Integrative Biology 200A *University of California, Berkeley*
PRINCIPLES OF PHYLOGENETICS *Spring 2008*
Updated by Nick Matzke

Lab 11: MrBayes Lab

Note: try downloading and installing MrBayes on your own laptop, and modify instructions accordingly.

Introduction

MrBayes uses a Markov Chain Monte Carlo (MCMC) approach to search for trees. There are two phases in an MCMC. The first phase, called "burn-in", is more like a normal heuristic search in which the program explores the parameter space trying to find parameter values near the maximum likelihood. During this phase the likelihood of the trees increases steadily. In the second phase, called stationarity, the program explores the parameter space around the maximum likelihood. During stationarity the likelihood seams to vary stochastically around a mean value.

The cool thing about an MCMC is that when it is in stationarity it samples the parameter values in a proportion that approximates the posterior probability. That is to say, if given your data and your model there is a 20% chance that Tree A is the right tree, then in stationarity a MCMC will produce that tree approximately 20% of the time. This is true not only for trees, but for all the parameters. Thus you can generate a distribution for the value of any parameter. It makes this approximation without holding any of the other parameters constant, but instead integrating over all values of those parameters. Of course this only works if your model is correct, your priors are correct and you are actually in stationarity for long enough. It is not always easy to tell if you are in stationarity, or how long 'long enough' is.

Today we will explore Bayesian phylogenetics as discussed in lecture by running analyses and interpreting results. The goals of the lab are the following:

I. Adding a MrBayes block to a nexus file

II. Conduct a Bayesian analysis

III. Analyze chain diagnostics and "burn-in"

IV. Summarize parameters from the posterior probability distribution

V. Generate a tree with posterior probability values

MrBayes is available free in Mac, PC or Unix from: http://mrbayes.csit.fsu.edu/index.php
The program comes with example data that we'll use for the following exercises. We'll use PAUP or a text editor to edit data matrices and EXCEL to analyze results from the Bayesian analysis.

I. Adding a MrBayes block to a nexus file

A MrBayes block is a string of commands at the end of a data matrix file that tells the MrBayes program what to do. In the MrBayes block you can specify the model you want to use, your prior distributions, the length of the analysis, the number of samples to keep, etc. Each line must end with a "**;**". Once written, you simply have to execute your matrix in the MrBayes program and it will do the rest. The alternative is to execute your data matrix in MrBayes and enter the commands there. The commands are the same no matter where you enter them.

1. First, go to the folder C:/Program Files/IB200A/MrBayes. Copy the MrBayes folder and paste it on your desktop.
2. Now look inside the folder, and open the **primates.nex** file from the MrBayes folder in a text editor. (If you want, you can use your own data matrix instead. Open it in Mesquite, export it for MrBayes, and substitute the name of your data file for *primates* in the rest of the handout. Ask me for help with any of these steps.)

Now you need to add a MrBayes block to the end of the nexus file. The first line of the MrBayes block should be **begin mrbayes;**

The first command line we'll add to the block specifies the substitution model and begins with the command "**lset**". There are many ways to choose a substitution model. For example there is a version of *Modeltest* called *MrModeltest*, which works the same as *Modeltest* and can be found on the *Modeltest* website, but we aren't going to bother with this in this lab. Instead of using a statistical test to pick the appropriate model for your data, just pick whichever one you want. You can try alternative models and see if they result in alternative trees. First, you have to specify the number of substitution types (**nst**), the choices are either 1 (all rates equal), 2 (transition: transversion ratio), or 6 (general-time reversible). Second, specify the rate variation (**rates**). Choose equal (no rate variation or proportion of invariable sites), gamma (rate variation), propinv (invariable sites), or invgamma (both rate variation and invariable sites). Here is an example:

lset nst=6 rates=invgamma;

This is equivalent to the GTR + I + gamma model.

The next line we'll add is the "**mcmcp**" command. This tells MrBayes how many chains to run, the duration of the run, how many sample to take, and how many to print to the screen. Here is an example that you can follow:

mcmcp ngen=100000 printfreq=100 samplefreq=100 nchains=4;

You could type **mcmc** instead of **mcmcp**, but this would start an MCMC run. By typing **mcmcp** you can set the MCMC parameters without starting a run.

ngen - This option sets the number of cycles for the MCMC algorithm. This should be a big number as you want the chain to first reach stationarity, and then remain there for enough time to take lots of samples. We have a time constraint that allows us to only run short analyses. You can experiment with different numbers of cycles

printfreq - This specifies how often info about the chain is printed to the screen.

samplefreq – This specifies how often the Markov Chain is sampled.

nchains – This specifies how many chains are run (one is always cold, the rest are heated)

The final line in the MrBayes block is **end;**

3. Here is the entire MrBayes block you should now have in your file:

   begin mrbayes;

   lset nst=6 rates=invgamma;

   mcmcp ngen=100000 printfreq=100 samplefreq=100 nchains=4;

   end;

4. Now save the file with a new name – for simplicity let's use primatesbayes.nex

II. Conduct a Bayesian analysis

5. In order to run MrBayes, the file you want to execute has to be the same folder as the MrBayes executable. This is not a problem right now, since primatesbayes.nex is already where it should be, but keep it in mind for times you may want to use MrBayes in the future (like on your final project.)

6. Open MrBayes and execute your data. Type:

   execute primatesbayes.nex

If everything is typed correctly in the Nexus file and MrBayes block everything should work. If not, double-check your nexus file and look for errors. (Note: it is also possible type the commands you put in the mrbayes PAUP block at the command line instead of putting them in the file.)

7. Let's take a quick look at what MrBayes has to say about all those commands you entered in the mrbayes PAUP block. Type **help lset**. This gives you an explanation of the different likelihood settings as well as what they are set to. Type **help mcmcp** and **help prset** to view the MCMC and prior settings.

8. OK, enough of that. Let's start the actual analysis. Type **mcmc** to start the run.

You will see a series of eight numbers in rows. The first number is the number of trees that the program has looked at so far. The program shows you every 100th tree, because you set **printfreq=100**. The next first four numbers are from the first run and the four numbers after the asterisk are the second run. The numbers in parentheses are the log likelihoods of the trees from the hot chains and the two numbers in brackets are the log likelihoods of the cold chains from each run. The last number in each row is an estimate of how long the run has left.

After every thousandth tree it shows you the average standard deviation of your split frequencies. This is a good way to tell if you have reached stationarity. When the two runs differ by **less than 0.01** you are probably all good.

III. Analyze chain diagnostics and determine "burn-in"

9. It will take a while for your analysis to run. In fact, it will take 100,000 generations (since we set **ngen=100000**.)
10. Eventually, the program will stop running and will ask you "Continue with analysis? (yes/no):" This is your chance to check and see if your analysis has reached stationarity. If not, you would want to continue with the chain for additional cycles. The output on the screen will summarize the acceptance rates for the moves in the chain.

The best way to tell if you're in stationarity is to see if your average standard deviation has been low for a long time. This is a good diagnostic, and it is ok to decide the run is finished based on it alone, but let's look at the output in excel so that you can see how the likelihoods change.

11. To double-check if you've reached stationarity, you need to look at the parameters that have been generated so far. Leave MrBayes open, but switch your view to the MrBayes folder on your desktop. You should see four new files:
primatesbayes.nex.run1.p
primatesbayes.nex.run1.t
primatesbayes.nex.run2.p
primatesbayes.nex.run2.t

The "dot-p" files contain the parameter estimates and likelihood scores for all of the sampled steps in the chain.

12. Open both "dot-p" files in EXCEL (but don't shut down MrBayes) and make X-Y scatter plots of the likelihood scores (LnL) versus time (Gen).

Did your analysis appear to reach stationarity? If so, at what generation did "burn-in" take place? You may need to change the scale of the X and/or Y-axis to really get a good look at this. How do the likelihoods of the two runs compare to each other? How does this compare to when the average standard deviation reached 0? You can also graph the parameter estimates of your substitution model. Did those estimates reach stationarity? If you need help graphing let me know.

13. Close your .p files without saving them, and go back to MrBayes. If you reached stationarity, you can type **no** to end the run and keep going. If you didn't reach stationarity, continue your chain for additional cycles by typing **yes**.

IV. Summarize parameters from the posterior probability distribution

14. Now, let's calculate the number of generations you want to discard as before burn-in. Here's an example of how to calculate the number of burn-in samples correctly:
    - You run your analysis for **100,000** generations.
    - You sampled every **100** generations.
    - 100,000 / 100 = 1,000: this means that you only have **1,000** samples to analyze.

- How many of those initial **1,000** samples do you want to discard as burn-in?
- Let's say you determined that burn-in occurred around generation 5,000. Then, you would want to discard 5,000/100 = 50 samples.

15. Assuming you have determined burn-in for your data, you are now ready to summarize the information from your analysis. The **sump** command will summarize the "dot-p" file and output a table showing the mean, variance, and 95% CI values for the parameters in your analysis. It is important, when summarizing these data, to exclude those samples retained prior to burn-in. Thus, part of the **sump** command entails indicating which samples to exclude from the summary. Here is an example of the **sump** command:

    sump burnin=*xxx*

V. Generate a tree with posterior probability values

16. To generate a tree you need to use the **sumt** command and know how many samples to discard as burn-in. You've already done this in the previous step. Use the following example to generate your tree with posterior probability values:

    sumt burnin=*xxx*

This command summarizes the trees in the *name*.nex.t file. All of the trees are read from the file and the proportion of the time any single clade is found it is counted. The proportion of the time that the clade is found is an approximation of the posterior probability of the bipartition. This command also generates three files. The ".con" file is a consensus tree that can be imported into PAUP, which shows the estimated branch lengths and posterior probabilities of each clade. The ".parts" file contains not only these values, but also their standard deviations. The ".trprobs" file contains a sorted list of all the completely resolved trees that were found during the analysis with each tree's posterior probability. This list can be used to construct a credible set of trees.

Type **Quit** to shut the program down.