Model adequacy using BEAST v2.4.2

Assessing clock and substitution models

David A. Duchene

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 dating 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 the practical exercises.

2 Programs used in this Exercise

- BEAST2.
- R programming environment.
  - R packages *ape* and *phangorn*.

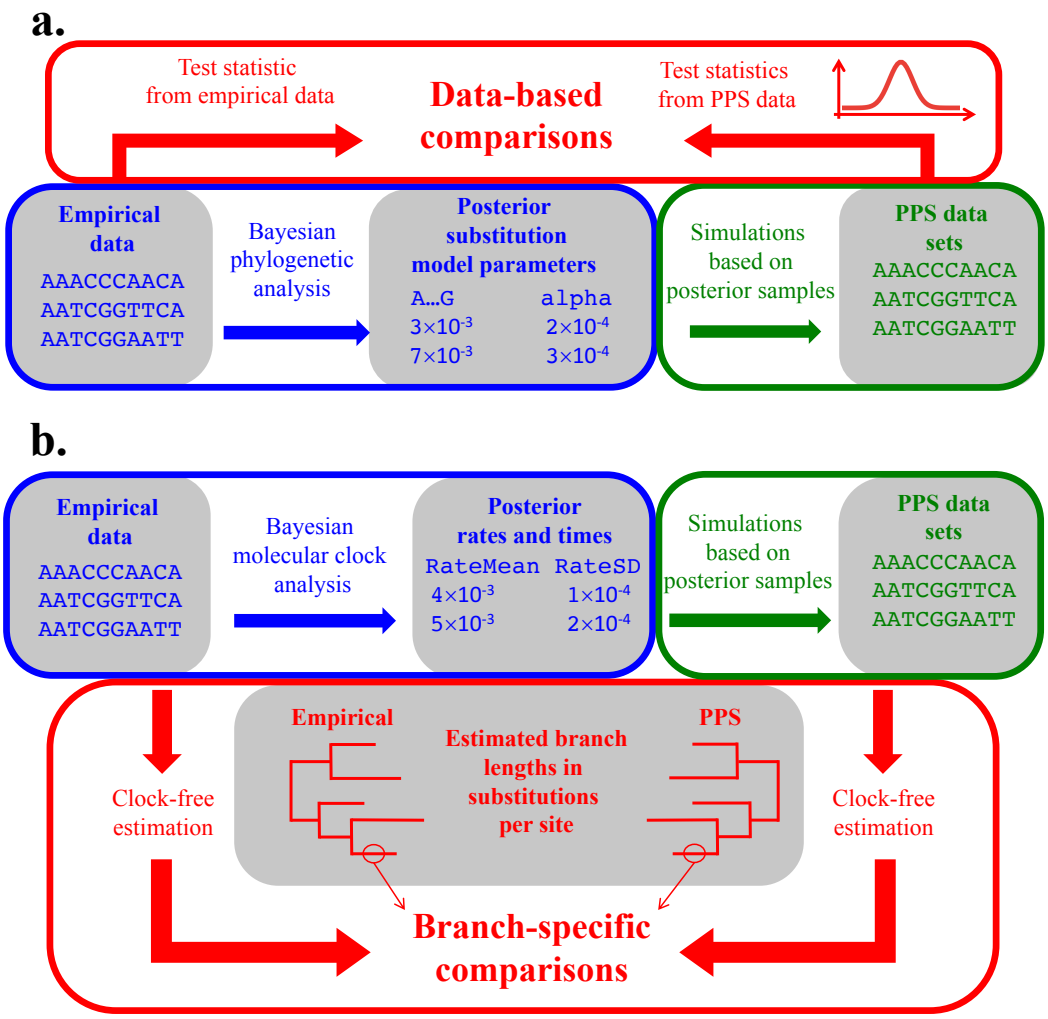

Figure 1: Two of the existing approaches to using posterior predictive simulations to assess model adequacy in Bayesian phylogenetics. (a) One group of methods use characteristics of the data for model assessment, like the multinomial likelihood or the GC content. (b) One method exists that can assess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

3 Steps for assessing model adequacy

3.1 Step 1: run the empirical data

In the data folder you will find a simulated sequence alignment with 2000 nucleotides in nexus format (al.nex) and a Yule-process chronogram in newick format (chrono.tre) with 50 taxa. The alignment was simulated along the chronogram under a Jukes-Cantor substitution model and with rate auto-correlation among lineages. You will also find an XML file (sim.xml) in the xml folder to run BEAST 2 for this data set under a strict clock and a Jukes-Cantor model of substitution and a root calibration.

In this first step, we will run the XML file using BEAST 2. The output of this run can also be found in the folder preekooked runs.

3.2 Step 2: reading the runs and simulating data

We will run the remainder of the model assessment in section 4 of the practical. We will first study the code in R (or the comments) in detail, to cement the steps required for assessing model adequacy. Specifically, we will explore the file adeq.R.

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)
}
```

In the second step we 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 make the posterior predictive simulations. You will find the code to read the posterior and simulate data in make.pps.als, which identifies the model being assessed and simulates data accordingly. For revision, the input required in this step includes:

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

Feel free to investigate the make.pps.als code if you are interested in how we can read and simulate data in R. 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. You might want to revise how an alignment can be simulated using data from the posterior.

3.2.1 Step 3: 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.

3.2.2 Step 4: calculate *P*-values

We can now compare the test statistic for the empirical data and each of the simulated data sets. 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 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 for which branches the molecular dating model can estimate the number of substitutions. We will explore the interpretation of these data after assessing model adequacy in the following section.

4 Model assessment practice

4.0.1 One step assessment

In this section we will run our model assessment. The results for this example can also be found in the precooked runs folder. You will also find the results for the BEAST 2 run of sim.xml, which are the output of the first step for model assessment. We will now run and save the assessment of clock model adequacy, assuming that you are using the output of BEAST 2 or have completed your own run.

Begin by opening R. The following will set the working directory to the scripts folder, and then source all the functions in the folder. Remember that we assume you have installed the package *phangorn* and its dependencies.

```
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 = "sim.trees", log.file ="sim.log", empdat.file = "↩
    ../data/al.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 uses the results from results.Rdata to produce a variety of useful graphics.

4.0.2 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 is to make 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)
```

This code should generate a plot that is identical or very similar to those in 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 model is a good description of the process that generated the data. This is unsurprising, since our empirical data were generated under the model!

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

```
assessment_provided[9]
```

4.0.3 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).

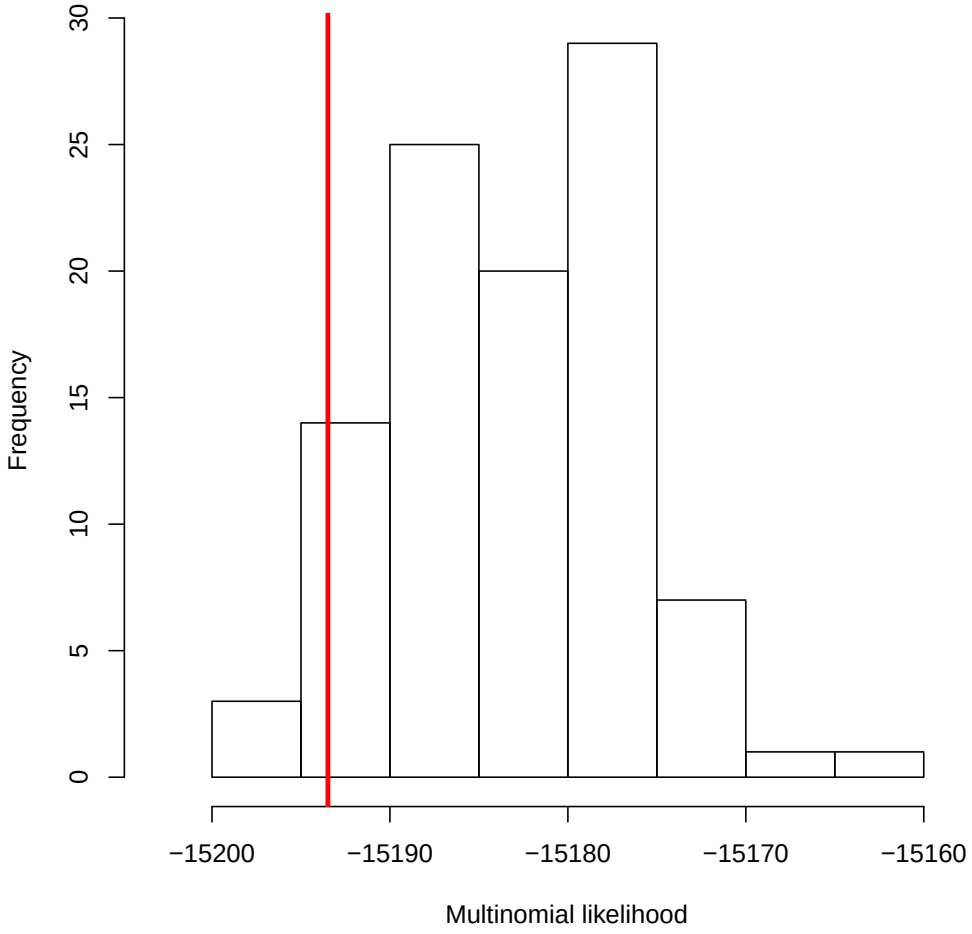

Figure 2: Distribution of PPS multinomial likelihood values with the value of the test statistic for the empirical data shown as a vertical line in red.

```
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 number 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, such that if this value is 0 branch-lengths are being underestimated with respect to the model. Similarly, branch-lengths are being overestimated with respect to the model if the value is 1.

Also investigate the A index:

```
assessment_provided[4]
```

This index provides an additional piece of information: it 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.

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 from simulations. The units are the length of the empirical branch-length estimate. The rates and times models can be considered adequate if this value is close to zero. 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 = 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.

5 Useful references

- Goldman, N. (1993). Statistical tests of models of DNA substitution. Journal of molecular evolution, 36(2), 182-198.
- Bollback, J. P. (2002). Bayesian model adequacy and choice in phylogenetics. Molecular Biology and Evolution, 19(7), 1171-1180.
- Duchene, D. A., Duchene, S., Holmes, E. C., Ho, S. Y. (2015). Evaluating the adequacy of molecular clock models using posterior predictive simulations. Molecular biology and evolution, 32(11), 2986-2995.

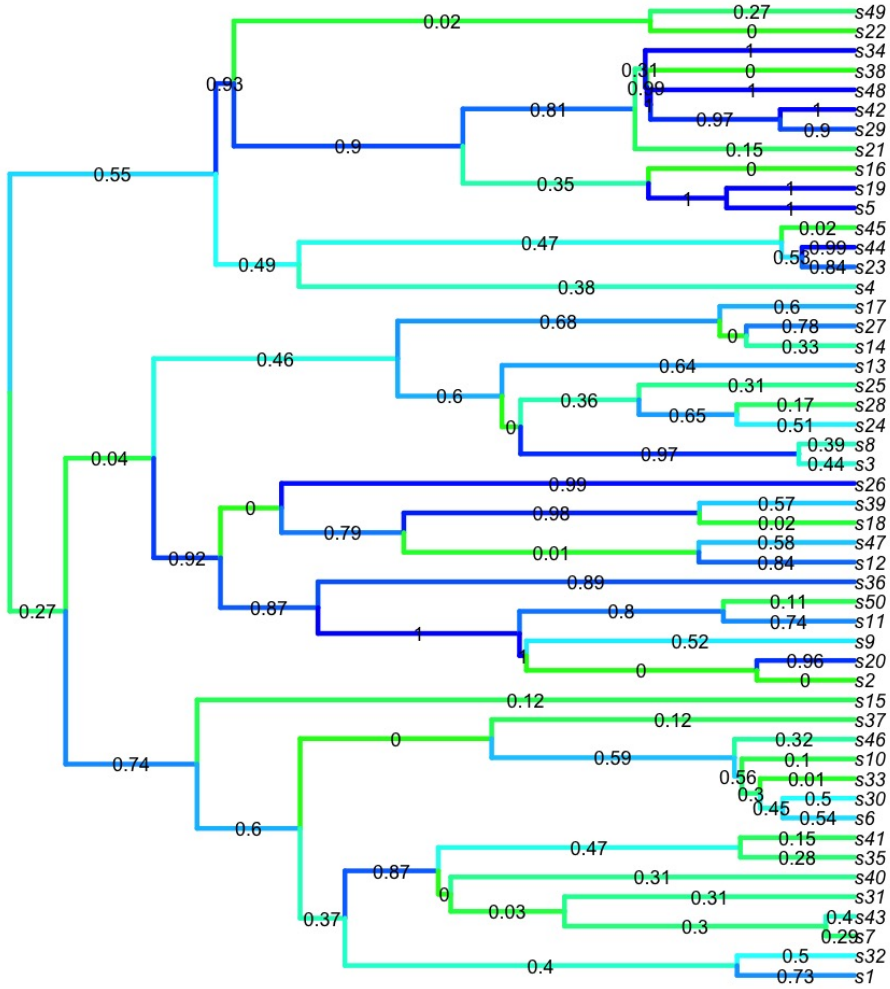

Figure 3: Estimated chronogram with branches coloured by their clock adequacy P-value.

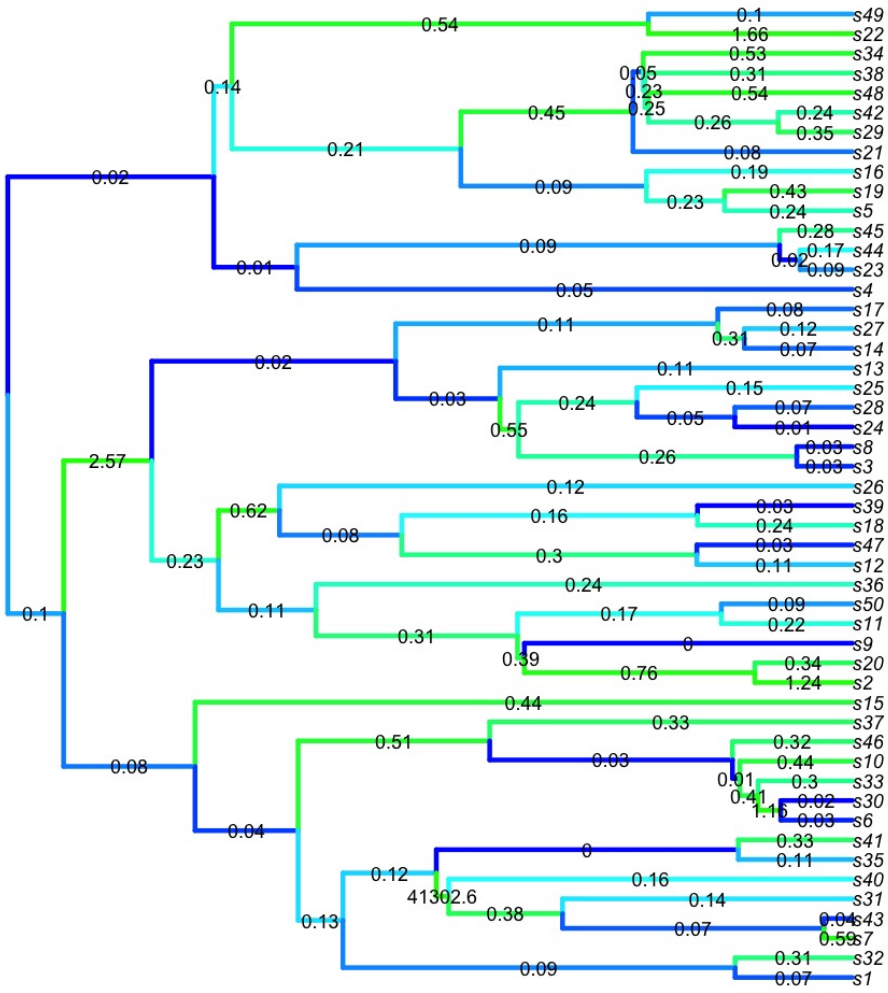

Figure 4: Estimated chronogram with branches coloured by their deviation between the empirical and simulated lengths.

Version dated: January 30, 2017