Phylogenetic Inference using RevBayes

Model Selection & Data Partitioning

Overview

This tutorial demonstrates how to set up and perform an analysis that calculates Bayes factors to select among different partition configurations of aligned DNA sequences. After selecting the model that is best supported by the data, the exercise continues with basic inference of an unrooted tree topology and branch lengths using Markov chain Monte Carlo (MCMC).

1 Exercise: Model Selection & Partitioning using Bayes Factors

1.1 Introduction

Variation in the evolutionary process across the sites of nucleotide sequence alignments is well established, and is an increasingly pervasive feature of datasets composed of gene regions sampled from multiple loci and/or different genomes. Inference of phylogeny from these data demands that we adequately model the underlying process heterogeneity; failure to do so can lead to biased estimates of phylogeny and other parameters (Brown and Lemmon, 2007). To accommodate process heterogeneity within and/or between various gene(omic) regions, we will evaluate the support for various partition schemes using Bayes factors to compare the marginal likelihoods of the candidate partition schemes.

Accounting for process heterogeneity involves adopting a 'mixed-model' approach, (Ronquist and Huelsenbeck, 2003) in which the sequence alignment is first parsed into a number of partitions that are intended to capture plausible process heterogeneity within the data. The determination of the partitioning scheme is guided by biological considerations regarding the dataset at hand. For example, we might wish to evaluate possible variation in the evolutionary process within a single gene region (*e.g.*, between stem and loop regions of ribosomal sequences), or among gene regions in a concatenated alignment (*e.g.*, comprising multiple nuclear loci and/or gene regions sampled from different genomes). The choice of partitioning scheme is up to the investigator and many possible partitions might be considered for a typical dataset.

Next, a substitution model is specified for each predefined process partition (using a given model-selection criterion, such as Bayes factors). In this exercise, we assume that each partition evolved under an independent general-time reversible model with gamma-distributed rates across sites (GTR+Γ). Under this model the observed data are conditionally dependent on the exchangeability rates (θ), stationary base frequencies (π), and the degree of gamma-distributed among-site rate variation (α), as well as the unrooted tree topology (Ψ) and branch lengths (ν). We show the graphical model representation of the GTR+Γ mode in Figure **??**. When we assume different GTR+Γ models for each partitions, this results in a composite model, in which all sites are assumed to share a common, unrooted tree topology and proportional branch lengths, but subsets of sites ('data partitions') are assumed to have independent substitution model parameters. This composite model is referred to as a *mixed model*.

Finally, we perform a separate MCMC simulation to approximate the joint posterior probability density of the phylogeny and other parameters. Note that, in this approach, the mixed model is a fixed assumption of the inference (*i.e.*, the parameter estimates are conditioned on the specified mixed model), and the parameters for each process partition are independently estimated.

For most sequence alignments, several (possibly many) partition schemes of varying complexity are plausible *a priori*, which therefore requires a way to objectively identify the partition scheme that balances estimation bias and error variance associated with under- and over-parameterized mixed models, respectively. Increasingly, mixed-model selection is based on *Bayes factors* (*e.g.*, Suchard, Weiss and Sinsheimer, 2001), which involves first calculating the marginal likelihood under each candidate partition scheme and then comparing the ratio of the marginal likelihoods for the set of candidate partition schemes (Brandley, Schmitz and Reeder, 2005; Nylander et al., 2004; McGuire et al., 2007). The analysis pipeline that we will use in this tutorial is depicted in Figure 1.

Given two models, M_0 and M_1, the Bayes factor comparison assessing the relative plausibility of each

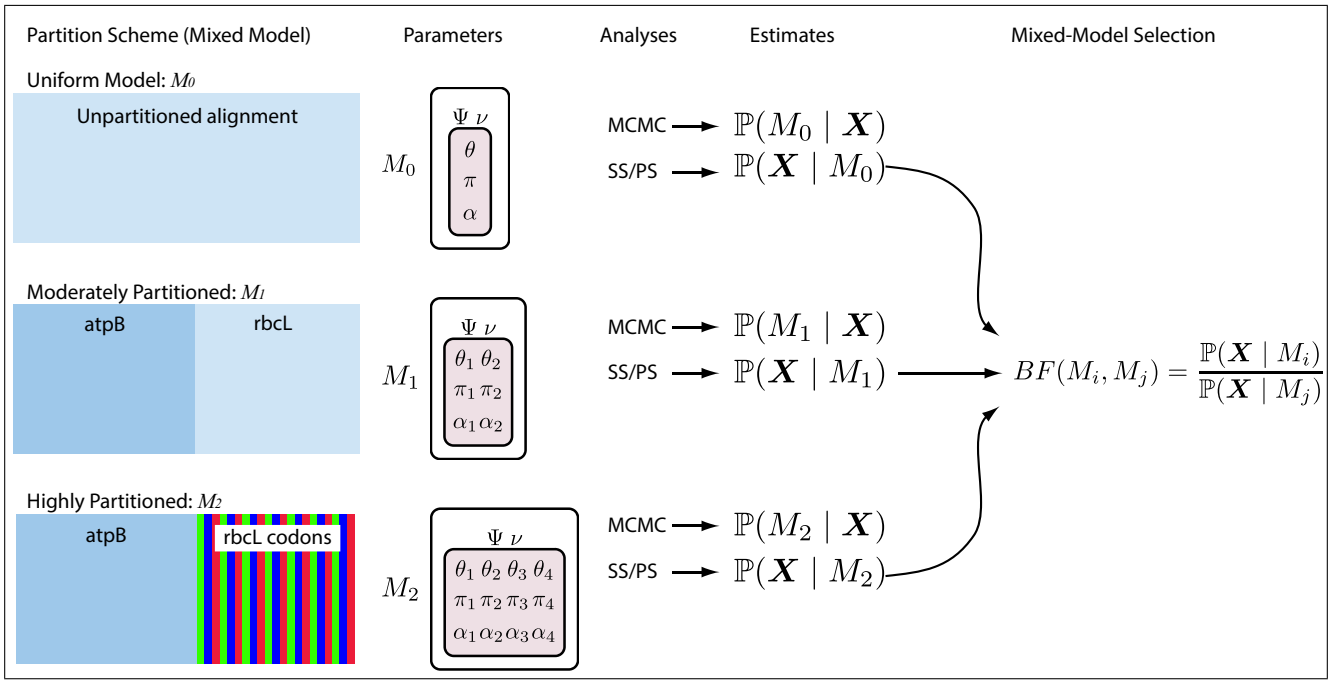

Figure 1: The analysis pipeline for Exercise 1. We will explore three partition schemes for the conifer dataset. The first model (the 'uniform model', M_0) assumes that all sites evolved under a common GTR+Γ substitution model. The second model (the 'moderately partitioned' model, M_1) invokes two data partitions corresponding to the two gene regions (atpB and rbcL), and assumes each subset of sites evolved under an independent GTR+Γ model. The final mixed model (the 'highly partitioned' model, M_2) invokes four data partitions—the first partition corresponds to the atpB gene region, and the remaining partitions correspond to the three codon positons of the rbcL gene region—and each data partition is assumed evolved under an independent GTR+Γ substitution model. Note that we assume that all sites share a common tree topology, Ψ, and branch-length proportions, ν, for each of the candidate partition schemes. We perform two separate sets of analyses for each mixed model—a Metropolis-coupled MCMC simulation to approximate the joint posterior probability density of the mixed-model parameters, and a 'stepping-stone' MCMC simulation to approximate the marginal likelihood for each mixed model. The resulting marginal-likelihood estimates are then evaluated using Bayes factors to assess the fit of the data to the three candidate mixed models.

model as an explanation of the data, $BF(M_0, M_1)$, is:

$$BF(M_0, M_1) = \frac{\text{posterior odds}}{\text{prior odds}}.$$

The posterior odds is the posterior probability of M_0 given the data, $\mathbf{X}$, divided by the posterior odds of M_1 given the data:

$$\text{posterior odds} = \frac{\mathbb{P}(M_0 \mid \mathbf{X})}{\mathbb{P}(M_1 \mid \mathbf{X})},$$

and the prior odds is the prior probability of M_0 divided by the prior probability of M_1:

$$\text{prior odds} = \frac{\mathbb{P}(M_0)}{\mathbb{P}(M_1)}.$$

Thus, the Bayes factor measures the degree to which the data alter our belief regarding the support for M_0 relative to M_1 (Lavine and Schervish, 1999):

$$BF(M_0, M_1) = \frac{\mathbb{P}(M_0 \mid \mathbf{X}, \theta_0)}{\mathbb{P}(M_1 \mid \mathbf{X}, \theta_1)} \div \frac{\mathbb{P}(M_0)}{\mathbb{P}(M_1)}. \quad (1)$$

This, somewhat vague, definition does not lead to clear-cut identification of the "best" model. Instead, *you* must decide the degree of your belief in M_0 relative to M_1. Despite the absence of any strict "rule-of-thumb", you can refer to the scale (outlined by Jeffreys, 1961) for interpreting these measures (Table 1).

Table 1: The scale for interpreting Bayes factors by Harold Jeffreys (1961).

$BF(M_0, M_1)$	Strength of evidence
$< 1:1$	Negative (supports M_1)
$1:1$ to $3:1$	Barely worth mentioning
$3:1$ to $10:1$	Substantial
$10:1$ to $30:1$	Strong
$30:1$ to $100:1$	Very strong
$> 100:1$	Decisive

For a detailed description of Bayes factors see Kass and Raftery (1995)

Unfortunately, direct calculation of the posterior odds to prior odds ratio is unfeasible for most phylogenetic models. However, we can further define the posterior odds ratio as:

$$\frac{\mathbb{P}(M_0 \mid \mathbf{X})}{\mathbb{P}(M_1 \mid \mathbf{X})} = \frac{\mathbb{P}(M_0)}{\mathbb{P}(M_1)} \frac{\mathbb{P}(\mathbf{X} \mid M_0)}{\mathbb{P}(\mathbf{X} \mid M_1)},$$

where $\mathbb{P}(\mathbf{X} \mid M_i)$ is the *marginal likelihood* of the data marginalized over all parameters for M_i; it is also referred to as the *model evidence* or *integrated likelihood*. More explicitly, the marginal likelihood is the probability of the set of observed data ($\mathbf{X}$) under a given model (M_i), while averaging over all possible values of the parameters of the model (θ_i) with respect to the prior density on θ_i

$$\mathbb{P}(\mathbf{X} \mid M_i) = \int \mathbb{P}(\mathbf{X} \mid \theta_i)\mathbb{P}(\theta_i)dt. \tag{2}$$

If you refer back to equation 1, you can see that, with very little algebra, the ratio of marginal likelihoods is equal to the Bayes factor:

$$BF(M_0, M_1) = \frac{\mathbb{P}(\mathbf{X} \mid M_0)}{\mathbb{P}(\mathbf{X} \mid M_1)} = \frac{\mathbb{P}(M_0 \mid \mathbf{X}, \theta_0)}{\mathbb{P}(M_1 \mid \mathbf{X}, \theta_1)} \div \frac{\mathbb{P}(M_0)}{\mathbb{P}(M_1)}. \tag{3}$$

Therefore, we can perform a Bayes factor comparison of two models by calculating the marginal likelihood for each one. Alas, exact solutions for calculating marginal likelihoods are not known for phylogenetic models (see equation 2), thus we must resort to numerical integration methods to estimate or approximate these values. In this exercise, we will estimate the marginal likelihood for each partition scheme using both the stepping-stone (Xie et al., 2011) and path sampling estimators (Gelman and Meng, 1998; Lartillot and Philippe, 2006; Friel and Pettitt, 2008).

1.2 Getting Started

This tutorial assumes that you have already downloaded, compiled, and installed RevBayes. We also recommend that—if you are working on a Unix machine—you put the `rb` binary in your path.

For the exercises outlined in this tutorial, we will use RevBayes interactively by typing commands in the command-line console. The format of this exercise uses `lavender blush shaded boxes` to delineate

important steps. The various RevBayes commands and syntax are specified using **typewriter text**. And the specific commands that you should type (or copy/paste) into RevBayes are indicated by shaded box and prompt. For example, after opening the RevBayes program, you can load your data file:

```
RevBayes > data_atpB <- readCharacterData("data/conifer_atpB.nex")[1]
```

For this command, type in the command and its options:
data_atpB <- readCharacterData("data/conifer_atpB.nex")[1]. **DO NOT** type in "**RevBayes >**", the prompt is simply included to replicate what you see on your screen.

Multi-line entries, particularly loops, will often be displayed in boxes without the **RevBayes >** prompt so that they can be copied and pasted wholly.

```
for( i in 1:12 ){
  x[i] ~ dnExponential(1.0)
}
```

This tutorial also includes hyperlinks: bibliographic citations are burnt orange and link to the full citation in the references, external URLs are cerulean, and internal references to figures and equations are purple.

The various exercises in this tutorial take you through the steps required to perform phylogenetic analyses of the example datasets. In addition, we have provided the output files for every exercise so you can verify your results. (Note that since the MCMC runs you perform will start from different random seeds, the output files resulting from your analyses *will not* be identical to the ones we provide you.)

- Download data and output files from: https://molevol.mbl.edu/index.php/RevBayes
- Also note that "pre-cooked" output files are provided in the download. Throughout this tutorial, you can use those files to summarize output if you do not have time to run the full analyses yourself.

1.3 Launch RevBayes

Execute the RevBayes binary. If this program is in your path, then you can simply type in your Unix terminal:

- **$ rb**

1.4 An Unpartitioned Analysis

The first model is a uniform model over all sequences. Hence, the model is equivalent to the exercise on substitution models. To specify the model please consult the previous exercise.

Estimating the Marginal Likelihood

Typically, model comparison is performed prior to running the full MCMC analysis under a model. If you calculated the Bayes factors to determine the relative support for the uniform model and found that there was strong evidence supporting this model over others (hint: this is not true if you proceed with this tutorial), then it would be worth your time to proceed with the MCMC steps outlined above. The following steps will describe using stepping-stone and path sampling methods on a set of power posteriors to estimate marginal likelihoods under the uniform model.

With a fully specified model, we can set up the **powerPosterior()** analysis to create a file of 'powers' and likelihoods from which we can estimate the marginal likelihood using stepping-stone or path sampling. This method computes a vector of powers from a beta distribution, then executes an MCMC run for each power step while raising the likelihood to that power. In this implementation, the vector of powers starts with 1, sampling the likelihood close to the posterior and incrementally sampling closer and closer to the prior as the power decreases.

Just to be safe, it is better to clear the workspace and re-load the data and model:

```
RevBayes > clear()
RevBayes > source("RevBayes_scripts/uniform_partition_model.Rev")
```

First, we create the variable containing the power posterior. This requires us to provide a model and vector of moves, as well as an output file name. The **cats** argument sets the number of power steps.

```
RevBayes > pow_p <- powerPosterior(mymodel, moves, "pow_p_uniform.out", cats
   =50)
```

We can start the power posterior by first burning in the chain and and discarding the first 10000 states.

```
RevBayes > pow_p.burnin(generations=10000,tuningInterval=1000)
```

Now execute the run with the **.run()** function:

```
RevBayes > pow_p.run(generations=1000)
```

Once the power posteriors have been saved to file, create a stepping stone sampler. This function can read any file of power posteriors and compute the marginal likelihood using stepping-stone sampling.

```
RevBayes > ss <- steppingStoneSampler(file="pow_p_uniform.out", powerColumnName="
   power", likelihoodColumnName="likelihood")
```

Compute the marginal likelihood under stepping-stone sampling using the member function **marginal()** of the **ss** variable and record the value in Table 2.

```
RevBayes > ss.marginal()
```

Path sampling is an alternative to stepping-stone sampling and also takes the same power posteriors as input.

```
RevBayes > ps <- pathSampler(file="pow_p_uniform.out", powerColumnName="power",
   likelihoodColumnName="likelihood")
```

Compute the marginal likelihood under stepping-stone sampling using the member function **marginal()** of the **ps** variable and record the value in Table 2.

```
RevBayes > ps.marginal()
```

Stop here or continue on to evaluate partitioned models...

1.5 Partitioning by Gene Region

The uniform model used in the previous section assumes that all sites in the alignment evolved under the same process described by a shared tree, branch length proportions, and parameters of the GTR+Γ substitution model. However, our alignment contains two distinct gene regions—atpB and rbcL—so we may wish to explore the possibility that the substitution process differs between these two gene regions. This requires that we first specify the data partitions corresponding to these two genes, then define an independent substitution model for each data partition.

Clear Workspace and Reload Data

```
RevBayes > clear()
```

Since we wish to avoid individually specifying each parameter of the GTR+Γ model for each of our data partitions, we can *loop* over our datasets and create vectors of nodes. To do this, we begin by creating a vector of data file names:

```
RevBayes > filenames <- v("data/conifer_atpB.nex", "data/conifer_rbcL.nex")
```

Set a variable for the number of partitions:

```
RevBayes > n_parts <- filenames.size()
```

And create a vector of data matrices called **data**:

```
for (i in 1:n_parts){
   data[i] <- readCharacterData(filenames[i])[1]
}
```

Next, we can initialize some important variables. This does require, however, that both of our alignments have the same number of species and matching tip names.

```
RevBayes > n_species <- data[1].ntaxa()
RevBayes > names <- data[1].names()
RevBayes > n_branches <- 2 * n_species - 3
```

Specify the Parameters by Looping Over Partitions

We can avoid creating unique names for every node in our model if we use a **for** loop to iterate over our partitions. Thus, we will only have to type in our entire GTR+Γ model parameters once. This will produce a vector for each of the unlinked parameters — e.g., there will be a vector of **shape** nodes where the stochastic node for the first partition (atpB) will be **shape[1]** and the stochastic node for the second partition (rbcL) will be called **shape[2]**.

```
mi <- 0 # an iterator for the move vector
for (i in 1:n_parts){
  ## index i=1 : atpB gene ##
  ## index i=2 : rbcL gene ##

  # Exchangeability rates #
  er_prior[i] <- v(1,1,1,1,1,1)
  er[i] ~ dnDirichlet(er_prior[i])
  moves[mi++] <- mvSimplexElementScale(er[i], alpha=10, tune=true, weight=3)

  # Stationary base frequencies #
  sf_prior[i] <- v(1,1,1,1)
  sf[i] ~ dnDirichlet(sf_prior[i])
  moves[mi++] <- mvSimplexElementScale(sf[i], alpha=10, tune=true, weight=2)

  # Instantaneous rate matrix (deterministic) #
  Q[i] := gtr(er[i],sf[i])

  # Gamma-dist site rates #
  shape_prior[i] <- 0.05
```

```
  shape[i] ~ dnExponential( shape_prior[i] )
  gamma_rates[i] := discretizeGamma( shape[i], shape[i], 4 )
  moves[mi++] <- mvScale(shape[i], lambda=0.8, tune=true, weight=3.0)
}
```

Uniform Topology and Branch Lengths

Our two genes evolve under different GTR rate matrices with different mean-one gamma distributions on the site rates. However, we do assume that they share a single topology and set of branch lengths.

```
# Unrooted tree topology distribution #
topology ~ dnUniformTopology(n_species, names)

# Tree topology moves #
moves[mi++] <- mvNNI(topology, weight=10.0)
moves[mi++] <- mvSPR(topology, weight=5.0)

#### Specify a prior and moves on the branch lengths ####
# Create a vector of branch-length variables using a for loop #
for (i in 1:n_branches) {
  br_lens[i] ~ dnExponential(10.0)
  moves[mi++] <- mvScale(br_lens[i],lambda=1,tune=true,weight=1)
}

# A deterministic node for the tree length #
tree_length := sum(br_lens)

# Build the tree by combining the topology with br_lens #
phylogeny := treeAssembly(topology, br_lens)
```

Putting it All Together

Since we have a rate matrix and a site-rate model for each partition, we must create a phylogenetic CTMC for each gene. Additionally, we must fix the values of these nodes by attaching their respective data matrices. These two nodes are linked by the **phylogeny** node and their log-likelihoods are added to get the likelihood of the whole DAG.

```
for (i in 1:n_parts){
  phyloSeq[i] ~ dnPhyloCTMC(tree=phylogeny, Q=Q[i], siteRates=gamma_rates[i
     ], nSites=data[i].nchar(1), type="DNA")
  phyloSeq[i].clamp(data[i])
}
```

And we can pass in a single, shared node to wrap up our model DAG:

```
RevBayes > mymodel <- model(topology)
```

Estimating the Marginal Likelihood

Now run the power posterior analysis on the two-gene model.

```
RevBayes > pow_p <- powerPosterior(mymodel, moves, file="pow_p_twogene.out",
    cats=50)
RevBayes > pow_p.burnin(generations=1000,tuningInterval=100)
RevBayes > pow_p.run(generations=1000)
```

Calculate the marginal likelihoods under stepping-stone sampling:

```
RevBayes > ss <- steppingStoneSampler(file="pow_p_twogene.out",
   powerColumnName="power", likelihoodColumnName="likelihood")
RevBayes > ss.marginal()
```

And under path sampling:

```
RevBayes > ps <- pathSampler(file="pow_p_twogene.out", powerColumnName="
   power", likelihoodColumnName="likelihood")
RevBayes > ps.marginal()
```

Record the marginal likelihoods in Table 2.

1.6 Partitioning by Codon Position and by Gene

Because of the genetic code, we often find that different positions within a codon (first, second, and third) evolve at different rates. Thus, using our knowledge of biological data, we can devise a third approach that further partitions our alignment. For this exercise, we will partition sites within the rbcL gene by codon position.

Clear Workspace and Reload Data

```
RevBayes > clear()
RevBayes > data[1] <- readCharacterData("data/conifer_atpB.nex")[1]
RevBayes > data_rbcL <- readCharacterData("data/conifer_rbcL.nex")[1]
```

Specify Data Matrices for Each Codon Position

We must now add our codon-partitions to the **data** vector that already contains the matrix for atpB in the first index. Thus, the second index will be the rbcL codon position 1. We can create this by calling the helper function **setCodonPartition()**, which is a member function of the data matrix. We are assuming that the gene is *in frame*, meaning the first column in your alignment is a first codon position. The **setCodonPartition()** function takes a single argument, the position of the alignment you wish to extract. It then returns every third column, starting at the index provided as an argument.

Before we can use the use the **setCodonPartition()** function, we must first populate the position in the **data** matrix with some sequences. Then we call the member function of **data[2]** to exclude all but the 1^{st} positions.

```
RevBayes > data[2] <- data_rbcL
RevBayes > data[2].setCodonPartition(1)
```

Assign the 2^{nd} codon positions to **data[3]**:

```
RevBayes > data[3] <- data_rbcL
RevBayes > data[3].setCodonPartition(2)
```

Assign the 3^{rd} codon positions to **data[4]**:

```
RevBayes > data[4] <- data_rbcL
RevBayes > data[4].setCodonPartition(3)
```

Now we can query the vector of data matrices to get the size, which is 4:

```
RevBayes > n_parts <- data.size()
```

And set the special variables from the data:

```
RevBayes > n_species <- data[1].ntaxa()
RevBayes > names <- data[1].names()
RevBayes > n_branches <- 2 * n_species - 3
```

Specify the Parameters by Looping Over Partitions

Setting up the GTR+Γ model is just like in the two-gene analysis, except this time **n_parts** is equal to 4, so now our vectors of stochastic nodes should all contain nodes for each of the partitions.

```
mi <- 0 # an iterator for the move vector
for (i in 1:n_parts){
  ## index i=1 : atpB gene
  ## index i=2 : rbcL gene position 1
  ## index i=3 : rbcL gene position 2
  ## index i=4 : rbcL gene position 3

  # Exchangeability rates #
  er_prior[i] <- v(1,1,1,1,1,1)
  er[i] ~ dnDirichlet(er_prior[i])
  moves[mi++] <- mvSimplexElementScale(er[i], alpha=10, tune=true, weight=3)

  # Stationary base frequencies #
  sf_prior[i] <- v(1,1,1,1)
  sf[i] ~ dnDirichlet(sf_prior[i])
  moves[mi++] <- mvSimplexElementScale(sf[i], alpha=10, tune=true, weight=2)

  # Instantaneous rate matrix (deterministic) #
  Q[i] := gtr(er[i],sf[i])

  # Gamma-dist site rates #
  shape_prior[i] <- 0.05
  shape[i] ~ dnExponential( shape_prior[i] )
  gamma_rates[i] := discretizeGamma( shape[i], shape[i], 4 )
  moves[mi++] <- mvScale(shape[i], lambda=0.8, tune=true, weight=3.0)
}
```

Uniform Topology and Branch Lengths

We are still assuming that the genes share a single topology and branch lengths.

```
# Unrooted tree topology distribution #
topology ~ dnUniformTopology(n_species, names)

# Tree topology moves #
moves[mi++] <- mvNNI(topology, weight=10.0)
moves[mi++] <- mvSPR(topology, weight=5.0)

#### Specify a prior and moves on the branch lengths ####
# Create a vector of branch-length variables using a for loop #
for (i in 1:n_branches) {
  br_lens[i] ~ dnExponential(10.0)
  moves[mi++] <- mvScale(br_lens[i],lambda=1,tune=true,weight=1)
}

# A deterministic node for the tree length #
tree_length := sum(br_lens)

# Build the tree by combining the topology with br_lens #
```

```
phylogeny := treeAssembly(topology, br_lens)
```

Putting it All Together

We must specify a phylogenetic CTMC node for each of our partition models.

```
for (i in 1:n_parts){
  phyloSeq[i] ~ dnPhyloCTMC(tree=phylogeny, Q=Q[i], siteRates=gamma_rates[i
     ], nSites=data[i].nchar(1), type="DNA")
  phyloSeq[i].clamp(data[i])
}
```

And then wrap up the DAG using the **`model()`** function:

```
RevBayes > mymodel <- model(topology)
```

Estimating the Marginal Likelihood

Sample likelihoods from the set of power posteriors:

```
RevBayes > pow_p <- powerPosterior(mymodel, moves, file="
   pow_posterior_genecodon.out", cats=50)
RevBayes > pow_p.burnin(generations=1000,tuningInterval=100)
RevBayes > pow_p.run(generations=1000)
```

Compute the stepping-stone estimate of the marginal likelihood:

```
RevBayes > ss <- steppingStoneSampler(file="pow_posterior_genecodon.out",
   powerColumnName="power", likelihoodColumnName="likelihood")
RevBayes > ss.marginal()
```

Compute the path-sampling estimate of the marginal likelihood:

```
RevBayes > ps <- pathSampler(file="pow_posterior_genecodon.out",
   powerColumnName="power", likelihoodColumnName="likelihood")
RevBayes > ps.marginal()
```

Now record the marginal likelihoods in Table 2.

1.7 Compute Bayes Factors and Select Model

Now that we have estimates of the marginal likelihood under each of our different models, we can evaluate their relative plausibility using Bayes factors. Use Table 2 to summarize the marginal log-likelihoods estimated using the stepping-stone and path-sampling methods.

Table 2: Estimated marginal likelihoods for different partition configurations*.

	Marginal lnL estimates	
Partition	*Stepping-stone*	*Path sampling*
1.4 uniform (M_1)		
1.5 moderate (M_2)		
1.6 extreme (M_3)		

*you can edit this table

Phylogenetics software programs log-transform the likelihood to avoid underflow, because multiplying likelihoods results in numbers that are too small to be held in computer memory. Thus, we must use a different form of equation 3 to calculate the ln-Bayes factor (we will denote this value $\mathcal{K}$):

$$\mathcal{K} = \ln[BF(M_0, M_1)] = \ln[\mathbb{P}(\mathbf{X} \mid M_0)] - \ln[\mathbb{P}(\mathbf{X} \mid M_1)], \tag{4}$$

where $\ln[\mathbb{P}(\mathbf{X} \mid M_0)]$ is the *marginal lnL* estimate for model M_0. The value resulting from equation 4 can be converted to a raw Bayes factor by simply taking the exponent of $\mathcal{K}$

$$BF(M_0, M_1) = e^{\mathcal{K}}. \tag{5}$$

Alternatively, you can interpret the strength of evidence in favor of M_0 using the $\mathcal{K}$ and skip equation 5. In this case, we evaluate the $\mathcal{K}$ in favor of model M_0 against model M_1 so that:

if $\mathcal{K} > 1$, then model M_0 wins
if $\mathcal{K} < -1$, then model M_1 wins.

Thus, values of $\mathcal{K}$ around 0 indicate ambiguous support.

Using the values you entered in Table 2 and equation 4, calculate the ln-Bayes factors (using $\mathcal{K}$) for the different model comparisons. Enter your answers in Table 3 using the stepping-stone and the path-sampling estimates of the marginal log likelihoods.

Once you complete Table 3, you will notice that the Bayes factor comparison indicates strong evidence in support of the highly partitioned model using both the stepping-stone and path sampling estimates of the marginal likelihoods. However, this does not mean that model M_3 is the *true* partition model. We only considered three out of the many, many possible partitions for 2,659 sites (the number of possible partitions can be viewed if you compute the 2659^{th} Bell number). Given the strength of support for the highly partitioned model, it is possible that further partitioning is warranted for these data. In particular, partitioning the dataset by codon position for both atpB *and* rbcL is an important next step for this exercise (consider taking some time on your own to test this model).

Table 3: Bayes factor calculation*.

	ln-Bayes Factor ($\mathcal{K}$)	
Model comparison	*Stepping-stone*	*Path sampling*
M_1, M_2		
M_2, M_3		
M_1, M_3		
Supported model?		

*you can edit this table

Because of the computational costs of computing marginal likelihoods and the vast number of possible partitioning strategies, it is not feasible to evaluate all of them. New methods based on nonparametric Bayesian models have recently been applied to address this problem (Lartillot and Philippe, 2004; Huelsenbeck and Suchard, 2007; Wu, Suchard and Drummond, 2013). These approaches use an infinite mixture model (the Dirichlet process; Ferguson, 1973; Antoniak, 1974) that places non-zero probability on *all* of the countably-infinite possible partitions for a set of sequences. Bayesian phylogenetic inference under these models is implemented in the program PhyloBayes (Lartillot, Lepage and Blanquart, 2009) and the subst-bma plug-in for BEAST2 (Wu, Suchard and Drummond, 2013).

Note that Bayes factors based on comparison of HM-based marginal likelihoods often *strongly* favor the most extremely partitioned mixed model. In fact, the harmonic mean estimator has been shown to provide unreliable estimates of marginal likelihoods, compared to more robust approaches (Lartillot and Philippe, 2006; Xie et al., 2011; Fan et al., 2011). Based on these studies, it is recommended that you avoid using HM-derived marginal likelihoods for Bayes factor comparisons. (The Canadian Bayesian Radford Neal says the harmonic mean is the "worst Monte Carlo method ever".)

1.8 Perform MCMC Analysis Under Preferred Model

Clear Workspace and Load the Data and Model

```
RevBayes > clear()
RevBayes > source("RevBayes_scripts/<preferred>_partition_model.Rev")
```

Specify Monitors

```
RevBayes > monitors[1] <- mnModel(filename="conifer_prefmodel_mcmc.log",
   printgen=100)
```

```
RevBayes > monitors[2] <- mnFile(filename="conifer_prefmodel_mcmc.trees",
   printgen=100, phylogeny)
```

```
RevBayes > monitors[3] <- mnScreen(printgen=10, separator = " | ",
   tree_length)
```

Initialize and Run MCMC

```
RevBayes > mymcmc <- mcmc(mymodel, monitors, moves)
```

```
RevBayes > mymcmc.burnin(generations=10000,tuningInterval=1000)
```

```
RevBayes > mymcmc.run(generations=30000)
```

1.9 Summarize and Analyze MCMC Output

```
RevBayes > treetrace <- readTreeTrace("conifer_prefmodel_mcmc.trees")
RevBayes > treetrace.summarize()
```

```
RevBayes > mapTree(treetrace,"conifer_prefmodel_MAP.tre")
```

The trees in these files are also annotated with various branch- or node-specific parameters or statistics in an extended Newick format called NHX. We can use FigTree to visualize these summary trees.

- Open the summary tree in FigTree: **`conifer_prefmodel_MAP.tre`**.
- Use the tools on the side panel to display the posterior probabilities as node labels.

Batch Mode

If you wish to run this exercise in batch mode, the files are provided for you.

You can carry out these batch commands by providing the file name when you execute the **`rb`** binary in your unix terminal (this will overwrite all of your existing run files).

- **`$ rb full_analysis.Rev`**

Useful Links

- RevBayes: https://github.com/revbayes/code
- MrBayes: http://mrbayes.sourceforge.net
- PhyloBayes: http://www.phylobayes.org
- Bali-Phy: http://www.bali-phy.org
- Tree Thinkers: http://treethinkers.org

Questions about this tutorial can be directed to:

- Tracy Heath (email: tracyh@berkeley.edu)
- Michael Landis (email: mlandis@berkeley.edu)
- Sebastian Höhna (email: sebastian.hoehna@gmail.com)

Version dated: August 26, 2014

Relevant References

Antoniak CE. 1974. Mixtures of Dirichlet processes with applications to non-parametric problems. Annals of Statistics. 2:1152–1174.

Brandley MC, Schmitz A, Reeder TW. 2005. Partitioned Bayesian analyses, partition choice, and the phylogenetic relationships of scincid lizards. Systematic Biology. 54:373–390.

Brown JM, Lemmon AR. 2007. The importance of data partitioning and the utility of Bayes factors in Bayesian phylogenetics. Systematic Biology. 56:643–655.

Fan Y, Wu R, Chen MH, Kuo L, Lewis PO. 2011. Choosing among partition models in Bayesian phylogenetics. Molecular Biology and Evolution. 28:523–532.

Ferguson TS. 1973. A Bayesian analysis of some nonparametric problems. Annals of Statistics. 1:209–230.

Friel N, Pettitt AN. 2008. Marginal likelihood estimation via power posteriors. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 70:589–607.

Gelman A, Meng XL. 1998. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. Statistical science. pp. 163–185.

Huelsenbeck JP, Suchard M. 2007. A nonparametric method for accommodating and testing across-site rate variation. Systematic Biology. 56:975–987.

Jeffreys H. 1961. Theory of Probability. Oxford: Oxford University Press.

Kass RE, Raftery AE. 1995. Bayes factors. Journal of the American Statistical Association. 90:773–795.

Lartillot N, Lepage T, Blanquart S. 2009. Phylobayes 3: a bayesian software package for phylogenetic reconstruction and molecular dating. Bioinformatics. 25:2286.

Lartillot N, Philippe H. 2004. A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. Molecular Biology and Evolution. 21:1095–1109.

Lartillot N, Philippe H. 2006. Computing Bayes factors using theromodynamic integration. Systematic Biology. 55:195–207.

Lavine M, Schervish MJ. 1999. Bayes factors: What they are and what they are not. American Statistician. 53:119–122.

McGuire JA, Witt CC, Altshuler DL, Remsen JV. 2007. Phylogenetic systematics and biogeography of hummingbirds: Bayesian and maximum likelihood analyses of partitioned data and selection of an appropriate partitioning strategy. Systematic Biology. 56:837–856.

Nylander JAA, Ronquist F, Huelsenbeck JP, Aldrey JLN. 2004. Bayesian phylogenetic analysis of combined data. Systematic Biology. 53:47–67.

Rannala B, Yang Z. 1996. Probability distribution of molecular evolutionary trees: A new method of phylogenetic inference. Journal of Molecular Evolution. 43:304–311.

Robert CP, Casella G. 2002. Monte Carlo Statistical Methods. New York: Springer.

Rodrigue N, Philippe H, Lartillot N. 2008. Uniformization for sampling realizations of Markov processes: Applications to Bayesian implementations of codon substitution models. Bioinformatics. 24:56–62.

Ronquist F, Huelsenbeck JP. 2003. Mrbayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics. 19:1572–1574.

Ronquist F, van der Mark P, Huelsenbeck JP. 2009. Bayesian analysis of molecular evolution using MrBayes. In: Lemey P, Salemi M, Vandamme AM, editors, The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing, Second Edition. Cambridge University Press, pp. 1–1.

Rubinstein R. 1981. Simulation and the Monte Carlo Method. John Wiley & Sons, Inc. New York, NY, USA.

Simon D, Larget B. 2001. Bayesian analysis in molecular biology and evolution (BAMBE). http://www.mathcs.duq.edu/larget/bambe.html.

Smith A, Roberts G. 1993. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. Journal of the Royal Statistical Society. Series B (Methodological). 55:3–23.

Suchard M, Weiss R, Sinsheimer J. 2005. Models for estimating bayes factors with applications to phylogeny and tests of monophyly. Biometrics. 61:665–673.

Suchard MA, Weiss RE, Sinsheimer JS. 2001. Bayesian selection of continuous-time Markov chain evolutionary models. Molecular Biology and Evolution. 18:1001–1013.

Sukumaran J, Holder MT. 2010. DendroPy: A Python library for phylogenetic computing. Bioinformatics. 26:1569–1571.

Verdinelli I, Wasserman L. 1995. Computing Bayes factors using a generalization of the Savage Dickey density ratio. Journal of the American Statistical Association. 90:614–618.

Wong KM, Suchard MA, Huelsenbeck JP. 2008. Alignment uncertainty and genomic analysis. Science. 319:473–476.

Wu CH, Suchard MA, Drummond AJ. 2013. Bayesian selection of nucleotide substitution models and their site assignments. Molecular Biology and Evolution. 30:669–688.

Xie W, Lewis PO, Fan Y, Kuo L, Chen MH. 2011. Improving marginal likelihood estimation for Bayesian phylogenetic model selection. Systematic Biology. 60:150–160.

Yang Z. 2007. Fair-balance paradox, star-tree paradox, and Bayesian phylogenetics. Molecular biology and evolution. 24:1639.

Yang Z, Rannala B. 1997. Bayesian phylogenetic inference using DNA sequences: A Markov chain Monte Carlo method. Molecular Biology and Evolution. 14:717–724.

Yang Z, Rannala B. 2005. Branch-length prior influences Bayesian posterior probability of phylogeny. Systematic Biology. 54:455–470.