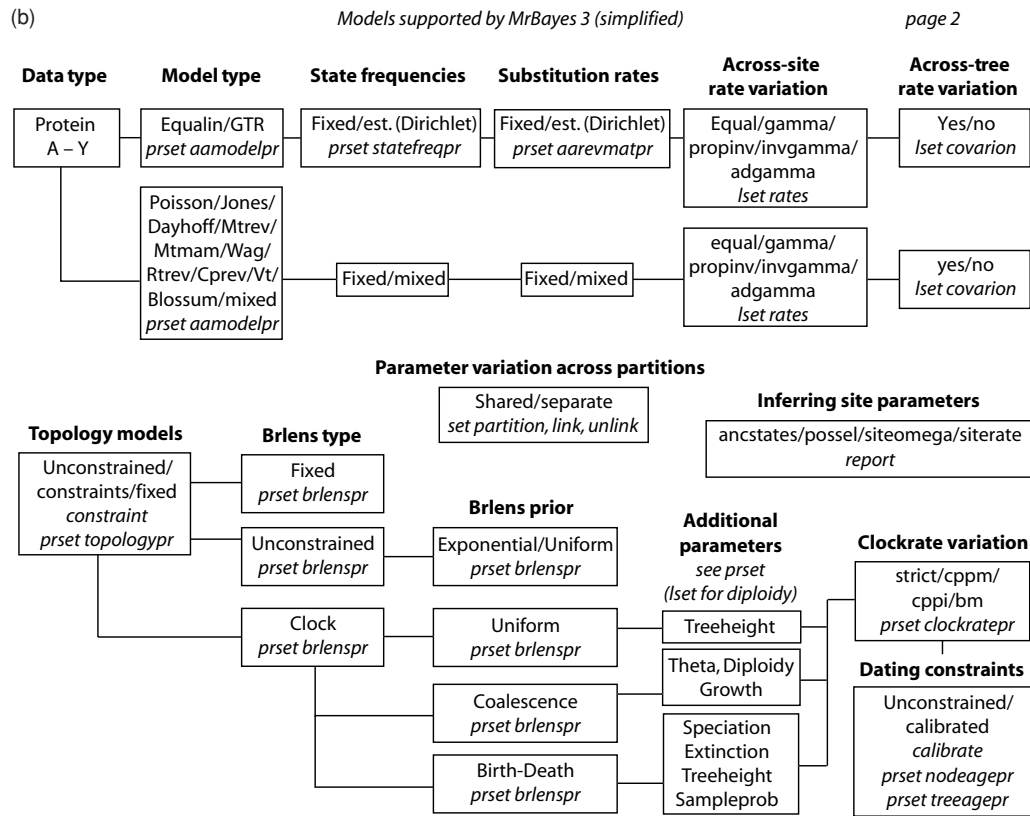


Fig. 7.8 (cont.)

the posterior odds when the prior odds are 1, that is, when we assign equal prior probabilities to the compared models.

Bayes factor comparisons are truly flexible. Unlike *likelihood ratio tests*, there is no requirement for the models to be nested. Furthermore, there is no need to correct for the number of parameters in the model, in contrast to comparisons based on the *Akaike Information Criterion* (Akaike, 1974) or the confusingly named *Bayesian Information Criterion* (Schwarz, 1978). Although it is true that a more parameter-rich model always has a higher *maximum likelihood* than a nested submodel, its model likelihood need not be higher. The reason is that a more parameter-rich model also has a larger parameter space and therefore a lower prior probability density. This can lead to a lower model likelihood unless it is compensated for by a sufficiently large increase in the likelihood values in the peak region.

The interpretation of a Bayes factor comparison is up to the investigator but some guidelines were suggested by Kass and Raftery (1995) (Table 7.2).

**235 Bayesian phylogenetic analysis using MRBAYES: theory**

**Table 7.2** Critical values for Bayes factor comparisons

$2 \ln B_{01}$	$B_{01}$	Evidence against $M_1$
0 to 2	1 to 3	Not worth more than a bare mention
2 to 6	3 to 20	Positive
6 to 10	20 to 150	Strong
>10	>150	Very strong

From Kass & Raftery (1995).

The easiest way of estimating the model likelihoods needed in the calculation of Bayes factors is to use the *harmonic mean* of the likelihood values from the stationary phase of an MCMC run (Newton & Raftery, 1994). Unfortunately, this estimator is unstable because it is occasionally influenced by samples with very small likelihood and therefore a large effect on the final result. A stable estimator can be obtained by mixing in a small proportion of samples from the prior (Newton & Raftery, 1994). Even better accuracy, at the expense of computational complexity, can be obtained by using thermodynamic integration methods (Lartillot & Philippe, 2006). Because of the instability of the harmonic mean estimator, it is good practice to compare several independent runs and only rely on this estimator when the runs give consistent results.

An alternative to running a full analysis on each model and then choosing among them using the estimated model likelihoods and Bayes factors is to let a single Bayesian analysis explore the models in a predefined model space (using *reversible-jump MCMC*). In this case, all parameter estimates will be based on an average across models, each model weighted according to its posterior probability. For instance, MRBAYES 3 uses this approach to explore a range of common fixed-rate matrices for amino acid data (see practice in Chapter 9 for an exercise).

Different topologies can also be considered different models and, in that sense, all Markov chains that integrate over the topology parameter also average across models. Thus, we can use the posterior sample of topologies from a single run to compare posterior probabilities of topology hypotheses.

For instance, assume that we want to test the hypothesis that group A is monophyletic against the hypothesis that it is not, and that 80% of the sampled trees have A monophyletic. Then the posterior model odds for A being monophyletic would be  $0.80/0.20 = 4.0$ . To obtain the Bayes factor, one would have to multiply this with the inverse of the prior model odds (see (7.13)). If the prior assigned equal prior probability to all possible topologies, then the prior model odds would be determined by the number of trees consistent with each of the two hypothesis, a ratio that is easy to calculate. If one class of trees is empty, a conservative estimate of the Bayes factor would be obtained by adding one tree of this class to the sample.

## 236 **Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

### **7.9 Prior probability distributions**

We will end with a few cautionary notes about priors. Beginners often worry excessively about the influence of the priors on their results and the subjectivity that this introduces into the inference procedure. In most cases however, the exact form of the priors (within rather generous bounds) has negligible influence on the posterior distribution. If this is a concern, it can always be confirmed by varying the prior assumptions.

The default priors used in MRBAYES are designed to be vague or uninformative probability distributions on the model parameters. When the data contain little information about some parameters, one would therefore expect the corresponding posterior probability distributions to be diffuse. As long as we can sample adequately from these distributions, which can be a problem if there are many of them (Nylander *et al.*, 2004), the results for other parameters should not suffer. We also know from simulations that the Bayesian approach does well even when the model is moderately overparameterized (Huelsenbeck & Rannala, 2004). Thus, the Bayesian approach typically handles weak data quite well.

However, the parameter space of phylogenetic models is vast and occasionally there are large regions with inferior but not extremely low likelihoods that attract the chain when the data are weak. The characteristic symptom is that the sample from the posterior is concentrated on parameter values that the investigator considers unlikely or unreasonable, for instance in comparison with the maximum likelihood estimates. We have seen a few examples involving models of rate variation applied to very small numbers of variable sites. In these cases, one can either choose to analyze the data under a simpler model (probably the best option in most cases) or include background information into the priors to emphasize the likely regions of parameter space.

## PRACTICE

Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck

### 7.10 Introduction to MRBAYES

The rest of this chapter is devoted to two tutorials that will get you started using MRBAYES 3 (Huelsenbeck & Ronquist, 2001; Ronquist & Huelsenbeck, 2003). For more help using the program, visit its website at <http://www.mrbayes.net>. The program has a command-line interface and should run on a variety of computer platforms, including clusters of Macintosh and UNIX computers. Note that the computer should be reasonably fast and should have a lot of RAM memory (depending on the size of the data matrix, the program may require hundreds of megabytes of memory). The program is optimized for speed and not for minimizing memory requirements.

Throughout the tutorial text, we will use `typewriter` font for what you see on the screen and what is in the input file. What you should type is given in **bold font**.

#### 7.10.1 Acquiring and installing the program

MRBAYES 3 is distributed without charge by download from the MRBAYES website (<http://mrbayes.net>). If someone has given you a copy of MRBAYES 3, we strongly suggest that you download the most recent version from this site. The site also gives information about the MRBAYES users' email list and describes how you can report bugs or contribute to the project.

MRBAYES 3 is a plain-vanilla program that uses a command-line interface and therefore behaves virtually the same on all platforms – Macintosh, Windows, and Unix. There is a separate download package for each platform. The Macintosh and Windows versions are ready to use after unzipping. If you decide to run the program under Unix/Linux, or in the Unix environment on a Mac OS X computer, then you will need to compile the program from the source code first. The MRBAYES website provides detailed instructions on how to do this.

In addition to the normal serial version, MRBAYES 3 is also available in a parallel version that uses MPI to distribute chains across two or more available processors. You can use this version to run MRBAYES on a computer cluster or on a single machine with several processors or processor cores available. See the MRBAYES website for detailed instructions.

All three packages of MRBAYES come with example data files. These are intended to show various types of analyses you can perform with the program, and you can use them as templates for your own analyses. Two of the files, `primates.nex` and `cynmix.nex`, will be used in the tutorials that follow.

## 238 **Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

### 7.10.2 Getting started

Start MRBAYES by double-clicking the application icon (or typing `./mb` in the Unix environment) and you will see the information below:

```
MrBayes 3.2.0

(Bayesian Analysis of Phylogeny)

by

John P. Huelsenbeck, Fredrik Ronquist, and Paul van der Mark

Section of Ecology, Behavior and Evolution
Division of Biological Sciences
University of California, San Diego
johnh@biomail.ucsd.edu

School of Computational Science
Florida State University
ronquist@scs.fsu.edu
paulvdm@scs.fsu.edu

Distributed under the GNU General Public License

Type 'help' or 'help <command>' for information
on the commands that are available.

MrBayes >
```

The order of the authors is randomized each time you start the program, so don't be surprised if the order differs from the one above. Note the `MrBayes >` prompt at the bottom, which tells you that MRBAYES is ready for your commands.

### 7.10.3 Changing the size of the MRBAYES window

Some MRBAYES commands will output a lot of information and write fairly long lines, so you may want to change the size of the MRBAYES window to make it easier to read the output. On Macintosh and Unix machines, you should be able to increase the window size simply by dragging the margins. On a Windows machine, you cannot increase the size of the window beyond the preset value by simply dragging the margins, but (on Windows XP, 2000 and NT) you can change both the size of the screen buffer and the console window by right-clicking on the blue title bar of the MRBAYES window and then selecting Properties in the menu that

## 239 Bayesian Phylogenetic Analysis Using MRBAYES: practice

appears. Make sure the Layout tab is selected in the window that appears, and then set the “Screen Buffer Size” and “Window Size” to the desired values.

### 7.10.4 Getting help

At the MrBayes > prompt, type **help** to see a list of the commands available in . Most commands allow you to set values (options) for different parameters. If you type **help <command>**, where <command> is any of the listed commands, you will see the help information for that command as well as a description of the available options. For most commands, you will also see a list of the current settings at the end. Try, for instance, **help lset** or **help mcmc**. The **lset** settings table looks like this:

Parameter	Options	Current Setting
Nucmodel	4by4/Doublet/Codon	4by4
Nst	1/2/6	1
Code	Universal/Vertmt/Mycoplasma/ Yeast/Ciliates/Metmt	Universal
Ploidy	Haploid/Diploid	Diploid
Rates	Equal/Gamma/Propinv/Invgamma/Adgamma	Equal
Ngammacat	<number>	4
Usegibbs	Yes/No	No
Gibbsfreq	<number>	100
Nbetacat	<number>	5
Omegavar	Equal/Ny98/M3	Equal
Covarion	No/Yes	No
Coding	All/Variable/Noabsencesites/ Nopresencesites	All
Parsmodel	No/Yes	No

Note that MRBAYES 3 supports abbreviation of commands and options, so in many cases it is sufficient to type the first few letters of a command or option instead of the full name.

A complete list of commands and options is given in the command reference, which can be downloaded from the program web site (<http://www.mrbayes.net>). You can also produce an ASCII text version of the command reference at any time by giving the command **manual** to MRBAYES. Further help is available in a set of hyperlinked html pages produced by Jeff Bates and available on the MRBAYES web site. Finally, you can get in touch with other MRBAYES users and developers through the mrbayes-users’ email list (subscription information on the MRBAYES website).

## 240 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck

### 7.11 A simple analysis

This section is a tutorial based on the `primates.nex` data file. It will guide you through a basic Bayesian MCMC analysis of phylogeny, explaining the most important features of the program. There are two versions of the tutorial. You will first find a Quick-Start version for impatient users who want to get an analysis started immediately. The rest of the section contains a much more detailed description of the same analysis.

#### 7.11.1 Quick start version

There are four steps to a typical Bayesian phylogenetic analysis using MRBAYES:

- (i) Read the NEXUS data file.
- (ii) Set the evolutionary model.
- (iii) Run the analysis.
- (iv) Summarize the samples.

In more detail, each of these steps is performed as described in the following paragraphs.

(1) At the `MrBayes >` prompt, type **execute primates.nex**. This will bring the data into the program. When you only give the data file name (`primates.nex`), the MRBAYES program assumes that the file is in the current directory. If this is not the case, you have to use the full or relative path to your data file, for example, **execute ../taxa/primates.nex**. If you are running your own data file for this tutorial, beware that it may contain some MRBAYES commands that can change the behavior of the program; delete those commands or put them in square brackets to follow this tutorial.

(2) At the `MrBayes >` prompt, type **lset nst=6 rates=invgamma**. This sets the evolutionary model to the GTR model with gamma-distributed rate variation across sites and a proportion of invariable sites. If your data are not DNA or RNA, if you want to invoke a different model, or if you want to use non-default priors, refer to the manual available from the program web site.

(3.1) At the `MrBayes >` prompt, type **mcmc ngen = 10 000 samplefreq = 10**. This will ensure that you get at least a thousand samples (10 000/10) from the posterior probability distribution. For larger data sets you probably want to run the analysis longer and sample less frequently (the default sampling frequency is every hundredth generation and the default number of generations is one million). During the run, MRBAYES prints samples of substitution model parameters to one or more files ending with the suffix “.p” and tree samples to one or more files ending with the suffix “.t”. You can find the predicted remaining time to completion of the analysis in the last column printed to screen.

## 241 Bayesian Phylogenetic Analysis Using MRBAYES: practice

(3.2) If the standard deviation of split frequencies is below 0.05 (or 0.01 for more precise results) after 10 000 generations, stop the run by answering **no** when the program asks *Continue the analysis?* (yes/no). Otherwise, keep adding generations until the value falls below 0.05 (or 0.01).

(4.1) Summarize the parameter values by typing **sump burnin = 250** (or whatever value corresponds to 25% of your samples). The program will output a table with summaries of the samples of the substitution model parameters, including the mean, mode, and 95% credibility interval of each parameter. Make sure that the potential scale reduction factor (PSRF) is reasonably close to 1.0 for all parameters (ideally below 1.02); if not, you need to run the analysis longer.

(4.2) Summarize the trees by typing **sumt burnin=250** (or whatever value corresponds to 25% of your samples). The program will output a *cladogram* with the posterior probabilities for each split and a *phylogram* with mean branch lengths. The trees will also be printed to a file that can be read by tree drawing programs such as TREEVIEW (see Chapter 5), MACCLADE, MESQUITE, and FIGTREE (see Chapter 5).

It does not have to be more complicated than this; however, as you get more proficient you will probably want to know more about what is happening behind the scenes. The rest of this section explains each of the steps in more detail and introduces you to all the implicit assumptions you are making and the machinery that MRBAYES uses in order to perform your analysis.

### 7.11.2 Getting data into MRBAYES

To get data into MRBAYES, you need a so-called NEXUS file that contains aligned nucleotide or amino acid sequences, morphological ("standard") data, restriction site (binary) data, or any mix of these four data types. The NEXUS data file is often generated by another program, such as MACCLADE or MESQUITE. Note, however, that MRBAYES version 3 does not support the full NEXUS standard, so you may have to do a little editing of the file for MRBAYES to process it properly. In particular, MRBAYES uses a fixed set of symbols for each data type and does not support user-defined symbols. The supported symbols are A, C, G, T, R, Y, M, K, S, W, H, B, V, D, N for DNA data; A, C, G, U, R, Y, M, K, S, W, H, B, V, D, N for RNA data; A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V, X for protein data; 0, 1 for restriction (binary) data; and 0, 1, 2, 3, 4, 5, 6, 5, 7, 8, 9 for standard (morphology) data. In addition to the standard one-letter ambiguity symbols for DNA and RNA listed above, ambiguity can also be expressed using the NEXUS parenthesis or curly braces notation. For instance, a taxon polymorphic for states 2 and 3 can be coded as (23), (2,3), 23, or 2,3 and a taxon with either amino acid A or F can be coded as (AF), (A,F), AF or A,F. Like most other statistical phylogenetics programs, MRBAYES effectively treats polymorphism and uncertainty the same

**242 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

way (as uncertainty), so it does not matter whether you use parentheses or curly braces. If you have symbols in your matrix other than the ones supported by MRBAYES, you will need to replace them before processing the data block in MRBAYES. You will also need to remove the “Equate” and “Symbols” statements in the “Format” line if they are included. Unlike the NEXUS standard, MRBAYES supports data blocks that contain mixed data types as described below.

To put the data into MRBAYES, type **execute <filename>** at the `MrBayes >` prompt, where **<filename>** is the name of the input file. To process our example file, type **execute primates.nex** or simply **exe primates.nex** to save some typing. Note that the input file must be located in the same folder (directory) where you started the MRBAYES application (or else you will have to give the path to the file) and the name of the input file should not have blank spaces. If everything proceeds normally, MRBAYES will acknowledge that it has read the data in the DATA block of the NEXUS file by outputting some information about the file read in.

**7.11.3 Specifying a model**

All of the commands are entered at the `MrBayes >` prompt. At a minimum two commands, `lset` and `prset`, are required to specify the evolutionary model that will be used in the analysis. Usually, it is also a good idea to check the model settings prior to the analysis using the `showmodel` command. In general, `lset` is used to define the structure of the model and `prset` is used to define the prior probability distributions on the parameters of the model. In the following, we will specify a GTR + I +  $\Gamma$  model (a General Time Reversible model with a proportion of invariable sites and a gamma-shaped distribution of rate variation across sites) for the evolution of the mitochondrial sequences and we will check all of the relevant priors. If you are unfamiliar with stochastic models of molecular evolution, we suggest that you consult Chapters 4 and 10 in this book or a general text, such as Felsenstein (2003).

In general, a good start is to type **help lset**. Ignore the help information for now and concentrate on the table at the bottom of the output, which specifies the current settings. It should look like this:

Model settings for partition 1:

Parameter	Options	Current Setting
Nucmodel	4by4/Doublet/Codon	4by4
Nst	1/2/6	1
Code	Universal/Vertmt/Mycoplasma/ Yeast/Ciliates/Metmt	Universal
Ploidy	Haploid/Diploid	Diploid

## 243 Bayesian Phylogenetic Analysis Using MRBAYES: practice

Rates	Equal/Gamma/Propinv/Invgamma/Adgamma	Equal
Ngammacat	<number>	4
Usegibbs	Yes/No	No
Gibbsfreq	<number>	100
Nbetacat	<number>	5
Omegavar	Equal/Ny98/M3	Equal
Covarion	No/Yes	No
Coding	All/Variable/Noabsencesites/ Nopresencesites	All
Parsmodel	No/Yes	No

-----

First, note that the table is headed by `Model settings` for partition 1. By default, MRBAYES divides the data into one partition for each type of data you have in your DATA block. If you have only one type of data, all data will be in a single partition by default. How to change the partitioning of the data will be explained in the second tutorial.

The `Nucmodel` setting allows you to specify the general type of DNA model. The `Doublet` option is for the analysis of paired stem regions of ribosomal DNA and the `Codon` option is for analyzing the DNA sequence in terms of its codons. We will analyze the data using a standard nucleotide substitution model, in which case the default `4by4` option is appropriate, so we will leave `Nucmodel` at its default setting.

The general structure of the substitution model is determined by the `Nst` setting. By default, all substitutions have the same rate (`Nst=1`), corresponding to the F81 model (or the JC model if the stationary state frequencies are forced to be equal using the `prset` command, see below). We want the GTR model (`Nst=6`) instead of the F81 model so we type `lset nst=6`. MRBAYES should acknowledge that it has changed the model settings.

The `Code` setting is only relevant if the `Nucmodel` is set to `Codon`. The `Plويدy` setting is also irrelevant for us. However, we need to change the `Rates` setting from the default `Equal` (no rate variation across sites) to `Invgamma` (gamma-shaped rate variation with a proportion of invariable sites). Do this by typing `lset rates = invgamma`. Again, MRBAYES will acknowledge that it has changed the settings. We could have changed both `lset` settings at once if we had typed `lset nst = 6 rates = invgamma` in a single line.

We will leave the `Ngammacat` setting (the number of discrete categories used to approximate the gamma distribution) at the default of four. In most cases, four rate categories are sufficient. It is possible to increase the accuracy of the likelihood calculations by increasing the number of rate categories. However, the time it will take to complete the analysis will increase in direct proportion to the number of

**244 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

rate categories you use, and the effects on the results will be negligible in most cases.

The default behavior for the discrete gamma model of rate variation across sites is to sum site probabilities across rate categories. To sample those probabilities using a Gibbs sampler, we can set the `Usegibbs` setting to `Yes`. The Gibbs sampling approach is much faster and requires less memory, but it has some implications you have to be aware of. This option and the `Gibbsfreq` option are discussed in more detail in the MRBAYES manual.

Of the remaining settings, it is only `Covarion` and `Parsmodel` that are relevant for single nucleotide models. We will use neither the parsimony model nor the covarion model for our data, so we will leave these settings at their default values. If you type `help lset` now to verify that the model is correctly set, the table should look like this:

Model settings for partition 1:

Parameter	Options	Current Setting
Nucmodel	4by4/Doublet/Codon	4by4
Nst	1/2/6	6
Code	Universal/Vertmt/Mycoplasma/ Yeast/Ciliates/Metmt	Universal
Ploidy	Haploid/Diploid	Diploid
Rates	Equal/Gamma/Propinv/Invgamma/Adgamma	Invgamma
Ngammacat	<number>	4
Usegibbs	Yes/No	No
Gibbsfreq	<number>	100
Nbetacat	<number>	5
Omegavar	Equal/Ny98/M3	Equal
Covarion	No/Yes	No
Coding	All/Variable/Noabsencesites/ Nopresencesites	All
Parsmodel	No/Yes	No

**7.11.4 Setting the priors**

We now need to set the priors for our model. There are six types of parameters in the model: the topology, the branch lengths, the four stationary frequencies of the nucleotides, the six different nucleotide substitution rates, the proportion of invariable sites, and the shape parameter of the gamma distribution of rate variation. The default priors in MRBAYES work well for most analyses, and we will not change any of them for now. By typing `help prset` you can obtain a list of the

## 245 Bayesian Phylogenetic Analysis Using MRBAYES: practice

default settings for the parameters in your model. The table at the end of the help information reads:

Model settings for partition 1:

Parameter	Options	Current Setting
Tratiopr	Beta/Fixed	Beta(1.0,1.0)
Revmatpr	Dirichlet/Fixed	Dirichlet (1.0,1.0,1.0,1.0,1.0,1.0)
Aamodelpr	Fixed/Mixed	Fixed(Poisson)
Aarevmatpr	Dirichlet/Fixed	Dirichlet(1.0,1.0,...)
Omegapr	Dirichlet/Fixed	Dirichlet(1.0,1.0)
Ny98omega1pr	Beta/Fixed	Beta(1.0,1.0)
Ny98omega3pr	Uniform/Exponential/Fixed	Exponential(1.0)
M3omegapr	Exponential/Fixed	Exponential
Codoncatfreqs	Dirichlet/Fixed	Dirichlet(1.0,1.0,1.0)
Statefreqpr	Dirichlet/Fixed	Dirichlet(1.0,1.0,1.0,1.0)
Shapepr	Uniform/Exponential/Fixed	Uniform(0.0,200.0)
Ratecorrpr	Uniform/Fixed	Uniform(-1.0,1.0)
Pinvarpr	Uniform/Fixed	Uniform(0.0,1.0)
Covswitchpr	Uniform/Exponential/Fixed	Uniform(0.0,100.0)
Symdirihyperpr	Uniform/Exponential/Fixed	Fixed(Infinity)
Topologypr	Uniform/Constraints	Uniform
Brlenspr	Unconstrained/Clock	Unconstrained:Exp(10.0)
Treeheightpr	Exponential/Gamma	Exponential(1.0)
Speciationpr	Uniform/Exponential/Fixed	Uniform(0.0,10.0)
Extinctionpr	Uniform/Exponential/Fixed	Uniform(0.0,10.0)
Sampleprob	<number>	1.00
Thetapr	Uniform/Exponential/Fixed	Uniform(0.0,10.0)
Nodeagepr	Unconstrained/Calibrated	Unconstrained
Treeagepr	Fixed/Uniform/ Offsetexponential	Fixed(1.00)
Clockratepr	Strict/Cpp/Bm	Strict
Cppratepr	Fixed/Exponential	Exponential(0.10)
Psigammapr	Fixed/Exponential/Uniform	Fixed(1.00)
Nupr	Fixed/Exponential/Uniform	Fixed(1.00)
Ratepr	Fixed/Variable=Dirichlet	Fixed

We need to focus on `Revmatpr` (for the six substitution rates of the GTR rate matrix); `Statefreqpr` (for the stationary nucleotide frequencies of the GTR rate matrix); `Shapepr` (for the shape parameter of the gamma distribution of rate variation); `Pinvarpr` (for the proportion of invariable sites); `Topologypr` (for the topology); and `Brlenspr` (for the branch lengths).

**246 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

The default prior probability density is a flat Dirichlet (all values are 1.0) for both `Revmatpr` and `Statefreqpr`. This is appropriate if we want to estimate these parameters from the data assuming no prior knowledge about their values. It is possible to fix the rates and nucleotide frequencies, but this is generally not recommended. However, it is occasionally necessary to fix the nucleotide frequencies to be equal, for instance, in specifying the JC and SYM models. This would be achieved by typing `prset statefreqpr = fixed(equal)`.

If we wanted to specify a prior that puts more emphasis on equal nucleotide frequencies than the default flat Dirichlet prior, we could, for instance, use `prset statefreqpr = Dirichlet(10,10,10,10)` or, for even more emphasis on equal frequencies, `prset statefreqpr = Dirichlet(100,100,100,100)`. The sum of the numbers in the Dirichlet distribution determines how focused the distribution is, and the balance between the numbers determines the expected proportion of each nucleotide (in the order A, C, G, and T). Usually, there is a connection between the parameters in the Dirichlet distribution and the observations. For example, you can think of a Dirichlet (150,100,90,140) distribution as one arising from observing 150 As, 100 Cs, 90 Gs, and 140 Ts in some set of reference sequences. If your set of sequences is independent of those reference sequences, but this reference set is clearly relevant to the analysis of your sequences, it might be reasonable to use those numbers as a prior in your analysis.

In our analysis, we will be cautious and leave the prior on state frequencies at its default setting. If you have changed the setting according to the suggestions above, you need to change it back by typing `prset statefreqpr = Dirichlet(1,1,1,1)` or `prst = Dir(1,1,1,1)` if you want to save some typing. Similarly, we will leave the prior on the substitution rates at the default flat Dirichlet(1,1,1,1,1,1) distribution.

The `Shapepr` parameter determines the prior for the  $\alpha$  (shape) parameter of the gamma distribution of rate variation. We will leave it at its default setting, a uniform distribution spanning a wide range of  $\alpha$  values. The prior for the proportion of invariable sites is set with `Pinvarpr`. The default setting is a uniform distribution between 0 and 1, an appropriate setting if we don't want to assume any prior knowledge about the proportion of invariable sites.

For topology, the default `Uniform` setting for the `Topologypr` parameter puts equal probability on all distinct, fully resolved topologies. The alternative is to constrain some nodes in the tree to always be present, but we will not attempt that in this analysis.

The `BrlenSpr` parameter can either be set to unconstrained or clock-constrained. For trees without a molecular clock (unconstrained) the branch length prior can be set either to exponential or uniform. The default exponential prior with parameter 10.0 should work well for most analyses. It has an expectation of  $1/10 = 0.1$ , but allows a wide range of branch length values (theoretically from 0 to

## 247 Bayesian Phylogenetic Analysis Using MRBAYES: practice

infinity). Because the likelihood values vary much more rapidly for short branches than for long branches, an exponential prior on branch lengths is closer to being uninformative than a uniform prior.

### 7.11.5 Checking the model

To check the model before we start the analysis, type **showmodel**. This will give an overview of the model settings. In our case, the output will be as follows:

Model settings:

```
Datatype = DNA
Nucmodel = 4by4
Nst      = 6
          Substitution rates, expressed as proportions
          of the rate sum, have a Dirichlet prior
          (1.00,1.00,1.00,1.00,1.00,1.00)
Covarion = No
# States = 4
          State frequencies have a Dirichlet prior
          (1.00,1.00,1.00,1.00)
Rates    = Invgamma
          Gamma shape parameter is uniformly dist-
          ributed on the interval (0.00,200.00).
          Proportion of invariable sites is uniformly dist-
          ributed on the interval (0.00,1.00).
          Gamma distribution is approximated using 4 categories.
          Likelihood summarized over all rate categories
          in each generation.
```

Active parameters:

Parameters

```
-----
Revmat      1
Statefreq   2
Shape       3
Pinvar      4
Topology    5
Brlens      6
-----
```

```
1 -- Parameter = Revmat
      Type      = Rates of reversible rate matrix
      Prior     = Dirichlet(1.00,1.00,1.00,1.00,1.00,1.00)
```

**248 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

```

2 -- Parameter = Pi
    Type      = Stationary state frequencies
    Prior     = Dirichlet

3 -- Parameter = Alpha
    Type      = Shape of scaled gamma distribution of site rates
    Prior     = Uniform(0.00,200.00)

4 -- Parameter = Pinvar
    Type      = Proportion of invariable sites
    Prior     = Uniform(0.00,1.00)

5 -- Parameter = Tau
    Type      = Topology
    Prior     = All topologies equally probable a priori
    Subparam. = V

6 -- Parameter = V
    Type      = Branch lengths
    Prior     = Unconstrained:Exponential(10.0)
    
```

Note that we have six types of parameters in our model. All of these parameters will be estimated during the analysis (to fix them to some estimated values, use the `prset` command and specify a fixed prior). To see more information about each parameter, including its starting value, use the `showparams` command. The `startvals` command allows one to set the starting values of each chain separately.

**7.11.6 Setting up the analysis**

The analysis is started by issuing the `mcmc` command. However, before doing this, we recommend that you review the run settings by typing `help mcmc`. In our case, we will get the following table at the bottom of the output:

Parameter	Options	Current Setting
Seed	<number>	144979379
Swapseed	<number>	1587146502
Ngen	<number>	10000
Nruns	<number>	2
Nchains	<number>	4
Temp	<number>	0.200000
Reweight	<number>, <number>	0.00 v 0.00 ^
Swapfreq	<number>	1
Nswaps	<number>	1

## 249 Bayesian Phylogenetic Analysis Using MRBAYES: practice

```

Samplefreq      <number>          10
Printfreq       <number>          100
Printall        Yes/No           Yes
Printmax        <number>           8
Mcmcdiag        Yes/No           Yes
Diagnfreq       <number>          1000
Diagnstat       Avgstddev/Maxstddev Avgstddev
Minpartfreq     <number>          0.20
Allchains       Yes/No           No
Allcomps        Yes/No           No
Relburnin       Yes/No           Yes
Burnin          <number>           0
Burninfrac      <number>          0.25
Stoprule        Yes/No           No
Stopval         <number>          0.05
Savetrees       Yes/No           No
Checkpoint      Yes/No           Yes
Checkfreq       <number>          100000
Filename        <name>             primates.nex.<p/t>
Startparams     Current/Reset      Current
Starttree       Current/Random     Current
Nperts          <number>           0
Data            Yes/No           Yes
Ordertaxa       Yes/No           No
Append          Yes/No           No
Autotune        Yes/No           Yes
Tunefreq        <number>          100
Scientific      Yes/No           Yes
    
```

The Seed is simply the seed for the random number generator, and Swapseed is the seed for the separate random number generator used to generate the chain swapping sequence (see below). Unless they are set to user-specified values, these seeds are generated from the system clock, so your values are likely to be different from the ones in the screen dump above. The Ngen setting is the number of generations for which the analysis will be run. It is useful to run a small number of generations first to make sure the analysis is correctly set up and to get an idea of how long it will take to complete a longer analysis. We will start with 10 000 generations. To change the Ngen setting without starting the analysis we use the `mcmc` command, which is equivalent to `mcmc` except that it does not start the analysis. Type `mcmc ngen = 10 000` to set the number of generations to 10 000. You can type `help mcmc` to confirm that the setting was changed appropriately.

**250 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

By default, MRBAYES will run two simultaneous, completely independent, analyses starting from different random trees (`Nruns = 2`). Running more than one analysis simultaneously allows MRBAYES to calculate convergence diagnostics on the fly, which is very helpful in determining when you have a good sample from the posterior probability distribution. The idea is to start each run from different randomly chosen trees. In the early phases of the run, the two runs will sample very different trees, but when they have reached convergence (when they produce a good sample from the posterior probability distribution), the two tree samples should be very similar.

To make sure that MRBAYES compares tree samples from the different runs, check that `McmcDiagn` is set to `yes` and that `DiagnFreq` is set to some reasonable value, such as every 1000th generation. MRBAYES will now calculate various run diagnostics every `DiagnFreq` generation and print them to a file with the name `<Filename>.mcmc`. The most important diagnostic, a measure of the similarity of the tree samples in the different runs, will also be printed to screen every `DiagnFreq` generation. Every time the diagnostics are calculated, either a fixed number of samples (`burnin`) or a percentage of samples (`burninfrac`) from the beginning of the chain is discarded. The `relburnin` setting determines whether a fixed `burnin` (`relburnin = no`) or a `burnin` percentage (`relburnin = yes`) is used. By default, MRBAYES will discard the first 25% samples from the cold chain (`relburnin = yes` and `burninfrac = 0.25`).

By default, MRBAYES uses Metropolis coupling to improve the MCMC sampling of the target distribution. The `SwapFreq`, `Nswaps`, `Nchains`, and `Temp` settings together control the Metropolis coupling behavior. When `Nchains` is set to 1, no heating is used. When `Nchains` is set to a value  $n$  larger than 1, then  $n - 1$  heated chains are used. By default, `Nchains` is set to 4, meaning that MRBAYES will use three heated chains and one “cold” chain. In our experience, heating is essential for some data sets but it is not needed for others. Adding more than three heated chains may be helpful in analyzing large and difficult data sets. The time complexity of the analysis is directly proportional to the number of chains used (unless MRBAYES runs out of physical RAM memory, in which case the analysis will suddenly become much slower), but the cold and heated chains can be distributed among processors in a cluster of computers and among cores in multicore processors using the MPI version of the program, greatly speeding up the calculations.

MRBAYES uses an incremental heating scheme, in which chain  $i$  is heated by raising its posterior probability to the power  $1/(1 + i\lambda)$ , where  $\lambda$  is the temperature controlled by the `Temp` parameter (see Section 7.5). Every `SwapFreq` generation, two chains are picked at random and an attempt is made to swap their states. For many analyses, the default settings should work nicely. If you are running many

## 251 Bayesian Phylogenetic Analysis Using MRBAYES: practice

more than three heated chains, however, you may want to increase the number of swaps (`Nswaps`) that is tried each time the chain stops for swapping. If the frequency of swapping between chains that are adjacent in temperature is low, you may want to decrease the `Temp` parameter.

The `Samplefreq` setting determines how often the chain is sampled. By default, the chain is sampled every 100th generation, and this works well for most analyses. However, our analysis is so small that we are likely to get convergence quickly. Therefore, it makes sense to sample the chain more frequently, say every 10th generation (this will ensure that we get at least 1000 samples when the number of generations is set to 10 000). To change the sampling frequency, type **mcmc samplefreq = 10**.

When the chain is sampled, the current values of the model parameters are printed to file. The substitution model parameters are printed to a `.p` file (in our case, there will be one file for each independent analysis, and they will be called `primates.nex.run1.p` and `primates.nex.run2.p`). The `.p` files are tab delimited text files that can be imported into most statistics and graphing programs (including `TRACER`, see Chapter 18). The topology and branch lengths are printed to a `.t` file (in our case, there will be two files called `primates.nex.run1.t` and `primates.nex.run2.t`). The `.t` files are NEXUS tree files that can be imported into programs like `PAUP*`, `TREEVIEW` and `FIGTREE`. The root of the `.p` and `.t` file names can be altered using the `Filename` setting.

The `Printfreq` parameter controls the frequency with which the state of the chains is printed to screen. You can leave `Printfreq` at the default value (print to screen every 100th generation).

The default behavior of `MRBAYES` is to save trees with branch lengths to the `.t` file. Since this is what we want, we leave this setting as it is. If you are running a large analysis (many taxa) and are not interested in branch lengths, you can save a considerable amount of disk space by not saving branch lengths.

When you set up your model and analysis (the number of runs and heated chains), `MRBAYES` creates starting values for the model parameters. A different random tree with predefined branch lengths is generated for each chain and most substitution model parameters are set to predefined values. For instance, stationary state frequencies start out being equal and unrooted trees have all branch lengths set to 0.1. The starting values can be changed by using the **Startvals** command. For instance, user-defined trees can be read into `MRBAYES` by executing a NEXUS file with a “trees” block and then assigned to different chains using the **Startvals** command. After a completed analysis, `MRBAYES` keeps the parameter values of the last generation and will use those as the starting values for the next analysis unless the values are reset using `mcmc starttrees = random startvals = reset`.

## 252 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck

Since version 3.2, MRBAYES prints all parameter values of all chains (cold and heated) to a checkpoint file every `Checkpointfreq` generations, by default every 100 000 generations. The checkpoint file has the suffix `.ckp`. If you run an analysis and it is stopped prematurely, you can restart it from the last checkpoint by using `mcmc append = yes`. MRBAYES will start the new analysis from the checkpoint; it will even read in all the old trees and include them in the convergence diagnostic if needed. At the end of the new run, you will obtain parameter and tree files that are indistinguishable from those you would have obtained from an uninterrupted analysis. Our data set is so small that we are likely to get an adequate sample from the posterior before the first checkpoint.

### 7.11.7 Running the analysis

Finally, we are ready to start the analysis. Type `mcmc`. MRBAYES will first print information about the model and then list the proposal mechanisms that will be used in sampling from the posterior distribution. In our case, the proposals are the following:

The MCMC sampler will use the following moves:

```
With prob. Chain will change
  3.45 % param. 1 (Revmat) with Dirichlet proposal
  3.45 % param. 2 (Pi) with Dirichlet proposal
  3.45 % param. 3 (Alpha) with Multiplier
  3.45 % param. 4 (Pinvar) with Sliding window
 17.24 % param. 5 (Tau) and 6 (V) with Extending subtree swapper
 34.48 % param. 5 (Tau) and 6 (V) with Extending TBR
 17.24 % param. 5 (Tau) and 6 (V) with Parsimony-based SPR
 17.24 % param. 6 (V) with Random brlen hit with multiplier
```

The exact set of proposals and their relative probabilities may differ depending on the exact version of the program that you are using. Note that MRBAYES will spend most of its effort changing the topology (Tau) and branch length (V) parameters. In our experience, topology and branch lengths are the most difficult parameters to integrate over and we therefore let MRBAYES spend a large proportion of its time proposing new values for those parameters. The proposal probabilities and tuning parameters can be changed with the `Propset` command, but be warned that inappropriate changes of these settings may destroy any hopes of achieving convergence.

After the initial log likelihoods, MRBAYES will print the state of the chains every 100th generation, like this:

## 253 Bayesian Phylogenetic Analysis Using MRBAYES: practice

Chain results:

```
1 -- [-5723.498] (-5729.634) (-5727.207) (-5731.104) * [-5721.779] (-5731.701) (-5737.807) (-5730.336)
100 -- (-5726.662) (-5728.374) (-5733.144) [-5722.257] * [-5721.199] (-5726.193) (-5732.098) (-5732.563) -- 0:03:18
200 -- [-5729.666] (-5721.116) (-5731.222) (-5731.546) * (-5726.632) [-5731.803] (-5738.420) (-5729.889) -- 0:02:27
300 -- [-5727.654] (-5725.420) (-5736.655) (-5725.982) * (-5722.774) (-5743.637) (-5729.989) [-5729.954] -- 0:02:09
400 -- [-5728.809] (-5722.467) (-5742.752) (-5729.874) * (-5723.731) (-5739.025) [-5719.889] (-5731.096) -- 0:02:24
500 -- [-5728.286] (-5723.060) (-5738.274) (-5726.420) * [-5724.408] (-5733.188) (-5719.771) (-5725.882) -- 0:02:13
600 -- [-5719.082] (-5728.268) (-5728.040) (-5731.023) * (-5727.788) (-5733.390) [-5723.994] (-5721.954) -- 0:02:05
700 -- [-5717.720] (-5725.982) (-5728.786) (-5732.380) * (-5722.842) (-5727.218) [-5720.717] (-5729.936) -- 0:01:59
800 -- (-5725.531) (-5729.259) (-5743.762) [-5731.019] * (-5729.238) [-5731.272] (-5722.135) (-5727.906) -- 0:02:06
900 -- [-5721.976] (-5725.464) (-5731.774) (-5725.830) * (-5727.845) [-5723.992] (-5731.020) (-5728.988) -- 0:02:01
1000 -- (-5724.549) [-5723.807] (-5726.810) (-5727.921) * (-5729.302) [-5730.518] (-5733.236) (-5727.348) -- 0:02:06
```

Average standard deviation of split frequencies: 0.000000

```
1100 -- [-5724.473] (-5726.013) (-5723.995) (-5724.521) * (-5734.206) (-5720.464) [-5727.936] (-5723.821) -- 0:02:01
...
```

```
9000 -- (-5741.070) (-5728.937) (-5738.787) [-5719.056] * (-5731.562) [-5722.514] (-5721.184) (-5731.386) -- 0:00:13
```

Average standard deviation of split frequencies: 0.000116

```
9100 -- (-5747.669) [-5726.528] (-5738.190) (-5725.938) * (-5723.844) (-5726.963) [-5723.221] (-5724.665) -- 0:00:11
9200 -- (-5738.994) (-5725.611) (-5734.902) [-5723.275] * [-5718.420] (-5724.197) (-5730.129) (-5724.800) -- 0:00:10
9300 -- (-5740.946) (-5728.599) [-5729.193] (-5731.202) * (-5722.247) [-5723.141] (-5729.026) (-5727.039) -- 0:00:09
9400 -- (-5735.178) (-5726.517) [-5726.557] (-5728.377) * (-5721.659) (-5723.202) (-5734.709) [-5726.191] -- 0:00:07
9500 -- (-5731.041) (-5730.340) [-5721.900] (-5730.002) * (-5724.353) [-5727.075] (-5735.553) (-5725.420) -- 0:00:06
9600 -- [-5726.318] (-5737.300) (-5725.160) (-5731.890) * (-5721.767) [-5730.250] (-5742.843) (-5725.866) -- 0:00:05
9700 -- [-5726.573] (-5735.158) (-5728.509) (-5724.753) * (-5722.873) [-5729.740] (-5744.456) (-5723.282) -- 0:00:03
9800 -- (-5728.167) (-5736.140) (-5729.682) [-5725.419] * (-5723.056) (-5726.630) (-5729.571) [-5720.712] -- 0:00:02
9900 -- (-5738.486) (-5737.588) [-5732.250] (-5728.228) * (-5726.533) (-5733.696) (-5724.557) [-5722.960] -- 0:00:01
10000 -- (-5729.797) (-5725.507) (-5727.468) [-5720.465] * (-5729.313) (-5735.121) (-5722.913) [-5726.844] -- 0:00:00
```

Average standard deviation of split frequencies: 0.000105

Continue with analysis? (yes/no):

If you have the terminal window wide enough, each generation of the chain will print on a single line.

The first column lists the generation number. The following four columns with negative numbers each corresponds to one chain in the first run. Each column corresponds to one physical location in computer memory, and the chains actually shift positions in the columns as the run proceeds. The numbers are the log likelihood values of the chains. The chain that is currently the cold chain has its value surrounded by square brackets, whereas the heated chains have their values

## 254 **Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

surrounded by parentheses. When two chains successfully change states, they trade column positions (places in computer memory). If the Metropolis coupling works well, the cold chain should move around among the columns; this means that the cold chain successfully swaps states with the heated chains. If the cold chain gets stuck in one of the columns, then the heated chains are not successfully contributing states to the cold chain, and the Metropolis coupling is inefficient. The analysis may then have to be run longer or the temperature difference between chains may have to be lowered.

The star column separates the two different runs. The last column gives the time left to completion of the specified number of generations. This analysis approximately takes 1 second per 100 generations. Because different moves are used in each generation, the exact time varies somewhat for each set of 100 generations, and the predicted time to completion will be unstable in the beginning of the run. After a while, the predictions will become more accurate and the time will decrease predictably.

### 7.11.8 When to stop the analysis

At the end of the run, MRBAYES asks whether or not you want to continue with the analysis. Before answering that question, examine the average standard deviation of split frequencies. As the two runs converge onto the stationary distribution, we expect the average standard deviation of split frequencies to approach zero, reflecting the fact that the two tree samples become increasingly similar. In our case, the average standard deviation is zero after 1000 generations, reflecting the fact that both runs sampled the most probable tree in the first few samples. As the runs pick up some of the less probable trees, the standard deviation first increases slightly and then decreases to end up at a very low value. In larger phylogenetic problems, the standard deviation is typically moderately large initially and then increases for some time before it starts to decrease. Your values can differ slightly because of stochastic effects. Given the extremely low value of the average standard deviation at the end of the run, there appears to be no need to continue the analysis beyond 10 000 generations so, when MRBAYES asks “Continue with analysis? (yes/no):”, stop the analysis by typing “no.”

Although we recommend using a convergence diagnostic, such as the standard deviation of split frequencies, there are also simpler but less powerful methods of determining when to stop the analysis. The simplest technique is to examine the log likelihood values (or, more exactly, the log probability of the data given the parameter values) of the cold chain, that is, the values printed to screen within square brackets. In the beginning of the run, the values typically increase rapidly (the absolute values decrease, since these are negative numbers). This is the “burn-in” phase and the corresponding samples typically are discarded. Once the likelihood

## 255 Bayesian Phylogenetic Analysis Using MRBAYES: practice

of the cold chain stops increasing and starts to randomly fluctuate within a more or less stable range, the run may have reached stationarity, that is, it may be producing a good sample from the posterior probability distribution. At stationarity, we also expect different, independent runs to sample similar likelihood values. Trends in likelihood values can be deceiving though; you're more likely to detect problems with convergence by comparing split frequencies than by looking at likelihood trends.

When you stop the analysis, MRBAYES will print several types of information useful in optimizing the analysis. This is primarily of interest if you have difficulties in obtaining convergence, which is unlikely to happen with this analysis. We give a few tips on how to improve convergence at the end of the chapter.

### 7.11.9 Summarizing samples of substitution model parameters

During the run, samples of the substitution model parameters have been written to the .p files every `samplefreq` generation. These files are tab-delimited text files that look something like this:

```
[ID: 9409050143]
Gen      LnL      TL      r(A<->C)  ...  pi(G)    pi(T)    alpha    pinvar
1        -5723.498  3.357  0.067486  ...  0.098794 0.247609 0.580820 0.124136
10       -5727.478  3.110  0.030604  ...  0.072965 0.263017 0.385311 0.045880
...
9990     -5727.775  2.687  0.052292  ...  0.086991 0.224332 0.951843 0.228343
10000    -5720.465  3.290  0.038259  ...  0.076770 0.240826 0.444826 0.087738
```

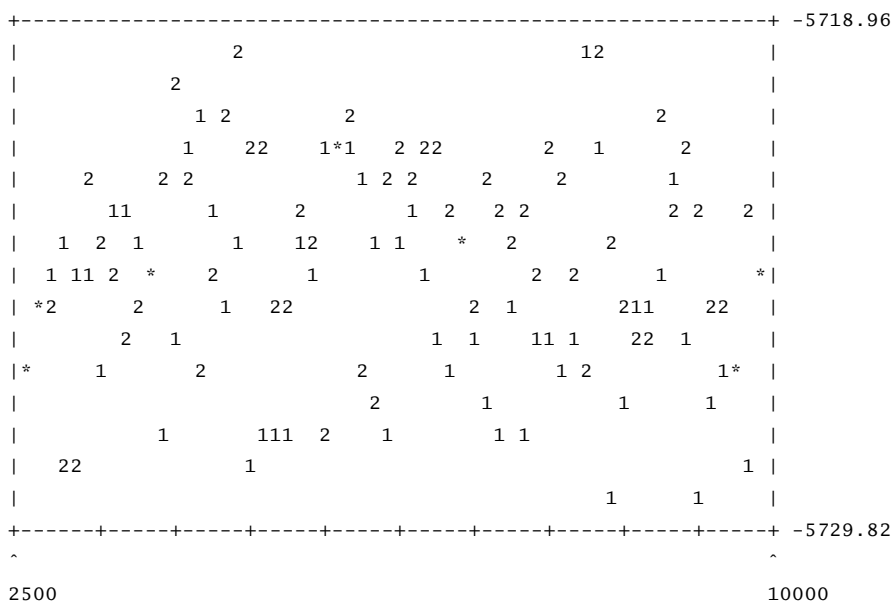
The first number, in square brackets, is a randomly generated ID number that lets you identify the analysis from which the samples come. The next line contains the column headers, and is followed by the sampled values. From left to right, the columns contain: (1) the generation number (`Gen`); (2) the log likelihood of the cold chain (`LnL`); (3) the total tree length (the sum of all branch lengths, `TL`); (4) the six GTR rate parameters (`r(A<->C)`, `r(A<->G)` etc); (5) the four stationary nucleotide frequencies (`pi(A)`, `pi(C)` etc); (6) the shape parameter of the gamma distribution of rate variation (`alpha`); and (7) the proportion of invariable sites (`pinvar`). If you use a different model for your data set, the .p files will, of course, be different.

MRBAYES provides the `sump` command to summarize the sampled parameter values. Before using it, we need to decide on the burn-in. Since the convergence diagnostic we used previously to determine when to stop the analysis discarded the first 25% of the samples and indicated that convergence had been reached after 10 000 generations, it makes sense to discard 25% of the samples obtained during the first 10 000 generations. Since we sampled every 10th generation, there are 1000

**256 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

samples (1001 to be exact, since the first generation is always sampled) and 25% translates to 250 samples. Thus, summarize the information in the . p file by typing “**sump burnin = 250.**” By default, sump will summarize the information in the . p file generated most recently, but the filename can be changed if necessary.

The sump command will first generate a plot of the generation versus the log probability of the data (the log likelihood values). If we are at stationarity, this plot should look like “white noise,” that is, there should be no tendency of increase or decrease over time. The plot should look something like this:



If you see an obvious trend in your plot, either increasing or decreasing, you probably need to run the analysis longer to get an adequate sample from the posterior probability distribution.

At the bottom of the sump output, there is a table summarizing the samples of the parameter values:

```

Model parameter summaries over the runs sampled in files
  'primates.nex.run1.p' and 'primates.nex.run2.p':
(Summaries are based on a total of 1502 samples from 2 runs)
(Each run produced 1001 samples of which 751 samples were included)

          95 % Cred. Interval
          -----
Parameter      Mean      Variance      Lower      Upper      Median      PSRF *
-----
TL              2.954334    0.069985    2.513000    3.558000    2.941000    1.242
    
```

**257 Bayesian Phylogenetic Analysis Using MRBAYES: practice**

r(A<->C)	0.044996	0.000060	0.030878	0.059621	0.044567	1.016
r(A<->G)	0.470234	0.002062	0.386927	0.557040	0.468758	1.025
r(A<->T)	0.038107	0.000073	0.023568	0.056342	0.037172	1.022
r(C<->G)	0.030216	0.000189	0.007858	0.058238	0.028350	1.001
r(C<->T)	0.396938	0.001675	0.317253	0.476998	0.394980	1.052
r(G<->T)	0.019509	0.000158	0.001717	0.047406	0.018132	1.003
pi(A)	0.355551	0.000150	0.332409	0.382524	0.357231	1.010
pi(C)	0.320464	0.000131	0.298068	0.343881	0.320658	0.999
pi(G)	0.081290	0.000043	0.067120	0.095940	0.080521	1.004
pi(T)	0.242695	0.000101	0.220020	0.261507	0.243742	1.030
alpha	0.608305	0.042592	0.370790	1.142317	0.546609	1.021
pinvar	0.135134	0.007374	0.008146	0.303390	0.126146	0.999

\* Convergence diagnostic (PSRF = Potential scale reduction factor [Gelman and Rubin, 1992], uncorrected) should approach 1 as runs converge. The values may be unreliable if you have a small number of samples. PSRF should only be used as a rough guide to convergence since all the assumptions that allow one to interpret it as a scale reduction factor are not met in the phylogenetic context.

For each parameter, the table lists the mean and variance of the sampled values, the lower and upper boundaries of the 95% credibility interval, and the median of the sampled values. The parameters are the same as those listed in the .p files: the total tree length (TL), the six reversible substitution rates (r(A<->C), r(A<->G), etc.), the four stationary state frequencies (pi(A), pi(C), etc.), the shape of the gamma distribution of rate variation across sites (alpha), and the proportion of invariable sites (pinvar). Note that the six rate parameters of the GTR model are given as proportions of the rate sum (the Dirichlet parameterization). This parameterization has some advantages in the Bayesian context; in particular, it allows convenient formulation of priors. If you want to scale the rates relative to the G-T rate, just divide all rate proportions by the G-T rate proportion.

The last column in the table contains a convergence diagnostic, the Potential Scale Reduction Factor (PSRF). If we have a good sample from the posterior probability distribution, these values should be close to 1.0. If you have a small number of samples, there may be some spread in these values, indicating that you may need to sample the analysis more often or run it longer. In our case, we can probably obtain more accurate estimates of some parameters easily by running the analysis slightly longer.

**7.11.10 Summarizing samples of trees and branch lengths**

Trees and branch lengths are printed to the .t files. These files are NEXUS-formatted tree files with a structure like this:

**258 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

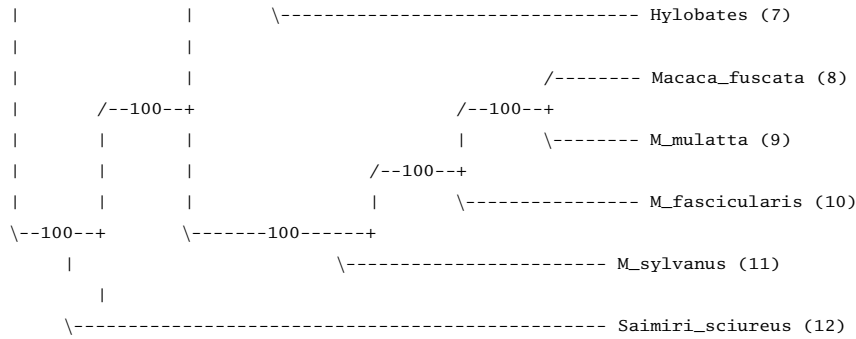
```
#NEXUS
[ID: 9409050143]
[Param: tree]
begin trees;
translate
  1 Tarsius_syrichta,
  2 Lemur_catta,
  3 Homo_sapiens,
  4 Pan,
  5 Gorilla,
  6 Pongo,
  7 Hylobates,
  8 Macaca_fuscata,
  9 M_mulatta,
  10 M_fascicularis,
  11 M_sylvanus,
  12 Saimiri_sciureus;
tree rep.1 = (((12:0.486148,((((3:0.042011,4:0.065025):0.034344,5:0.051939...
...
tree rep.10000 = (((((10:0.087647,(8:0.013447,9:0.021186):0.030524):0.0568...
end;
```

To summarize the tree and branch length information, type “sumt burnin = 250.” The `sumt` and `sump` commands each have separate burn-in settings, so it is necessary to give the burn-in here again. Most MRBAYES settings are persistent and need not be repeated every time a command is executed, but the settings are typically not shared across commands. To make sure the settings for a particular command are correct, you can always use **help <command>** before issuing the command.

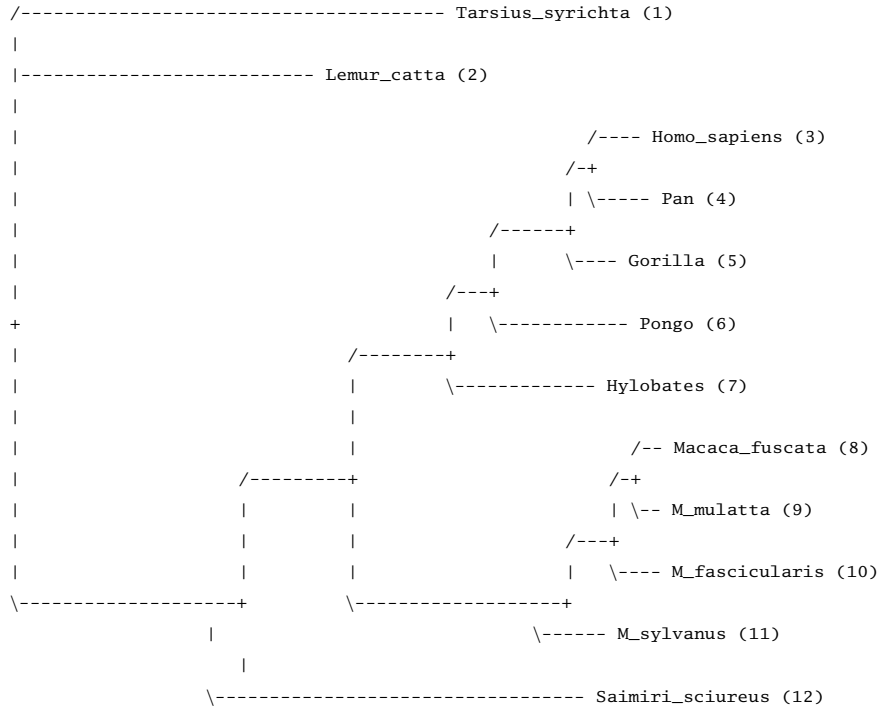
The `sumt` command will output, among other things, summary statistics for the taxon bipartitions, a tree with clade credibility (posterior probability) values, and a phylogram (if branch lengths have been saved). The summary statistics (see below) describe each split (clade) in the tree sample (dots for the taxa that are on one side of the split and stars for the taxa on the other side; for instance, the sixth split (ID 6) is the terminal branch leading to taxon 2 since it has a star in the second position and a dot in all other positions). Then it gives the number of times the split was sampled (`\#obs`), the probability of the split (`Probab.`), the standard deviation of the split frequency (`Stdev(s)`) across runs, the mean (`Mean(v)`) and variance (`Var(v)`) of the branch length, the Potential Scale Reduction Factor (PSRF), and finally the number of runs in which the split was sampled (`Nruns`). In our analysis, there is overwhelming support for a single tree, so almost all splits in this tree have a posterior probability of 1.0.



**260 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**



Phylogram (based on average branch lengths):



|-----| 0.200 expected changes per site

In the background, the `sumt` command creates three additional files. The first is a `.parts` file, which contains the list of taxon bipartitions, their posterior probability (the proportion of sampled trees containing them), and the branch lengths associated with them (if branch lengths have been saved). The branch length values are based only on those trees containing the relevant bipartition.

## 261 Bayesian Phylogenetic Analysis Using MRBAYES: practice

The second generated file has the suffix `.con` and includes two consensus trees. The first one has both the posterior probability of clades (as interior node labels) and the branch lengths (if they have been saved) in its description. A graphical representation of this tree can be generated in the program `TREEVIEW` by Rod Page or `FIGTREE` by Andrew Rambaut (see Chapter 5 and Chapter 18). The second tree only contains the branch lengths and it can be imported into a wide range of tree-drawing programs such as `MACCLADE` and `MESQUITE`. The third file generated by the `sumt` command is the `.trprobs` file, which contains the trees that were found during the MCMC search, sorted by posterior probability.

### 7.12 Analyzing a partitioned data set

`MRBAYES` handles a wide variety of data types and models, as well as any mix of these models. In this example we will look at how to set up a simple analysis of a combined data set, consisting of data from four genes and morphology for 30 taxa of gall wasps and outgroups. A similar approach can be used, for example, to set up a partitioned analysis of molecular data coming from different genes. The data set for this tutorial is found in the file `cynmix.nex`.

#### 7.12.1 Getting mixed data into MRBAYES

First, open up the NEXUS data file in a text editor. The `DATA` block of the NEXUS file should look familiar but there are some differences compared to the `primates.nex` file in the format statement:

```
Format datatype = mixed(Standard:1-166,DNA:167-3246)
interleave=yes gap=- missing=?;
```

First, the `datatype` is specified as `datatype = mixed(Standard:1--166, DNA:167-3246)`. This means that the matrix contains standard (morphology) characters in columns 1-166 and DNA characters in the remaining columns. The `mixed` `datatype` is an extension to the NEXUS standard. This extension was originated by `MRBAYES 3` and may not be compatible with other phylogenetics programs.

Second, the matrix is interleaved. It is often convenient to specify mixed data in interleaved format, with each block consisting of a natural subset of the matrix, such as the morphological data or one of the gene regions.

#### 7.12.2 Dividing the data into partitions

By default, `MRBAYES` partitions the data according to data type. There are only two data types in the matrix, so this model will include only a morphology (standard) and a DNA partition. To divide the DNA partition into gene regions, it is convenient

**262 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

to first specify character sets. In principle, this can be done from the command line, but it is more convenient to do it in a MRBAYES block in the data file. With the MRBAYES distribution, we added a file `cynmix-run.nex` with a complete MRBAYES block. For this section, we are going to create a command block from scratch, but you can consult the `cynmix-run.nex` for reference.

In your favorite text editor, create a new file called `cynmix-command.nex` in the same directory as the `cynmix.nex` file and add the following new MRBAYES block (note that each line must be terminated by a semicolon):

```
#NEXUS

begin mrbayes;
execute cynmix.nex;
charset morphology = 1-166;
charset COI = 167-1244;
charset EF1a = 1245-1611;
charset LWRh = 1612-2092;
charset 28S = 2093-3246;
```

The first line is required to comply with the NEXUS standard. With the `execute` command, we load the data from the `cynmix.nex` file and the `charset` command simply associates a name with a set of characters. For instance, the character set COI is defined above to include characters 167 to 1244. The next step is to define a partition of the data according to genes and morphology. This is accomplished with the line (add it after the lines above):

```
partition favored = 5: morphology, COI, EF1a, LWRh, 28S;
```

The elements of the `partition` command are: (1) the name of the partitioning scheme (`favored`); (2) an equal sign (`=`); (3) the number of character divisions in the scheme (`5`); (4) a colon (`:`); and (5) a list of the characters in each division, separated by commas. The list of characters can simply be an enumeration of the character numbers (the above line is equivalent to `partition favored = 5: 1-166, 167-1244, 1245-1611, 1612-2092, 2093-3246;`) but it is often more convenient to use predefined character sets as we did above. The final step is to tell MRBAYES that we want to work with this partitioning of the data instead of with the default partitioning. We do this using the `set` command:

```
set partition = favored;
```

Finally, we need to add an end statement to close the MRBAYES block. The entire file should now look like this:

## 263 Bayesian Phylogenetic Analysis Using MRBAYES: practice

```
#NEXUS

begin mrbayes;
  execute cynmix.nex;
  charset morphology = 1-166;
  charset COI = 167-1244;
  charset EF1a = 1245-1611;
  charset LWRh = 1612-2092;
  charset 28S = 2093-3246;
  partition favored = 5: morphology, COI, EF1a, LWRh, 28S;
  set partition = favored;
end;
```

When we read this block into MRBAYES, we will get a partitioned model with the first character division being morphology, the second division being the COI gene, etc. Save the data file, exit your text editor, and finally launch MRBAYES and type **execute cynmix-command.nex** to read in your data and set up the partitioning scheme.

### 7.12.3 Specifying a partitioned model

Before starting to specify the partitioned model, it is useful to examine the default model. Type “**showmodel**” and you should get this table as part of the output:

Active parameters:

Parameters	Partition(s)				
	1	2	3	4	5
Statefreq	1	2	2	2	2
Topology	3	3	3	3	3
Brlens	4	4	4	4	4

There is a lot of other useful information in the output of `showmodel` but this table is the key to the partitioned model. We can see that there are five partitions in the model and four active (free) parameters. There are two stationary state frequency parameters, one for the morphological data (parameter 1) and one for the DNA data (parameter 2). Then there is also a topology parameter (3) and a set of branch length parameters (4). Both the topology and branch lengths are the same for all partitions.

Now, assume we want a separate GTR +  $\Gamma$  + I model for each gene partition. All the parameters should be estimated separately for the individual genes. Assume further that we want the overall evolutionary rate to be (potentially) different

**264 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

across partitions, and that we want to assume gamma-shaped rate variation for the morphological data. We can obtain this model by using `lset` and `prset` with the `applyto` mechanism, which allows us to apply the settings to specific partitions. For instance, to apply a GTR +  $\Gamma$  + I model to the molecular partitions, we type **`lset applyto = (2,3,4,5) nst = 6 rates = invgamma`**. This will produce the following table when `showmodel` is invoked:

Active parameters:

Parameters	Partition(s)				
	1	2	3	4	5
Revmat	.	1	1	1	1
Statefreq	2	3	3	3	3
Shape	.	4	4	4	4
Pinvar	.	5	5	5	5
Topology	6	6	6	6	6
Brlen	7	7	7	7	7

As you can see, all molecular partitions now evolve under the correct model but all parameters (`statefreq`, `revmat`, `shape`, `pinvar`) are shared across partitions. To unlink them such that each partition has its own set of parameters, type: **`unlink statefreq = (all) revmat = (all) shape = (all) pinvar = (all)`**. Gamma-shaped rate variation for the morphological data is enforced with **`lset applyto = (1) rates = gamma`**. The trickiest part is to allow the overall rate to be different across partitions. This is achieved using the `ratepr` parameter of the `prset` command. By default, `ratepr` is set to `fixed`, meaning that all partitions have the same overall rate. By changing this to `variable`, the rates are allowed to vary under a flat Dirichlet prior. To allow all our partitions to evolve under different rates, type **`prset applyto = (all) ratepr = variable`**.

The model is now essentially complete but there is one final thing to consider. Typically, morphological data matrices do not include all types of characters. Specifically, morphological data matrices do not usually include any constant (invariable) characters. Sometimes, *autapomorphies* are not included either, and the matrix is restricted to parsimony-informative characters. For MRBAYES to calculate the probability of the data correctly, we need to inform it of this ascertainment (coding) bias. By default, MRBAYES assumes that standard data sets include all variable characters but no constant characters. If necessary, one can change this setting using `lset coding`. We will leave the `coding` setting at the default, though. Now, `showmodel` should produce this table:

## 265 Bayesian Phylogenetic Analysis Using MRBAYES: practice

Active parameters:

Parameters	Partition(s)				
	1	2	3	4	5
Revmat	.	1	2	3	4
Statefreq	5	6	7	8	9
Shape	10	11	12	13	14
Pinvar	.	15	16	17	18
Ratemultiplier	19	19	19	19	19
Topology	20	20	20	20	20
Brlens	21	21	21	21	21

### 7.12.4 Running the analysis

When the model has been completely specified, we can proceed with the analysis essentially as described above in the tutorial for the `primates.nex` data set. However, in the case of the `cynmix.nex` dataset, the analysis will have to be run longer before it converges.

When looking at the parameter samples from a partitioned analysis, it is useful to know that the names of the parameters are followed by the character division (partition) number in curly braces. For instance,  $\pi(A)\{3\}$  is the stationary frequency of nucleotide A in character division 3, which is the EF1a division in the above analysis.

In this section we have used a separate NEXUS file for the MRBAYES block. Although one can add this command block to the data file itself, there are several advantages to keeping the commands and the data blocks separate. For example, one can create a set of different analyses with different parameters in separate “command” files and submit all those files to a job scheduling system on a computer cluster.

### 7.12.5 Some practical advice

As you continue exploring Bayesian phylogenetic inference, you may find the following tips helpful:

(1) If you are anxious to get results quickly, you can try running without Metropolis coupling (heated chains). This will save a large amount of computational time at the risk of having to start over if you have difficulties getting convergence. Turn off heating by setting the `mcmc` option `nchains` to 1 and switch it on by setting `nchains` to a value larger than 1.

(2) If you are using heated chains, make sure that the acceptance rate of swaps between adjacent chains are in the approximate range of 10% to 70% (the

**266 Fredrik Ronquist, Paul van der Mark, and John P. Huelsenbeck**

acceptance rates are printed to the `.mcmc` file and to screen at the end of the run). If the acceptance rates are lower than 10%, decrease the temperature constant (`mcmc temp=<value>`); if the acceptance rates are higher than 70%, increase it.

(3) If you run multiple simultaneous analyses or use Metropolis coupling and have access to a machine with several processors or processor cores, or if you have access to a computer cluster, you can speed up your analyses considerably by running MRBAYES in parallel under MPI. See the MRBAYES website for more information about this.

(4) If you are using automatic optimization of proposal tuning parameters, and your runs are reasonably long so that MRBAYES has sufficient time to find the best settings, you should not have to adjust proposal tuning parameters manually. However, if you have difficulties getting convergence, you can try selecting a different mix of topology moves than the one used by default. For instance, the random SPR move tends to do well on some data sets, but it is switched off by default because, in general, it is less efficient than the default moves. You can add and remove topology moves by adjusting their relative proposal probabilities using the `propset` command. Use `showmoves allavailable = yes` first to see a list of all the available moves.

For more information and tips, turn to the MRBAYES website (<http://mrbayes.net>) and the MRBAYES users' email list.