

MrBayes version 3.2 Tutorial

Fredrik Ronquist

July 29, 2010

This tutorial walks you through two MrBayes example analyses. The first one describes a simple analysis, and the second one describes how you set up a more complex analysis of a partitioned data set. The tutorial ends with some practical tips and advice, as well as some pointers to additional sources of information.

Throughout the tutorial, we use `typewriter font` for things you see on screen or in a data file, and **bold font** for things you should type in. Alternative commands you could have typed in, but should not type in to follow the tutorial, are also given in `typewriter font`.

1 A Simple Analysis

This section is 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.

1.1 Quick Start Version

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

1. Read the Nexus data file
2. Set the evolutionary model

3. Run the analysis

4. 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`), MrBayes 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 full MrBayes manual and its Appendix.

3.1. At the `MrBayes >` prompt, type `mcmc ngen=20000`. This will ensure that you get at least 200 samples from the posterior probability distribution, since the default sampling frequency is every 100th generation. For larger data sets you probably want to run the analysis longer and sample less frequently. You can find the predicted remaining time to completion of the analysis in the last column printed to screen.

3.2. If the standard deviation of split frequencies is below 0.01 after 20,000 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 0.01. If you are interested mainly in the well-supported parts of the tree, a standard deviation below 0.05 may be adequate.

4.1. Type `sump relburnin=yes burninfrac=0.25` to summarize the parameter values using the same burn-in as the `mcmc` command. The program will output a table with summaries of the samples of the substitution model parameters, including the mean, mode, and 95 % credibility interval (region of Highest Posterior Density, HPD) of each parameter. Make sure that the potential scale

reduction factor (PSRF) is reasonably close to 1.0 for all parameters; if not, you need to run the analysis longer.

4.2. Summarize the trees by typing **sumt relburnin=yes burnfrac=0.25**. 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 FigTree and other tree-drawing programs, such as TreeView and Mesquite.

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.

1.2 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 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 way (as uncertainty), so it does not matter whether you use parentheses or curly braces. If you have other symbols in your matrix than

the ones supported by MrBayes, you need to replace them before processing the data block in MrBayes. You 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 (MrBayes allows you to use the shortest unambiguous version of a command). 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, or it will have to be quoted. 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.

1.3 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 + Γ model (a General Time Reversible model with a proportion of invariable sites and a gamma-shaped distribution of rates across sites) for the evolution of the mitochondrial sequences and we will check all of the relevant priors. We assume that you are familiar with the common stochastic models of molecular evolution.

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
-----------	---------	-----------------

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

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 `Ploidy` 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 4. 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 rate categories you use, and the effects on the results will be negligible in most cases.

The default behaviour 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

1.4 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 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/Fixed	Uniform
Brlenspr	Unconstrained/Clock/Fixed	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/Ibr	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 `Brlenpr` (for the branch lengths).

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 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 put 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 (roughly) 150 A's, 100 C's, 90 G's and 140 T's in some set of reference sequences. If the reference sequences are independent but 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) distribution.

The `Shapepr` parameter determines the prior for the α (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 α 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 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.

1.5 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)

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.

1.6 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>	1141605343
Swapseed	<number>	1280372787
Ngen	<number>	1000000
Nruns	<number>	2
Nchains	<number>	4
Temp	<number>	0.200000
Reweight	<number>, <number>	0.00 v 0.00 ^
Swapfreq	<number>	1
Nswaps	<number>	1
Samplefreq	<number>	100
Printfreq	<number>	100
Printall	Yes/No	Yes
Printmax	<number>	8
Mcmcdiagn	Yes/No	Yes
Diagnfreq	<number>	1000
Diagnstat	Avgstddev/Maxstddev	Avgstddev
Minpartfreq	<number>	0.10

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/ Parsimony	Current
Nperts	<number>	0
Data	Yes/No	Yes
Ordertaxa	Yes/No	No
Append	Yes/No	No
Autotune	Yes/No	Yes
Tunefreq	<number>	100

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 20,000 generations but you may want to start with an even smaller number for a larger data set. To change the **Ngen** setting without starting the analysis we use the **mcmcp** command, which is equivalent to **mcmc** except that it does not start the analysis. Type **mcmcp ngen=20000** to set the number of generations to 20,000. You can type **help mcmc** to confirm that the setting was changed appropriately.

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 (`burnfrac`) 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 `burnfrac=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 3 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 λ is the temperature controlled by the `Temp` parameter. 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 more than three heated chains, however, you may want to increase the number of swaps (**Nswaps**) that are 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, including ours. If you have a large data set, it may take longer to converge and you may want to sample less frequently or you will end up with very large files containing tree and parameter samples.

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. 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* and TreeView. 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).

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**.

Since version 3.2, MrBayes prints all parameter values of all chains (cold and heated) to a checkpoint file every `Checkfreq` 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. At the end of the new run, you will 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.

1.7 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 use move
1.79 %	Dirichlet(Revmat)
1.79 %	Slider(Revmat)
1.79 %	Dirichlet(Pi)
1.79 %	Slider(Pi)
3.57 %	Multiplier(Alpha)
17.86 %	eSS(Tau,V)
17.86 %	eTBR(Tau,V)
35.71 %	pSPR(Tau,V)
17.86 %	Multiplier(V)

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:

```
Chain results:

  1 -- (-7515.474) (-7815.502) (-7571.894) [-7511.216] * (-7912.443) (-7430.324) (-7722.968) [-7559.768]
 100 -- (-6457.486) (-6443.204) (-6362.653) [-6380.948] * (-6452.131) (-6412.384) (-6460.409) [-6335.541] -- 0:00:00
 200 -- (-6372.894) (-6284.653) [-6212.481] (-6320.671) * (-6326.804) [-6206.832] (-6368.248) (-6274.370) -- 0:01:39
 300 -- (-6215.251) (-6238.351) [-6173.761] (-6215.648) * [-6175.548] (-6162.354) (-6295.342) (-6170.237) -- 0:01:05
 400 -- (-6169.260) [-6106.352] (-6157.140) (-6134.522) * [-6044.984] (-6105.105) (-6239.651) (-6126.382) -- 0:01:38
 500 -- (-6132.093) [-6045.345] (-6071.921) (-6105.350) * [-6027.764] (-6052.897) (-6122.643) (-6054.535) -- 0:01:18
 600 -- (-6086.736) [-5966.605] (-6022.943) (-6048.775) * (-6005.907) (-6050.838) (-6052.809) [-5987.512] -- 0:01:04
 700 -- (-6071.156) [-5949.411] (-6001.893) (-6028.975) * (-5969.434) (-6034.590) (-5985.207) [-5962.131] -- 0:01:22
 800 -- (-6043.289) [-5919.917] (-5955.320) (-5990.842) * (-5934.204) (-5998.712) [-5917.514] (-5957.886) -- 0:01:12
 900 -- (-6036.192) [-5915.292] (-5940.829) (-5928.622) * (-5916.117) (-5974.419) [-5885.179] (-5947.285) -- 0:01:03
1000 -- (-6033.926) [-5879.274] (-5930.137) (-5912.750) * (-5919.677) (-5979.409) [-5849.042] (-5893.568) -- 0:01:16

Average standard deviation of split frequencies: 0.000000

1100 -- (-6015.382) (-5879.918) (-5932.478) [-5850.292] * (-5845.497) (-5970.688) [-5835.631] (-5882.916) -- 0:01:08
...
19000 -- (-5725.208) (-5728.059) (-5723.771) [-5720.516] * (-5725.163) (-5733.313) (-5731.771) [-5733.018] -- 0:00:03

Average standard deviation of split frequencies: 0.000000

19100 -- (-5721.777) (-5731.432) [-5724.683] (-5719.899) * (-5724.676) [-5725.091] (-5728.996) (-5742.658) -- 0:00:03
19200 -- (-5725.644) [-5723.736] (-5730.977) (-5718.788) * [-5732.428] (-5725.226) (-5733.051) (-5741.748) -- 0:00:02
19300 -- (-5722.932) (-5727.877) (-5729.790) [-5719.233] * [-5728.970] (-5732.444) (-5730.074) (-5731.851) -- 0:00:02
19400 -- (-5722.253) (-5732.094) (-5733.256) [-5721.040] * (-5731.382) [-5726.897] (-5734.551) (-5733.469) -- 0:00:02
19500 -- [-5723.923] (-5732.401) (-5726.903) (-5722.455) * (-5727.740) (-5722.413) (-5736.126) [-5727.055] -- 0:00:01
19600 -- (-5731.034) (-5729.754) (-5732.244) [-5725.747] * (-5725.214) (-5722.015) (-5733.053) [-5723.926] -- 0:00:01
19700 -- (-5738.424) (-5731.187) (-5728.800) [-5728.881] * (-5725.340) [-5720.537] (-5734.678) (-5725.685) -- 0:00:01
19800 -- (-5732.570) (-5732.026) (-5729.572) [-5727.604] * [-5721.525] (-5718.952) (-5741.802) (-5722.740) -- 0:00:00
19900 -- (-5724.326) (-5728.367) [-5725.441] (-5726.584) * (-5723.621) (-5730.548) (-5746.447) [-5716.807] -- 0:00:00
20000 -- (-5723.983) (-5727.877) [-5724.582] (-5720.923) * (-5730.172) (-5728.091) (-5748.344) [-5716.947] -- 0:00:00

Average standard deviation of split frequencies: 0.000520

Continue with analysis? (yes/no): 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 correspond 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 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. You can also try to reduce the temperature difference between chains, which may increase the efficiency of the Metropolis coupling.

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.

1.8 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 down to 0.0 already after 1,000 generations and then stays at very low values throughout the run. Your values can differ slightly because of stochastic effects but should show a similar trend.

In larger and more difficult analyses, you will typically see the standard deviation of split frequencies come down much more slowly towards 0.0; the standard deviation can even increase temporarily, especially in the early part of the run. A rough guide is that an average standard deviation below 0.01 is very good indication of convergence, while values between 0.01 and 0.05 may be adequate depending on the purpose of your analysis. The `sumt` command (see below) allows you to examine the error (standard deviation) associated with each clade in the tree. Typically, most of the error is associated with clades that are not very well supported (posterior probabilities well below 0.95), and getting accurate estimates of those probabilities may not be an important concern.

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 20,000 generations so when MrBayes asks `Continue with analysis? (yes/no): stop`

the analysis by typing `no`.

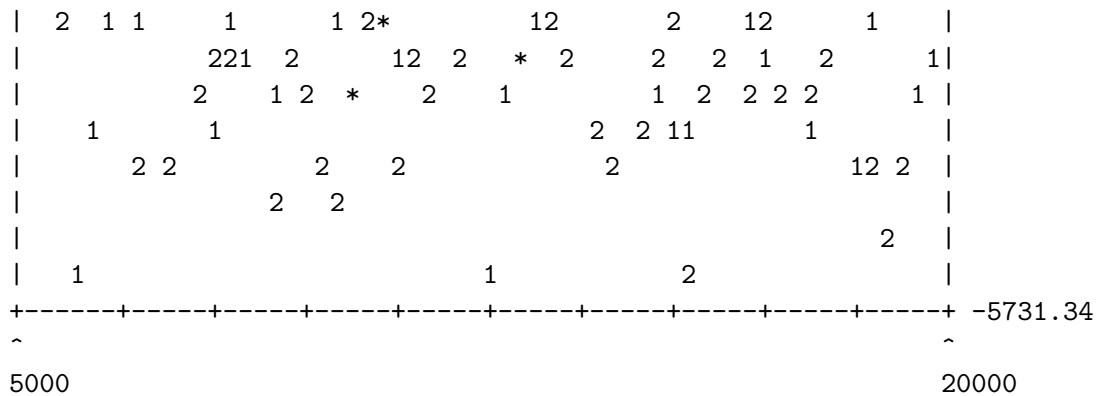
Although we recommend using a convergence diagnostic, such as the standard deviation of split frequencies, there are also simpler but less powerful methods of determining when to stop the analysis. The simplest technique is to examine the log likelihood values (or, more exactly, the log probability of 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). In our case, the values increase from below -7500 to around -5725 in the first few thousand generations. This is the "burn-in" phase and the corresponding samples are typically discarded. Once the likelihood of the cold chain stops to increase 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 tutorial.

1.9 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 (numbers are actually given in scientific format by default, so the files do not look quite as nice as the one below although they are structurally equivalent):

```
[ID: 9409050143]
Gen      LnL          TL      r(A<->C)  ...  pi(T)      alpha      pinvar
1        -7821.374    2.100    0.166667  ...  0.250000   0.500000   0.000000
100      -6328.159    2.091    0.166667  ...  0.307263   0.842091   0.036693
```

If you see an obvious trend in your plot, either increasing or decreasing, you very likely 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:

Parameter	Mean	Variance	95% HPD Interval		Median	PSRF+
			Lower	Upper		
TL	2.925036	0.067504	2.415013	3.379712	2.928130	0.997
r(A<->C)	0.046638	0.000087	0.025996	0.061754	0.047223	1.033
r(A<->G)	0.472949	0.001944	0.380185	0.538198	0.475733	1.093
r(A<->T)	0.038436	0.000070	0.024704	0.057508	0.037291	1.067
r(C<->G)	0.028027	0.000145	0.005312	0.049157	0.028217	1.023
r(C<->T)	0.394395	0.001729	0.319370	0.473755	0.390730	1.110
r(G<->T)	0.019556	0.000169	0.000078	0.044754	0.017799	1.018
pi(A)	0.355732	0.000173	0.333218	0.383760	0.354468	1.064
pi(C)	0.317528	0.000128	0.299947	0.342945	0.317121	1.031
pi(G)	0.082920	0.000045	0.072361	0.096352	0.082537	1.164
pi(T)	0.243820	0.000085	0.224679	0.260468	0.244298	1.014
alpha	0.693416	0.065342	0.336440	1.174297	0.655553	1.061
pinvar	0.165968	0.009690	0.001575	0.310722	0.178146	1.090

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. A reasonable goal might be to aim for values between 1.00 and 1.02 but it can be difficult to achieve this for all parameters in the model in larger and more complicated analyses. In our case, we can probably easily obtain more accurate estimates by running the analysis slightly longer.

1.10 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 (the real files have branch lengths printed in scientific format so they look slightly more messy but the structure is the same):

```
#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 gen.1 = [&U] (((12:0.100000,((((3:0.100000,4:0.100000):0.100000...
...
  tree gen.20000 = [&U] (((((10:0.087647,(8:0.013447,9:0.021186):0.030...
end;
```

To summarize the tree and branch length information, type **sumt relburnin = yes burninfrac = 0.25**. 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 output first gives a key to each partition in the tree sample using dots for the taxa that are on one side of the partition and stars for the taxa on the other side. For instance, the 14th partition (ID 14) in the output below represents the clade *Homo* (taxon 3) and *Pan* (taxon 4), since there are stars in the third and fourth positions and a dot in all other positions.

List of taxa in bipartitions:

```

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

```

Key to taxon bipartitions (saved to file "primates.nex.parts"):

```

ID -- Partition
-----
1 -- .*****
2 -- .*.....
3 -- ..*.....
4 -- ...*.....

```

```

5 -- .....*.....
6 -- .....*.....
7 -- .....*.....
8 -- .....*....
9 -- .....*...
10 -- .....*..
11 -- .....*.
12 -- .....*
13 -- .....****.
14 -- ..**.....
15 -- ..*****
16 -- .....**...
17 -- ..*****.
18 -- ..*****.
19 -- ..****.
20 -- ..***.
21 -- .....***.
-----

```

Then it gives a table over the informative bipartitions (the ones with more than one taxon included), specifying the number of times the partition was sampled (#obs), the probability of the partition (Probab.), the standard deviation of the partition frequency (Sd(s)) across runs, the min and max of the standard deviation across runs (Min(s) and Max(s)) and finally the number of runs in which the partition was encountered. In our analysis, there is overwhelming support for a single tree, so all partitions in this tree have a posterior probability of 1.0.

Summary statistics for informative taxon bipartitions
(saved to file "primates.nex.tstat"):

ID	#obs	Probab.	Sd(s)+	Min(s)	Max(s)	Nruns
13	302	1.000000	0.000000	1.000000	1.000000	2
14	302	1.000000	0.000000	1.000000	1.000000	2
15	302	1.000000	0.000000	1.000000	1.000000	2
16	302	1.000000	0.000000	1.000000	1.000000	2
17	302	1.000000	0.000000	1.000000	1.000000	2
18	302	1.000000	0.000000	1.000000	1.000000	2
19	302	1.000000	0.000000	1.000000	1.000000	2
20	302	1.000000	0.000000	1.000000	1.000000	2
21	302	1.000000	0.000000	1.000000	1.000000	2

We then get a table summarizing branch and node parameters, in our case the branch lengths. The indices in this table refer to the key to partitions. For instance, `length[14]` is the length of the branch corresponding to partition ID 14. As we noted above, this is the branch grouping humans and chimps. The meaning of most of the values in this table is obvious. The last two columns give a convergence diagnostic, the Potential Scale Reduction Factor (PSRF), and the number of runs in which the partition was encountered. The PSRF diagnostic is the same used for the regular parameter samples, and it should approach 1.0 as runs converge.

Summary statistics for branch and node parameters
(saved to file "primates.nex.vstat"):

Parameter	Mean	Variance	95% HPD Interval		Median	PSRF+	Nruns
			Lower	Upper			
length[1]	0.486115	0.007159	0.349117	0.660632	0.477699	0.997	2
length[2]	0.335038	0.003829	0.222216	0.448005	0.331021	1.008	2
length[3]	0.050689	0.000114	0.033718	0.072349	0.049417	1.001	2
length[4]	0.060501	0.000144	0.039651	0.082690	0.060417	0.997	2
length[5]	0.057754	0.000183	0.031723	0.081064	0.056059	1.001	2
length[6]	0.143419	0.000537	0.100539	0.189667	0.140537	1.000	2
length[7]	0.172066	0.001072	0.112808	0.233264	0.172907	1.007	2
length[8]	0.016107	0.000031	0.005941	0.026377	0.015679	1.001	2
length[9]	0.023164	0.000045	0.011955	0.037226	0.022580	0.999	2
length[10]	0.056704	0.000147	0.033104	0.079370	0.056479	0.999	2
length[11]	0.069330	0.000366	0.029295	0.103081	0.070148	1.012	2
length[12]	0.433951	0.005270	0.305526	0.572054	0.426316	0.998	2
length[13]	0.248133	0.002680	0.148162	0.338245	0.243948	0.997	2
length[14]	0.029261	0.000142	0.008043	0.052766	0.028329	0.997	2
length[15]	0.273555	0.003600	0.163707	0.403779	0.268913	1.011	2
length[16]	0.035972	0.000125	0.016521	0.059122	0.035263	0.998	2
length[17]	0.118515	0.001761	0.044026	0.199364	0.119746	0.998	2
length[18]	0.124953	0.001162	0.052839	0.178939	0.122538	0.997	2
length[19]	0.057618	0.000424	0.017331	0.091541	0.057151	1.000	2
length[20]	0.082425	0.000486	0.051259	0.137057	0.080014	0.997	2
length[21]	0.049766	0.000398	0.018269	0.094330	0.047661	0.997	2

This table is followed by two trees. The clade credibility tree (upper tree) gives the probability of each partition or clade in the tree, and the phylogram (lower tree) gives the branch lengths measured in expected substitutions per site:

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, e.g., 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`.

2.1 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.

2.2 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 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 like we did above. The final step is to tell MrBayes that we want to work with this partitioning of the data instead of the default partitioning. We do this using the `set` command:

```
set partition = favored;
```

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

```
#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. Note that this command causes MrBayes to read in the data file because it contains the command `execute cynmix.nex`.

2.3 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 + Γ + 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 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 + Γ + 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
Brlens	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 cal-

culate 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, which is `variable` for standard (morphology) data. Now, `showmodel` should produce this table:

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

2.4 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. It is important to remember, though, that MrBayes uses the name of the file containing the character matrix as the default for all output files. Thus, if you run all your analyses in the same directory, results from different analyses will overwrite each other.

To change this behavior, include the command **mcmc filename=<filename>;** in each of your run files, just before issuing the **mcmc** command, using a different file name for each run file. For instance, if you wish to name the output files from one analysis using the root **analysis1**, you use the line **mcmc filename=analysis1;**. The files will then be named **analysis1.run1.t**, **analysis1.run2.t**, etc. An alternative approach is to run each analysis in a separate directory, in which case the naming of the output files will not be an issue.

2.5 Some Practical Advice

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

1. The convergence diagnostics provided by MrBayes are quite powerful but they certainly do not exhaust the possibilities. Several programs will read MrBayes output files and will provide you with a number of additional ways in which you can examine the output from your analysis. Two of the most popular tools are Tracer and AWTY. Among other things, they provide nice graphical representations of the output from MCMC analyses.

2. 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.

3. If you are using heated chains, try to make sure that the acceptance rates of swaps between adjacent chains are in the approximate range of 10 to 70 %. The acceptance rates are printed to the **.mcmc** file and to screen at the end of the run. The latter output contains a table that looks like this (you find the critical values in a different format in the **.mcmc** file):

1	2	3	4

1		0.53	0.26	0.12
2	3347		0.55	0.31
3	3295	3337		0.53
4	3332	3396	3293	

It is the values just above the diagonal, $\{0.53, 0.55, 0.53\}$ in this case, that you should focus on. 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. Acceptance rates outside the optimal range do not invalidate the results from your analysis, they only mean that you could make your analysis more efficient.

4. 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 web site for more information about this.

5. 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, or change the frequency with which they are used, 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 web site (<http://mrbayes.net>), Fredrik's MrBayes resources page (<http://www.sc.fsu.edu/~fronquis/mrbayes>), the MrBayes home on SourceForge (<http://www.sf.net/projects/mrbayes>) and the MrBayes users email list.