Phylogenetic Inference using RevBayes

Model adequacy testing using Posterior Prediction

Sebastian Höhna and Lyndon M. Coghill

1 Overview

This tutorial demonstrates some general principles of Bayesian model comparison, which is based on estimating the marginal likelihood of competing models and then comparing their relative fit to the data using Bayes factors. We consider the specific case of calculating Bayes factors to select among different substitution models.

1.1 Requirements

We assume that you have read and hopefully completed the following tutorials:

- RB_Getting_Started
- RB_Data_Tutorial
- RB_CTMC_Tutorial

This means that we will assume that you know how to execute and load data into `RevBayes`, are familiar with some basic commands, and know how to perform an analysis of a single-gene dataset (assuming an unconstrained/unrooted tree).

2 Data and files

We provide several data files that we will use in this tutorial. Of course, you may want to use your own dataset instead. In the **`data`** folder, you will find the following files

- **`primates_cytb.nex`**: Alignment of the *cytochrome b* subunit from 23 primates representing 14 of the 16 families (*Indriidae* and *Callitrichidae* are missing).

3 Introduction

For most sequence alignments, several (possibly many) substitution models of varying complexity are plausible *a priori*. We therefore need a way to objectively identify the model that balances estimation bias and inflated error variance associated with under- and over-parameterized models, respectively. Increasingly, model selection is based on *Bayes factors* (*e.g.,* **?????**), which involves first calculating the marginal likelihood of each candidate model and then comparing the ratio of the marginal likelihoods for the set of candidate models.

Given two models, M_0 and M_1, the Bayes-factor comparison assessing the relative fit of each model to 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 (**?**):

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

Note that interpreting Bayes factors involves some subjectivity. That is, it is up to *you* to decide the degree of your belief in M_0 relative to M_1. Despite the absence of an absolutely objective model-selection threshold, we can refer to the scale (outlined by **?**) that provides a "rule-of-thumb" for interpreting these measures (Table **??**).

Table 1: The scale for interpreting Bayes factors by Harold **?**.

Strength of evidence	$BF(M_0, M_1)$	$\log(BF(M_0, M_1))$	$\log_{10}(BF(M_0, M_1))$
Negative (supports M_1)	< 1	< 0	< 0
Barely worth mentioning	1 to 3.2	0 to 1.16	0 to 0.5
Substantial	3.2 to 10	1.16 to 2.3	0.5 to 1
Strong	10 to 100	2.3 to 4.6	1 to 2
Decisive	> 100	> 4.6	> 2

For a detailed description of Bayes factors see **?**

Unfortunately, it is generally not possible to directly calculate the posterior odds to prior odds ratio. 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 (this may be familiar to you as the denominator of Bayes Theorem, which is variously referred to as the *model evidence* or *integrated likelihood*). Formally, the marginal likelihood is the probability of the observed data ($\mathbf{X}$) under a given model (M_i) that is averaged over all possible values of the parameters of the model (θ_i) with respect to the prior density on θ_i

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

This makes it clear that more complex (parameter-rich) models are penalized by virtue of the associated prior: each additional parameter entails integration of the likelihood over the corresponding prior density.

If you refer back to equation **??**, 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 **??**), 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 (**??**) and path sampling estimators (**??**).

3.1 Substitution Models

The models we use here are equivalent to the models described in the previous exercise on substitution models (continuous time Markov models). To specify the model please consult the previous exercise. Specifically, you will need to specify the following substitution models:

- Jukes-Cantor (JC) substitution model (**?**)
- Hasegawa-Kishino-Yano (HKY) substitution model (**?**)
- General-Time-Reversible (GTR) substitution model (**?**)
- Gamma (+G) model for among-site rate variation (**?**)
- Invariable-sites (+I) model (**?**)

3.2 Estimating the Marginal Likelihood

We will estimate the marginal likelihood of a given model using a ‘stepping-stone’ (or ‘path-sampling’) algorithm. These algorithms are similar to the familiar MCMC algorithms, which are intended to sample from (and estimate) the joint posterior probability of the model parameters. Stepping-stone algorithms are like a series of MCMC simulations that iteratively sample from a specified number of discrete steps between the posterior and the prior probability distributions. The basic idea is to estimate the probability of the data for all points between the posterior and the prior—effectively summing the probability of the data over the prior probability of the parameters to estimate the marginal likelihood. Technically, the steps correspond to a series of **`powerPosteriors()`**: a series of numbers between 1 and 0 that are iteratively applied to the posterior. When the posterior probability is raised to the power of 1 (typically the first stepping stone), samples are drawn from the (untransformed) posterior. By contrast, when the posterior probability is raised to the power of 0 (typically the last stepping stone), samples are drawn from the prior. To perform a stepping-stone simulation, we need to specify (1) the number of stepping stones (power posteriors) that we will use to traverse the path between the posterior and the prior (*e.g.,* we specify 50 or 100 stones), (2) the spacing of the stones between the posterior and prior (*e.g.,* we may specify that the stones are distributed according to a beta distribution), (3) the number of samples to (and thinning) of samples to be drawn from each stepping stone, and (4) the direction we will take (*i.e.,* from the posterior to the prior or vice versa).

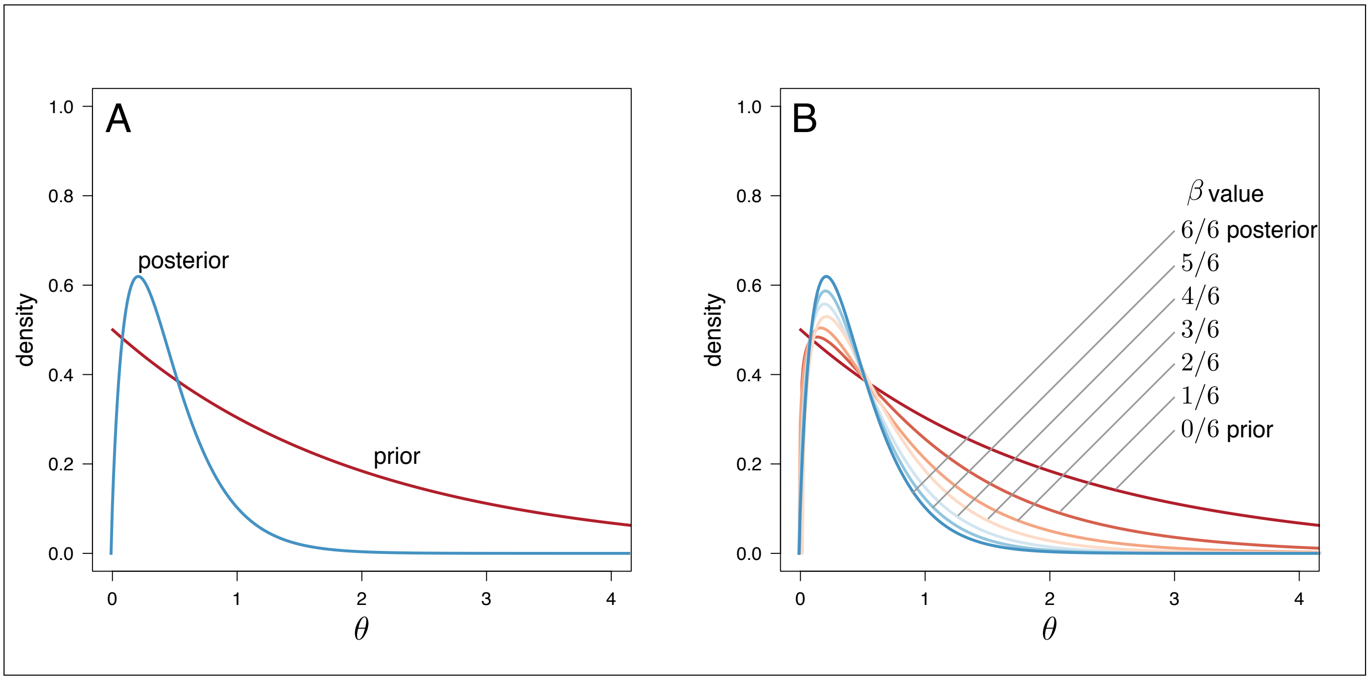

Figure 1: Estimating marginal likelihoods using stepping-stone simulation. Estimating the marginal likelihood involves integrating the likelihood of the data over the entire prior probability density for the model parameters. MCMC algorithms target the posterior probability density, which is typically concentrated in a small region of the prior probability density (A). Accordingly, standard MCMC simulation cannot provide unbiased estimates of the marginal likelihood because it will typically fail to explore most of the prior density. (B) Stepping-stone algorithms estimate the marginal likelihood by means of a series of MCMC-like simulations, where the likelihood is iteratively raised to a series of powers, effectively forcing the simulation to more fully explore the prior density of the model parameters. Here, six uniformly spaced stones span the posterior, where the power posterior is $\beta = 6/6 = 1$, to the prior, where the power posterior is $\beta = 0/6 = 0$.

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 (if you did not just restart `RevBayes`)

```
clear()
```

Now set up the model as in the previous exercise. You should start with the simple Jukes-Cantor substitution model. Setting up the model requires:

1. Loading the data and retrieving useful variables about it (*e.g.*, number of sequences and taxon names).
2. Specifying the instantaneous-rate matrix of the substitution model.
3. Specifying the tree model including branch-length variables.
4. Creating a random variable for the sequences that evolved under the `PhyloCTMC`.

5. Clamping the data.
6. Creating a model object.
7. Specifying the moves for parameter updates.

The following procedure for estimating marginal likelihoods is valid for any model in RevBayes. You will need to repeat this later for other models. First, we create the variable containing the power-posterior analysis. This requires that we provide a model and vector of moves, as well as an output file name. The **cats** argument sets the number of stepping stones.

```
pow_p = powerPosterior(mymodel, moves, monitors, "model1.out", cats=50)
```

We can start the power-posterior analysis by first burning in the chain and and discarding the first 10000 states. This will help ensure that analysis starts from a region of high posterior probability, rather than from some random point.

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

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

```
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.

```
ss = steppingStoneSampler(file="model1.out", powerColumnName="power", likelihoodColumnName="
    likelihood")
```

These commands will execute a stepping-stone simulation with 50 stepping stones, sampling 1000 states from each step. Compute the marginal likelihood under stepping-stone sampling using the member function **marginal()** of the **ss** variable and record the value in Table **??**.

```
ss.marginal()
```

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

```
ps = pathSampler(file="model1.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 **??**.

```
ps.marginal()
```

→ As an example we provide the file **RevBayes_scripts/marginalLikelihood_JukesCantor.Rev**.

3.3 Exercises

- Compute the marginal likelihoods of the *cytb* alignment for the following substitution models:
  - Jukes-Cantor (JC) substitution model
  - Hasegawa-Kishino-Yano (HKY) substitution model
  - General-Time-Reversible (GTR) substitution model
  - GTR with gamma distributed-rate model (GTR+G)
  - GTR with invariable-sites model (GTR+I)
  - GTR+I+G model
- Enter the marginal likelihood estimate for each model in the corresponding cell of Table **??**.
- Repeat the above marginal likelihood analyses for the *mt-COX2 gene* and enter results in Table **??**.
- Which is the best fitting substitution model?

Table 2: Estimated marginal likelihoods for different substitution models for the cytb alignment*.

	Marginal lnL estimates	
Substitution Model	*Stepping-stone*	*Path sampling*
JC (M_1)		
HKY (M_2)		
GTR (M_3)		
GTR+Γ (M_4)		
GTR+I (M_5)		
GTR+Γ+I (M_6)		

*you can edit this table

4 Computing Bayes Factors and Model Selection

Now that we have estimates of the marginal likelihood for each of our the candidate substitution models, we can evaluate their relative fit to the datasets using Bayes factors. Phylogenetic programs log-transform the likelihood values to avoid underflow: multiplying likelihoods (numbers < 1) generates numbers that are too small to be held in computer memory. Accordingly, we need to use a different form of equation **??** 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 **??** 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 directly interpret the strength of evidence in favor of M_0 in log space by comparing the values of $\mathcal{K}$ to the appropriate scale (Table refbftable, second column). In this case, we evaluate $\mathcal{K}$ in favor of model M_0 against model M_1 so that:

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

Thus, values of $\mathcal{K}$ around 0 indicate that there is no preference for either model.

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

Table 3: Bayes factor calculation*.

Model comparison	M_1	M_2	M_3	M_4	M_5	M_6
M_1	-					
M_2		-				
M_3			-			
M_4				-		
M_5					-	
M_6						-

*you can edit this table

5 For your consideration...

In this tutorial you have learned how to use `RevBayes` to assess the *relative* fit of a pool of candidate substitution models to a given sequence alignment. Typically, once we have identified the "best" substitution model for our alignment, we would then proceed to use this model for inference. Technically, this is a decision to condition our inferences on the selected model, which explicitly assumes that it provides a reasonable description of the process that gave rise to our data. However, there are several additional issues to consider before proceeding along these lines, which we briefly mention below.

5.1 Accommodating Process Heterogeneity

In the analyses that we performed in this tutorial, we assumed that all sites in an alignment evolved under an identical substitution process. It is well established that this assumption is frequently violated in real datasets. Various aspects of the substitution process—the stationary frequencies, exchangeability rates, degree of ASRV, etc.—may vary across sites of our sequence alignment. For example, the nature of the substitution process may vary between codon positions of protein-coding genes, between stem and loop regions of ribosomal genes, or between different gene and/or genomic regions. It is equally well established that failure to accommodate this *process heterogeneity*—variation in the nature of the substitution process across the alignment—will cause biased estimates of the tree topology, branch lengths and other phylogenetic model parameters. We can accommodate process heterogeneity by adopting a *mixed-model* approach, where two or more subsets of sites are allowed to evolve under distinct substitution processes. We will demonstrate how to specify—and select among—alternative mixed models using `RevBayes` in a separate tutorial, RB_PartitionedData_Tutorial.

5.2 Assessing Model Adequacy

In this tutorial, we used Bayes factors to assess the fit of various substitution models to our sequence data, effectively establishing the *relative* rank of the candidate models. Even if we have successfully identified the very best model from the pool of candidates, however, the preferred model may nevertheless be woefully inadequate in an *absolute* sense. For this reason, it is important to consider *model adequacy*: whether a given model provides a reasonable description of the process that gave rise to our sequence data. We can assess the absolute fit of a model to a given dataset using *posterior predictive simulation.* This approach is based on the following premise: if the candidate model provides a reasonable description of the process that gave rise to our dataset, then we should be able to generate data under this model that resemble our observed data. We will demonstrate how to assess model adequacy using `RevBayes` in a separate tutorial, RB_ModelAdequacy_Tutorial.

5.3 Accommodating Model Uncertainty

Even when we have carefully assessed the relative and absolute fit of candidate models to our dataset, it may nevertheless be unwise to condition our inference on the best model. Imagine, for example, that there are several (possibly many) alternative models that provide a similarly good fit to our given dataset. In such scenarios, conditioning inference on *any* single model (even the 'best') ignores uncertainty in the chosen model, which will cause estimates to be biased. This is the issue of *model uncertainty.* The Bayesian framework provides a natural approach for accommodating model uncertainty by means of *model averaging*; we simply adopt the perspective that models (like standard parameters) are random variables, and integrate the inference over the distribution of candidate models. We will demonstrate how to accommodate model uncertainty using `RevBayes` in a separate tutorial, RB_ModelAveraging_Tutorial.

Version dated: September 22, 2016