Tutorial using BEAST v2.6.7

Introduction to substitution and molecular clock models

Louis du Plessis

This is a simple introductory tutorial to help you get started with using BEAST2 and its accomplices.

Contents

1 Background

Before diving into performing complex analyses with BEAST2 one needs to understand the basic workflow and concepts. While BEAST2 tries to be as user-friendly as possible, the amount of possibilities can be overwhelming.

In this simple tutorial you will get acquainted with the basic workflow of BEAST2 and the software tools most commonly used to interpret the results of analyses. Bear in mind that this tutorial is designed only to help you get started using BEAST2. This tutorial does not discuss all the choices and concepts in detail, as they are discussed in other tutorials. Interspersed throughout the tutorial are topics for discussion. These discussion topics are optional, however if you work through them you will have a better understanding of the concepts discussed in this tutorial. Feel free to skip the discussion topics and come back to them later, while running the analysis file, or after finishing the whole tutorial.

This tutorial is adapted from https://taming-the-beast.org/tutorials/Introduction-to-BEAST2/ (by Jūlija Pečerska, Veronika Bošková and Louis du Plessis), which itself was adapted from Divergence Dating Tutorial with BEAST 2.0 (by Alexei Drummond, Andrew Rambaut and Remco Bouckaert).

In this tutorial the focus was changed to genomic epidemiology. The dataset and analyses are based on those presented at http://beast.community/ebov_local_clocks.html (by Andrew Rambaut, JT McCrone and Guy Baele).

2 Programs used in this Exercise

BEAST2 - Bayesian Evolutionary Analysis Sampling Trees 2

BEAST2 (http://www.beast2.org) is a free software package for Bayesian evolutionary analysis of molecular sequences using MCMC and strictly oriented toward inference using rooted, time-measured phylogenetic trees. This tutorial is written for BEAST v2.6.7 (Bouckaert et al. 2014; Bouckaert et al. 2019).

BEAUti2 - Bayesian Evolutionary Analysis Utility

BEAUti2 is a graphical user interface tool for generating BEAST2 XML configuration files.

Both BEAST2 and BEAUti2 are Java programs, which means that the exact same code runs on all platforms. For us it simply means that the interface will be the same on all platforms. The screenshots used in this tutorial are taken on a Mac OS X computer; however, both programs will have the same layout and functionality on both Windows and Linux. BEAUti2 is provided as a part of the BEAST2 package so you do not need to install it separately.

TreeAnnotator

TreeAnnotator is used to summarise the posterior sample of trees to produce a maximum clade credibility tree. It can also be used to summarise and visualise the posterior estimates of other tree parameters (e.g. node height).

TreeAnnotator is provided as a part of the BEAST2 package so you do not need to install it separately.

Tracer

Tracer (http://beast.community/tracer) is used to summarise the posterior estimates of the various parameters sampled by the Markov Chain. This program can be used for visual inspection and to assess convergence. It helps to quickly view median estimates and 95% highest posterior density intervals of the parameters, and calculates the effective sample sizes (ESS) of parameters. It can also be used to investigate potential parameter correlations. We will be using Tracer v1.7.2

FigTree

FigTree (http://beast.community/figtree) is a program for viewing trees and producing publication-quality figures. It can interpret the node-annotations created on the summary trees by TreeAnnotator, allowing the user to display node-based statistics (e.g. posterior probabilities). We will be using FigTree v1.4.4.

DensiTree

Bayesian analysis using BEAST2 provides an estimate of the uncertainty in tree space. This distribution is represented by a set of trees, which can be rather large and difficult to interpret. DensiTree is a program for qualitative analysis of sets of trees. DensiTree allows to quickly get an impression of properties of the tree set such as well-supported clades, distribution of tree heights and areas of topological uncertainty.

DensiTree is provided as a part of the BEAST2 package so you do not need to install it separately.

3 Practical

This tutorial will guide you through the analysis of an alignment of 15 Zaïre Ebola virus (EBOV) genomes. The main aim of this tutorial is to use molecular clock models to estimate the rate of evolution and the date of the most recent common ancestor of all EBOV lineages that have spilled over into humans. More generally, this tutorial aims to introduce new users to a basic workflow and point out the steps towards performing a full analysis of sequencing data within a Bayesian framework using BEAST2.

After completing this tutorial you should be able to:

- Set up all the components of a simple BEAST2 analysis in BEAUti2
- Use different substitution and clock models
- Use monophyly constraints to constrain the topology
- Use Tracer to check convergence
- Use FigTree to visualise the results

3.1 The Data

Before we can start, we need to download the input data for the tutorial. For this tutorial we will use two Fasta files, `EBOV_reference_set_15_cds.fasta` and `EBOV_reference_set_15_ig.fasta`, containing respectively the aligned coding and non-coding sequences of 15 EBOV genomes. The dataset contains one genome from 15 human EBOV outbreaks between 1976 and 2018. More details on the genomes can be found at http://beast.community/ebov_local_clocks.html.

If you cloned the Github repository you should already have the alignments on your drive. Otherwise, you can download the files from https://github.com/laduplessis/viroinf-hiddensee/tree/main/datasets/ebov. Please make sure you download the raw files and that your browser doesn't insert HTML code into the file!

3.2 Creating a simple analysis file with BEAUti

To run analyses with BEAST, we first need to prepare a configuration file in XML format that contains everything BEAST2 needs to run the analysis. A BEAST2 XML file contains:

- The namespace (which BEAST2 packages to load)
- The data (typically a sequence alignment)
- The model specification
- Initial values and parameter constraints
- Settings of the MCMC algorithm
- Output options

Even though it is possible to create such files from scratch in a text editor, it can be complicated and is not exactly straightforward. BEAUti is a user-friendly program that provides a graphical user interface to aid you in producing a valid configuration file for BEAST.

Sometimes it is easier to modify the file by hand than to make modifications in BEAUti. For more complex analyses it is also usually necessary to edit the XML file by hand, since not all models or settings are available in BEAUti. Although the XML file can look intimidating it has a fairly simple structure and it is recommended to always investigate the XML file after creating the file to ensure that everything is

correctly specified.

Begin by starting **BEAUti2**.

3.2.1 Importing the alignment

To give BEAST2 access to the data, the alignment has to be added to the configuration file.

Drag and drop the file `EBOV_reference_set_15_cds.fasta` into the open BEAUti window (it should be open on the **Partitions** tab). When the query box pops up asking for the data type select **nucleotide**.

Alternatively, use **File > Import Alignment** or click on the + in the bottom left-hand corner of the window, then locate and click on the alignment file.

Now also load `EBOV_reference_set_15_ig.fasta`.

Once you have done that, the data should appear in the BEAUti window which should look as shown in Figure 1.

3.2.2 Setting up partition models

A common way to account for site-to-site rate heterogeneity (variation in substitution rates between different sites) is to use a Gamma site model. In this model, we assume that rate variation follows a Gamma distribution. To make the analysis tractable the Gamma distribution is discretised into a small number of bins (usually 4-6). The mean of each bin then acts as a multiplier for the overall substitution rate. The transition probabilities are then calculated for each scaled substitution rate. To calculate the likelihood for a site we marginalise over all rates, i.e. **P(data | tree, substitution model)** is calculated under each Gamma rate category and the results are averaged over all rates. This is a handy approach if we suspect that some sites are evolving faster than others but the precise position of these sites in the alignment is unknown. We will look at Gamma site models later in this tutorial.

Another, more straightforward, way to account for site-to-site rate heterogeneity is to split the alignment into explicit partitions, and specify an independent substitution model for each partition. This is useful when we have a good *a priori* intuition about which positions in the alignment have different substitution rates from the rest. In our example, our alignment is already split into coding and non-coding regions, and we further split the coding region into two partitions: 1st and 2nd, and 3rd codon positions. This is because most mutations at 3rd codon positions are synonymous and we therefore expect them to have a faster substitution rate than 1st and 2nd codon positions.

Select the `EBOV_reference_set_15_cds` partition (alignment). It should be the top partition in the **Partitions** tab. Next click on **Split** at the bottom and select {1,2} + 3 to partition the alignment into two partitions:

- 1st and 2nd codon positions
- 3rd codon positions

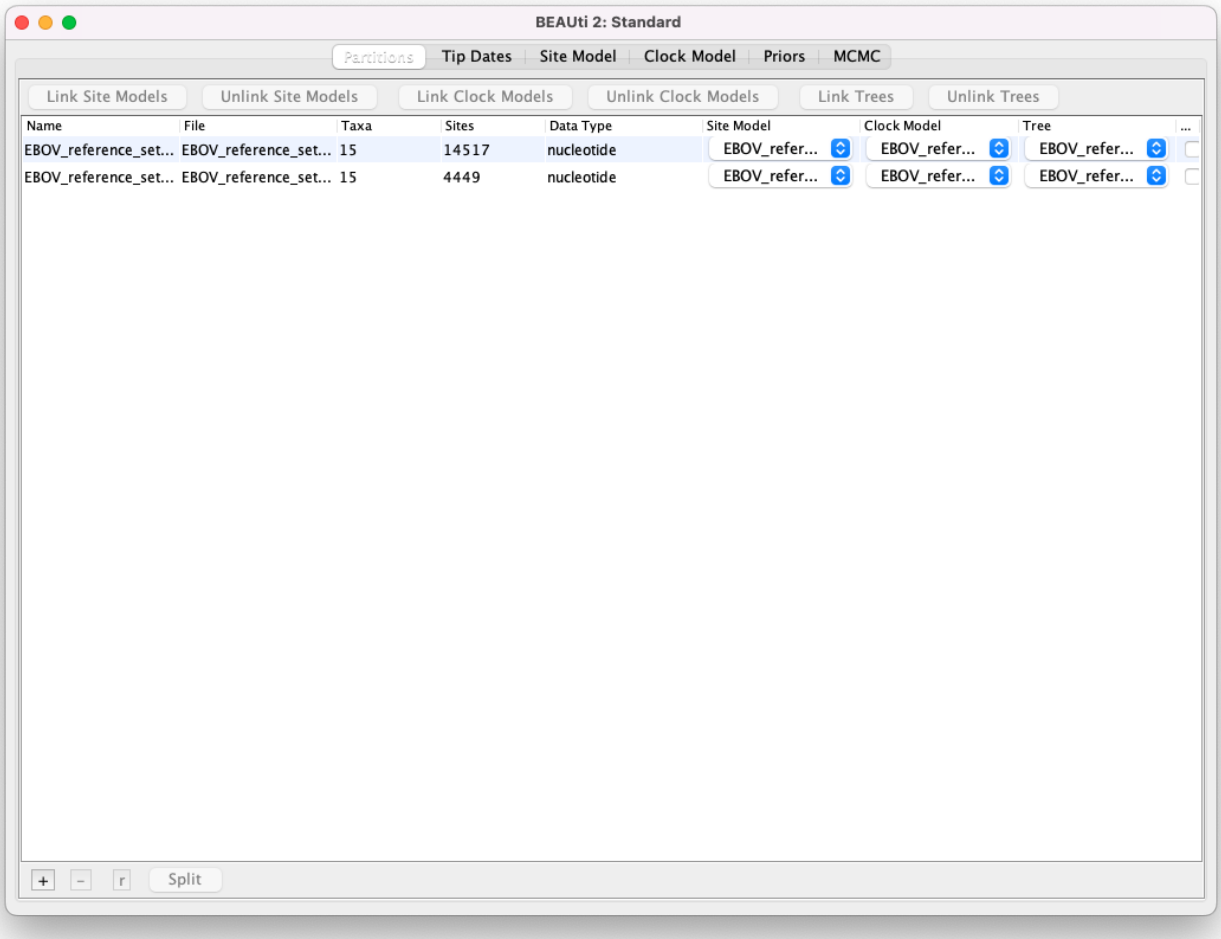

Figure 1: Data imported into BEAUti2.

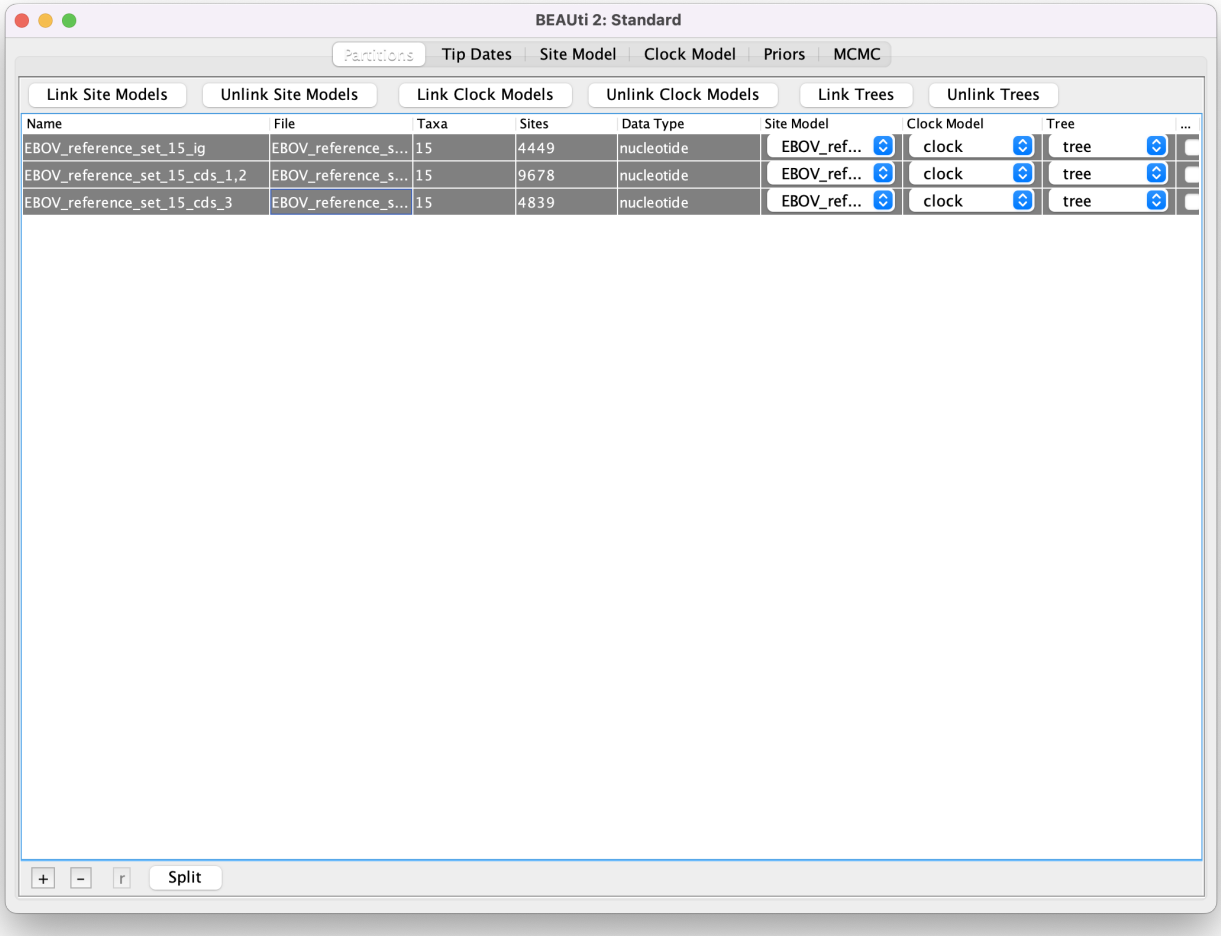

Figure 2: Partitioning the alignment and linking the clock and tree models.

Now select all three partitions (use **shift+click**) and click **Link Clock Models** and **Link Trees**.

You can also look at the individual alignments for each partition by double-clicking on the partition under the **File** column.

We linked trees and clock models, because by default BEAST2 would use independent trees and clock models for each partition. There are cases (e.g. segmented viruses or when recombination is believed to play a large role) when we would not link the trees and clock models of different partitions.

You will see that the **Clock Model** and the **Tree** columns in the table both changed to `EBOV_reference_set_15_ig`↩. Now we will rename both models such that the following options and generated log files are easier to read. The resulting setup should look as shown in Figure 2.

Click on the first drop-down menu in the **Clock Model** column and rename the shared clock model to `clock`.

Likewise, rename the shared tree to `tree`.

3.2.3 Setting the sampling dates

The dataset contains genomes collected from outbreaks between 1976 and 2018. Since Ebola viruses evolve on the same timescale as the period over which the genomes were collected, we can use this information to calibrate the molecular clock.

In the **Tip Dates** panel, check the **Use tip dates** option

The panel should now show the headers of the sequences in the Fasta file. Each sequence header follows a regular format containing the Genbank identifier, sequence name, country of origin and collection date separated by vertical bars.

In order for BEAST2 to use this information we must specify the format of the date string and tell BEAST2 where to find the data.

- Set **Dates specified** to the `as dates with format` option.
- Select `yyyy-M-dd` from the dropdown box.
- Click the **Auto-configure** button. A window will appear where you can specify how BEAUti can find the collection dates in the sequence headers (Figure 3).
- Select **use everything** and specify **after last |** and click **OK**.

This should throw a date parsing error and the panel should now look as in Figure 4. The collection dates for the three highlighted sequences could not be automatically parsed because they are only known up to the month and so don't follow the same format as the rest. To correct this we could edit the sequence headers in the Fasta files or we can simply manually edit the dates for these 3 sequences. Since we have no extra information we will use the middle of the month for the day. Alternatively we could enter the first day of the month. As the sequences in the dataset were collected over more than 40 years it is unlikely that a difference of less than 30 days will result in a big change to parameter estimates. Note that it is also possible to estimate the collection dates of sequences in BEAST2, but this cannot be set in BEAUti.

Double-click on each of the red highlighted sequences under the `Date (raw value)` column and enter **15** as the day. Note that more errors will be thrown until all three dates are corrected! When you're done the panel should look as in Figure 5.

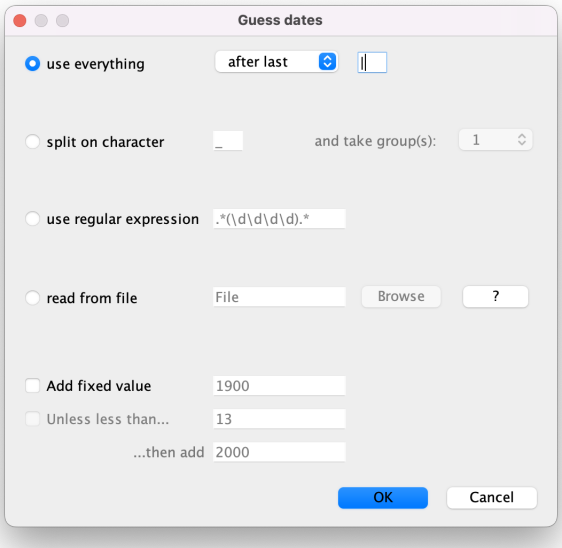

Figure 3: Auto-configure tip dates.

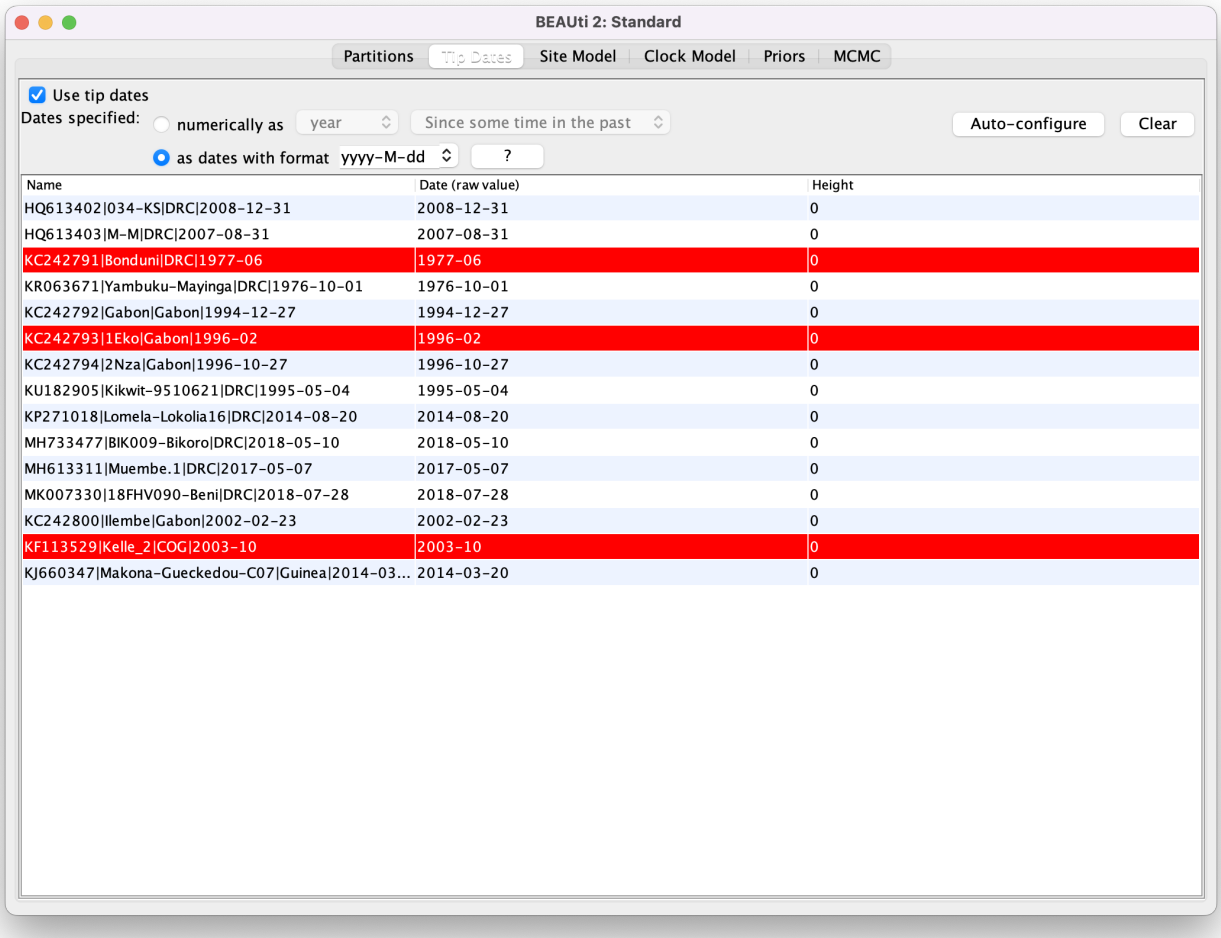

Figure 4: Error parsing these collection dates!

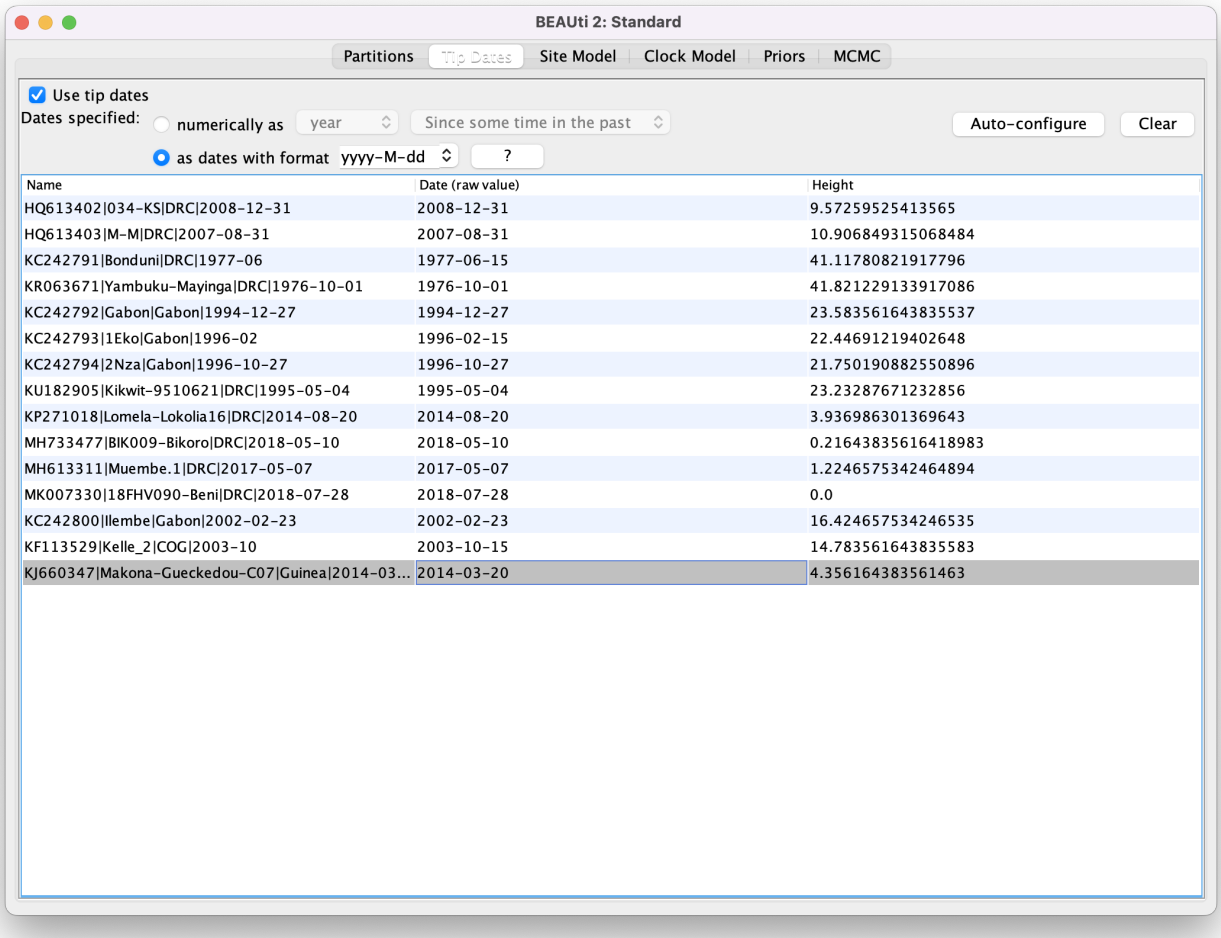

Figure 5: Setting the tip dates.

3.2.4 Setting up the substitution model

Next, we need to set up the substitution models for each partition in the **Site Model** tab.

Select the **Site Model** tab.

The options available in this panel depend on whether the alignment data are in nucleotides, amino-acids, binary data or general data. The settings available after loading the alignment will contain the default values which we normally want to modify.

The panel on the left shows each partition. Remember that we did not link the substitution models in the previous step for the different partitions, so each partition evolves under a different substitution model, i.e. we assume that different positions in the alignment accumulate substitutions differently. We will need to set the site substitution model separately for each part of the alignment as these models are unlinked. However, we think that all partitions evolve according to the same model (but with different parameter values).

Make sure that `EBOV_reference_set_15_ig` is selected.

- Check the **estimate** checkbox for **Substitution Rate**.
- Select **HKY** in the **Subst Model** drop-down menu.
- Select **Empirical** from the **Frequencies** drop-down menu.

Note that when you checked **estimate** for the substitution rate a yellow circle with a cross appeared to the right of **Fix mean substitution rate**. If you hover your cursor above the circle you will see a warning. **Ignore** the warning and continue with the next step.

The panel should look like in Figure 6.

We are using an HKY substitution model with empirical frequencies. This will fix the frequencies to the proportions observed in the partition. This approach means that we can get a good fit to the data without explicitly estimating these parameters. Next we *could* repeat the above steps for each of the remaining partitions or we can take a shortcut.

Select the remaining two partitions (use **shift+click**). The window will now look like Figure 7.

Click **OK** to to clone the site model for the other three partitions from `EBOV_reference_set_15_ig`.

If you did everything correctly the yellow circle with a cross to the right of **Fix mean substitution rate** should have disappeared.

Topic for discussion: Can you figure out the reason for the warning when you checked **estimate** for the substitution rate? Don't worry if you can't figure it out, the reason for the warning is explained in detail in later tutorials.

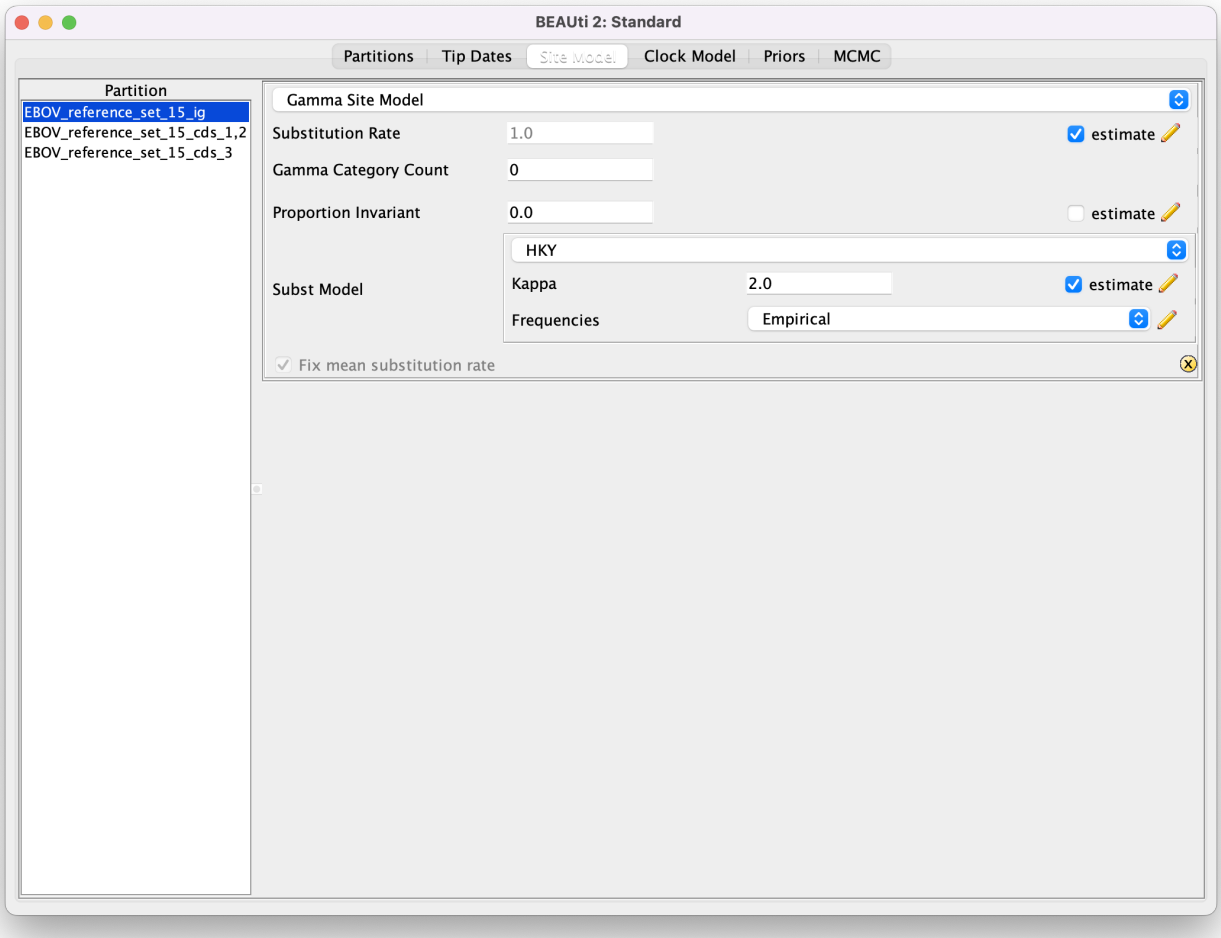

Figure 6: Site model setup.

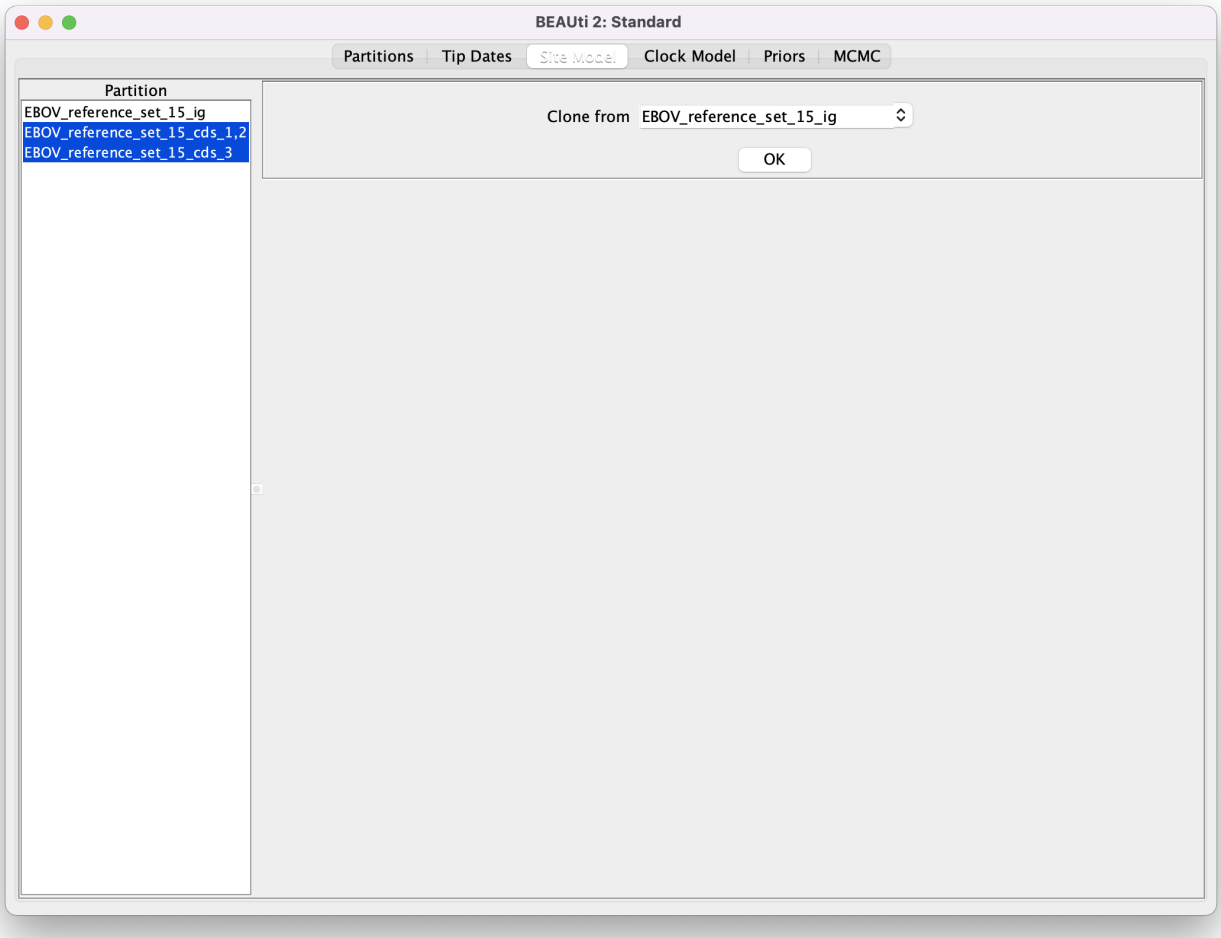

Figure 7: Shortcut to clone site models between partitions.

3.2.5 Setting the clock model

Next, select the **Clock Model** tab at the top of the main window. This is where we set up the molecular clock model. For this initial analysis we are going to leave the selection at the default value of a strict molecular clock, which assumes a steady linear accummulation of mutations over time, with no variation among branches in the tree.

Click on the **Clock Models** tab and view the setup *(but don't do anything)*.

3.2.6 Setting priors

The **Priors** tab allows prior distributions to be specified for each model parameter. The model selections made in the **Site Model** and **Clock Model** tabs determine which parameters are included in the model. For each of these parameters a prior distribution needs to be specified. It is also possible to specify hyperpriors (and hyper-hyperpriors etc.) for each of the model parameters. We also need to specify a prior for the **Tree** that describes the prior expectation for how the tree grows over time.

In this example we use a basic Kingman coalescent model for the tree prior. This simple model assumes a constant effective population size through time. This makes sense for our scenario, as we are modelling the virus within its reservoir species (since we are only including one representative genome per outbreak and we believe each outbreak was started by a single spillover from the reservoir). As we believe the virus to be endemic in the reservoir we would expect the effective population size to remain constant over time.

Go to the **Priors** tab and select **Coalescent Constant Population** in the drop-down menu next to **Tree.t:tree**.

The **popSize** parameter measures the effective population size of the virus. We will leave its prior at the default one-on-X ($\frac{1}{X}$) prior.

By default there is a Uniform prior on the **clockRate** parameter of the clock model, between 0 and ∞. This is a very bad idea for a clock rate prior, because molecular clock rates are in general very small. Thus, we would want to change this prior to a distribution that places more weight on biologically realistic values. We know from human EBOV outbreaks that the molecular clock rate is approximately 1×10^{-3} substitutions per site per year (s/s/y). However, the rate can be elevated during an exponentially growing human outbreak and we expect the long-term substitution rate in the animal reservoir would likely be a little slower.

For **clockRate.c:clock** select **Log Normal** from the drop-down menu

- Expand the options for **clockRate.c:clock** using the arrow button on the left.
- Set the **M** parameter to **1E-3**.
- Set the **S** parameter to **0.5**
- Check the **Mean in Real Space** box

Note that BEAUti displays a plot of the prior distribution on the right, as well as a few of its quantiles. This is for easy reference and can help us to decide if a prior is appropriate. The lognormal prior we are

using is defined on $(0, \infty)$ and would easily allow the rate to be slower than 1×10^{-3} s/s/y, but would also penalise much faster and biologically unrealistic rates.

If we wanted to add a hyperprior on one of the parameters of the lognormal prior we would check the **estimate** box on the right of the parameter. We could also change the initial values or limits of the model parameters by clicking on the boxes next to the drop-down menus. Do **not** do this here, as we are **not** adding any hyperpriors or changing limits in this analysis!

The only remaining model parameters are the transition-transversion ratios for the substitution models on each partition (the **kappa** parameters). If we had chosen to estimate the nucleotide frequencies there would also be priors for the frequency parameters. **We will leave the rest of the priors on their default values!** The BEAUti panel should look as shown in Figure 8.

Please note that in general using default priors is frowned upon as priors are meant to convey your prior knowledge of the parameters. It is important to know what information the priors add to the MCMC analysis and whether this fits your particular situation. In our case the default priors are suitable for this particular analysis, however for further, more complex analyses, we will require a clear idea of what the priors mean. Getting this understanding is difficult and comes with experience.

3.2.7 Setting the MCMC options

Finally, the **MCMC** tab allows us to control the length of the MCMC chain and the frequency of stored samples. It also allows one to change the output file names.

Go to the **MCMC** tab.

The **Chain Length** parameter specifies the number of steps the MCMC chain will make before finishing (i.e. the number of accepted proposals). This number depends on the size of the dataset, the complexity of the model and the precision of the answer required. The default value of 10'000'000 is arbitrary and should be adjusted accordingly. For this initial analysis we will leave the chain length as is, so that it will finish in a few minutes. We also leave the **Store Every** and **Pre Burnin** fields at their default values.

Below these general settings you will find the logging settings. Each particular option can be viewed in detail by clicking the arrow to the left of it. You can control the names of the log files and how often values will be stored in each of the files.

Start by expanding the **tracelog** options. This is the log file you will use later to analyse and summarise the results of the run. The **Log Every** parameter for the log file should be set relative to the total length of the chain. Sampling too often will result in very large files with little extra benefit in terms of the accuracy of the analysis. Sampling too sparsely will mean that the log file will not record sufficient information about the distributions of the parameters. We normally want to aim to store no more than 10'000 samples so this should be set to no less than chain length/10'000.

Expand the **tracelog** options.

- Leave the **Log Every** parameter at **1000**.
- Change the file name to `EBOV_SC.log`

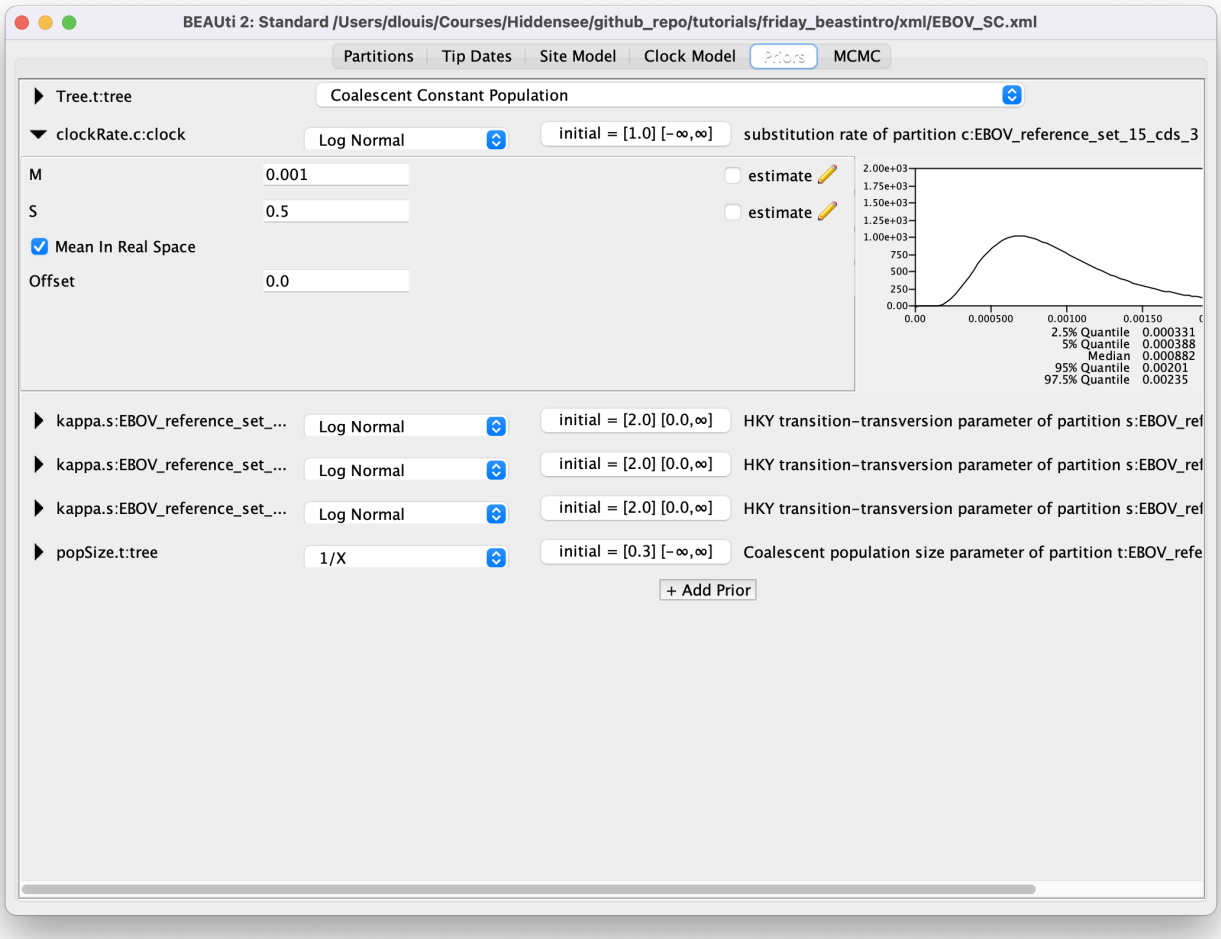

Figure 8: Prior setup.

Next, expand the **screenlog** options. The screen output is simply for monitoring the program's progress. Since it is not so important, especially if you run your analysis on a remote server or on a cluster, the **Log Every** can be set to any value. However, if it is set too small, the sheer quantity of information being displayed on the screen will actually slow the program down. For this analysis we will make BEAST2 log to screen every 10'000 samples, which will be easier to follow than the default setting of every 1'000 samples.

Expand the **screenlog** options.

- Set the **Log Every** parameter to **10'000**

Finally, we can also change the tree logging frequency by expanding **treelog.t:tree**. For big trees with many taxa each individual tree will already be quite large, thus if you log many trees the tree files can easily become extremely large. You would be amazed at how quickly BEAST can fill up even the biggest of drives if the tree logging frequency is too high! For this reason it is often a good idea to set the tree logging frequency lower than the trace log (especially for analyses with many taxa). However, be careful, as the post-processing steps of some models (such as the Bayesian skyline plot) require the trace and tree logging frequencies to be identical!

Expand the **treelog.t:tree** options.

- Set the **File Name** to `EBOV_SC.trees`.
- Leave the **Log Every** parameter at the default value of 1'000.

3.2.8 Generating the XML file

We are now ready to create the BEAST2 XML file. This is the final configuration file BEAST2 can use to execute the analysis.

Save the XML file under the name `EBOV_SC.xml` using **File > Save**.

Do **NOT** close BEAUti, as we will return to it in the following sections!

3.3 Running the analysis

Now open BEAST2 and provide your newly created XML file as input. You can also change the **random number seed** for the run. This number is the starting point of a pseudo-random number chain BEAST2 will use to generate the samples. As computers are unable to generate truly random numbers, we have to resort to generating deterministic sequences of numbers that only look random, but will be identical when the starting seed is the same. If your MCMC run converges to the true posterior then you will be able to draw the same conclusions regardless of which random seed is provided. However, if you want to exactly reproduce the results of a run you need to start it with the same random number seed. For the results below we used the random seed 777.

Run the **BEAST2** program.

- Select `EBOV_SC.xml` as the **Beast XML File**.
- Set the **Random number seed** to **777** (or pick your favourite number).
- Check the **Use BEAGLE library if available** checkbox. If you have previously installed BEAGLE this will make the analysis run faster.

The BEAST2 window should look as shown in Figure 9.

Run **BEAST2** by clicking the `Run` button.

BEAST2 will run until the specified number of steps in the chain is reached. While it is running, it will print the screenlog values to a console and store the tracelog and tree log values to files located in the same folder as the configuration XML file. The screen output will look approximately as shown in Figure 10.

The window will remain open when BEAST2 finished running the analysis. When you try to close it, you may see BEAST2 asking the question: "Do you wish to save?". Note that your log and trees files are always saved, no matter what answer you choose for this question. Thus, the question is only restricted to saving the BEAST2 screen output (which contains some information about the hardware configuration, initial values, operator acceptance rates and running time that are not stored in the other output files).

Topic for discussion: While the analysis is running see if you can identify which parts of the setup in BEAUti are concerned with the data, the model and the MCMC algorithm.

Open the XML file in your favourite text editor. Can you recognize any of the values you set in BEAUti? Can you identify the data, model specification and MCMC settings in the XML file?

Can you find the likelihood, priors and hyperpriors in the XML file?

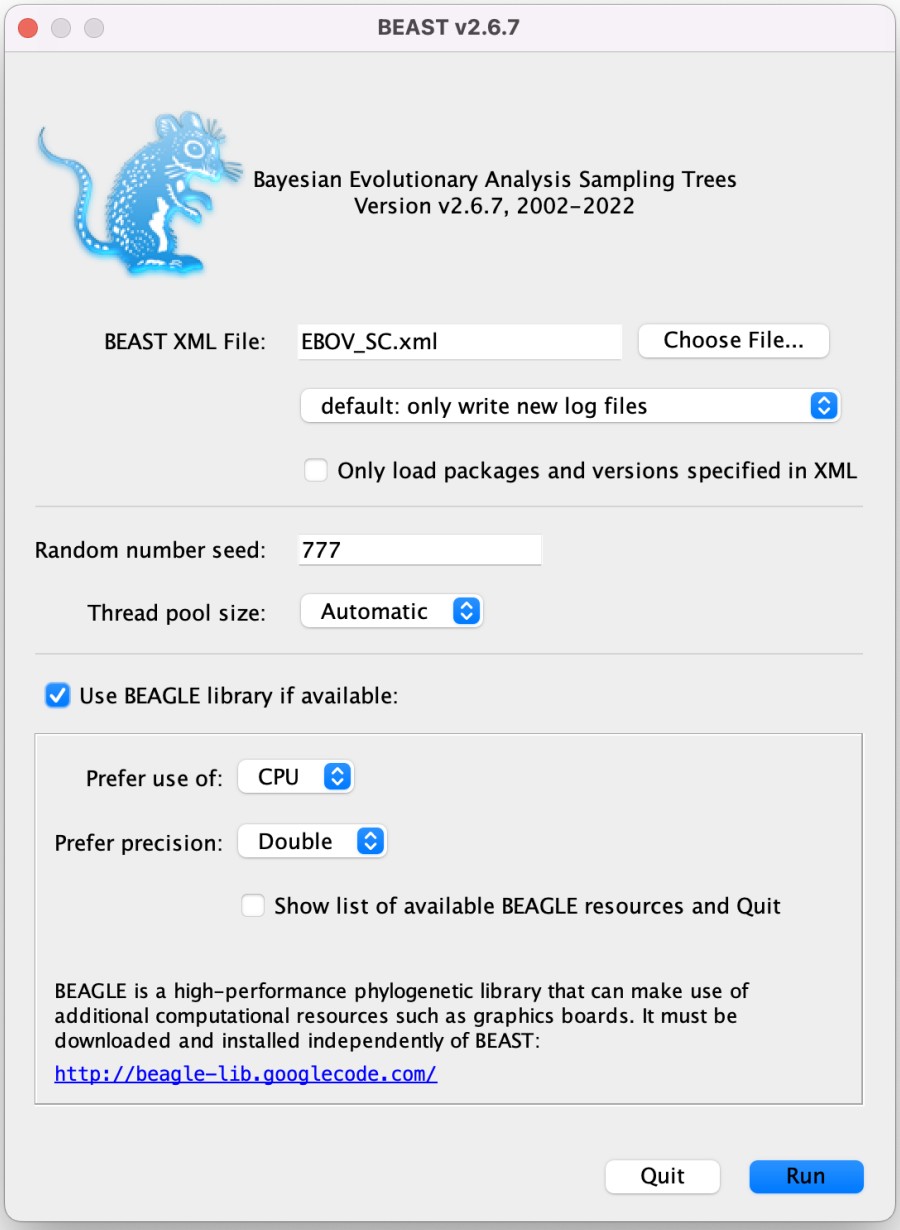

Figure 9: BEAST2 setup for the analysis.

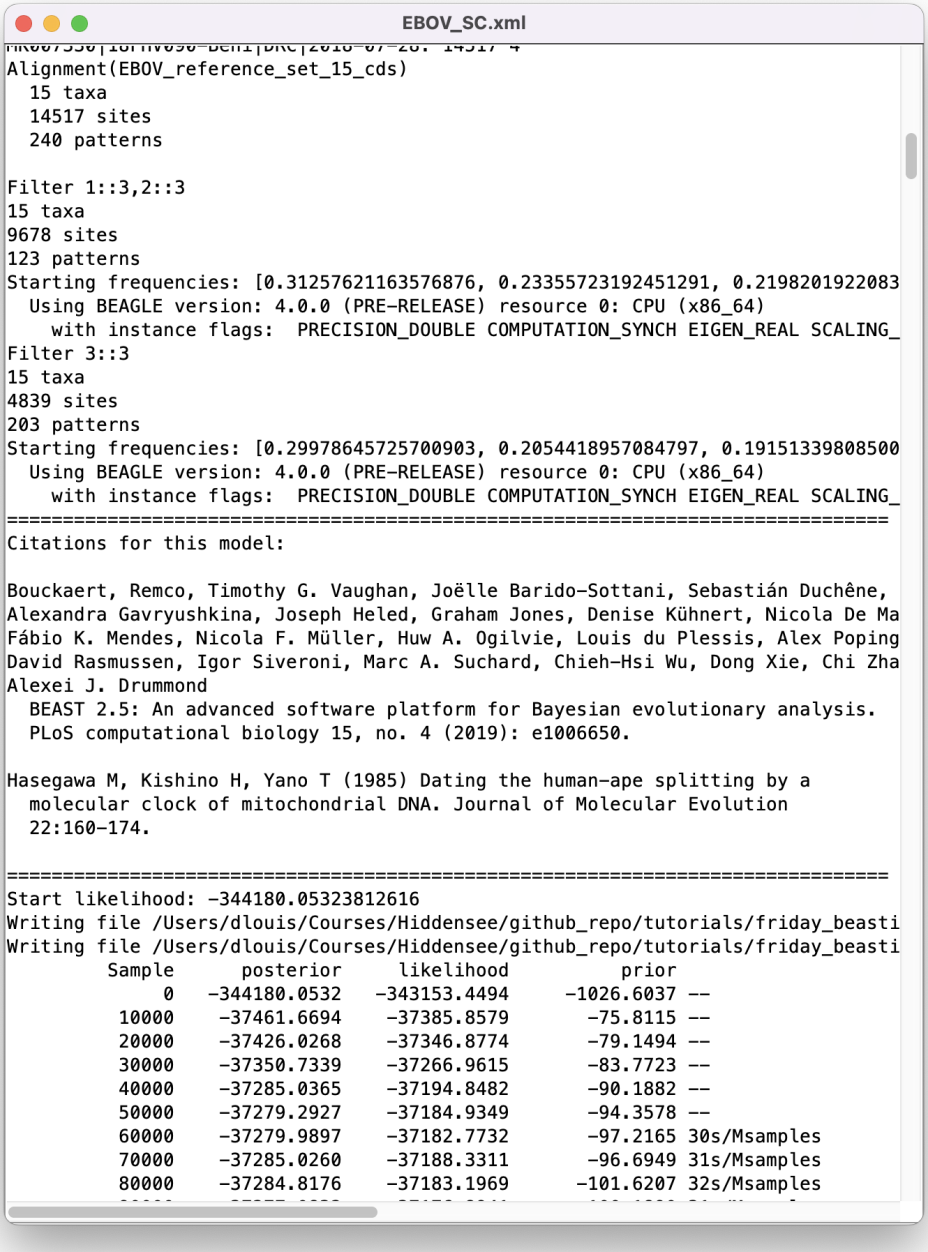

Figure 10: BEAST2 screen output for the analysis.

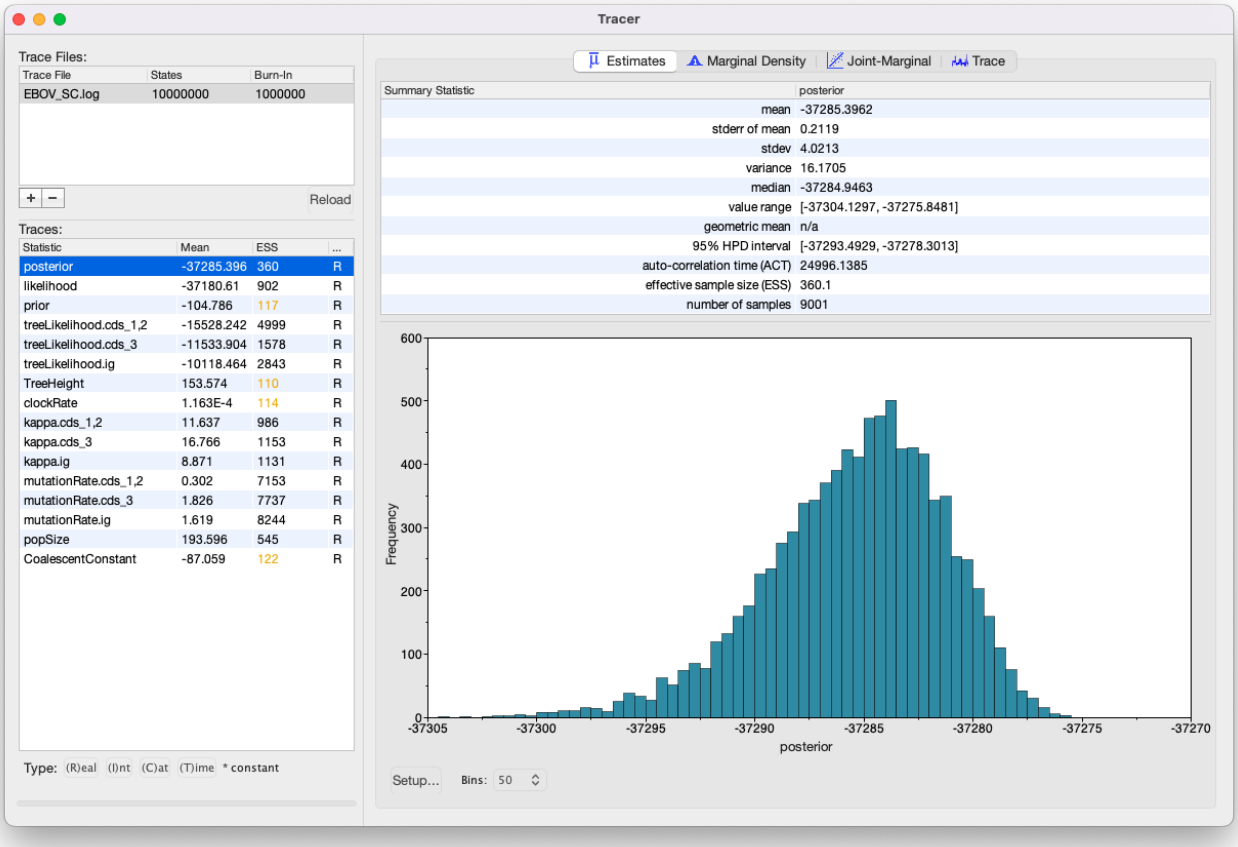

Figure 11: Tracer showing a summary of the BEAST2 run of the strict clock analysis with an MCMC chain length of 10'000'000 and no constraints.

3.4 Analysing the results

Once BEAST2 has finished running, open Tracer to get an overview of BEAST2 output. When the main window has opened, choose `File` > `Import Trace File...` and select the file called `EBOV_SC.log` that BEAST2 has created, or simply drag the file from the file manager window into Tracer.

> Open **Tracer**. Drag and drop the `EBOV_SC.log` file into the open Tracer window.
>
> Alternatively, use **File > Import Trace File...** (or press the + button below the **Trace Files** panel) then locate and click on `EBOV_SC.log`.

The Tracer window should look as shown in Figure 11.

Tracer provides a few useful summary statistics on the results of the analysis. On the left side in the top window it provides a list of log files loaded into the program. The window below shows the list of statistics logged in each file. For each statistic it gives a list of summary values such as the mean, standard error, median, and others it can compute from the data. The summary values are displayed in the top right

window and a histogrom showing the distribution of the statistic is in the bottom right window.

The log file contains traces for the posterior (this is the natural logarithm of the product of the tree likelihood and the prior density), prior, the likelihood, tree likelihoods and other continuous parameters. Selecting a trace on the left brings up the summary statistics for this trace on the right hand side. When first opened, the **posterior** trace is selected and various statistics of this trace are shown under the **Estimates** tab.

For each loaded log file we can specify a **Burn-In**, which is shown in the file list table (top left) in Tracer. The burn-in is intended to give the Markov Chain time to reach its equilibrium distribution, particularly if it has started from a bad starting point. A bad starting point may lead to over-sampling regions of the posterior that actually have very low probability under the equilibrium distribution, before the chain settles into the equilibrium distribution. Burn-in allows us to simply discard the first N samples of a chain and not use them to compute the summary statistics. Determining the number of samples to discard is not a trivial problem and depends on the size of the dataset, the complexity of the model and the length of the chain. A good rule of thumb is to always throw out at least the first 10% of the whole chain length as the burn-in (however, in some cases it may be necessary to discard as much as 50% of the MCMC chain).

Select the **TreeHeight** statistic in the left hand list to look at the tree height estimated jointly for all partitions in the alignment. Tracer plots the (marginal posterior) histogram for the selected statistic and also gives you summary statistics such as the mean and median. The 95% HPD stands for *highest posterior density interval* and represents the most compact interval on the selected statistic that contains 95% of the posterior density. It can be loosely thought of as a Bayesian analogue to a confidence interval. The **TreeHeight** statistic gives the marginal posterior distribution of the age of the root of the entire tree (that is, the tMRCA; the time to the most recent common ancestor).

Select **TreeHeight** in the bottom left hand list in Tracer and view the different summary statistics on the right.

You can also compare estimates of different parameters in Tracer. Once a trace file is loaded into the program you can, for example, compare estimates of the different mutation rates corresponding to the different partitions in the alignment.

Select all three mutation rates by clicking the first mutation rate (**mutationRate.cds_1,2**), then holding **Shift** and clicking the last mutation rate (**mutationRate.ig**).

Select the **Marginal Density** tab on the right to view the four distributions together.

Select different options in the **Display** drop-down menu to display the posterior distributions in different ways.

You will be able to see all four distributions in one plot, similar to what is shown in Figure 12.

Topic for discussion: What can you deduce from the marginal densities of the 4 mutation rates? Does this make biological sense?

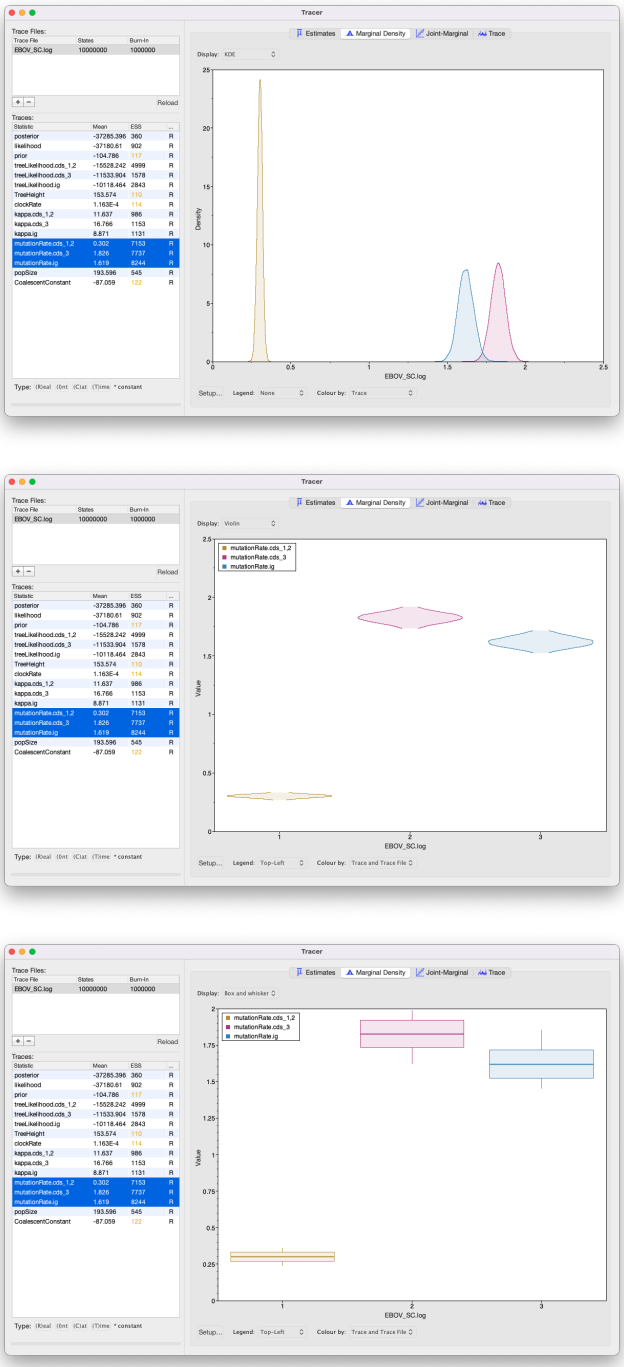

Figure 12: Tracer showing the four marginal probability distributions of the mutation rates in each partition of the alignment. The figure at the top shows the marginal distributions plotted with a Kernel Density Estimation (KDE) in the middle as violin plots and at the bottom as box and whisker plots. Note that you can also display a legend.

3.4.1 Analysing tree estimates

Besides producing a sample of parameter estimates, BEAST2 also produces a posterior sample of time-calibrated phylogenetic trees. These need to be summarised too before any conclusions about the quality of the posterior estimate can be made.

One way to summarise the trees is by using the program TreeAnnotator. This will take the set of trees and find the *maximum clade credibility* tree, which is one particular estimate of the "best supported" tree. The nodes in this tree are also annotated with the corresponding 95% HPD ranges in the posterior set of trees and each clade is annotated with its posterior probability.

Open **TreeAnnotator**.

Set the **Burnin percentage** to **10%** to discard the first 10% of trees in the tree file.

The next option, the **Posterior probability limit**, specifies a limit such that if a node is found at less than this frequency in the sample of trees (i.e. has a posterior probability less than this limit), it will not be annotated. For example, setting it to 0.5 means that only nodes seen in the majority (more than 50%) of trees will be annotated. The default value is 0, which we will leave as is, and which means that TreeAnnotator will annotate all nodes.

Leave the **Posterior probability limit** at the default value of **0**.

For the **Target tree type** option you can either choose a specific tree from a file or ask TreeAnnotator to find a tree in your sample. The default option which we will leave, **Maximum clade credibility tree**, finds the tree with the highest product of the posterior probability of all its nodes.

Leave the **Target tree type** at the default value of **Maximum clade credibility tree**.

Next, select **Mean heights** for the **Node heights**. This sets the heights (ages) of each node in the tree to the mean height across the entire sample of trees for that clade.

Select **Mean heights** in the **Node heights** drop-down menu.

Finally, we have to select the input tree log file and set an output file.

Click **Choose File** next to **Input Tree File** and choose `EBOV_SC.trees`.

Set the **Output File** to `EBOV_SC.MCC.tree`.

The setup should look as shown in Figure 13. You can now run the program.

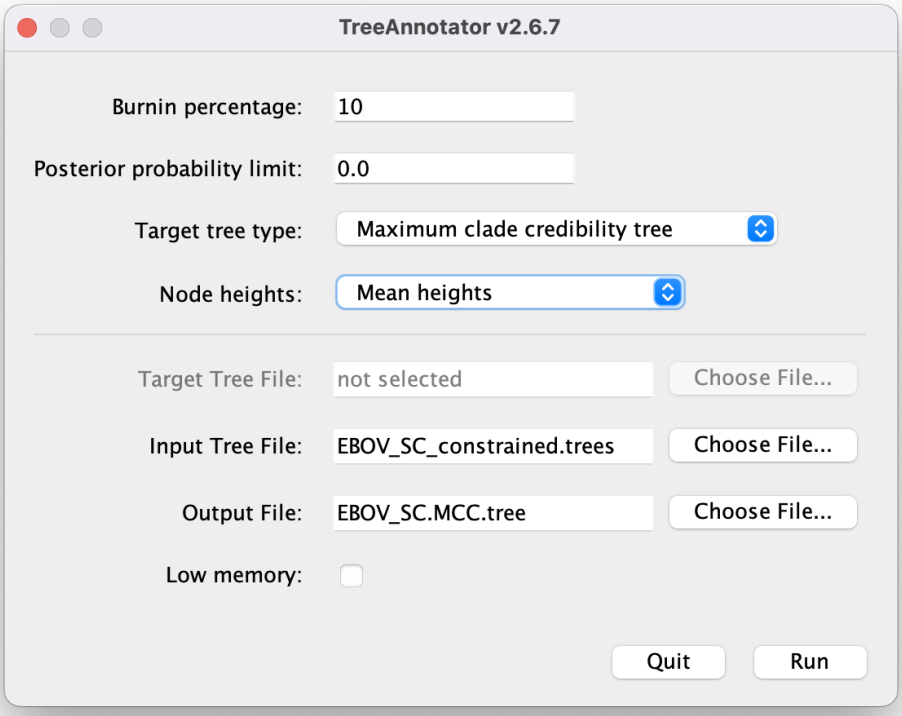

Figure 13: TreeAnnotator setup

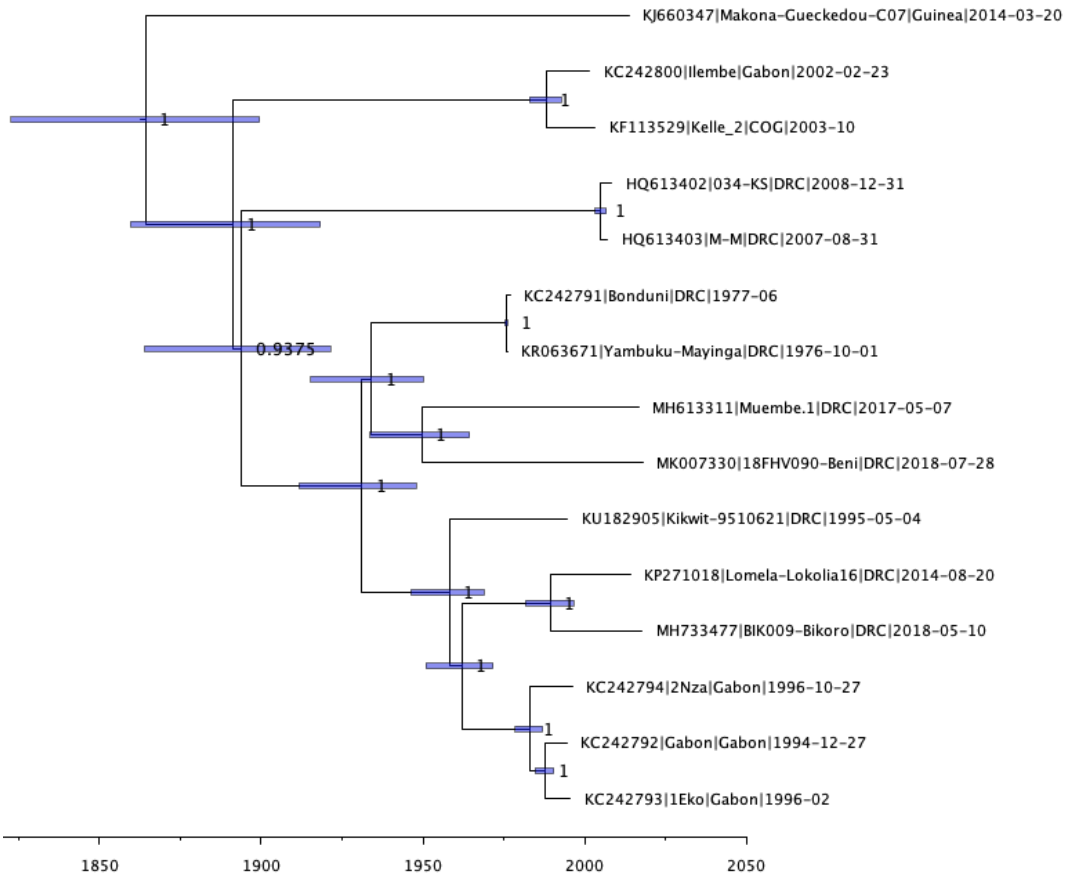

Figure 14: FigTree visualisation of the estimated tree.

3.4.2 Visualising the tree estimate

Finally, we can visualize the tree with one of the available pieces of software, such as FigTree.

Open **FigTree**. Use **File** > **Open** then locate and click on `EBOV_SC.MCC.tree`.

- Expand **Trees** options, check **Order nodes** and select **decreasing** from the drop-down menu.
- Expand the **Tip Labels** options and increase the **Font Size** until it is readable.
- Check the **Node Bars** checkbox, expand the options and select `height_95%_HPD` from the **Display** drop-down menu.
- Check the **Node Labels** checkbox, expand the options and select `posterior` from the **Display** drop-down menu.
- Increase the **Font Size** until it is readable.
- Uncheck the **Scale Bar** checkbox.
- Check the **Scale Axis** checkbox, expand the options, check **Reverse axis** and increase the **Font Size**.
- Expand the **Time Scale** options and set the offset to 2018 (approximately the most recent collection date).

Your tree should now look something like Figure 14. We first ordered the tree nodes. Because there are many ways to draw the same tree, ordering nodes makes it easier for us to compare different trees to each other. The scale bars we added represent the 95% HPD interval for the age of each node in the tree, as estimated by the BEAST2 analysis. The node labels we added gives the posterior probability for a node in the posterior set of trees (that is, the trees logged in the tree log file, after discarding the burn-in). We can also use FigTree to display other statistics, such as the branch lengths, the 95% HPD interval of a node etc. The exact statistics available will depend on the model used.

We can see that the tree topology is very highly supported, although there is some uncertainty in the age of the deeper nodes. We also see that the representative genome of the 2014 West African Ebola virus disease epidemic (Makona-Gueckedou-C07) is an outgroup to all other genomes in our dataset and that the tMRCA is estimated to be around the middle of the 19th century.

3.5 Adding topology constraints

Dudas and Rambaut (2014) found in a different analysis that the root most likely lies between the outbreaks of the 1970s and the other outbreaks. In other words, they found that the MRCA (most recent common ancestor) of all known human EBOV outbreaks was closest to the viruses that caused the 1970s outbreaks. Armed with this prior knowledge, how can we incorporate it in our analysis? We can set a monophyly constraint to tell BEAST2 that a certain group of sequences should form a single clade in the tree with a common ancestor! Implicitly that will also constrain the MRCA of this clade to be younger than the tree's root. We can do this in BEAUti.

Either set up a new XML file in BEAUti and follow the same steps as above until you've specified the priors or else simply go back to BEAUti and edit the previous analysis file.

To add an extra prior to the model, Click on the **Priors** tab and click the **+ Add Prior** button below the list of priors and select **MRCA Prior** from the drop-down menu.

You will see a dialogue box that allows you to select a subset of taxa from the phylogenetic tree.

- Set the **Taxon set label** to `ingroup`.
- Select all sequences on the left hand side list and click the >> button to add them to the `ingroup` taxon set.
- Locate `Bonduni` and `Yambuku-Mayinga` sequences on the right hand side and click << to move them out of the taxon set.

The taxon set should now look like Figure 15.

Click the **OK** button to add the newly defined taxon set to the prior list.

In order to constrain the tree topology to keep our ingroup monophyletic during the course of the MCMC analysis we have to select monophyletic.

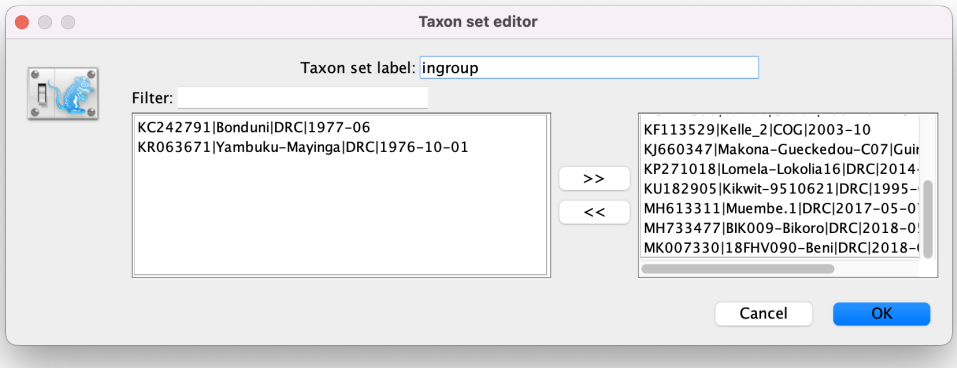

Figure 15: Ingroup taxon set.

Check the **monophyletic** checkbox next to **ingroup.prior**.

We could now also add calibration information for its most recent common ancestor (MRCA). We can do this by adding a prior on the age of the MRCA node of our taxon set. However, we can only do this if we have some prior information about its age, which we do not have here so any such prior would be pure guesswork.

Leave the MCMC settings as before, but change the filenames to `EBOV_SC_constrained.log` and `EBOV_SC_constrained.trees`. Now save the XML file as `EBOV_SC_constrained.xml` and run it in BEAST2 (with seed 777).

3.6 Setting up a relaxed clock analysis

While the constrained strict clock analysis is running we will set up a similar analysis, but using a relaxed clock this time. Whereas the strict clock enforced the same molecular clock rate on all branches in the tree, a relaxed clock allows for rate variation among branches.

We have a good intuition that there should be rate variation. When the first genomes from the 2014 Ebola virus disease outbreak in the Democratic Republic of the Congo were sequenced, it was noted that they were much less divergent from earlier EBOV genomes than those sequenced from the 2014 West African Ebola virus disease epidemic (Maganga et al. 2014). Long-term periods of latency had also been observed during the West African epidemic (Blackley et al. 2016; Diallo et al. 2016). These observations suggested that latency or possibly different animal reservoirs (Lam et al. 2015) could be responsible for the observed discrepancies in genetic distances that were observed for the 2014 genomes. The same lower divergence was also observed for genomes from teh 2017 and 2018 outbreaks in the Democratic Republic of the Congo (http://beast.community/ebov_local_clocks.html).

Click on the **Clock Models** tab and select `Relaxed Clock Log Normal` from the dropdown box.

This will use an uncorrelated lognormally distributed relaxed clock model. This model allows each branch in the tree to have a different rate, independently drawn from a lognormal distribution (Drummond et

al. 2006). This is a very flexible model and is the most popular relaxed clock model for viral genomes. However, it should not be used as a default model for every analysis!

The relaxed clock has two parameters, for which we need to set priors. These parameters describe the mean and the standard deviation of the lognormal distribution from which the clock rates of the tree branches are drawn.

Click on the **Priors** tab.

For **ucldMean.c:clock** we will use the same exponential prior we used for the clock rate in the strict clock analysis. We will leave the prior for **ucldStdev** as the default.

Leave the MCMC settings as before, but change the filenames to `EBOV_UCLD_constrained.log` and `EBOV_UCLD_constrained.trees`. Now save the XML file as `EBOV_UCLD_constrained.xml` and run it in BEAST2 (with seed 777).

3.7 Comparing results and checking convergence

Two very important summary statistics that we should pay attention to are the Auto-Correlation Time (ACT) and the Effective Sample Size (ESS). ACT is the average number of states in the MCMC chain that two samples have to be separated by for them to be uncorrelated, i.e. for them to be independent samples from the posterior. The ACT is estimated from the samples in the trace (excluding the burn-in). The ESS is the number of independent samples that the trace is equivalent to. This is calculated as the chain length (excluding the burn-in) divided by the ACT.

The ESS is regarded as a good quality-measure of the resulting sample sequence. It is unclear how to determine exactly how large should the ESS be for the analysis to be trustworthy. In general, an ESS of 200 is considered high enough to make the analysis useful. However, this is an arbitrary number and you should always use your own judgment to decide if the analysis has converged or not. ESS values below 100 are coloured in red, which means that we should (probably) not trust the value of the statistics, and ESS values between 100 and 200 are coloured in yellow.

If a lot of statistics have red or yellow coloured ESS value, we have not sufficiently explored the posterior space. This is most likely a result of the chain not running long enough.

Load the log files of all three XML files into Tracer.

Click on the **Trace** tab to look at the traces of parameters.

You'll notice that both of the constrained analyses have several parameters with very low ESS values below 100 (Figure 16). Looking at the traces of these parameters it is clear that they are having some trouble mixing and that we definitely need to run our analyses longer. Luckily it doesn't look like any of these parameters have "sticky" chains!

Select all three log files in the left-hand panel to show all shared parameters between them.

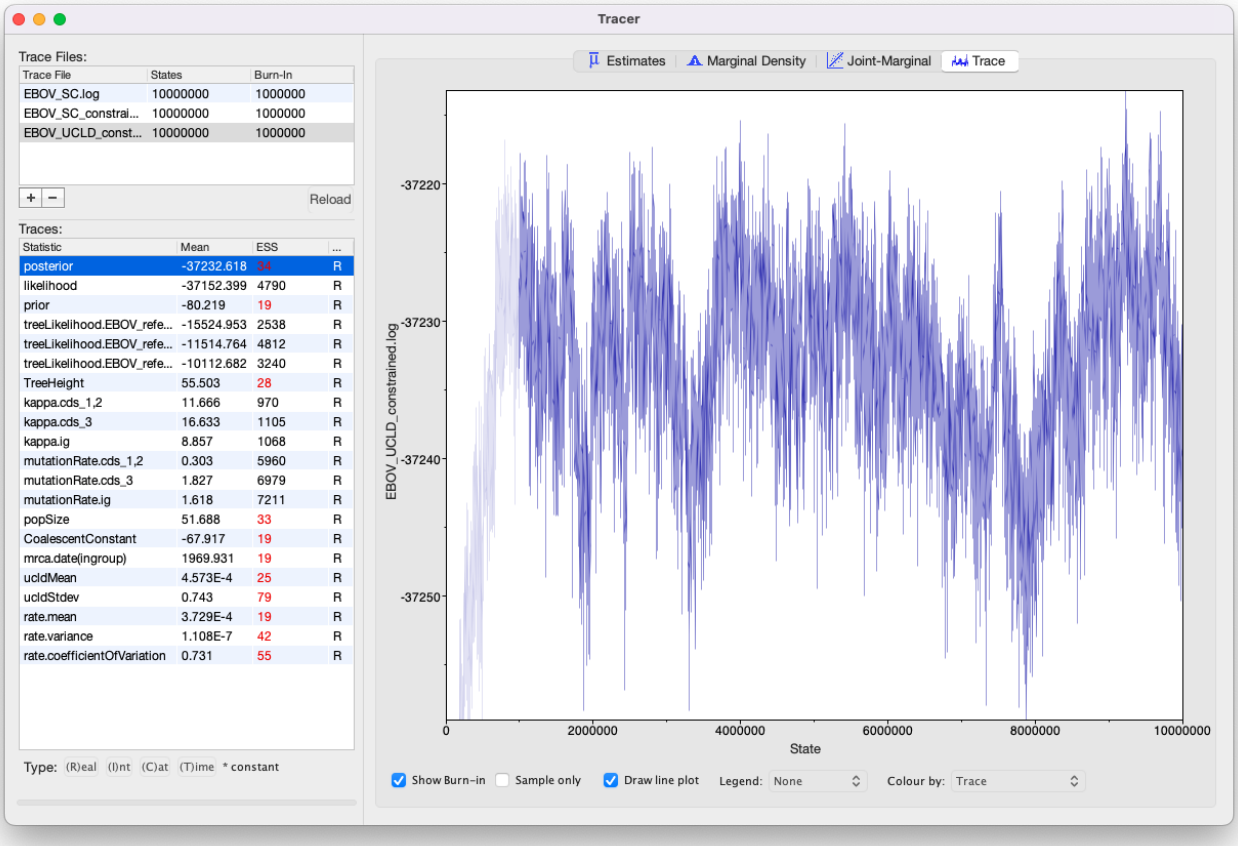

Figure 16: Trace with a poor ESS value.

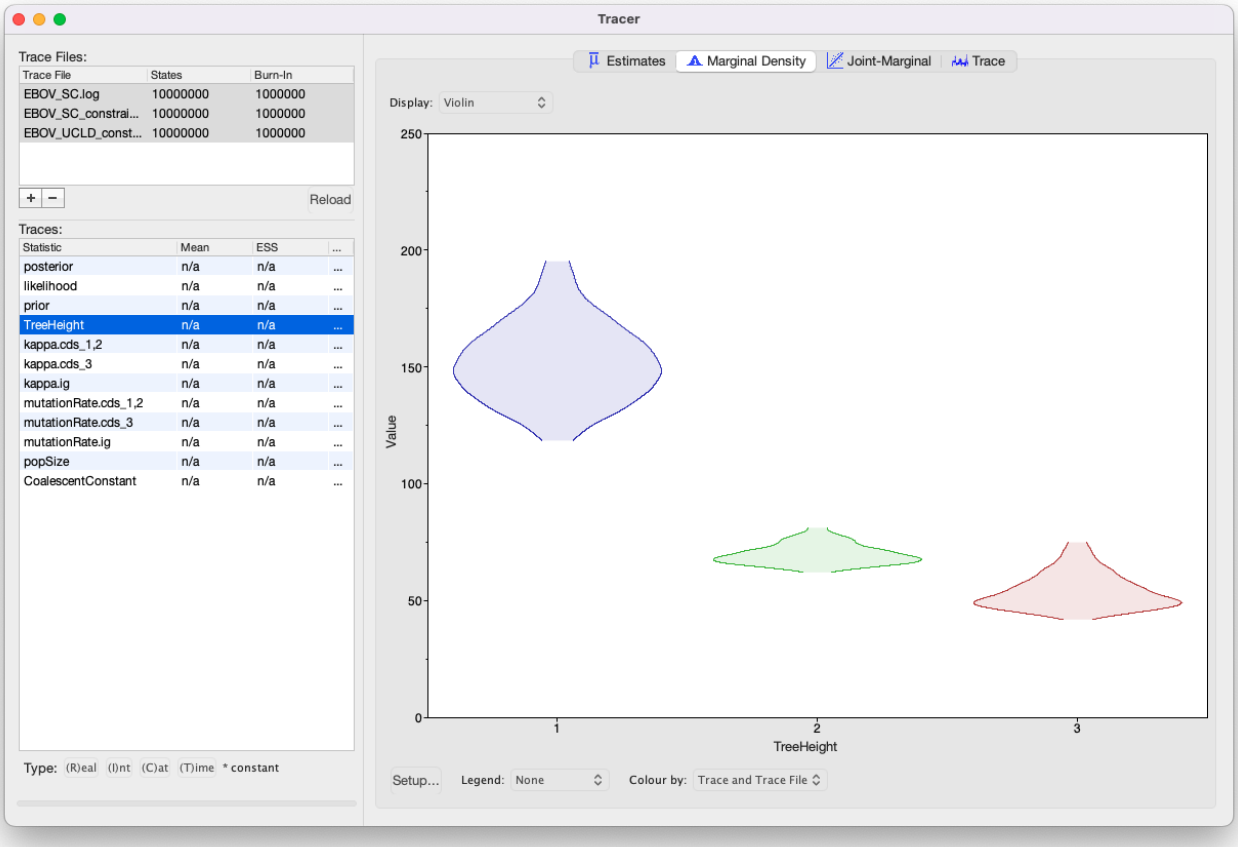

Figure 17: The posterior height estimates of the three trees.

We note that both of the constrained analyses resulted in much younger estimates for the tMRCA (Figure 17. We also note that these two analyses also logged the age of the tMRCA of the ingroup, with the relaxed clock analysis estimating the youngest age (Figure 18).

We also note that the coefficient of variation of the clock rate of the relaxed clock model has an HPD interval that does not include 0 (Figure 19). If the coefficient of variation is 0 all rates are equal and the relaxed clock model reduces to a strict clock model. Thus, we can conclude that there is posterior evidence for variable rates among branches in the tree.

Tracer also allows us to look for correlations between parameters under the **Joint Marginal** tab, as shown in Figure 20. When two parameters are highly correlated this can lead to poor convergence of the MCMC chain.

We can also look at correlations between more than two parameters.

Select all 3 mutation rates again

- Navigate to the **Joint Marginal** tab

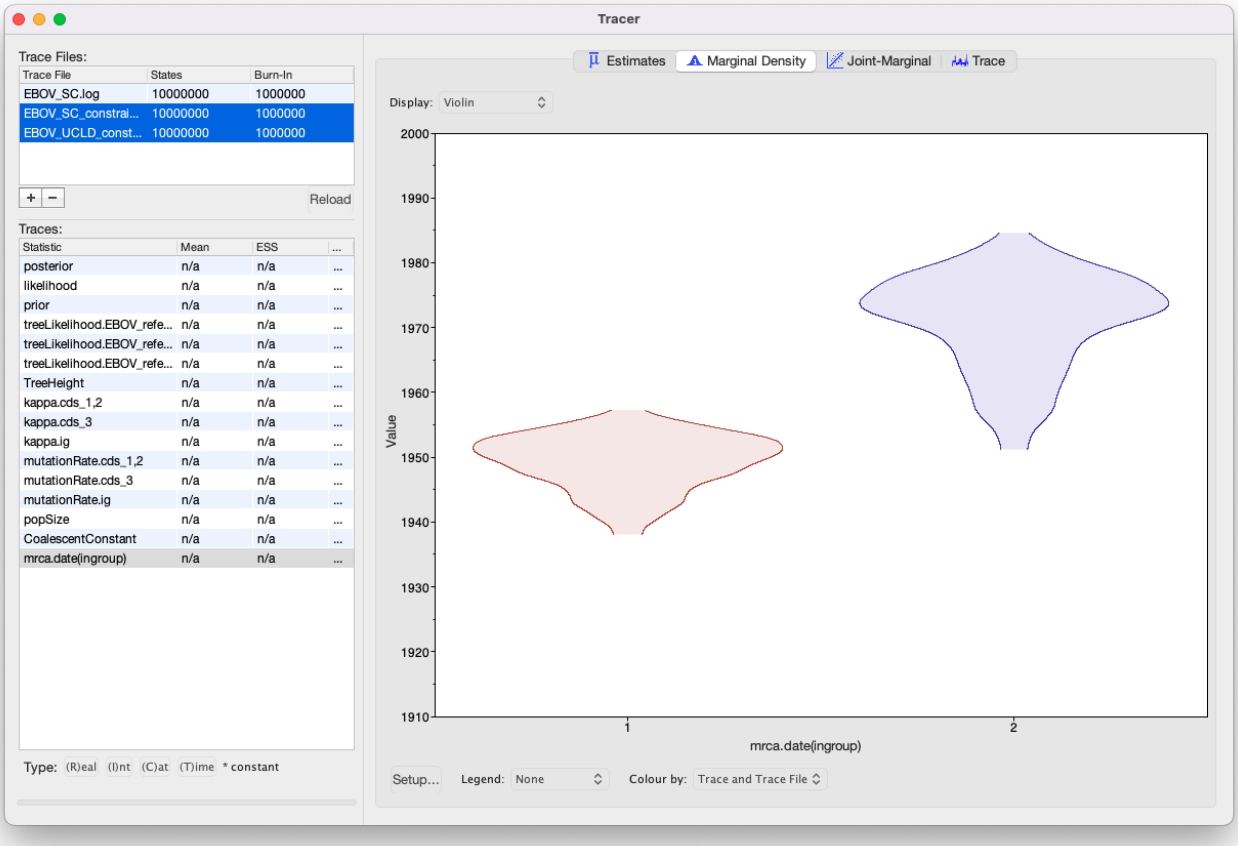

Figure 18: The posterior tMRCAs of the ingroup.

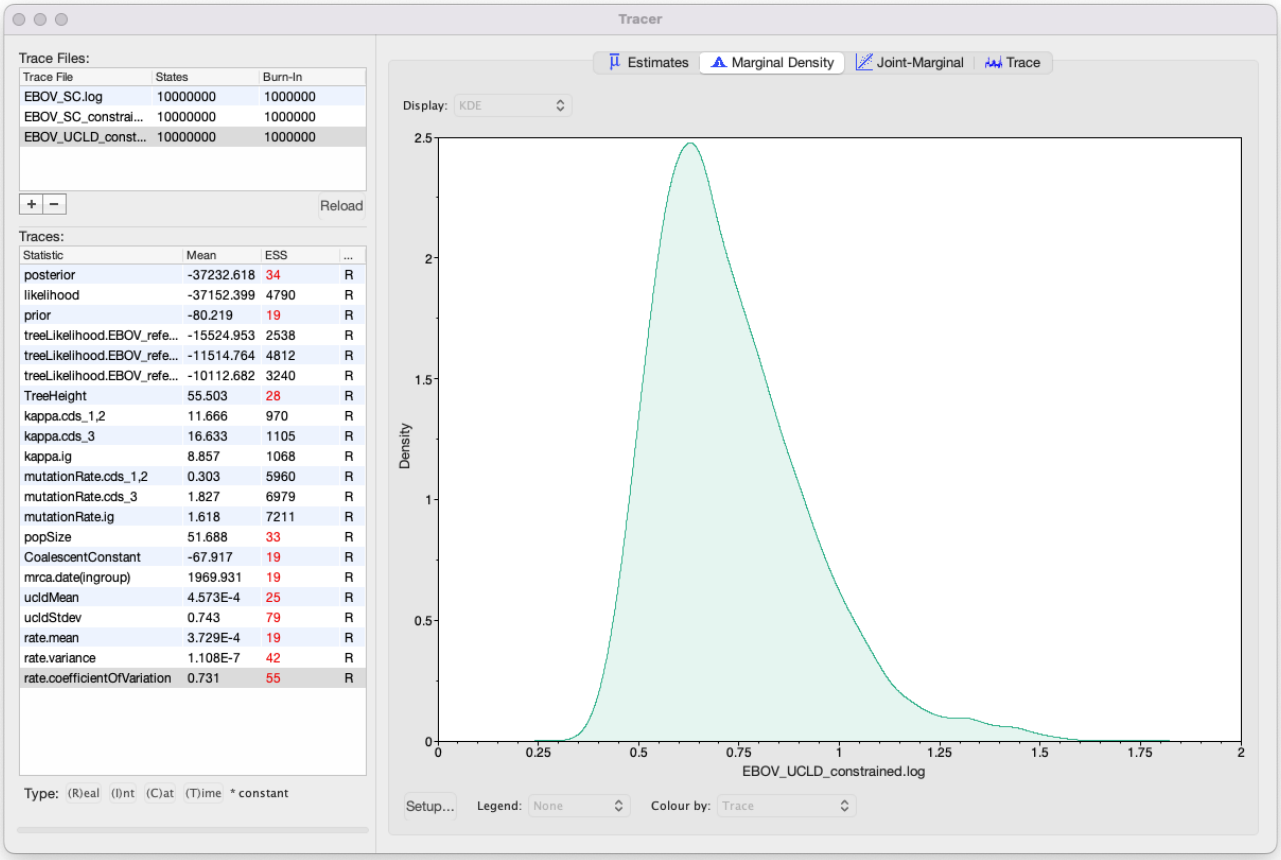

Figure 19: The coefficient of variation of the clock rate of the relaxed clock model.

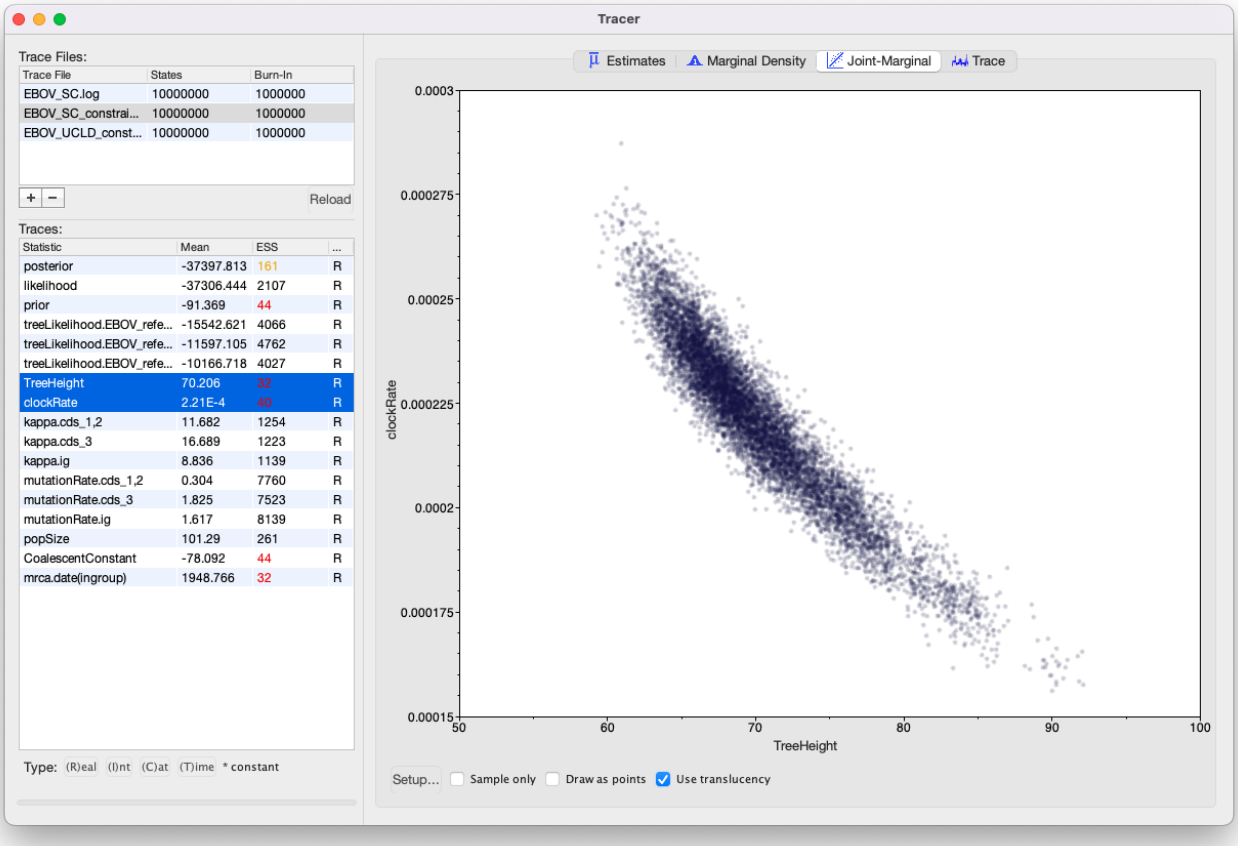

Figure 20: Correlation between the tree height and clock rate estimates.

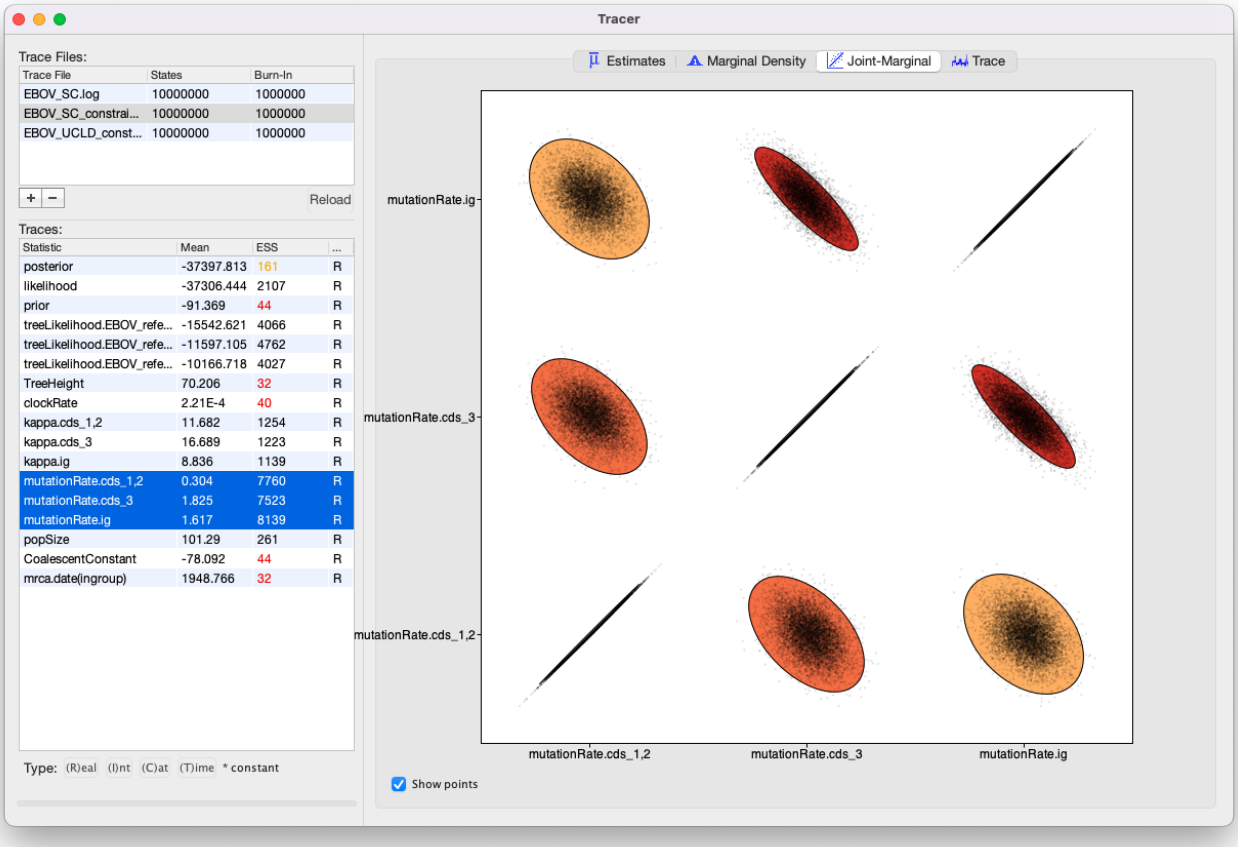

Figure 21: Correlations between the mutation rate parameters.

- Check **Show points**

The panel should look like Figure 21. The ellipses represent the covariance between pairs of parameters and make it easy to identify which pairs are correlated or anti-correlated. Is there a strong correlation or anti-correlation between some of our mutation rate parameters?

3.7.1 Visualising rates on trees

We can usue FigTree to investigate which branches on the tree are inferred to have elevated substitution rates under the relaxed clock analysis.

Open **TreeAnnotator**.

Use the same settings as before to create the MCC tree for the relaxed clock analysis and save it as `EBOV_UCLD_constrained.MCC.tree`.

Now open the MCC tree in Figtree.

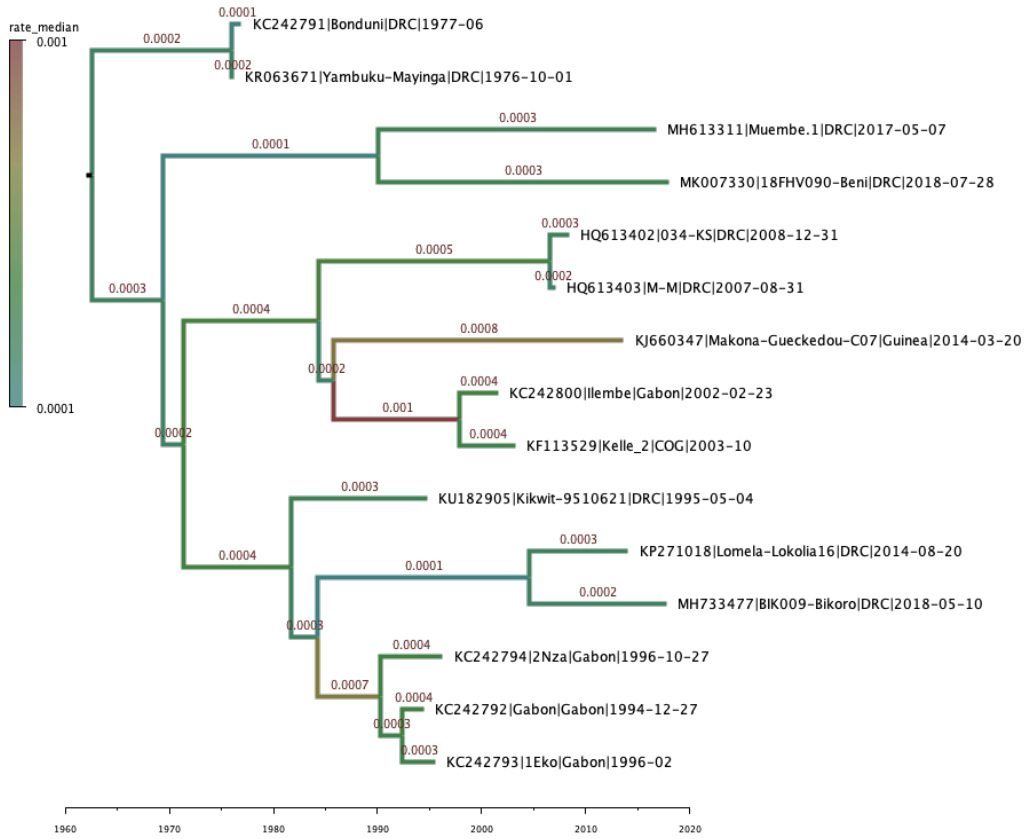

Figure 22: FigTree visualisation of the tree estimated under the relaxed clock model.

- Expand **Trees** options, check **Order nodes** and select **decreasing** from the drop-down menu.
- Expand the **Appearance** options and increase the **Line Weight** to **4**. Now open the **Colour by** drop-down menu and select `rate_median` Finally, click the **Colours** button and reverse the Hue spectrum (so faster rates are hotter and slower rates cooler colours).
- Expand the **Tip Labels** options and increase the **Font Size** until it is readable.
- Check the **Branch Labels** checkbox, expand the options and select `rate_median` from the **Display** drop-down menu.
- Increase the **Font Size** until it is readable.
- Uncheck the **Scale Bar** checkbox.
- Check the **Scale Axis** checkbox, expand the options, check **Reverse axis** and increase the **Font Size**.
- Expand the **Time Scale** options and set the offset to 2018 (approximately the most recent collection date).
- Check the **Legend** checkbox, expand the options and select `rate_median` from the **Attribute** drop-down menu

The tree should now look something like Figure 22. Note that the stem branches leading to the 2014, 2017 and 2018 outbreaks in the DRC (Muembe.1, 18FHV090-Beni, Lomela-Lokolia16 and BIK009-Bikoro have low median clock rates of 1×10^{-4} s/s/y, while the branch leading to the 2014 West African Ebola virus disease epidemic has a much faster median rate of 8×10^{-4} s/s/y.

3.8 Setting up a Gamma site model

Finally, we will return to the discrete Gamma model for modeling site-to-site rate heterogeneity. We will edit the relaxed clock analysis to also incorporate rate heterogeneity within each of the three partitions. In addition, we will also estimate the nucleotide frequencies.

Select the **Site Model** tab.

Make sure that `EBOV_reference_set_15_ig` is selected.

- Set the **Gamma Category Count** to 4.
- Check the **estimate** box for the **Shape** parameter (it should already be checked).
- Select **Estimated** from the **Frequencies** drop-down menu.
- Double-check that the **estimate** checkbox is ticked for the **Substitution Rate** and that the **Subst Model** is set to **HKY**

Now use the shortcut to clone the substitution model to all partitions and double-check that each partition now has an HKY model, with 4 Gamma categories, with the shape parameter, frequencies, Kappa and substitution rates estimated.

The site model setup should look as in Figure 23. No changes need to be made to the priors (we will use the default priors for the shape and frequency parameters). Change the log and tree file names, save the file as `EBOV_UCLD_constrained_gamma.xml` and run the analysis in BEAST2 (with seed 777).

Note that this analysis is taking a lot longer to run than the previous analyses. That is because with a Gamma category count of 4 BEAST2 needs to do approximately 4 times as many operations to calculate the likelihood. This is why we don't usually use a lot of rate categories to discretize the Gamma distribution. In practice 4 categories allow a substantial amount of variation while also not increasing the overhead too much.

There is already a pre-cooked log file of this analysis that was run a lot longer in the `precooked_runs/` folder. Load this file into Tracer and compare it to the earlier relaxed clock analysis without the Gamma model.

The **Trace** tab is primarily a diagnostic tool for checking convergence to the posterior, assessing the length of the burn-in and whether or not the chain is mixing well. There is a good argument to be made for this being the *most important* tab in the Tracer program and that it is the first tab users should look at.

Have a look at the individual parameter traces in the **Trace** tab, in both the short and long log files. Can you figure out why ESS values for some parameters are higher than others?

Do you think a burn-in of 10% is sufficient for this analysis?

Do you think adding the Gamma model and estimating the frequencies added any new insights here?

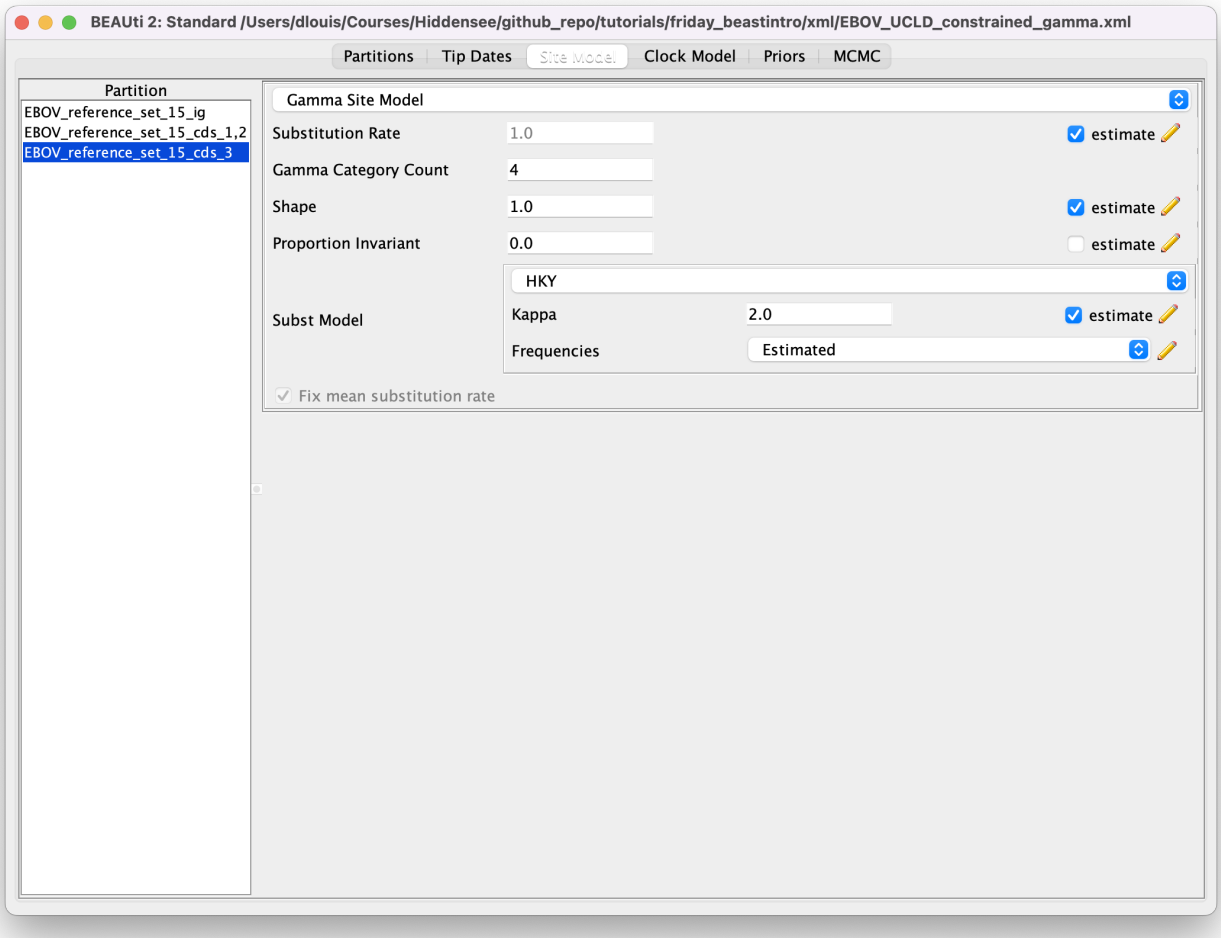

Figure 23: The site model setup with a Gamma site model.

3.9 Visualising tree posteriors (optional)

The MCC tree is one way of summarising the posterior distribution of trees as a single tree, annotated with extra information on some nodes to represent the uncertainty in the tree estimates. Just as summarising the posterior distributions of a continuous parameter as a median and credible interval throws away a lot of information (such as the shape of the distribution) a lot of information is lost when summarising a set of trees as an MCC tree. However, it is significantly more difficult to visualise the set of posterior trees.

One possibility is to use the program **DensiTree**. DensiTree does not need a summary tree (so we do not need to run TreeAnnotator prior to using DensiTree) to be able to visualise the estimates.

Open **DensiTree**. Use **File > Load** then locate and click on `EBOV_UCLD_constrained_gamma_long.trees`.

Expand the **Show** options and check the **Consensus Trees** checkbox.

You should now see many lines corresponding to all the individual trees sampled by the MCMC chain. You can also clearly see a pattern across all of the posterior trees.

In order to see the support for the topology, select the **Central** view mode.

Now expand the **Clades** menu, check the **Show clades** checkbox and the **text** checkbox for the **Support**.

The tree should look as shown in Figure 24.

You can also view all of the different clades and their posterior probabilities by selecting **Help > View clades**. In this particular run there is little uncertainty in the tree estimate with respect to clade grouping, as almost every clade has close to 100% support.

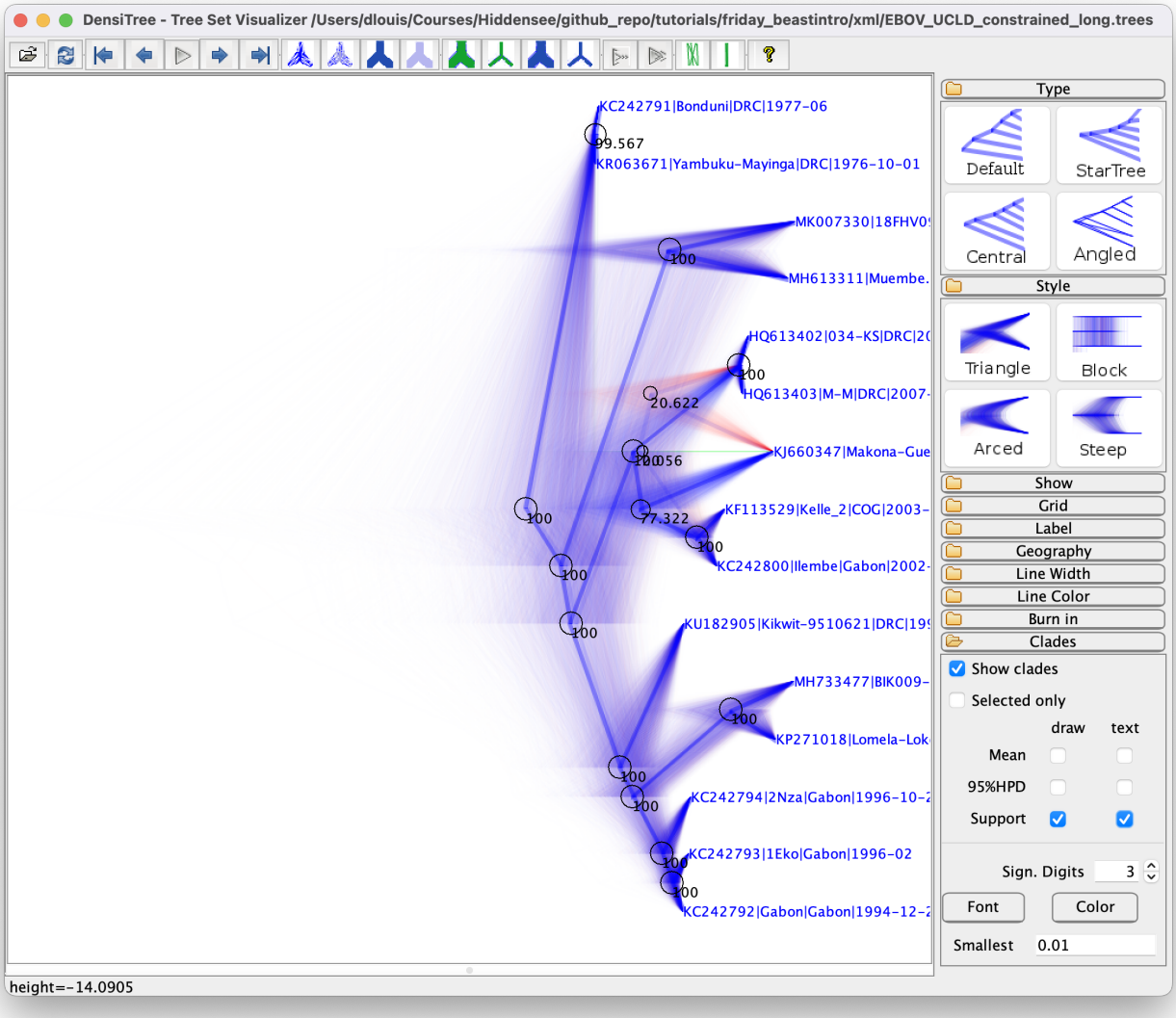

Figure 24: DensiTree visualisation of the tree sample.

Version dated: September 29, 2022

References

Blackley, DJ et al. 2016. Reduced evolutionary rate in reemerged Ebola virus transmission chains. *Science Advances* 2: e1600378.

Bouckaert, R, J Heled, D Kühnert, T Vaughan, CH Wu, D Xie, MA Suchard, A Rambaut, and AJ Drummond. 2014. Beast 2: a software platform for bayesian evolutionary analysis. *PLoS computational biology* 10: e1003537.

Bouckaert, R et al. 2019. Beast 2.5: an advanced software platform for bayesian evolutionary analysis. *PLOS Computational Biology* 15:

Diallo, B et al. 2016. Resurgence of Ebola Virus Disease in Guinea Linked to a Survivor With Virus Persistence in Seminal Fluid for More Than 500 Days. *Clinical Infectious Diseases* 63: 1353–1356.

Drummond, AJ and RR Bouckaert. 2014. *Bayesian evolutionary analysis with BEAST 2*. Cambridge University Press,

Drummond, AJ, SYW Ho, MJ Phillips, and A Rambaut. 2006. Relaxed Phylogenetics and Dating with Confidence. *PLOS Biology* 4: e88.

Dudas, G and A Rambaut. 2014. Phylogenetic Analysis of Guinea 2014 EBOV Ebolavirus Outbreak. *PLoS Currents* 6: ecurrents.outbreaks.84eefe5ce43ec9dc0bf0670f7b8b417d.

Lam, TTY, H Zhu, YL Chong, EC Holmes, and Y Guan. 2015. Puzzling Origins of the Ebola Outbreak in the Democratic Republic of the Congo, 2014. en. *Journal of Virology* 89: (Ed.) Dermody, TS, 10130–10132.

Maganga, GD et al. 2014. Ebola Virus Disease in the Democratic Republic of Congo. en. *New England Journal of Medicine* 371: 2083–2091.