

# Model adequacy using BEAST2

## Assessing clock and substitution models

*David Duchêne*

## 1 Background

This tutorial will guide you through methods to assess model adequacy in BEAST v2.4.2. In common practice, evolutionary models are selected based on their statistical fit *relative to each other*. This might be sufficient in some cases, but there is a risk that all of the candidate models lead to inferences that poorly represent the true evolutionary process. Indeed, even the most complex or best fitting model from a set of candidates can produce highly erroneous estimates of parameters of interest. In this tutorial we will explore methods to investigate the absolute merits of the model. This kind of assessment of the absolute performance of models is also known as model checking or assessment of model adequacy or plausibility.

Before starting the tutorial, it is important that you understand the methods used in Bayesian inference for assessing model adequacy. A typical assessment of model adequacy is done by comparing a test statistic, calculated from the empirical data set, with the values of the statistic calculated from a large number of data sets simulated under the model. The simulated data from the model are often referred to as posterior predictive simulations (PPS), and they represent future or alternative data sets under the candidate model. The test statistic should be informative about the assumptions of the model in question.

A large number of test statistics have been proposed to assess the components of phylogenetic analyses. In this tutorial we will investigate two test statistics, one for assessing the substitution model, and one for assessing the priors on rates and times used for molecular dating (Figure 1).

- The multinomial likelihood, which is the likelihood of the data under a model with only a single general assumption: that substitution events are independent and identically distributed. This statistic can be used to assess overall substitution model fit. Note that sites with missing data or indels should not be included when estimating the multinomial likelihood.
- The *A* index assesses the power of the molecular-clock model to estimate the number of substitutions across branches, assuming a tree topology and an adequate substitution model. We will go through this statistic in more detail during this exercise.

## 2 Programs used in this Exercise

- BEAST v2.4.2.
- R programming environment.
- R packages *ape* and *phangorn*. Before starting the exercise, install these packages by typing the following in your R console:

```
install.packages("phangorn", dependencies = T)
```

### 3 Run the empirical data

In the `data` folder you will find a simulated sequence alignment with 2000 nucleotides in nexus format (`a1.1.nex`) and a Yule-process chronogram in newick format (`chron.tre`) with 50 taxa. The alignment was simulated along the chronogram under a Jukes-Cantor substitution model and an **uncorrelated log-normal clock model**. You will also find an XML file (`sim.1.xml`) in the `xml` folder to run BEAST 2 for this data set. The settings in this XML file include a Jukes-Cantor substitution model, a **strict clock**, a birth-death tree prior, and a normally distributed root calibration with standard deviation of 5 and an offset of 50.

Importantly, this XML file has been modified directly such that the starting tree has been provided and it remains fixed throughout the run. Continue in this section if you wish to learn how to modify the XML, or proceed to the next section to assess model adequacy.

Load the data `a1.1.nex` to BEAUti, setting the model as described above, and saving the XML file. Then open the XML file in your preferred text editor, and add a `tree` statement as follows above line with the `<run ...` statement.

```
<tree id="Tree.t:a1" spec='beast.util.TreeParser' newick='((A, B), (C, D), E);' taxa="@a1"/>
```

Make sure that the newick tree is correct (you will find this in `chron.tre` in the `data` folder). Also make sure that the names of the tree and the taxa are correct. You will find the name of your tree after `"Tree.t:` elsewhere in the XML. The name of the taxa is an `@` followed by your data `id` as shown immediately above the alignment.

Now remove the `tree` statement, which is inside the `state` and looks like the following.

```
<tree id="Tree.t:treename" name="stateNode">
  <taxonset id="TaxonSet.alignmentname" spec="TaxonSet">
    <alignment idref="alignmentname"/>
  </taxonset>
</tree>
```

In the place of that `tree` statement, type an `input` statement, which will lead to a new `statenode` as follows.

```
<input name='stateNode' idref='Tree.t:treename' />
```

Once again, make sure that the tree name is correct.

Now remove the whole random tree initialiser, which begins and ends with the following lines.

```
<init id="RandomTree.t:treename" ...
...
</init>
```

Lastly, remove the lines (or set the weights to 0) of the operators on the tree topology. These include the lines that have any of `SubtreeSlide`, `Narrow`, `Wide`, and `WilsonBalding`. You might find these words together with the tree model selected (in this case `BirthDeath`).

Now run the XML file using BEAST 2. The output of this run can also be found in the folder `precooked_runs`.

## 4 One step model assessment

In this section we will run substitution and clock model assessment, using the output from the BEAST 2 analysis that we ran in the previous section. We will perform a one step model assessment in this section, and in following section we will explore in detail each of the steps undergone in this assessment. The results for this example can also be found in the folder `precooked_runs`. In that folder you will also find the results for the BEAST 2 run of `sim1.xml`.

Begin by opening R. The following will set the working directory to the scripts folder, and then source all the functions in the folder.

```
setwd('[INSERT THE PATH TO SCRIPTS FOLDER]')
for(i in dir()) source(i)
```

Next, we set the directory to `precooked_runs`, and run the function `adeq()`. The arguments for this function are the posterior of trees in nexus format, the log file, the alignment in nexus format, and the number of posterior predictive simulations to be performed. You can use different path arguments if you ran your own BEAST 2 analyses in another folder.

```
setwd('../precooked_runs')
clock_adequacy_example <- adeq(trees.file = "sim1.trees", log.file = "sim1.log", empdat.file =↵
  "../data/al.1.nex", Nsim = 100)
names(clock_adequacy_example)
```

The contents of `clock_adequacy_example` should appear after the final line of code, and are each of the components that can be used to assess clock and substitution model adequacy.

The `clock_adequacy_example` object should have the same contents as object “assessment\_provided” in the file `results.Rdata`:

```
load('results.Rdata')
names(assessment_provided)
```

The following section describes the steps for assessing model adequacy in detail.

## 5 Steps for assessing model adequacy

### 5.1 Reading the runs and simulating data

We will study the code in R and Figure 1 in detail. This section is mainly for reading and discussion, and is aimed for you to cement the steps required for assessing model adequacy.

Open the `adeq.R` file in a text editor of your preference and you should see the following:

```
adeq <- function(trees.file, log.file, empdat.file, Nsim = 100){
  empdat <- as.phyDat(as.DNABin(read.nexus.data(empdat.file)))
  seqlen <- ncol(as.matrix(as.DNABin(empdat)))
  tree.topo <- read.nexus(trees.file)[[1]]
  sims <- make.pps.als(trees.file, log.file, Nsim, seqlen)
  sims <- make.pps.tr(sims, empdat, tree.topo)
  bls <- compile.results(sims)
  return(bls)
}
```

The analysis of empirical data appears in blue in Figure 2. We then need to read the posterior trees and parameter estimates of the BEAST 2 analysis. The R package *phangorn* allows us to take these data and run the posterior predictive simulations, which is shown in green in Figure 1. You will find the code to read data from the posterior and for simulating genetic alignments in `make.pps.als`. This script identifies the model being assessed and simulates data accordingly. The input required in this step includes:

- The paths of the posterior files for your analysis.
- The number of simulations you want to perform.
- The sequence length (number of sites in your alignment).

If you are interested in how we read and simulate data in R, you can investigate the `make.pps.als` code. The following is the line in `adeq.R` in question:

```
sims <- make.pps.als(trees.file, log.file, Nsim, seqlen)
```

If the input file has a greater number of samples than the number of simulations requested, this will randomly select samples from the posterior. This is a good moment to discuss or read about how an alignment can be simulated using data from the posterior.



Figure 1: Two of the existing approaches to using posterior predictive simulations to assess model adequacy in Bayesian phylogenetics. (a) One group of methods use characteristics of the data for model assessment, like the multinomial likelihood or the GC content. (b) Another method can assess clock models using estimates from clock-free methods. Under this approach, the number of substitutions per site expected along each branch under the clock hierarchical model are compared with those inferred in a clock-free analysis of the empirical data.

### 5.1.1 Calculate test statistics

Once we have simulated data sets, we can calculate the test statistics. The function `make.pps.trs` takes the assumed tree, substitution model, and the empirical and simulated data sets. The function estimates phylogenetic branch lengths for the empirical data set and each of the simulated data sets. In this step we also estimate the multinomial likelihood test statistic for the empirical data and each of the simulated data sets.

```
sims <- make.pps.tr(sims, empdat, tree.topo)
```

The output of this function is what we need for model assessment: the test statistics for the empirical data, and the distribution of test statistics for the simulated data sets.

### 5.1.2 Calculate *P*-values

We can now compare the test statistic for the empirical data and each of the simulated data sets, which is the step shown in red in Figure 1. The most common way to do this is to calculate the tail area probability, which is the number of simulations with a test statistic greater than the value for the empirical data.

```
bls <- compile.results(sims)
```

This function will provide the test statistics for simulations, as well as *P*-values for each of the test statistics. Following practice from frequentist statistics, we can consider the model to be inadequate if the *P*-value for a given test statistic is below 0.05. Importantly, the assessment of the clock model allows us to identify the branches for which the molecular clock model can estimate the number of substitutions. We will explore the interpretation of these data in the following section.

## 6 Interpreting substitution model assessment

It is strongly recommended to use qualitative checks of models using graphical analyses. This section uses the results in `precooked_runs/results.Rdata` to graph different components for assessing clock model adequacy using posterior predictive simulations.

We will first visualise the results for assessing substitution model adequacy. The following code makes a histogram of the distribution of the multinomial likelihood for the PPS data, and will show the position of the value for empirical data on this distribution.

```
hist(assessment_provided[[8]],
     xlim = c(min(assessment_provided[[8]]) - sd(assessment_provided[[8]]),
              max(assessment_provided[[8]]) + sd(assessment_provided[[8]])),
     main = "", xlab = "Multinomial likelihood")

abline(v = assessment_provided[[7]], col = 2, lwd = 3)
```

Your plot will be identical or very similar to Figure 2. When assessing model adequacy, we consider the model to be an adequate representation of the evolutionary process if the test statistic for the empirical data is a typical value arising from the model. The multinomial likelihood for the empirical data falls inside the distribution of values for simulated data (Figure 2), so our substitution model is a good description of

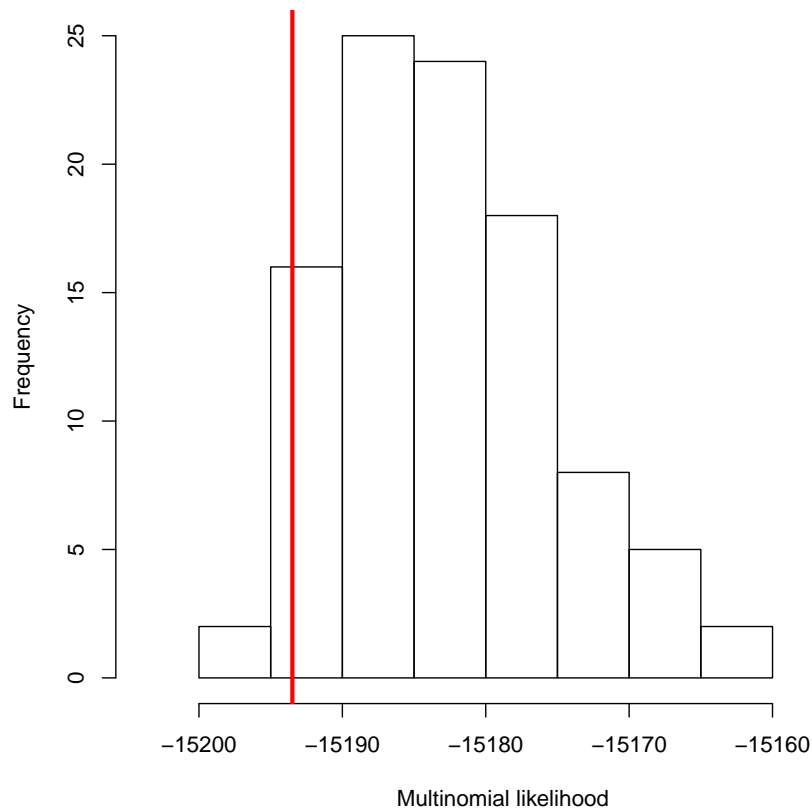


Figure 2: Distribution of PPS multinomial likelihood values with the value of the test statistic for the empirical data shown as a vertical line in red.

the process that generated the data. This is unsurprising, since our empirical data were actually generated under the same substitution model used for analysis!

This result can also be observed in the  $P$ -value for the multinomial likelihood in R:

```
assessment_provided[9]
```

### 6.0.1 Interpreting clock model assessment

The following script shows a simple example to explore the branch-wise posterior predictive  $P$ -values. We will first load the tree. In this example we will use the original tree provided, but usually the tree with the median posterior branching times would be appropriate. We will colour the branches with the best accuracy in blue, and the branches that have the lowest accuracy in green (Figure 3).

```
tr <- read.tree("../data/chrono.tre")
plot(tr, edge.col = rainbow(length(assessment_provided$branch_wise_pppvalues),
  start = 2/6, end = 4/6)[rank(assessment_provided$branch_wise_pppvalues)],
  edge.width = 6, cex = 1.5)
```

```
edgelabels(assessment_provided$branch_wise_pppvalues, bg = "white", cex = 1.5, frame = "none")
```

The values along each branch indicate the proportion of simulations in which the branch-length was greater than the length estimated using the empirical data. The expected value under the model is 0.5. If this value is 0 branch-lengths are being underestimated with respect to the model. Similarly, if the value is 1 branch-lengths are being overestimated with respect to the model.

You can also investigate the *A* index:

```
assessment_provided[4]
```

This index is the proportion of branches in the tree for which the branch-wise posterior predictive *P*-values are inside the central 95 percent of the distribution. The rates and times models can be considered adequate when the *A* index is high.

You might find it surprising that the *A* index in these data is not so close to 1. The reason for this might be evident when you open the data alignment in a text editor. These data have a very large amount of variation, and have possibly undergone substantial substitutional saturation. For comparison, you might want to repeat all of the analyses, or explore the provided results for the second data set provided, `a1.2.nex`. This data set evolved through the same tree but with lower rates of molecular evolution.

The following script shows a simple example to explore the branch-wise length deviation. This metric is a proxy for the difference between the branch-length estimate using empirical data and the mean branch-length estimate using posterior predictive simulations. To make the values comparable across branches, the difference between the empirical branch-length and mean PPS branch-length has been divided by the empirical branch length. If this value is close to zero, the priors for times and molecular evolutionary rates can be considered adequate. We apply the same colouring system as the plot above, but note that in the case of branch length deviation larger numbers indicate greater deviation from the empirical branch length, and therefore lower accuracy (Figure 4).

```
plot(tr, edge.col = rev(rainbow(length(assessment_provided$branch_length_deviation),
  start = 2/6, end = 4/6)[rank(assessment_provided$branch_length_deviation)]),
  edge.width = 6, cex = 1.5)
edgelabels(round(assessment_provided$branch_length_deviation, 2), bg = "white", cex = 1.5,
  frame = "none")
```

Note that in this simple method to graph the results, the branches in the two plots above have been coloured by their rank, rather than their magnitude.



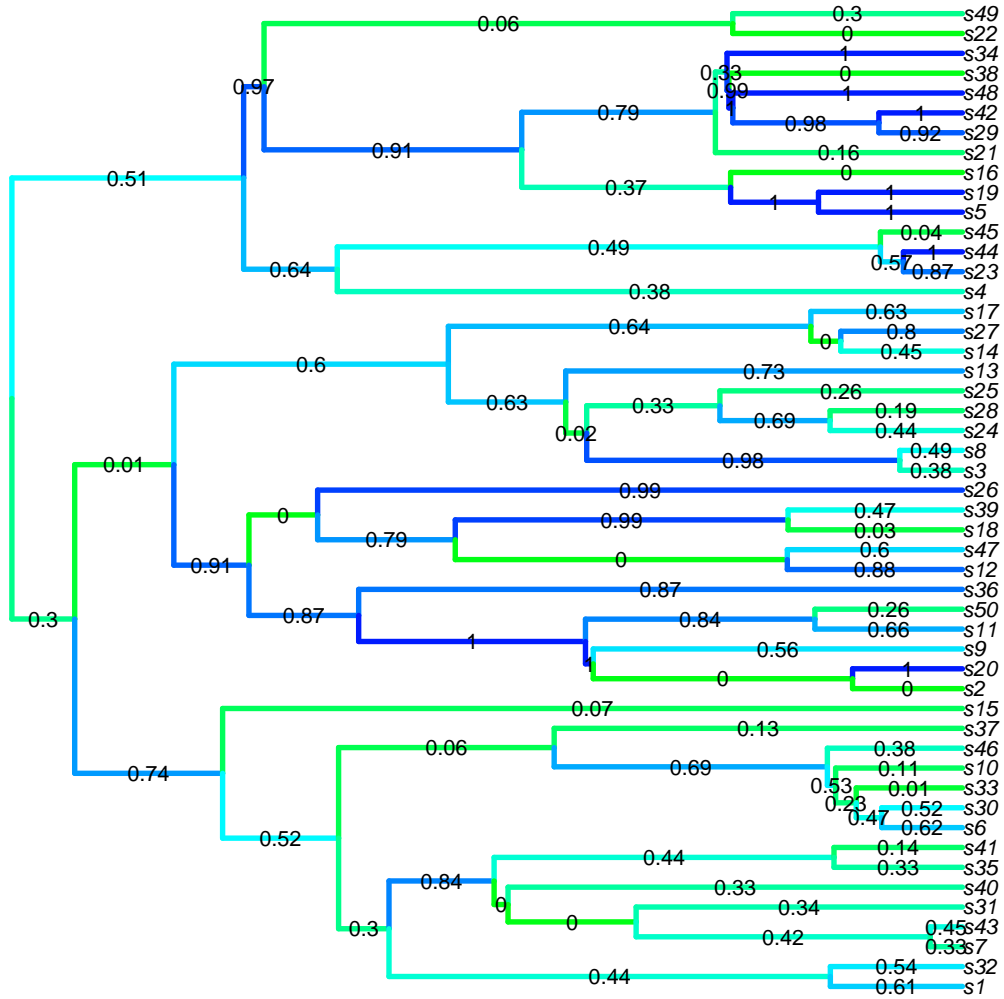


Figure 3: Estimated chronogram with branches coloured by their clock adequacy P-value.

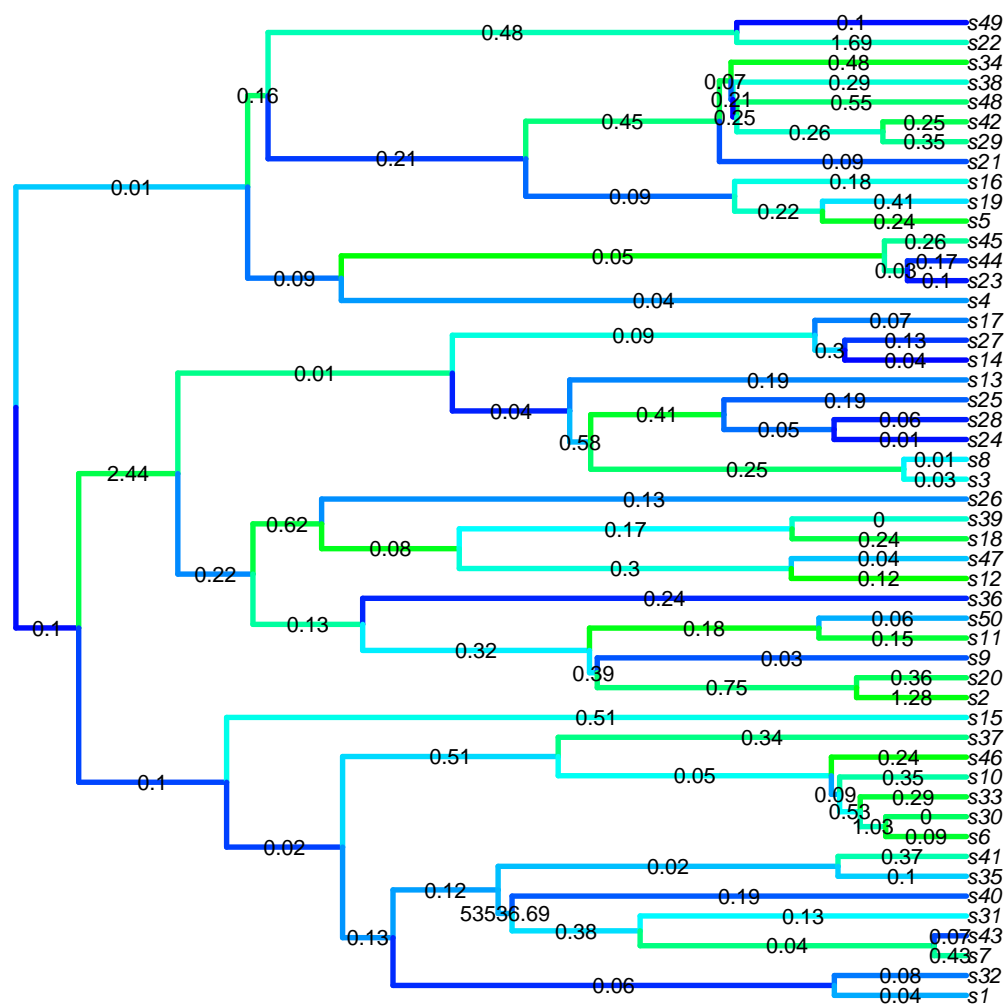


Figure 4: Estimated chronogram with branches coloured by their deviation between the empirical and simulated lengths.



This tutorial was written by David Duchêne for [Taming the BEAST](#) and is licensed under a [Creative Commons Attribution 4.0 International License](#).

Version dated: February 5, 2017

## Relevant References

- Bollback, JP. 2002. Bayesian model adequacy and choice in phylogenetics. *Molecular Biology and Evolution* 19: 1171.
- Duchêne, DA, S Duchêne, EC Holmes, and SY Ho. 2015. Evaluating the adequacy of molecular clock models using posterior predictive simulations. *Molecular Biology and Evolution* 32: 2986.
- Goldman, N. 1993. Statistical tests of models of dna substitution. *Journal of Molecular Evolution* 36: 182–198.