MrBayes Lab

Note: This computer lab exercise was written by Paul O. Lewis. Paul has graciously allowed us to use and modify the lab for the Summer Institute in Statistical Genetics. Thanks, Paul!

The main goal is to become familiar with running MrBayes (Ronquist and Huelsenbeck, 2003) and interpreting its output. Don't rush things. If you don't finish the lab, you can always download MrBayes when you get home and finish it there.

Stylistic conventions used in this document:

Commands understood by the PAUP* and/or MrBayes programs are in `this fixed width font`. Questions for you to answer are in *italics*. Important things to note are in **bold face**. Names of files are in this font. Web site URLs look `http://like_this` and are clickable links.

Exercise 1: Basics

1. You can download MrBayes 3.2.1 (Ronquist et al., 2012) for your platform from `http://mrbayes.net`. If you are using Windows, you have to install a .NET framework from Microsoft in order to run the MrBayes installer.

2. On Mac, you can then run MrBayes by launching the /Applications/Utilities/Terminal and then typing mb. It is easiest to copy the algaemb.nex file into your home directory.

   On Windows, it is easiest to put the algaemb.nex file into the C:\Program Files\mrbayes32 directory which contains the mrbayes executable. Then you can launch MrBayes by double-clicking on the icon.

   MrBayes by double-clicking the file name. When you see the "MrBayes >" prompt, type the following and then press the Enter key:

   ```
   execute algaemb.nex
   ```

   The algaemb.nex file is essentially the same as the one used for the PAUP* lab, and you should refer to that handout for a description of the data file. This version differs only in formatting: some features of the NEXUS file format that are understood by PAUP* are not handled cleanly by MrBayes. For example, the CHARACTERS block is not recognized by MrBayes, and the `symbols` and `labels` statements within the `format` command are not recognized.

   Assuming MrBayes reads the file without errors, set up an MCMC run using the HKY85 model with discrete gamma among site rate heterogeneity. There are three commands in particular that you will need to know something about in order to set up a typical run: `lset`, `prset` and `mcmc`. The command `mcmcp` is identical to `mcmc` except that it does not actually start a run.

   For each of these commands you can obtain online information by typing help followed by the command name: for example, `help prset`. This may spew more information than will fit on your screen, however. If this happens you can tell the Windows command prompt to remember more of the text that it displays:

   (a) click the icon at the left edge of the title bar of the MrBayes window,

   (b) choose "Properties",

   (c) click the "Layout" tab, and

(d) change “Screen Buffer Size, Height” to a large number (e.g. 10000).

After describing every parameter of a particular command, the `help` command, displays a table of the current values for the parameters.

3. First, establish the prior distributions that will be used for the model parameters. Here is a table of the parameters and the prior distributions that will be used:

Base frequencies	“flat” Dirichlet distribution
Shape parameter	Exponential distribution, mean 1.0
TRatio	“flat” Beta distribution – also known as Uniform[0,1.0]

The prior distribution over the tree topologies and branch lengths are:

Topologies	Uniform prior over all (fully-resolved) tree topologies.
Branch lengths	Exponential distribution, mean 0.1

```
prset statefreqpr=dirichlet(1.0, 1.0, 1.0, 1.0)
prset shapepr=exp(1)
prset tratiopr=beta(1,1)
prset topologypr = uniform
prset brlenspr=unconstrained:exp(10)
```

A Dirichlet distribution with all four parameters 1.0 assigns equal prior probability density to all (legal) combinations of base frequencies. To make the prior on base frequencies favor equal base frequencies, you could use a value such as Dirichlet(50.0, 50.0, 50.0, 50.0). Such a prior would tend to tie down the base frequencies close to 0.25 unless the data strongly contradicted equal base frequencies.

The `tratiopr` is poorly named. In previous versions of MrBayes, the setting controlled the prior on the rate ratio between transitions and transversions. This parameter, κ, is defined over the range $[0,\infty)$. In recent versions of MrBayes, the prior is described in terms of a [0,1.0] parameter that is a transformation of κ. If we call this parameter x then $\frac{x}{1-x} = \kappa$. Thus, using a Beta(10,1) prior for “`tratiopr`” has an expectation that the rate of a transition is 10 times the rate of a transversion.

The `unconstrained` keyword in the prior for branch lengths indicates that a non-clock model is being used (i.e. the branch lengths are not constrained to obey the molecular clock). Note that MrBayes expects you to specify the hazard parameter for exponential distributions, not the mean. The hazard parameter is the inverse of the mean, so specifying `exp(10)` is the same as specifying an exponential prior distribution having mean 0.1.

4. Next, set up the HKY-gamma model. This amounts to telling MrBayes that we will be using a two-parameter model (a model that distinguishes between transitions and transversions) and that we want gamma-distributed rate heterogeneity with 4 categories.

```
lset nst=2 rates=gamma ngammacat=4
```

5. Specifying the outgroup uses the `outgroup` command (similar to PAUP*):

```
outgroup Anacystis_nidulans
```

6. Now set up the MCMC run itself: `mcmcp ngen=50000 filename=algaeout samplefreq=50 printfreq=200 savebrlens=yes nruns = 1`

The command parameters used here (and their meanings) are:

- `ngen` – the total number of generations,
- `filename` – the **prefix** to use for output file names. Trees will be written toalgaeout.t; samples of parameter values will be in a tab-delimited format in algaeout.p; and statistics about the MCMC algorithm itself will be written in algaeout.mcmc
- `samplefreq` – the frequency with which to sample trees and parameters,
- `printfreq` – the frequency with which to print a one-line progress report.
- `savebrlens` – tells MrBayes to save branch length information in the tree file that is output.
- `nruns=1` – tells MrBayes to run only one simultaneous MCMCMC analysis. This disables the automatic run stopping criterion (more on this feature later) and means that MrBayes will pay attention to the `ngen` parameter setting that we give it. Note: this parameter does *not* control the number of heated chains in MCMCMC (that is the `nchains` parameter and the default is to run 4 chains: 3 "heated" chains, and the "cold" chain that is sampled).

7. All that is left now is to start the run:

   ```
   mcmc
   ```

8. While MrBayes runs, it shows one-line progress reports. The first column is the generation number. The next few columns show the log-likelihoods of the separate chains that are running, with the cold chain indicated by square brackets rather than parentheses. The last complete column is the best estimate of the time remaining until the run completes.

   After it finishes, type `no` in response to the query "Continue with chain? (yes/no):" and then press the Enter key.

9. MrBayes will now report various statistics about the run, such as the percentage of the time it was able to accept proposed changes of various sorts, or the percentage of proposed swaps between chains that it accepted. These percentages should, ideally, all be between about 20% and 40%, but as long as they are not extreme (e.g. 1% or 99%) then things went well.

   MrBayes saves information in several files. One of these should be called algaeout.p. This is the file in which the model parameter values were saved. This file is saved in tab-delimited format so that it is easy to import into a spreadsheet program such as Excel. This would allow you to create history plots showing variation in the parameter values over the course of the run. This file is important because it shows the likelihood values (which can be used as a crude indication of how long to make the burnin period). My own examination of the file I produced using MrBayes suggests that the burnin should end at or around 2,000 iterations ("generations" in MrBayes lingo). By this point, MrBayes has wandered into the heart of the posterior distribution after starting with random tree topology (which is almost certainly a very poor fit to the data).

   The second file of importance is the algaeout.t file, which contains the sampled trees. This is an ordinary NEXUS tree file, and could be read into another program (e.g. PAUP* or TreeView) that understands the NEXUS file format. We will ask MrBayes to summarize this file for us today using the `sumt` command:

   ```
   sumt file=algaeout burnin=41 nruns=1
   ```

Why did we specify 41 as the burnin?[1]

10. You should now see a tree showing all compatible splits (a.k.a. bipartitions) with branches labeled with their split posterior probabilities. The last tree shown will be the same topology, but will make the lengths of the branches proportional to the posterior mean branch length for each split present in the tree. The posterior mean branch length is a simple average. Every time a tree with some split S is visited during the running of the chain, its current branch length is added to a sum and the posterior mean branch length is obtained by simply dividing this sum by the total number of times that split S was present in the current tree during the run.

    Do organisms with green plant chloroplasts group together in this tree? What is the posterior probability of the split separating the green plant chloroplast group from the chromophyte + cyanobacterium group?[2]

    The `sumt` produces 3 useful files:

    (a) algaeout.con has the majority-rule consensus of the sampled trees in NEXUS format. The tree occurs twice. In the first version, internal nodes are given name that are the estimates of the posterior probability that the branch occurs in the true tree. The branch lengths are the posterior mean branch lengths described above. PAUP*, Mesquite, or TreeView can be used to display these trees.

    (b) algaeout.parts contains a table of splits and their associated posterior probabilities (and branch lengths). This file can be used to get probabilities for groupings that do not appear in the majority rule consensus tree.

    (c) algaeout.trprobs contains the sampled topologies. Redundant trees are excluded and the trees are sorted by their posterior probability. The trees are sorted so that the tree with the highest posterior probability is the first tree (the last tree has the lowest probability among the sampled trees – of course there are many unsampled trees that are not written to the file). After the tree name there will be a NEXUS comment such as `[p = 0.015, P = 0.922]`. `p` is the (estimate) of the posterior probability of this tree. `P` is the cumulative posterior probability that the true tree is **this tree or any one of the trees before it in the file**. Thus, `P` is simply the sum of all of the `p`'s before and including this tree. The tree's posterior probabilities are also stored as NEXUS tree weights.

11. To terminate the MrBayes program.

    ```
    quit
    ```

Exercise 2 (optional): Bayes Factors

1. Edit the algaemb.nex file (PAUP* provides a nice text editor), adding the following MrBayes block at the end of the file:

    ```
    begin MrBayes;
      set autoclose=yes;
    ```

[1]We ran the chain 50,000 steps, sampling every 50. Thus, there should be 50,000 / 50 = 1,000 trees saved in the tree file. Actually, there will be 1001 trees in the file because MrBayes saves the starting tree too. I said that burnin should stop at 2,000 steps, which translates to 40 sampled trees, so I am telling MrBayes to not include the starting tree and the next 40 trees in its summary.

[2]The answer to the first question is yes, and the posterior probability I obtained was 0.88 (your results may differ a little).

```
  outgroup Anacystis_nidulans ;
  prset statefreqpr=fixed(equal);
  lset nst=1;
  mcmcp ngen=510000 samplefreq=100 printfreq=1000 nruns=1;
  mcmcp nchains=4 savebrlens=yes;
  mcmc file=jc;
  sump file=jc burnin=101 nruns=1;
end;
```

This sets up the Jukes-Cantor model in MrBayes. The `sump` command estimates the marginal likelihood of the JC model for this data set. You need to make sure that the `nruns` option of the `sump` command is set to the same number as the number of runs you requested in the `mcmc` command. The marginal likelihood of the model represents the average probability of the data using the model, where the average is a weighted average and the weights are provided by the prior distribution. Think of this as the average ability of the model to explain the data. Models that explain the data poorly over the parameter space defined by the prior have a lower marginal likelihood than models that explain the data better on average.

2. Execute this file in MrBayes, and write down the estimated marginal likelihood (harmonic mean), labeling it "log JC marginal likelihood". Type
`quit` to terminate MrBayes.

3. Edit the algaemb.nex file again, this time replacing
```
        lset nst=1;
```
with
```
        lset nst=2;
```
and replacing
```
        mcmc file=jc;
        sump file=jc burnin=101 nruns=1;
```
with
```
        mcmc file=k2p;
        sump file=k2p burnin=101 nruns=1;
```
This time we will be using the K2P model, which differs from the JC model only in allowing transitions to occur at a different rate than transversions. The `nst=2` tells MrBayes that you want the rate matrix to have two different rates (one for transitions and the other for transversions) rather than just one (`nst=1`)[3]. If the transition rate is indeed different than the transversion rate, then on average the K2P model should be able to explain the data better than the JC model, and the Bayes Factor in favor of the K2P model should be greater than 1. The `prset tratiopr` prior specifiication will now be used (this setting is simply ignored when `nst` is not set to 2).

4. Execute this file again in MrBayes, and again write down the estimated marginal likelihood (harmonic mean), labeling it "log K2P marginal likelihood". Type `quit`
to terminate MrBayes.

5. Compute the Bayes Factor in favor of the K2P model over the JC model as follows:

$$\text{BF} = \frac{\bar{L}_1}{\bar{L}_0} = e^{\ln \bar{L}_1 - \ln \bar{L}_0}$$

[3] "nst" stands for number of substitution types.

where $\ln \bar{L}_1$ is the log harmonic mean you wrote down for the K2P model, and $\ln \bar{L}_0$ is the log harmonic mean you wrote down for the JC model. In words, simply subtract the log harmonic mean for JC from the log harmonic mean for K2P. The Bayes Factor is the constant e raised to this value.[4]

What is the Bayes Factor for this comparison of models? Does the K2P model explain the data better than the JC model for this data set?[5]

6. If you have time and wish to explore Bayes Factors further, you can compare the K2P model to the HKY model. This involves simply replacing the
```
prset statefreqpr=fixed(equal)
```
with
```
prset statefreqpr=dirichlet(1.0, 1.0, 1.0, 1.0)
```
Rather than fixing the base frequencies all at 0.25, they will now be allowed to vary during the MCMC run, which effectively switches the model from K2P to HKY. I found that the difference in log marginal likelihoods was about 6.34 for this comparison (with HKY on top), and thus the Bayes Factor in favor of HKY over K2P is about 567. Much less impressive than the previous comparison, but the fact that HKY explains the data more than 500 times better than K2P means that the base frequencies really aren't equal and a model (such as HKY) that allows unequal base frequencies is preferable.

Exercise 3: Marginal Distributions

One of the benefits of the Bayesian approach is the ability to obtain marginal distributions of just about any quantity of interest. For example the marginal distribution of the transition/transversion rate ratio, κ, is the distribution of this parameter with all other parameters in the model integrated out.

If we pretend that the field the MCRobot explores is a bivariate parameter space, with κ on one axis and the evolutionary distance v on the other, then the joint probability density of these two parameters would be represented by a single bivariate normal hill in the field. The marginal distribution of κ would be a univariate normal distribution, and can be visualized by imagining that the "hill" is made up by many tiny balls inside a box, and marginalization involves tilting the field so that the marbles pile up against one side of the box.

The data in `jc.nex` was simulated using a JC69 model, so we know in this case that the true value of κ is 1.0. How certain of this could we be, however, if we didn't know the truth but only had these data to work with.? What is the 95% credible region for κ given this dataset? This is the type of question that can be answered easily using MrBayes.

1. Create a MrBayes block at the bottom of the `jc.nex` file so that the Kimura 2-parameter (1980) model is specified.

```
begin MrBayes;
```

[4]Unfortunately this simple harmonic mean estimator for the marginal likelihood is a poor estimator – it has high variance. Despite the fact that these estimates of the marginal likelihoods are not necessarily very reliable, they have been widely used. See (Lartillot and Philippe, 2006) and the Stepping-stone method of Paul Lewis and collaborators for more reliable methods of estimating the marginal likelihood. The version 3.2 of MrBayes (Ronquist et al., 2012) supports stepping-stone sampling. The manual pdf that comes with MrBayes describes how to conduct a stepping-stone sampling analysis using the `ss` command in MrBayes.

[5]The difference in log marginal likelihoods was 80.68, making the Bayes Factor equal to $e^{80.68} = 1.09 \times 10^{35}$, which is an impressive number! This indicates that the K2P model does indeed explain the data much better than the JC model.

```
  set autoclose=yes;
  lset nst=2;
  prset statefreqpr=fixed(equal) tratiopr=beta(1,1);
  mcmcp ngen=110000 samplefreq=100 printfreq=1000 nruns=1;
  mcmc;
  sump burnin=101 nruns=1;
end;
```

2. Execute this file and at the end of the output you should find a parameter summary that shows the posterior mean, variance and 95% credible intervals for both tree length (TL) and κ (kappa).

   Does the 95% credible region for κ include 1.0? Is this result consistent with the fact that JC was the generating model?[6]

3. Read the jc.nex.p file into Excel if Excel is available. For the 95% credible region, we need to find the 2.5 and 97.5 percentiles. Assuming D is the column containing the κ values, the following formulas can be used to obtain, respectively, the number of sampled values, the posterior mean of kappa, the 2.5 percentile and the 97.5 percentile. The formulas assume that you have copied the entire contents of the jc.nex file and pasted it into the upper left corner of an Excel worksheet. The cell D104 thus corresponds to generation 10100, the first cell considered if the top 101 samples are discared as burnin.

   ```
   =COUNT(D104:D1103)
   =AVERAGE(D104:D1103)
   =PERCENTILE(D104:D1103,0.025)
   =PERCENTILE(D104:D1103,0.975)
   ```

   Do these values agree with those computed by MrBayes?[7]

4. If Tracer (http://tree.bio.ed.ac.uk/software/tracer) is available, exploring marginal distributions of parameters is easy. Just choose File > Import... from the main menu of Tracer, then navigate to the jc.nex.p file, select it and click the Open button. Then, click on kappa in the "Traces" panel on the lower left. (If you do not see the work kappa anywhere on the left, it may be that the "Traces" panel is off the screen on the bottom. In this case, look for a horizontal bar on the lower left; drag this bar upward to reveal the Traces panel.)

5. If Tracer is not available, you can create a histogram of column D (the column labeled kappa) in Excel. This histogram represents the marginal distribution of kappa. In Excel, you must first create a column of bin boundaries. I used the series of 40 values 0.05, 0.1, 0.15, ..., 2.0 for this purpose. Then, from the main menu choose Tools > Data Analysis > Histogram and then specify `$D$104:$D$1103` for the input range (again, assuming κ is in column D) and `$J$1:$J$40` for the bin range (assuming the 40 bin boundaries are in column J).

Exercise 4: Stopping rules

MCMC algorithms can produce arbitrarily precise estimates of the distributions that they sample from **if they are run long enough**. Assessing when an MCMC simulation has been run long enough can be very

[6] Yes, the credible region includes 1.0 (the interval I obtained stretched from 0.906 to 1.149), which is consistent with JC being the model used to generate the data.

[7] If not, be sure you did not include the first 101 values, which were discarded as burnin when the `sump` command was run.

difficult. There are wide variety of tools that can detect failure to converge (e.g. the Google search for "convergence diagnostics" + MCMC returned 17,200 hits for me in June 2006). An inherent weakness in these diagnostics is that they look at where the chain(s) has/have been. If a whole region of high posterior probability has been missed in every run, this error will not be detected. Unfortunately, MCMC errors of this type usually lead us to be too confident in our results (we will underestimate the variance because we have not sampled the landscape thoroughly).

The general defense against inflated confidence due to MCMC error is to run several independent MCMC analyses and compare the results. In particular, if the starting points for different runs are chosen such that they are spread out more than by random ("over-dispersed") then different runs are expected to yield different answers as long as they have been run for an insufficient number of iterations. Given that the runs are all sampling the same posterior distribution, they will all agree *eventually.* Thus, a general strategy for generating reliable results from MCMC analyses is to perform multiple runs making them all long enough for them to agree with each other.

Fortunately, versions of MrBayes after version 3 include an automatic stopping rule. Instead of specifying a fixed number of generations and then assessing whether or not the analysis was sufficiently long, we can ask MrBayes to run the chain until a condition has been met. The convergence diagnostic implemented in MrBayes 3.1.2 is the **average standard deviation in split frequencies**. The standard deviation of the frequency of a particular split (or "clade" or "grouping") across all runs is calculated. These standard deviations are then averaged over all splits which had a frequency (in at least one run) above a user-specified cutoff frequency. As the runs proceed, they should converge to similar split frequencies for all of the splits in the tree. When the average standard deviation in split frequencies drops below the `stopval`, then MrBayes will terminate the MCMC runs.

1. Add the command to use automatic stopping rule to the last version of the algaemb.nex. Replace:
```
        mcmcp ngen=510000 samplefreq=100 printfreq=1000 nruns=1;
```
   with
```
        mcmcp samplefreq=100 printfreq=1000;
        mcmcp nruns=2 stoprule=YES burninfrac=.25 ;
        mcmcp stopval=0.01 minpartfreq=0.05;
        mcmcp mcmcdiagn=YES diagnfreq=100;
```

   These new options mean:

   - `nruns=2` – the automatic stopping rule needs more than one run so that it can compare split frequencies between runs, so 2 is the minimum we can get away with.
   - `stoprule=YES` – *Guess what this one means.*[8]
   - `burninfrac=.25` – every time MrBayes calculates convergence diagnostics, it will discard the first 25% of the run. Alternatively you can specify `burnin=1000` to specify a constant burnin length.
   - `stopval=0.01` – Stop the runs when the average standard deviation of split frequencies (across different runs) is ≤ 0.01.
   - `minpartfreq=0.05` – Determines the (lower) cutoff for split frequencies included when calculating the average standard deviation of split frequencies.
   - `mcmcdiagn=YES` – Write MCMC diagnostics in <prefix>.mcmc

[8] Yup, you're right.

    - `diagnfreq=100` – Calculate the MCMC diagnostics every 100 iterations.

2. Redirect the output to a new set of files by replacing:
   ```
   mcmc file=k2p;
   ```
   with
   ```
   mcmc file=autoStop;
   ```

3. Finally comment out the `sump` command by replacing:
   ```
   sump burnin=101 nruns=1;
   ```
   with
   ```
   [sump burnin=101 nruns=2;]
   ```

4. Execute the file with the `Execute` command.
   How does the progress reported by MrBayes during the run look different from previous runs? What is the trend in the average standard deviation of split frequencies during the run?[9]

5. Unfortunately, if we are using `burninfrac` we do not know the number of samples to discard from the beginning of the run, so we cannot put the `sump` in our MrBayes block. After the chain has run, look at one of the `.t` or `.p` files and see how many samples were taken. Multiply this number by the `burninfrac`. This is your burnin

6. Comment out the MrBayes block in algaemb.nex by surrounding it in [and] or by changing the name of the block so that it is skipped (e.g. change it to `xMrBayes`).

7. Launch MrBayes, execute algaemb.nex and then type:
   `sumt file=autoStop nruns=2 burnin=X`
   where `X` is the burnin that you calculated above. This should give you results that are similar to those you obtained in earlier when we used a (somewhat arbitrary) number of generations.

8. If you have time, you can repeat the run with the automatic stopping rule. You will probably get a different chain length, but the results should be compatible.

The web site http://ceb.csit.fsu.edu/awty is a useful tool for producing graphics that help you assess convergence. Due to bandwidth concerns, we may not all be able to use the site at the same time, but I encourage you to check it out on your own time.

Exercise 5: Partitioned models

MrBayes is extremely flexible in terms of the number of models that it supports. It allows you to partition the data into subsets and apply different models to different subsets of data. Beyond that you can even force some model parameters to be shared across the different model. For instance you could allow two different subsets to have different rate heterogeneity parameters, but force them to share base frequency parameters.

[9]There are now lines like "Average standard deviation of split frequencies: 0.010775" reported every 100 iterations. The average standard deviation of split frequencies may bounce around, but the trend should be to decrease. The run will terminate when the diagnostic falls below the stopval (0.01)

This will be a very brief introduction to how to set up partitioned model analyses in MrBayes. For simplicity sake, we'll use the same `algaemb.nex` that we used above. This is a data set is a group of ribosomal RNA sequences. Unfortunately, I do not know which sites code for stem and which code for loops (if we did we could use MrBayes models for coevolution of nucleotides that pair in stems in the rRNA secondary structures). Instead we'll use an arbitrary partitioning scheme. This will lead to an unsatisfying example, but focus on the commands that we use – you can apply these commands to your own data with more sensible partitions.

1. Relaunch MrBayes and `Execute bglobin.nex`

2. The command to define a partition of the data is very similar to the `CharPartition` used in PAUP*. The general form is `Partition <partition name> = <number of subsets> : <subset definition> , <subset definition> ;` Where everything in the `<>` is replaced with an appropriate value. Unlike the commands we used with PAUP*, the subsets are not named, and we have to tell the program "up front" how many subset there are. Open `bglobin.nex`
   in a text editor. You should see a MrBayes block with the following content:

   ```
   charset non_coding = 1-90 358-432 ;

   charset first_pos  = 91 - 357 \ 3 ;

   charset second_pos = 92 - 357 \ 3 ;

   charset third_pos  = 93 - 357 \ 3 ;

   partition region = 4:non_coding,first_pos,second_pos,third_pos;

   set partition = region;
   ```

   This sets up 4 subsets. The syntax

   ```
   91 - 357 \ 3
   ```

   means "every third character from 91 up to 357." So

   ```
   91 - 357 \ 3
   ```

   means characters $91, 94, 97, 100, \ldots$. This partitioning command would be the one you would use if you wanted to use different models for different coding positions (and the entire data matrix was an in-frame protein coding gene sequences).

3. To tell MrBayes that it needs to use this partition in the next analysis, we use the command:
   `set partition = bc`

4. I find that the key to setting up partitioned models correctly is to verify after every step that you and MrBayes agree about the model that you have set. Fortunately, you can enter:
   `ShowModel`

to see a report from MrBayes about what models are currently in effect. The report ends with a section that lists all of the parameters of any active model. It shows what subsets (referred to "Partitions" in the report) use that parameter.

5. MrBayes uses the `LSet` and `PrSet` to adjust the model itself (what parameters are used) and the prior distributions for each of these parameters. Because there are multiple models in effect, we have to qualify these commands with an `applyto` command parameter that lists which models the command is intended for. The `applyto` uses the number of the data partition as the identifier of the model. So if you enter:

   ```
   LSet ApplyTo=(1) nst= 6;
   ShowModel;
   ```

   The first partition is now assigned a 6 substitution type model (the GTR model). Make sure to put the `applyto` option *before* the other options in the `LSet` and `PrSet` commands.

6. If you examine the report from the previous `ShowModel` command you will see that we have MrBayes configured to use 3 different models, but partitions 2 and 3 are sharing all of their parameters. In MrBayes terminology, these parameters are linked. We can use the `Unlink` command to give the different partitions their own parameters to estimate. This means that we will actually be using different models for each subset. To make MrBayes infer separate equilibrium state frequencies for each subset use the command:

   ```
   Unlink StateFreq=(all);
   ShowModel;
   ```

7. There are several other model changing commands shown in a comment in the data file. Particularly important is the `prset ratepr = Dirichlet(3, 1, 1, 3)` which says that the different subsets of the data can have different rates of evolution. In our example of codon positions, it is very likely that the different codon positions have different mean rates, so we certainly want to let the software explore models where these mean rates vary. In this we think that the non-coding region and the third-base positions are likely to evolve at a higher rate, so we are putting a prior on the rate multiplier for each subset such that the 1st and 4th subset are expected to be evolving three times faster than the other two subsets.

8. Note that these model conditions may be overparameterized – I just set up this model to demonstrate the syntax needed to configure MrBayes

9. Now you can use the `MCMC`, `SumT`, and `SumP` commands to perform the MCMC approximation, and summarize the results.

References

Lartillot, N. and Philippe, H. (2006). Computing Bayes factors using thermodynamic integration. *Systematic Biology*, 55(2):195–207.

Ronquist, F., Teslenko, M., van der Mark, P., Ayres, D. L., Darling, A., Höhna, S., Larget, B., Liu, L., Suchard, M. A., and Huelsenbeck, J. P. (2012). Mrbayes 3.2: Efficient bayesian phylogenetic inference and model choice across a large model space. *Systematic Biology*, page *in press*.

Ronquist, F. R. and Huelsenbeck, J. P. (2003). Mrbayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19(12):1574–1575.