Feedback — Quiz 2: Bayesian Phylogeny (computer)

Help Center

You submitted this quiz on **Sun 28 Jul 2013 7:26 AM PDT**. You got a score of **10.00** out of **10.00**.

Overview

Today's exercise will focus on phylogenetic analysis using Bayesian methods.

As was the case for likelihood methods, Bayesian analysis is founded on having a probabilistic model of how the observed data is produced. (This means that, for a given set of parameter values, you can compute the probability of any possible data set). You will recall from the lecture that in Bayesian statistics the goal is to obtain a full probability distribution over all possible parameter values. To find this so-called posterior probability distribution requires combining the likelihood and the prior probability distribution.

The prior probability distribution shows your beliefs about the parameters before seeing any data, while the likelihood shows what the data is telling about the parameters. Specifically, the likelihood of a parameter value is the probability of the observed data given that parameter value. (This is the measure we have previously used to find the maximum likelihood estimate). If the prior probability distribution is flat (i.e., if all possible parameter values have the same prior probability) then the posterior distribution is simply proportional to the likelihood distribution, and the parameter value with the maximum likelihood then also has the maximum posterior probability. However, even in this case, using a Bayesian approach still allows one to interpret the posterior as a probability distribution. If the prior is NOT flat, then it may have a substantial impact on the posterior although this effect will diminish with increasing amounts of data. A prior may be derived from the results of previous experiments. For instance one can use the posterior of one analysis as the prior in a new, independent analysis.

In Bayesian phylogeny the parameters are of the same kind as in maximum likelihood phylogeny. Thus, typical parameters include tree topology, branch lengths, nucleotide frequencies, and substitution model parameters such as for instance the transition/transversion ratio or the gamma shape parameter. (The difference is that while we want to find the best point estimates of parameter values in maximum likelihood, the goal in Bayesian phylogeny is instead to find a full probability distribution over all possible parameter values). The observed data is again usually taken to be the alignment, although it would of course be more reasonable to say that the sequences are what have been observed (and the alignment should then be inferred along with the phylogeny).

In this exercise we will explore how one can determine and use posterior probability distributions over trees, over clades, and over substitution

parameters. We will also touch upon the difference between marginal and joint probability distributions.

Getting started

- **Construct working directory:**

```
cd ~student
```

```
mkdir bayes
```

```
cd bayes
```

- **Copy files for today's exercise:**

```
cp ~/data/primatemitDNA.nexus ./primatemitDNA.nexus
```

```
cp ~/data/neandertal_aligned.nxs ./neandertal_aligned.nxs
```

```
cp ~/data/hcvsmall.nexus ./hcvsmall.nexus
```

```
cp ~/data/mbplot ./mbplot
```

You have analyzed (versions of) all these data files previously in this course. We will now use Bayesian phylogenetic analysis to complement what we learned in those analyses.

- **Install software:**

```
sudo apt-get install gawk
```

```
sudo apt-get install r-base-core
```

This installs gawk, a UNIX utility that we will use to extract information from data files, and R a freeware statistics package (warmly recommended).

Question 1

Posterior probability of trees

In today's exercise we will be using the program "MrBayes" to perform Bayesian phylogenetic analysis. MrBayes is a program that, like PAUP*, can be controlled by giving commands at a command line prompt. In fact, there is a substantial overlap between the commands used to control MrBayes and the PAUP command language. This should be a help when you are trying to understand how to use the program.

Note that the command "`help`" will give you a list of all available commands. Issuing "`help` *`command`*" will give you a more detailed description of the specified command along with current option values. This is similar to how "*`command`* `?`" works in PAUP. There is also a very useful MrBayes Wiki with a manual and frequently asked questions.

- **Start program:**

  ```
  mb
  ```

  This starts the program, giving you a prompt ("**MrBayes>** ") where you can enter commands.

- **Get a quick overview of available commands:**

  ```
  help
  ```

- **Load your sequences:**

```
execute primatemitDNA.nexus
```

This file contains mitochondrial DNA sequences from 5 different primates. Note that MrBayes accepts input in nexus format, and that this is the same command that was used to load sequences in PAUP*. In general, you can use many of the PAUP commands in MrBayes also.

- **Inspect data set:**

```
showmatrix
```

- **Define outgroup:**

```
outgroup Gibbon
```

- **Specify your model of sequence evolution:**

```
lset nst=2 rates=gamma
```

This command is again very much like the corresponding one in PAUP. You are specifying that you want to use a model with two substitution types (`nst=2`), and this is automatically taken to mean that you want to distinguish between transitions and transversions. Furthermore, `rates=gamma` means that you want the model to use a gamma distribution to account for different rates at different sites in the sequence.

- **Start Markov chain Monte Carlo sampling:**

Make sure to make the shell window as wide as possible and then issue the following commands to start the run:

```
mcmc ngen=100000 samplefreq=100 nchains=3 diagnfreq=5000
```

What you are doing here is to use the method known as MCMCMC ("Metropolis-coupled Markov chain Monte Carlo") to empirically

determine the posterior probability distribution of trees, branch lengths and substitution parameters. Recall that in the Bayesian framework this is how we learn about parameter values: instead of finding the best point estimates, we typically want to quantify the probability of the entire range of possible values. An estimate of the time left is shown in the last column of output.

Let us examine the command in detail. First, `ngen=100000 samplefreq=100` lets the search run for 100,000 steps ("generations") and saves parameter values once every 100 rounds (meaning that a total of 1000 sets of parameter values will be saved). The option `nchains=3` means that the MCMCMC sampling uses 3 parallel chains (but see below): one "cold" from which sampling takes place, and two "heated" that move around in the parameter space more quickly to find additional peaks in the probability distribution.

The option `diagnfreq=5000` has to do with testing whether the MrBayes run is succesful. Briefly, MrBayes will start two entirely independent runs starting from different random trees. In the early phases of the run, the two runs will sample very different trees but when they have reached convergence (when they produce a good sample from the posterior probability distribution), the two tree samples should be very similar. Every `diagnfreq` generations, the program will compute a measure of how similar the tree-samples are (specifically, the measure is the average standard deviation of split frequencies. A "split" is the same as a bipartition, i.e., a division of all leaves in the tree in two groups, by cutting an internal branch). As a rule of thumb, you may want to run until this value is less than 0.01.

During the run you will see reports about the progress of the two sets of four chains. Each line of output lists the generation number and the log likelihoods of the current tree/parameter combination for each of the two groups of three chains (a column of asterisks separate the results for the independent runs). The cold chains are the ones enclosed in brackets [...], while the heated chains are enclosed in parentheses (...). Occasionally the chains will swap so one of the heated chains now becomes cold (and sampling then takes place from this chain).

- **Continue run until parallel runs converge on same solution:**

  At the end of the run, Mrbayes will print the average standard deviation of split frequencies (which is a measure of how similar the tree samples of the two independent runs are). We recommend that you continue with the analysis until the value gets below 0.01 (if the value is larger than 0.01 then you should answer "yes" when the program asks "`Continue the analysis? (yes/no)`".)

- **Question:**

  Once you have reached convergence (and answered "no" to continue the analysis): How many generations did you have to run?

You entered:

100000

Your Answer		Score	Explanation
100000	✔	1.00	
Total		1.00 / 1.00	

Question 2

- **Have a look at the resulting sample files:**

  Open a new Terminal and cd to the `bayes` directory. Open one of the parameter sampling files in an nedit window:

  ```
  nedit primatemitDNA.nexus.run1.p &
  ```

  This file contains one line for each sampled point (you may want to turn off line-wrapping in nedit under the preferences menu). Each *row* corresponds to a certain sample time (or generation). Each *column* contains the sampled values of one specific parameter. The first line contains headings telling what the different columns are: "`lnL`" is the log likelihood of the current parameter estimates, "`TL`" is the tree length (sum of all branch lengths), "`kappa`" is the transition/transversion rate ratio, "`pi(A)`" is the frequency of A (etc.), and "`alpha`" is the shape parameter for the gamma distribution. (Column headings may be shifted relative to their corresponding columns). Note how the values of most parameters change a lot during the initial "burnin" period, before they settle near their most probable values. Now, close the nedit window and have a look at the file containing sampled trees:

  ```
  nedit primatemitDNA.nexus.run1.t &
  ```

  Tree topology is also a parameter in our model, and exactly like for the other parameters we also get samples from tree-space. One tree is printed per line in the parenthetical format used by most phylogeny software. There are 5 taxa in the present data set, meaning that the tree-space consists of only 15 different possible trees. Since we have taken more than 15 sample points, there must be several lines containing the same tree topology. Close the nedit window when you are done.

- **Examine MCMC trajectory for nucleotide frequency:**

Recall, that the idea in MCMCMC sampling is to move around in parameter space in such a way that the points will be visited according to their posterior probability (i.e., a region with very high posterior probability will be visited frequently). Now,plot the sampled values for the frequency of A for one of the run files:

```
./mbplot primatemitDNA.nexus.run1.p 5 0 10000
```

mbplot is a small script written by me, that extracts the relevant columns from the .p file and uses gnuplot to produce a plot of how the value changes with generation number. These options mean that you will get a plot of *[Math Processing Error]* (column 5 in the file) from generation 0 to generation 10,000. Note how the Markoc chain moves around in parameter space, sampling different possible values of *[Math Processing Error]*. You can experiment with plotting other columns as well, or plotting other ranges of generations if you want.

- **Investigate posterior probability distribution over trees:**

  MrBayes provides the sumt command to summarize the sampled trees. Before using it, we need to decide on the burn-in: The burn-in is the initial set of samples that are typically discarded, because we want to ensure that the MCMC has moved away from the random starting values, and has found the peaks of the probability landscape. Since the convergence diagnostic we used previously to determine when to stop the analysis discarded the first 25% of the samples, it makes sense to also discard 25% of the samples obtained during the analysis.

  Return to the shell window where you have MrBayes running. In the command below `relburnin=yes` and `burninfrac=0.25` tells MrBayes to discard 25% of the samples as burnin (you could also have explicitly given the number of samples to discard - `help sumt` will give you details about the command and the current option settings).

  ```
  sumt contype=halfcompat conformat=simple relburnin=yes burninfrac=0.25 showtreeprobs=yes
  ```

  (Scroll back so you can see the top of the output when the command is done). This command gives you a summary of the trees that are in the file you examined manually above. The option `contype=halfcompat` requests that a majority rule consensus tree is calculated from the set of trees that are left after discarding the burnin. This consensus is the first tree plotted to the screen. Below the consensus cladogram, a consensus phylogram is plotted. The branch lengths in this have been averaged over the trees in which that branch was present (a particular branch corresponds to a bi-partition of the data, and will typically not be present in every sampled tree). The cladogram also has "clade credibility" values. We will return to the meaning of these later in today's exercise.

  What most interests us right now is the list of trees that is printed after the phylogram. These trees are labeled "Tree 1", "Tree 2", etc, and are sorted according to their posterior probability which is indicated by a lower-case p after the tree number. (The upper-case P gives the

cumulated probability of trees shown so far, and is useful for constructing a credible set). This list highlights how Bayesian phylogenetic analysis is different from maximum likelihood: Instead of finding the best tree(s), we now get a full list of how probable *any* possible tree is.

The list of trees and probabilities was printed because of the option `showtreeprobs=yes`. Note that you probably do not want to issue that command if you have much more than 5 taxa! In that case you could instead inspect the file named `primatemitDNA.nexus.trprobs` which is now present in the same directory as your other files (this file is automatically produced by the `sumt` command).

NOTE: Annoyingly, there is a bug in the version of mrbayes we are using here, which means leaf names are not printed on the list of trees with probabilities. However, the most probable tree here in fact is identical to the consensus tree printed above it.

- **Question:** What is the posterior probability of the most probable tree?

You entered:

0.979

Your Answer		Score	Explanation
0.979	✔	1.00	
Total		1.00 / 1.00	

Question 3

Analysis of Neanderthal data (posterior probability of clades)

The classical view emerging from anatomical and archaeological studies places Neanderthals as a different species from Homo sapiens.

This is in agreement with the "Out-of-Africa hypothesis", which states that Neanderthals coexisted without mating with modern humans who

originated in Africa somewhere between 100,000 to 200,000 years ago. There is, however, also anatomical and paleontological research which supports the so-called "multi-regional hypothesis", which propounds that some populations of archaic Homo evolved into modern human populations in many geographical regions. Consequently, Neanderthals could have contributed to the genetic pool of present-day Europeans. We will use the present data set to consider this issue.

- **Load Neanderthal data set:**

  In the Terminal where you have MrBayes running:

  ```
  execute neandertal_aligned.nxs
  ```

  ```
  delete 5-40
  ```

  As we did for the maximum likelihood analysis, we will discard some of the human sequences in order to speed up the analysis. The command `delete 5-40` removes sequence number 5 to sequence number 40 from the active data set.

- **Investigate data:**

  ```
  showmatrix
  ```

  This data set consists of an alignment of mitochondrial DNA from human (17 sequences), chimpanzee (1 sequence), and Neanderthal (1 sequence). The Neanderthal DNA was extracted from archaeological material, specifically bones found at Vindija in Croatia.

- **Start analysis:**

  ```
  outgroup Pan_troglodytes
  lset nst=2
  mcmc ngen=200000 nchains=3 diagnfreq=10000
  ```

- **Find posterior probability of clades:**

  ```
  sumt contype=halfcompat showtreeprobs=no relburnin=yes burninfrac=0.25
  ```

Examine the consensus tree that is plotted to screen: On the branches that are resolved, you will notice that numbers have been plotted. These are clade-credibility values, and are in fact the posterior probability that the clade is real (based on the present data set). These numbers are different from bootstrap values: unlike bootstrap support (which have no clear statistical meaning) these are actual probabilities. Furthermore, they have been found using a full probabilistic model, instead of neighbor joining, and have still finished in a reasonable amount of time. These features make Bayesian phylogeny very useful for assessing hypotheses about monophyly.

- **Question:**

  What is the clade probability for Homo sapiens being a monophyletic group excluding the Neanderthal? (Express as decimal number, e.g., 0.67).

You entered:

1.0

Your Answer		Score	Explanation
1.0	✔	1.00	
Total		1.00 / 1.00	

Question 4

Probability distributions over other parameters

As the last thing, we will now turn away from the tree topology, and instead examine the other parameters that also form part of the probabilistic model. We will do this using a reduced version of the Hepatitis C virus data set that we have examined previously. Stay in the shell window where you just performed the analysis of Neanderthal sequences.

- **Load data set:**

```
execute hcvsmall.nexus
```

- **Define site partition:**

```
charset 1stpos=1-.\3
charset 2ndpos=2-.\3
charset 3rdpos=3-.\3
partition bycodon = 3:1stpos,2ndpos,3rdpos
set partition=bycodon
prset ratepr=variable
```

This is an alternative way of specifying that different sites have different rates. Instead of using a gamma distribution and learning which sites have what rates from the data, we are instead using our prior knowledge about the structure of the genetic code to specify that all 1st codon positions have the same rate, all 2nd codon positions have the same rate, and all 3rd codon positions have the same rate. Specifically, `charset 1stpos=1-.\3` means that we define a character set named "1stpos" which includes site 1 in the alignment followed by every third site ("`\3`", meaning it includes sites 1, 4, 7, 11, ...) until the end of the alignment (here denoted ".").

- **Specify model:**

```
lset nst=6
```

This specifies that we want to use a model of the General Time Reversible (GTR) type, where all 6 substitution types have separate rate parameters.

When the `lset` command was discussed previously, a few issues were glossed over. Importantly, and unlike PAUP, the lset command in MrBayes gives no information about whether nucleotide frequencies are equal or not, and whether they should be estimated from the data or not. In MrBayes this is instead controlled by defining the prior probability of the nucleotide frequencies (the command `prset` can be used to set priors). For instance, a model with equal nucleotide frequencies corresponds to having prior probability 1 (one) for the frequency vector (A=0.25, C=0.25, G=0.25, T=0.25), and zero prior probability for the infinitely many other possible vectors. As you will see below, the default prior is not this limited, and the program will therefore estimate the frequencies from the data.

- **Inspect model details:**

```
showmodel
```

This command gives you a summary of the current model settings. You will also get a summary of how the prior probabilities of all model parameters are set. You will for instance notice that the nucleotide frequencies (parameter labeled "`Statefreq`") have a "Dirichlet" prior. We will not go into the grisly details of what exactly the Dirichlet distribution looks like, but merely note that it is a distribution over many variables, and that depending on the exact parameters the distribution can be more or less flat. The Dirichlet distribution is a handy way of specifying the prior probability distribution of nucleotide (or amino acid) frequency vectors. The default statefreq prior in MrBayes is the flat or un-informative prior dirichlet(1,1,1,1).

We will not go into the priors for the remaining parameters in any detail, but you may notice that by default all topologies are taken to be equally likely (a flat prior on trees).

- **Start MCMC sampling:**

```
mcmc ngen=300000 samplefreq=100 diagnfreq=10000 nchains=3
```

The run will take a few minutes to finish (you may want to ensure that the average standard deviation of split frequencies is less than 0.01 before ending the analysis).

- **Compute summary of parameter values:**

```
sump relburnin=yes burninfrac=0.25
```

The `sump` command works much like the `sumt` command for the non-tree parameters. Again, we are using 25% of the total number of samples as burnin.

First, you get a plot of the lnL as a function of generation number. Values from the two independent runs are labeled "1" and "2" respectively. If the burnin is suitable, then the points should be randomly scattered over a narrow lnL interval.

Secondly, the posterior probability distribution of each parameter is summarized by giving the mean, variance, median, and 95% credible interval.

- **Question:** Report the mean of the relative substitution rate parameters `r(AC)` and `r(CG)`. The values must be entered in this order and

separated by spaces.

You entered:

0.137309 0.041772

Your Answer		Score	Explanation
0.137309	✔	0.50	
0.041772	✔	0.50	
Total		1.00 / 1.00	

Question 5

Based on the reported posterior means, does it seem that `r(CG)` is different from `r(AC)`?

Your Answer		Score	Explanation
○ There is no difference between the means of `r(CG)` and `r(AC)`.			
◉ It seems that `r(CG)` *[Math Processing Error]* `r(AC)`	✔	1.00	
○ It seems that `r(CG)` *[Math Processing Error]* `r(AC)`			
Total		1.00 / 1.00	

Question 6

- **Marginal vs. joint distributions:**

  Strictly speaking the comparison above was not entirely appropriate. We first found the overall distribution of the `r(CG)` parameter and then compared its mean to the mean of the overall distribution of the `r(AC)` parameter. By doing things this way, we are ignoring the possibility that the two parameters might be associated in some way. For instance, one parameter might always be larger than the other in any individual sample, even though the total distributions overlap. We should instead be looking at the distribution over both parameters simultaneously. A probability distribution over several parameters simultaneously is called a "joint distribution" over the parameters.

  By looking at one parameter at a time, we are summing its probability over all values of the other parameters. This is called the marginal distribution.

- **Examine marginal and joint distributions:**

  Again find a shell window where MrBayes is not running and issue the following command:

  ```
  cat hcvsmall.nexus.run1.p | grep -v '^Gen' | gawk '{if (NR > 750) print $7}' > rCG.data
  ```

  This command takes the parameter file, removes the header line, and for all lines that were sampled after the burnin period (set to 750 here) it prints the `r(CG)` parameter to the file named "rCG.data". Now extract a few extra interesting columns in a similar manner.

  ```
  cat hcvsmall.nexus.run1.p | grep -v '^Gen' | gawk '{if (NR > 750) print $4}' > rAC.data
  cat hcvsmall.nexus.run1.p | grep -v '^Gen' | gawk '{if (NR > 750) print $14}' > rm1.data
  cat hcvsmall.nexus.run1.p | grep -v '^Gen' | gawk '{if (NR > 750) print $15}' > rm2.data
  cat hcvsmall.nexus.run1.p | grep -v '^Gen' | gawk '{if (NR > 750) print $16}' > rm3.data
  ```

- **Start R, plot marginal distributions of rCG and rAC**

  ```
  R
  ```

  This will give you a prompt

```
>
```

where you can enter commands. R is an excellent freeware program for doing statistical analysis, and it has a range of very useful plotting features, that we here will use to plot the marginal distributions of rCG and rAC. At the R-prompt, enter the following commands, which will read the data from the files we just created, and plot histograms (in the form of so-called density plots);

```
AC = scan("rAC.data")
```

```
CG = scan("rCG.data")
```

```
plot(density(AC), col="red", xlim=c(0,0.3), ylim=c(0,40),main="Marginal posterior distributions",xlab="Substitution ra
te")
```

```
lines(density(CG), col="blue")
```

```
legend(x="topright", legend=c("rAC", "rCG"),lty=c(1,1), col=c("red","blue"))
```

- **Question:** How do the marginal distributions behave?

Your Answer		Score	Explanation
◉ The two marginal distributions have a small overlap. The `r(CG)` distribution has the highest peak.	✔	1.00	
○ The two marginal distributions have a large overlap. The `r(CG)` distribution has the highest peak.			
○ The two marginal distributions have no overlap. The `r(CG)` distribution has the highest peak.			
○ The two marginal distributions have a large overlap. The `r(AC)` distribution has the highest peak.			

The two marginal distributions have a small overlap. The `r(AC)` distribution has the highest peak.

Total	1.00 / 1.00

Question 7

- **Quit R:**

```
ctrl-D
```

Press and hold the `ctrl` and `d` keys at the same time. Answer `n` when asked whether you want to save workspace image.

- **Count total number of post-burnin sample points:**

Above, we used R to plot the marginal distributions of rCG and rAC. We now want to examine the joint distribution to find the real probability that rAC > rCG. Close the plot window and issue this command to count the total number of sample points left after discarding the burnin.

```
wc -l rCG.data
```

- **Question:** How many sample points are there in all after the burnin has been discarded?

You entered:

2252

Your Answer	Score	Explanation

2252	✔	1.00
Total		1.00 / 1.00

Question 8

- **Count sample points where rAC *[Math Processing Error]* rCG:**

```
paste rAC.data rCG.data | gawk '{if ($1>$2) print $0}' | wc -l
```

- **Question:** This counts the number of sample points where rAC *[Math Processing Error]* rCG. How many such points are there?

You entered:

2252

Your Answer		Score	Explanation
2252	✔	1.00	
Total		1.00 / 1.00	

Question 9

What is the joint probability that rAC > rCG? (Hint: You can calculate this by dividing the last number by the first).

You entered:

1.0

Your Answer		Score	Explanation
1.0	✔	1.00	
Total		1.00 / 1.00	

Question 10

Note how examining the joint distribution provides you with information that you could not get from simply comparing the marginal distributions. This very simple procedure can also be performed using spread-sheet programs, and can obviously be used to answer many different questions.

- **Start R, plot relative substitution rates at codon positions 1, 2, and 3:**

```
R
```

```
m1 = scan("rm1.data")
```

```
m2 = scan("rm2.data")
```

```
m3 = scan("rm3.data")
```

```
plot(density(m1), col="red", xlim=c(0,2.5), ylim=c(0,7),main="Marginal posterior distribution", xlab="Substitution rat
```

```
e")
```

```
lines(density(m2), col="blue")
```

```
lines(density(m3), col="green")
```

```
legend(x="topright", legend=c("pos 1", "pos 2", "pos 3"),lty=c(1,1,1), col=c("red","blue","green"))
```

- **Question:** This command plots the relative substitution rates at the first, second, and third codon positions. Since random mutations presumably hit all three codon positions with the same frequency, any differences must be caused by subsequent selection. How does the result fit with your knowledge of the structure of the genetic code? (more than one answer may be correct)

Your Answer		Score	Explanation
☐ Codon position 3 is the most conserved codon position.	✔	0.17	
☐ Codon position 1 is the most conserved codon position.	✔	0.17	
☑ Codon position 2 is the most conserved codon position.	✔	0.17	
☐ Codon position 2 is the most degenerate of the codon positions.	✔	0.17	
☐ Codon position 1 is the most degenerate of the codon positions.	✔	0.17	
☑ Codon position 3 is the most degenerate of the codon positions.	✔	0.17	
Total		1.00 / 1.00	

Question Explanation

We see that codon position 3 is experiencing the highest mutation rate and position 2 the lowest. Position 1 is in between. This makes sense since position 3 is the most degenerate of the codon positions and mutations at this position will often not change the encoded amino acid.

Mutations on codon position 2, on the other hand, will always change the encoded amino acid (except for changes between the stop codons UAA and UGA) and therefore this is the most conserved codon position. Mutations at codon position 1 will often, but not always, change the encoded amino acid.