

MrBayes 3: A Tutorial

Fredrik Ronquist
School of Computational Science
Florida State University
Tallahassee, FL 32306-4120, U.S.A.
ronquist@csit.fsu.edu

John P. Huelsenbeck
Division of Biological Sciences
University of California at San Diego
La Jolla, CA 92093, USA
johnh@biomail.ucsd.edu

1. Introduction

MrBayes 3 is a program for the Bayesian inference of phylogeny. The program has a command-line interface and should run on a variety of computer platforms, including clusters of Macintosh or 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.

This tutorial explains briefly how to use the program. We will walk you through a simple analysis, which will get you started, and a more complex analysis that uses more of the program's capabilities. Finally, we will give some instructions on how to run the parallel versions of the program. For more detailed information about commands and options, see the command reference that can either be downloaded from the program web site or generated from the program itself (see the section *Getting Help* below). All the information in the command reference is also available on-line when using the program.

1.1. Conventions used in this manual

What you see on the screen and what is in the input file is in `plain typewriter font`. What you type is in **bold typewriter font**.

1.2. Acquiring and installing MrBayes

MrBayes 3 is distributed without charge by download from <http://mrbayes.net>. If someone has given you a copy of MrBayes 3, we strongly suggest that you download the current version from the above site. Downloading the program requires that you give your name and email address; these addresses will help us keep track of the number of users of the different versions of the program and they will be used to announce future releases of the program and to send information about major bugs.

MrBayes 3 is a plain-vanilla program that uses a command line interface and therefore behaves the same on all platforms - Macintosh, Windows and Unix. There is a separate download package for each platform. The Macintosh package contains two versions of the program: the standard serial version, named MrBayes3 (program icon one copy of Reverend Bayes's portrait), and a version for running the program in parallel on clusters of Macintoshes, named MrBayes3p (program icon four portraits of Reverend Bayes). For more information on the parallel Macintosh version of MrBayes, which requires the installation of POOCH, see the fourth section of this user manual. The Windows package only contains the serial version of the program and is ready to run after unzipping, just like the Macintosh serial version.

If you decide to run the program under UNIX/LINUX, then you need to compile the program first. Simply untar the file `mrBayesSrc.tar` by typing `tar -xvf mrBayesSrc.tar`. This command extracts all of the files from the `.tar` archive. You then need to compile the program. We have included a Makefile that will automatically compile the program. You simply type `make` to compile the program. We assume as the default C compiler `gcc`. However, if you do not have `gcc` installed on your machine, you may need to change the compiler information in the Makefile. The executable is called "mb". To execute the program type `mb` or `./mb`. The procedure described above produces the serial version of the program. How to compile the parallel version of the program on UNIX clusters is described in the third section of this manual.

1.3. Getting Started

Start MrBayes by double-clicking the application icon (or typing `mb`) and you will see the information below:

```
MrBayes v3.0B5
(Bayesian Analysis of Phylogeny)
by
John P. Huelsenbeck and Fredrik Ronquist
Section of Ecology, Behavior and Evolution
Division of Biological Sciences
University of California, San Diego
johnh@biomail.ucsd.edu
School of Computational Science
and Information Technology
Florida State University
ronquist@csit.fsu.edu
Type "help" or "help <command>" for information
on the commands that are available.
```

MrBayes >

1.4. Getting Help

At the `MrBayes >` prompt, type **help** to see a list of the commands available in MrBayes. Most commands allow you to set values (options) for a range of 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. The help facility also provides a way to see the current settings. For instance, typing **help lset** results in a list of the options and, at the end, a table giving the current settings.

A complete list of commands and options is given in the command reference, which can be downloaded from the program web site. You can also produce an ASCII text version of the command reference by giving the command **manual** to MrBayes. 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. If the abbreviation is ambiguous, MrBayes will not execute the command, so there is no risk involved in trying to abbreviate commands and options when working interactively with the program.

2. 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 a phylogenetic model.

2.1. 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 data, or any mix of these four data types. The Nexus file that we will use for this tutorial, `primates.nex`, contains 12 mitochondrial DNA sequences of primates.

A Nexus file is a simple text (ASCII) file that begins with `#NEXUS` on the first line. The rest of the file is divided into different blocks. The `primates.nex` file looks like this:

```
#NEXUS
begin data;
  dimensions ntax=12 nchar=898;
  format datatype=dna interleave=no gap=-;
  matrix
Saimiri_sciureus AAGCTTCATAGGAGC ... ACTATCCCTAAGCTT
Tarsius_syrichta AAGCTTCACCGGCGC ... ATTATGCCTAAGCTT
Lemur_catta      AAGCTTCACCGGCGC ... ACTATCTATTAGCTT
Macaca_fuscata   AAGCTTCACCGGCGC ... CCTAACGCTAAGCTT
M_mulatta        AAGCTTCACCGGCGC ... CCTAACACTAAGCTT
M_fascicularis  AAGCTTTACAGGTGC ... CCTAACACTAAGCTT
M_sylvanus       AAGCTTTTCCGGCGC ... CCTAACATTAAGCTT
Homo_sapiens     AAGCTTTTCTGGCGC ... GCTCTCCCTAAGCTT
Pan              AAGCTTCTCCGGCGC ... GCTCTCCCTAAGCTT
Gorilla          AAGCTTCTCCGGTGC ... ACTCTCCCTAAGCTT
Pongo           AAGCTTCACCGGCGC ... ACTTCTACTAAGCTT
Hyllobates       AAGTTTCATTGGAGC ... ACTCTCCCTAAGCTT
  ;
end;
```

The file contains only one block, a DATA block. The DATA block is initialized with `begin data;` followed by the `dimensions` statement, the `format` statement, and the `matrix` statement. The `dimensions` statement must contain `ntax`, the number of taxa, and `nchar`, the number of characters in each aligned sequence. The `format` statement must specify the type of data, for instance `datatype=DNA` (or RNA or Protein or Standard or Restriction). The `format` statement may also contain `gap=-` (or whatever symbol is used for a gap in your alignment), `missing=?` (or whatever symbol is used for missing data in your file), `interleave=yes` when the data matrix is interleaved sequences, and `match=.` (or whatever symbol is used for matching characters in the alignment). The `format` statement is followed by the `matrix` statement, usually started by the word `matrix` on a separate line, followed by the aligned sequences. Each sequence begins with the sequence name separated from the sequence itself by at least one space. The data block is completed by an `end;` statement. Note that the `begin`, `dimensions`, `format`, and `end` statements all end in a semicolon. That semicolon is essential and must not be left out. Note that, although it occupies many lines in the file, the matrix description is also a regular statement, a `matrix` statement, which ends with a semicolon just as any other statement. Our example file contains sequences in non-interleaved (block) format. Non-interleaved is the default but you can put `interleave=no` in the `format` statement if you want to be sure.

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!: The input file must be located in the same folder (directory) as the MrBayes application 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.

2.2. Specifying a Model

All of the commands can be entered at the command line 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. New users are also recommended to check the model settings prior to 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 + Γ model for the evolution of the mitochondrial sequences and we will check all of the relevant priors.

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
Rates	Equal/Gamma/Propinv/Invgamma/Adgamma	Equal
Ngammacat	<number>	4
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 partition data will be explained in the next section of the manual, *Analyzing a Partitioned Data Set*.

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). 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 Gamma. Do this by typing **lset rates=gamma**. 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=gamma** 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 difficult or impossible to detect in most cases.

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	Gamma
Ngammacat	<number>	4
Nbetacat	<number>	5
Omegavar	Equal/Ny98/M3	Equal
Covarion	No/Yes	No
Coding	All/Variable/Noabsencesites/ Nopresencesites	All
Parsmodel	No/Yes	No

We now need to set the priors for our model. There are five types of parameters in the model: the topology, the branch lengths, the four stationary frequencies of the nucleotides, the six different nucleotide substitution rates, and the shape parameter of the gamma distribution of rate variation. It is a good idea to type **help prset** to obtain a list of the default settings for these parameter types. The table at the end of the help information reads:

Model settings for partition 1:

Parameter	Options	Current Setting
Tratioopr	Beta/Fixed	Beta(1.0,1.0)
Revmatpr	Dirichlet/Fixed	Dirichlet(1.0,1.0,1.0,...
Aamodelpr	Fixed/Mixed	Fixed(Poisson)
Omegapr	Dirichlet/Fixed	Dirichlet(1.0,1.0)
Ny98omegalpr	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
Ratepr	Fixed/Variable=Dirichlet	Fixed
Shapepr	Uniform/Exponential/Fixed	Uniform(0.1,50.0)
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)
Symmetricbetapr	Uniform/Exponential/Fixed	Fixed(Infinity)
Topologypr	Uniform/Constraints	Uniform
Brlenspr	Unconstrained/Clock	Unconstrained:Exp(10.0)
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)
Growthpr	Uniform/Exponential/	

Fixed/Normal

Fixed(0.0)

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), `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)`**. Note that 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).

In our analysis, we will 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 **`prset st = Dir(1,1,1,1)`** if you want to save some typing.

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.

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 `Brlenpr` 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 the distribution is still sufficiently vague that it will have little influence on the posterior unless the data are very weak.

In summary, we will not change the default priors. To check the model before we start the analysis, type **`showmodel`**. This will produce the following output:

Model settings:

```

Datatype = DNA
Nucmodel = 4by4
Nst = 6
      Substitution rates, expressed as proportions
      of the rate sum, follow a Dirichlet
      (1.00,1.00,1.00,1.00,1.00,1.00)
Covarion = No
# States = 4
      State frequencies have a Dirichlet prior
Rates = Gamma
      Gamma shape parameter is uniformly dist-
      ributed on the interval (0.05,50.00).
      Gamma distribution is approximated using 4 categories.

```

Active parameters:

```

Parameters
-----
Revmat          1
Statefreq       2
Shape           3
Topology        4
Brlens          5
-----

1 -- Parameter = Revmat
   Prior       = Dirichlet(1.00,1.00,1.00,1.00,1.00,1.00)
2 -- Parameter = Statefreq
   Prior       = Dirichlet
3 -- Parameter = Shape
   Prior       = Uniform(0.05,50.00)
4 -- Parameter = Topology
   Prior       = All topologies equally probable a priori
5 -- Parameter = Brlens
   Prior       = Branch lengths are Unconstrained:Exponential(10.0)

```

This gives a nice overview of the model settings.

2.3. Running 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`. One of the reasons for this is that you cannot stop a MrBayes run once it is started without abandoning the session completely (in the worst case, use `ctrl-Z` on UNIX systems or `ctrl-c` on Windows machines to accomplish this). The `help mcmc` command will produce the following table at the bottom of the output:

Parameter	Options	Current Setting
Seed	<number>	1091393476
Ngen	<number>	1000000
Samplefreq	<number>	100
Swapfreq	<number>	1
Printfreq	<number>	100
Nchains	<number>	4
Temp	<number>	0.200000

Reweight	<number> , <number>	0.00 v 0.00 ^
Filename	<name>	primates.nex.<p/t>
Burnin	<number>	0
Startingtree	Random/User	Random
Nperts	<number>	0
Savebrlens	Yes/No	Yes

The `Seed` is simply the seed for the random number generator. 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 that 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 only 1,000 generations. To change the `Ngen` setting, we *cannot* use the `mcmc` command because this will start the analysis. Instead we use the `mcmcpr` command, which is equivalent to `mcmc` except that it does not start the analysis. Type **`mcmcpr ngen=1000`** to set the number of generations to 1,000.

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. 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, the file will be called `primates.nex.p`), which is a tab delimited text file that can be imported into most statistics and graphing programs. The topology and branch lengths are printed to a `.t` file (in our case, the file will be called `primates.nex.t`), which is a Nexus tree file that can be imported into programs like PAUP*. The root of the `.p` and `.t` file names can be altered using the `Filename` setting.

By default, MrBayes uses Metropolis coupling to improve the MCMC sampling of the target distribution. The `Swapfreq`, `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 problems with more than about 50 taxa, whereas smaller problems often can be analyzed successfully without heating. It is still an open question whether adding more than three heated chains is helpful in analyzing large and difficult data sets. The time complexity of the analysis is directly proportional to the number of chains used.

MrBayes uses an incremental heating scheme, in which chain i is heated by raising its posterior probability by the power $1 / (1 + i\lambda)$, where λ is the temperature controlled by the `Temp` parameter. The effect of the heating is to flatten out the posterior probability, such that the heated chains more easily find isolated peaks in the posterior distribution and can help the cold chain move more rapidly between these peaks. 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.

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

The default behavior of MrBayes is to save trees without branch lengths to the `.t` file. We want to save the branch lengths so we type `mcmc savebrlens=yes`. If the branch lengths are not saved to file during the run, they cannot be retrieved after the analysis has been completed.

The `Startingtree` parameter can be used to feed the chain(s) with a user-specified starting tree. The default behavior is to start each chain with a different, random tree.

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 are used in sampling from the posterior distribution. In our case, the proposals are the following:

```
The chain will use the following moves:
  With prob. Chain will change
    3.03 \% param. 1 (revmat) with Dirichlet proposal
    3.03 \% param. 2 (state frequencies) with Dirichlet proposal
    3.03 \% param. 3 (gamma shape) with multiplier
   30.30 \% param. 4 (topology and branch lengths) with LOCAL
   30.30 \% param. 4 (topology and branch lengths) with extending TBR
   30.30 \% param. 5 (branch lengths) with nodeslider
```

Note that MrBayes will spend most of its effort changing topology and branch lengths. 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 these parameters. The proposal probabilities can be changed with the `props` command but be warned that inappropriate changes of proposal probabilities 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 -- [-7763.732] (-7691.348) (-7639.743) (-7524.970)
 100 -- (-7019.448) [-6444.645] (-6860.922) (-6847.998) -- 0:00:09
 200 -- (-6623.351) [-6357.934] (-6611.587) (-6691.889) -- 0:00:04
 300 -- (-6238.951) (-6263.390) [-6208.577] (-6293.287) -- 0:00:02
 400 -- (-6103.836) (-6180.572) [-6065.113] (-6171.450) -- 0:00:03
 500 -- (-6051.326) (-6054.049) [-5992.884] (-6103.330) -- 0:00:02
 600 -- (-6003.443) (-6009.621) [-5974.573] (-6080.474) -- 0:00:02
 700 -- (-5981.916) (-5998.243) [-5939.707] (-6013.063) -- 0:00:01
 800 -- (-5946.161) (-5934.670) [-5933.896] (-5993.630) -- 0:00:00
 900 -- [-5885.150] (-5920.244) (-5929.708) (-5948.639) -- 0:00:00
1000 -- [-5887.430] (-5917.988) (-5919.648) (-5934.146) -- 0:00:00
```

Continue with chain? (yes/no):

The first column lists the generation number. The four columns with negative numbers each correspond to one chain (or rather, one physical location in computer memory). 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. When two chains successfully change states, they trade places in computer memory because this is by far the most computationally efficient implementation. This results in their trading column positions. In a healthy run, the cold chain should move around among the columns because 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. The analysis may then have to be run longer or the temperature difference between chains may have to be lowered.

The last column gives the time left to completion of the specified number of generations. This analysis approximately takes 1 second per 100 generations. At the end of the run, MrBayes asks whether you want to continue with the chain.

2.4. When to Stop the Analysis

Perhaps the most difficult aspect of a Bayesian MCMC analysis is to decide how long to run it. An initial guess may be obtained by observing the log likelihood (or, more exactly, the log probability of the data given the parameter values) of the cold chain. In the beginning of the run, the log likelihood of the cold chain typically increases rapidly. This phase of the run is referred to as the burn-in and the samples from this phase are 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. At stationarity, we typically also observe that the cold and heated chains are exchanging states frequently, resulting in the cold chain moving around freely among the columns.

Try extending the chain to 10,000 generations by first answering the above question with **yes** and then requesting **9000** additional generations.

```

1100 -- (-5894.562) (-5834.792) [-5810.720] (-5890.157) -- 0:00:32
1200 -- (-5890.664) (-5833.120) [-5786.851] (-5872.152) -- 0:00:36
1300 -- (-5877.270) (-5824.168) [-5786.244] (-5873.336) -- 0:00:33
1400 -- (-5863.694) (-5816.845) [-5775.970] (-5888.859) -- 0:00:36
1500 -- (-5810.522) (-5799.613) [-5772.893] (-5847.262) -- 0:00:34
1600 -- [-5784.771] (-5798.489) (-5773.695) (-5851.832) -- 0:00:31
1700 -- (-5781.192) (-5794.808) [-5771.801] (-5833.122) -- 0:00:34
1800 -- (-5782.544) (-5779.421) [-5765.648] (-5817.788) -- 0:00:31
1900 -- (-5754.121) (-5771.894) [-5754.554] (-5822.383) -- 0:00:29
2000 -- [-5743.643] (-5769.566) (-5755.029) (-5805.316) -- 0:00:32
2100 -- [-5745.199] (-5753.541) (-5753.955) (-5795.461) -- 0:00:30
2200 -- [-5737.376] (-5751.863) (-5756.222) (-5795.651) -- 0:00:31
2300 -- [-5738.891] (-5760.954) (-5753.764) (-5809.397) -- 0:00:30
2400 -- [-5728.769] (-5766.453) (-5748.446) (-5801.951) -- 0:00:28
2500 -- [-5726.714] (-5768.640) (-5744.112) (-5785.119) -- 0:00:30
2600 -- (-5731.979) [-5748.859] (-5743.775) (-5774.670) -- 0:00:28
2700 -- (-5729.670) (-5736.618) [-5738.046] (-5768.434) -- 0:00:29
2800 -- (-5725.783) [-5737.256] (-5737.002) (-5776.507) -- 0:00:28
2900 -- [-5729.521] (-5734.973) (-5735.556) (-5763.342) -- 0:00:26
3000 -- [-5730.476] (-5742.189) (-5735.242) (-5772.966) -- 0:00:28
3100 -- [-5730.839] (-5740.678) (-5741.042) (-5767.070) -- 0:00:26
3200 -- [-5727.365] (-5741.383) (-5733.023) (-5767.109) -- 0:00:27

```

```

3300 -- (-5734.934) (-5740.855) [-5724.519] (-5760.213) -- 0:00:26
3400 -- (-5742.378) (-5730.592) [-5729.447] (-5749.677) -- 0:00:25
3500 -- (-5746.513) [-5721.187] (-5729.353) (-5749.642) -- 0:00:26
3600 -- (-5735.164) [-5721.605] (-5730.368) (-5744.788) -- 0:00:24
3700 -- (-5730.734) [-5724.544] (-5733.356) (-5740.770) -- 0:00:23
3800 -- (-5727.313) [-5727.291] (-5726.547) (-5736.770) -- 0:00:24
3900 -- (-5722.848) [-5725.822] (-5729.850) (-5736.852) -- 0:00:23
4000 -- [-5725.284] (-5730.743) (-5730.579) (-5740.580) -- 0:00:24
...
9000 -- (-5732.493) (-5732.830) (-5727.788) [-5722.522] -- 0:00:03
9100 -- (-5731.995) [-5724.999] (-5735.144) (-5728.063) -- 0:00:03
9200 -- (-5727.035) [-5723.285] (-5729.986) (-5735.920) -- 0:00:03
9300 -- (-5726.853) (-5720.256) (-5727.794) [-5724.041] -- 0:00:02
9400 -- [-5724.084] (-5724.416) (-5733.757) (-5731.239) -- 0:00:02
9500 -- [-5725.277] (-5728.837) (-5737.014) (-5743.613) -- 0:00:01
9600 -- [-5722.750] (-5729.779) (-5730.720) (-5730.701) -- 0:00:01
9700 -- [-5721.116] (-5724.920) (-5725.364) (-5730.869) -- 0:00:01
9800 -- [-5725.791] (-5729.472) (-5728.671) (-5728.843) -- 0:00:00
9900 -- (-5727.117) (-5730.515) [-5727.425] (-5732.889) -- 0:00:00
10000 -- (-5727.182) [-5728.934] (-5725.323) (-5738.342) -- 0:00:00

```

It looks like the likelihood values reach a plateau at around 4,000 generations and then stay there. To be on the safe side, we will decide to use an initial burn-in of 5,000 generations but we will not extend the chain beyond 10,000 generations. Answer **no** to the question of whether the chain should be extended. MrBayes will respond by printing various run statistics. Pay particular attention to the table of acceptance rates:

```

Acceptance rates for the moves in the "cold" chain:
With prob. Chain accepted changes to
30.74 \% param. 1 (revmat) with Dirichlet proposal
10.98 \% param. 2 (state frequencies) with Dirichlet proposal
33.33 \% param. 3 (gamma shape) with multiplier
16.24 \% param. 4 (topology and branch lengths) with LOCAL
16.28 \% param. 4 (topology and branch lengths) with extending TBR
25.11 \% param. 5 (branch lengths) with nodeslider

```

As a rough rule of thumb, an efficient Metropolis-Hastings MCMC sampler will have acceptance rates somewhere in the range 10 % to 70 %. If the acceptance rate is too high, then the proposal mechanism is making too modest suggestions of new states. If the rate is too low, on the other hand, then the proposals are too bold. Both situations are likely to lead to slow mixing, which means that you will have to run the chain longer to obtain convergence. The acceptance rates can be changed by modifying the tuning parameters using the `props` command. The topology proposals are difficult, however, and one often has to be satisfied with relatively low acceptance rates for these. In our analysis, the acceptance rates are OK for all proposals.

Finally, let's examine the state exchange information:

```

State exchange information:
          1      2      3      4
-----
1 |          0.48  0.28  0.14
2 | 1644          0.58  0.32
3 | 1716 1680          0.56
4 | 1589 1680 1691

```

The bottom row of the upper diagonal contains the acceptance rates for the swaps between chains separated by only one heating step. Again, a rough rule of thumb is that these acceptance rates should lie in the range 10 % to 70 %. If the acceptance rates are too low, you can try to lower the temperature difference; if the rates are too high, the temperature should be increased instead. In our case, the acceptance rates are all within the suggested range.

2.5. Summarizing Samples of Model Parameters

During the run, samples of the substitution model parameters have been written to the `.p` file. It looks something like this:

```
[ID: 5848203808]\\
  Gen  LnL      TL      r(G<->T)  ...  pi(G)      pi(T)      alpha\\
    1  -7433.991  2.098  0.166667  ...  0.250000  0.250000  0.500000\\
   100 -6601.275  1.939  0.143314  ...  0.226052  0.282844  0.732008\\
   ...
  9900 -5727.425  2.709  0.026515  ...  0.079022  0.249481  0.382468\\
 10000 -5728.934  3.177  0.015705  ...  0.084921  0.243379  0.387159\\
```

From left to right, the columns contain: (1) the generation number; (2) the log likelihood of the cold chain; (3) the total tree length (the sum of all branch lengths); (4) the six GTR rate parameters; (5) the four stationary nucleotide frequencies; and (6) the shape parameter of the gamma distribution of rate variation.

To summarize the information in the `.p` file, type `sump burnin=50`. By default, `sump` will summarize the information in the `.p` file generated most recently, but the filename can be changed if necessary. Beware that the burn-in value is given as the number of samples to be discarded as the burn-in, not as the number of generations to be discarded. Since we have been sampling every 100th generation, a burn-in of 5,000 generations is equivalent to a burn-in of 50 samples (to be exact, it is equivalent to 51 samples since the first generation is always sampled). The `sump` command will generate a plot of the generation versus the log probability of the data. If we are at stationarity, this plot should look like ‘white noise’, that is, there should be no tendency of steady increase or decrease. At the bottom of the output, there is a table summarizing the samples of the parameter values:

Parameter	Mean	Variance	95% Cred. Interval		Median
			Lower	Upper	
TL	2.877627	0.035346	2.606000	3.283000	2.849000
r(G<->T)	0.014772	0.000110	0.001623	0.037111	0.012585
r(C<->T)	0.400122	0.000659	0.341913	0.444436	0.406398
r(C<->G)	0.032839	0.000111	0.009279	0.058555	0.032048
r(A<->T)	0.039095	0.000059	0.027768	0.055970	0.038032
r(A<->G)	0.465614	0.000896	0.412362	0.532919	0.463180
r(A<->C)	0.047557	0.000051	0.037260	0.063988	0.047255
pi(A)	0.347618	0.000174	0.325877	0.376409	0.347966
pi(C)	0.323955	0.000134	0.297147	0.341814	0.324429

pi(G)	0.082582	0.000032	0.071130	0.093080	0.082649
pi(T)	0.245844	0.000121	0.226496	0.268155	0.243598
alpha	0.407082	0.000691	0.339477	0.448983	0.405698

The table should be self-explanatory. Note that the six rate parameters of the GTR model are given as proportions of the rate sum (the Dirichlet parameterization). This parameterization has many advantages in the Bayesian context. In particular, it allows convenient formulation of priors. If we multiply the rate proportions with the tree length, we get a Dirichlet summary of the posterior that could be used as the prior for a subsequent analysis. If you want to scale the rates relative to the G-T rate, just divide all rate proportions by the G-T rate proportion. You can also ask MrBayes to print the rates in this G-T scaled manner to the `.p` file by changing the output setting for the GTR rates in the `report` command prior to the run. However, internally in MrBayes the rates are always represented using the Dirichlet parameterization (from version 3 beta 4).

2.6. Summarizing Samples of Trees and Branch Lengths

Trees and branch lengths are printed to the `.t` file. This file looks something like this:

```
#NEXUS
[ID: 5848203808]
begin trees;
  translate
    1 Lemur_catta,
    2 Homo_sapiens,
    3 Pan,
    4 Gorilla,
    5 Pongo,
    6 Hylobates,
    7 Macaca_fuscata,
    8 M_mulatta,
    9 M_fascicularis,
    10 M_sylvanus,
    11 Saimiri_sciureus,
    12 Tarsius_syrichta;
  tree rep.1 =
  (((((2:0.100000,(8:0.100000,7:0.000748):0.197415):0.100000,(5:0.100000,6:0.100000):0.100000):0.100000,(9:0.100000,(10:0.100000,3:0.100000):0.100000):0.100000):0.100000,12:0.100000):0.100000,11:0.100000):0.100000,4:0.100000,1:0.100000);
  ...
  tree rep.10000 =
  (((6:0.126475,((4:0.079300,(3:0.049876,2:0.066929):0.040517):0.103191,5:0.138220):0.105224):0.147212,(10:0.059671,((8:0.035939,7:0.012332):0.037992,9:0.069477):0.060859):0.221336):0.149243,11:0.627561):0.180115,12:0.525606,1:0.339476);
end;
```

Note that branch lengths are printed to file only if this is requested when starting the analysis (in the `mcmc` or `mcmcpr` command).

To summarize the tree and branch length information, type `sumt burnin=50`. The `sumt` and `sump` commands each have separate burn-in settings so it is necessary to give the burn-in here again. Otherwise, many of the settings in MrBayes are persistent and

need not be repeated every time a command is executed. To make sure the settings for a particular command are correct, you can always use `help <command>`.

The `sumt` command will output, among other things, a tree with clade credibility (posterior probability) values and a phylogram (if branch lengths have been saved). In the background, the command will create 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. 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 Rod Page's program TreeView. The second tree only contains the branch lengths and it can be imported into a wide range of phylogenetics programs. 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.

2.7. Assessing Convergence

Before we leave this example, we need to revisit the question of whether or not our sample is likely to be representative of the posterior probability distribution; that is, if we have run the analysis long enough. Unfortunately, there is no bullet-proof way of determining this. Inspecting the chain probabilities and how they change over time, as we did above, is only a rough guide that can often be misleading. We can do better by examining the sampling behavior for all model parameters, but there is still a risk that a single chain that appears well-behaved is actually not sampling from all relevant regions of parameter space. We therefore recommend using three to four independent runs, each started from different, randomly chosen trees. If all of these run give similar estimates of substitution model parameters, topology and branch lengths, this suggests that each of the runs produces a reasonable sample from the posterior probability distribution.

To repeat our analysis without overwriting previous results, type `mcmc filename = <filename>`, where `<filename>` should not coincide with an existing file in the working directory. The model settings, the number of generations, that we want to save branch lengths and start each run from new, randomly chosen starting trees: all these settings are persistent and need not be specified again. After repeating the analysis three times, run `sump` and `sumt` on each of the resulting files and compare the results.

2.8. Running in Batch Mode

When you become more familiar with MrBayes, you will undoubtedly want to run it in batch mode instead of typing all commands at the prompt. This is done by adding a MRBAYES block to a Nexus file, either the same file containing the DATA block or a separate Nexus file. For instance, a MRBAYES block that performs three analyses of the kind outlined above would be specified as follows if entered after the appropriate DATA block:

```
begin mrbayes;
  set autoclose=yes nowarn=yes;
  lset nst=6 rates=gamma;
  mcmc ngen=10000 savebrlens=yes file=primates.nex1;
  mcmc file=primates.nex2;
  mcmc file=primates.nex3;
end;
```

You start the analysis simply by typing **execute <filename>**, where filename is the name of the file containing the DATA and MRBAYES blocks. The `set` command is needed to change the behavior of MrBayes such that it is appropriate for batch mode. When `autoclose = yes`, MrBayes will finish the MCMC analysis without asking you whether you want to add more generations. When `nowarn = yes`, MrBayes will overwrite existing files without warning you, so make sure that your batch file does not inadvertently cause the deletion of previous result files that should be saved for future reference.

The UNIX version of MrBayes can execute batch files in the background from the command prompt. Just type **mb <file> > log.txt &** at the UNIX prompt, where <file> is the name of your Nexus batch file, to have MrBayes run in the background, logging its output to the file `log.txt`. Alternatively, the UNIX version of MrBayes can also be run in batch mode using input redirection. For that you need a text file containing the commands exactly as you would have typed them from the command line. For instance, assume that your data set is in `primates.nex` and that you want to perform the same analyses specified above. Then type **mb < batch.txt > log.txt &** with the `batch.txt` file containing this text:

```
set autoclose=yes nowarn=yes
execute primates.nex
lset nst=6 rates=gamma
mcmc ngen=10000 savebrlens=yes file=primates.nex1
mcmc file=primates.nex2
mcmc file=primates.nex3
quit
```

The `quit` command forces MrBayes to terminate.

3. 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. The data set is found in the file `cynmix.nex`.

3.1. Getting Mixed Data into MrBayes

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 of 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.

3.2. 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 the default 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. The following lines in a MRBAYES block (or entered from the command line without the trailing semicolon) will specify one character set for each of the four gene regions:

```
charset morphology = 1-166;
charset COI = 167-1244;
charset EF1a = 1245-1611;
charset LWRh = 1612-2092;
charset 28S = 2093-3246;
```

The next step is to define a partition of the data according to genes and morphology. This is accomplished with the line:

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

Finally, we need 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;
```

We now have a partitioned model in MrBayes, with the first partition being morphology, the second partition being COI, etc.

3.3. 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)
-----
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:

```

                Partition(s)
Parameters      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 statefreqs=(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 (see the help info for `prset` to modify this 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 coding bias. By default, MrBayes assumes that standard data sets include all variable 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:

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
Pinvar	.	14	15	16	17
Topology	18	18	18	18	18
Brlens	19	19	19	19	19

If this is the case, the model is complete and we can proceed with the analysis as described above. In this case, however, the analysis will have to be run longer than the previous one before adequate results are obtained.

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

4. Running MrBayes in Parallel

Metropolis coupling or heating is well suited for parallelization. MrBayes 3 takes advantage of this and uses MPI to distribute heated and cold chains among available processors. There are two MPI versions of MrBayes. The first is the parallel version for Macintosh computers distributed as part of the Macintosh package. It is intended for use on clusters of Macintosh computers and runs under POOCH, which must be installed first. For more instructions on how to install and run this version of MrBayes, see the program web site.

The second MPI version of MrBayes is intended for use on clusters running UNIX and must be compiled from the source code. To tell the compiler that you want the MPI version, you need to change the first line of the `mb.h` source file first. The line originally reads:

```
#undef MPI_ENABLED          /* define or undefined to switch MPI on or off */
```

Change the initial `#undef` to `#define` such that the line reads:

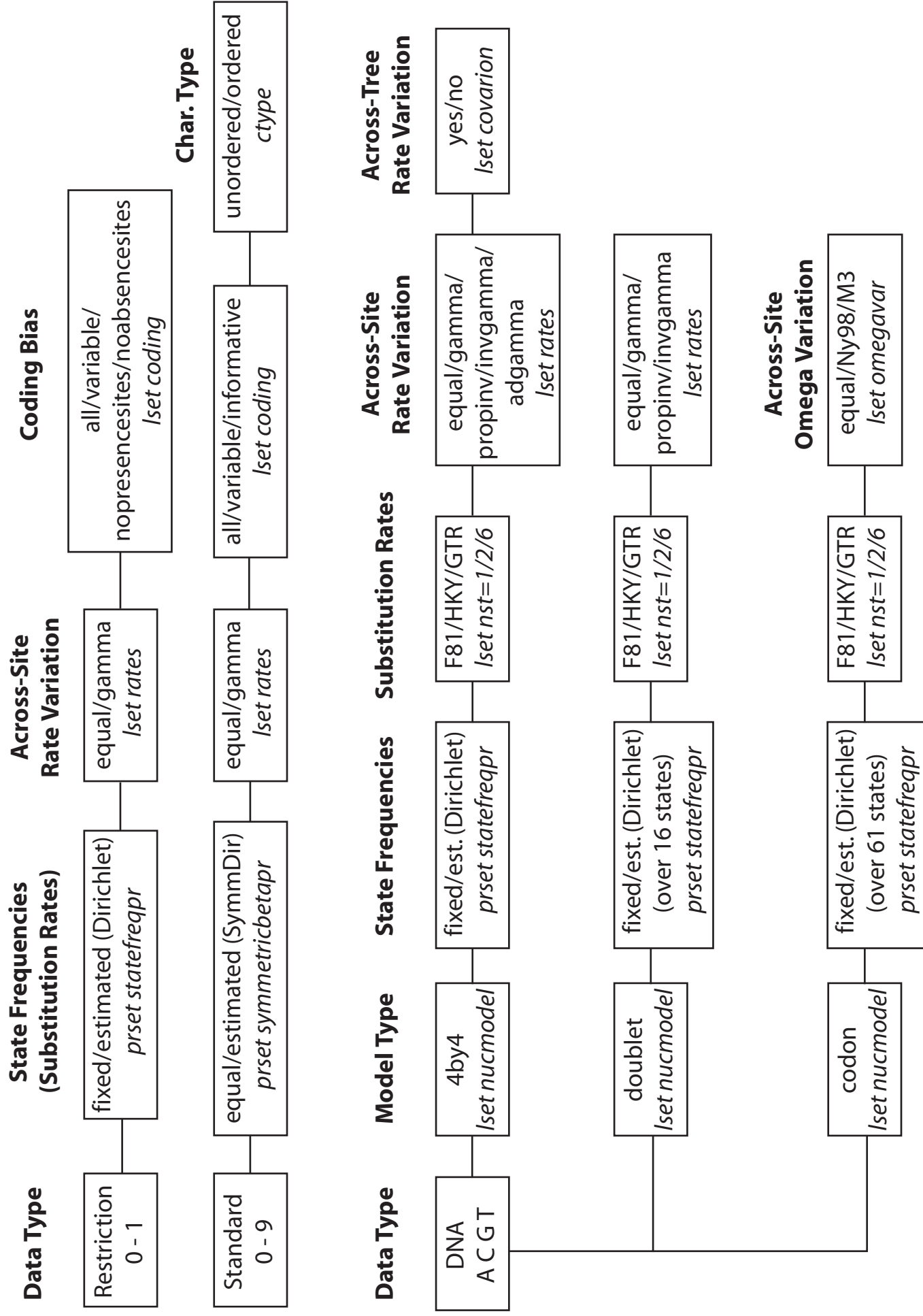
```
#define MPI_ENABLED          /* define or undefined to switch MPI on or off */
```

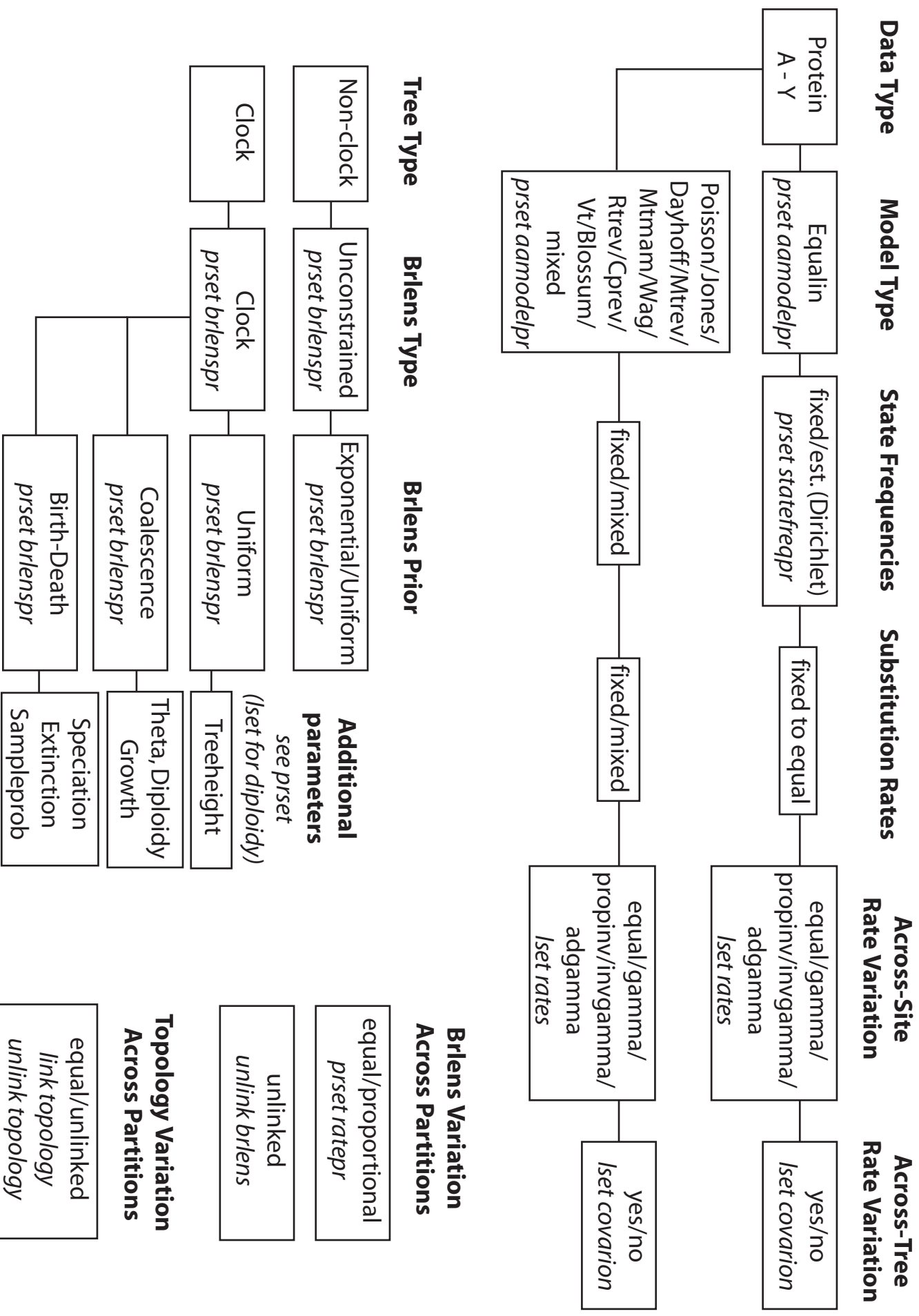
If your environment is set up correctly, among other things you need to have an appropriate `mpi.h` header file in your path, you should now be able to compile the MPI version of MrBayes. How you run it depends on the MPI implementation on your cluster; ask your system administrator if you need help.

5. Where to Go from Here

This brief tutorial covers only the basic capabilities of MrBayes 3. The command reference gives you more complete information about the commands and options. It can be obtained from the program web site but you can also produce a text-only copy of the command reference at any time by typing **manual** at the `MrBayes >` prompt. However, we want to emphasize that all of the information in the command reference is available on-line through the `help` command, and this provides the most convenient way of answering questions that arise when using the program.

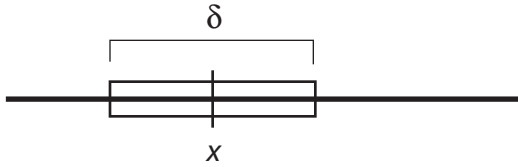
Appended to this manual you will find a graphical overview of the models supported by MrBayes 3, as well as graphical summaries of the most common types of proposals. The download packages also include sample data files that illustrate different types of analyses. These files contain explanatory comments and are well worth exploring as an additional way of learning more about MrBayes 3.





equal/unlinked
link topology
unlink topology

Sliding Window Proposal



New values are picked uniformly from a sliding window of size δ centered on x .

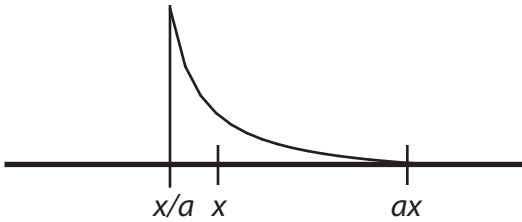
Tuning parameter: δ

Bolder proposals: increase δ

More modest proposals: decrease δ

Works best when the effect on the probability of the data is similar throughout the parameter range

Multiplier Proposal



New values are picked from the equivalent of a sliding window on the log-transformed x axis.

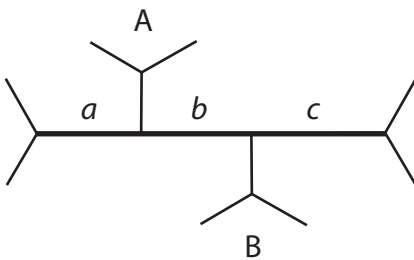
Tuning parameter: $\lambda = 2 \ln a$

Bolder proposals: increase λ

More modest proposals: decrease λ

Works well when changes in small values of x have a larger effect on the probability of data than changes in large values of x . Example: branch lengths.

LOCAL



Three internal branches - a , b , and c - are chosen at random. Their total length is changed using a multiplier with tuning parameter λ .

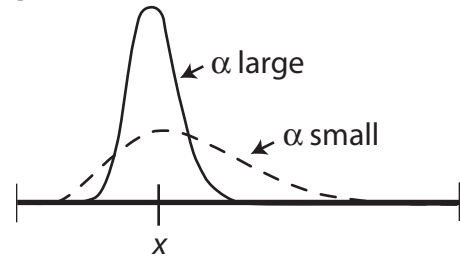
One of the subtrees A or B is picked at random.

It is randomly reinserted on $a + b + c$ according to a uniform distribution

Bolder proposals: increase λ

More modest proposals: decrease λ

Dirichlet proposal



New values are picked from a Dirichlet (or Beta) distribution centered on x .

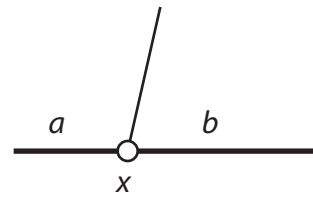
Tuning parameter: α

Bolder proposals: decrease α

More modest proposals: increase α

Works well for proportions, such as revmat and statefreqs.

Node Slider



Two adjacent branches a and b are chosen at random

The length of $a + b$ is changed using a multiplier with tuning parameter λ

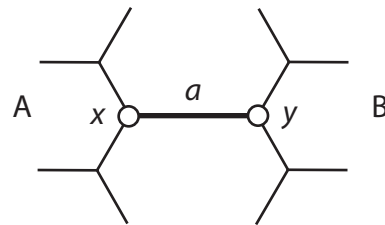
The node x is randomly inserted on $a + b$ according to a uniform distribution

Bolder proposals: increase λ

More modest proposals: decrease λ

The boldness of the proposal depends heavily on the uniform reinsertion of x , so changing λ may have limited effect

Extending TBR



An internal branch a is chosen at random

The length of a is changed using a multiplier with tuning parameter λ

The node x is moved, with one of the adjacent branches, in subtree A, one node at a time, each time the probability of moving one more branch is p (the extension probability).

The node y is moved similarly in subtree B.

Bolder proposals: increase p

More modest proposals: decrease p

Changing λ has little effect on the boldness of the proposal.