

# Estimating rates and dates from time-stamped sequences

## A hands-on practical

This chapter provides a step-by-step tutorial for analyzing a set of virus sequences which have been isolated at different points in time (heterochronous data). The data are 71 sequences from the *prM/E* gene of yellow fever virus (YFV) from Africa and the Americas with isolation dates ranging from 1940-2009. The sequences represent a subset of the data set analyzed by Bryant *et al.* (Bryant JE, Holmes EC, Barrett ADT, 2007 Out of Africa: A Molecular Perspective on the Introduction of Yellow Fever Virus into the Americas. PLoS Pathog 3(5): e75. doi:10.1371/journal.ppat.0030075).

The most commonly cited hypothesis of the origin of yellow fever virus (YFV) in the Americas is that the virus was introduced from Africa, along with *Aedes aegypti* mosquitoes, in the bilges of sailing vessels during the slave trade. Although the hypothesis of a slave trade introduction is often repeated prior to paper by Bryant *et al.* (2007), it had not been subject to rigorous examination using gene sequence data and modern phylogenetic techniques for estimating divergence times. The aim of this exercise is to obtain an estimate of the rate of molecular evolution, an estimate of the date of the most recent common ancestor and to infer the phylogenetic relationships with appropriate measures of statistical support.

The first step will be to convert a NEXUS file with a DATA or CHARACTERS block into a BEAST XML input file. This is done using the program BEAUti (this stands for Bayesian Evolutionary Analysis Utility). This is a user-friendly program for setting the evolutionary model and options for the MCMC analysis. The second step is to actually run BEAST using the input file that contains the data, model and settings. The final step is to explore the output of BEAST in order to diagnose problems and to summarize the results.

To undertake this tutorial, you will need to download three software packages in a format that is compatible with your computer system (all three are available for Mac OS X, Windows and Linux/UNIX operating systems):

- **BEAST** - this package contains the BEAST program, BEAUti and a couple of utility programs. At the time of writing, the current version is v1.8.4. *BEAST* releases are generally available for download from <http://beast.bio.ed.ac.uk/>, but the latest release can now be found at <https://github.com/beast-dev/beast-mcmc/releases>.
- **Tracer** - this program is used to explore the output of **BEAST** (and other Bayesian MCMC programs). It graphically and quantitatively summarizes the empirical distributions of continuous parameters and provides diagnostic information. At the time of writing, the current version is v1.6. It is available for download from <http://beast.bio.ed.ac.uk/>.
- **FigTree** - this is an application for displaying and printing molecular phylogenies, in particular those obtained using **BEAST**. At the time of writing, the current version is v1.4.2. It is available for download from <http://tree.bio.ed.ac.uk/>.

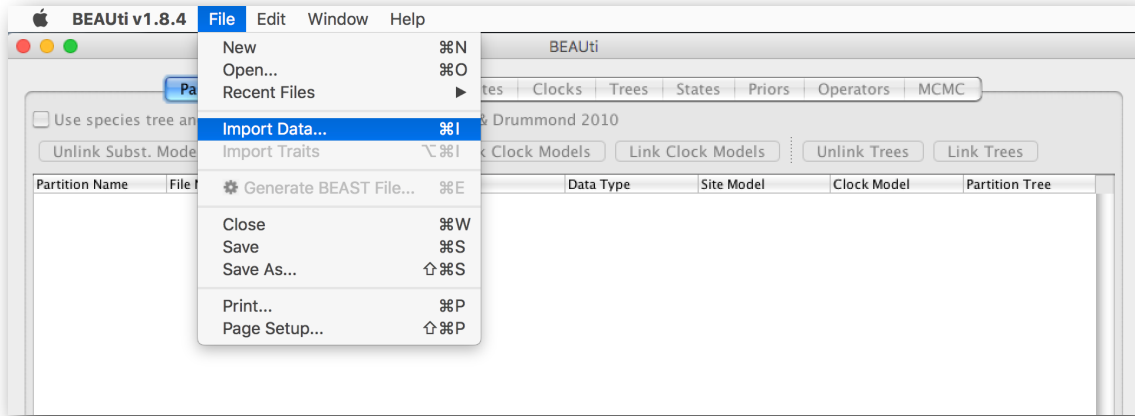
Prior to **BEAST** analysis of the YFV data, it is advisable to confirm that the sequences contain sufficient 'temporal signal' for reliable estimation of evolutionary rates and divergence times. This can be achieved using a simple exploratory regression approach as implemented in **TempEst**: <http://tree.bio.ed.ac.uk/software/tempest/> (Rambaut et al., 2016). **TempEst** takes as input a 'non-clock' phylogeny, which can be estimated using a standard neighbour-joining, maximum likelihood or Bayesian approach.

## Running BEAUti

The program **BEAUti** is a user-friendly program for setting the model parameters for BEAST. Run BEAUti by double clicking on its icon.

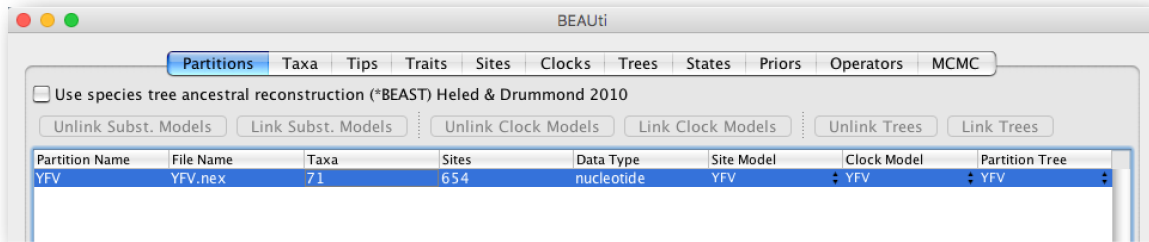
## Loading the NEXUS file

To load a NEXUS format alignment, simply select the **Import Data...** option from the **File** menu.

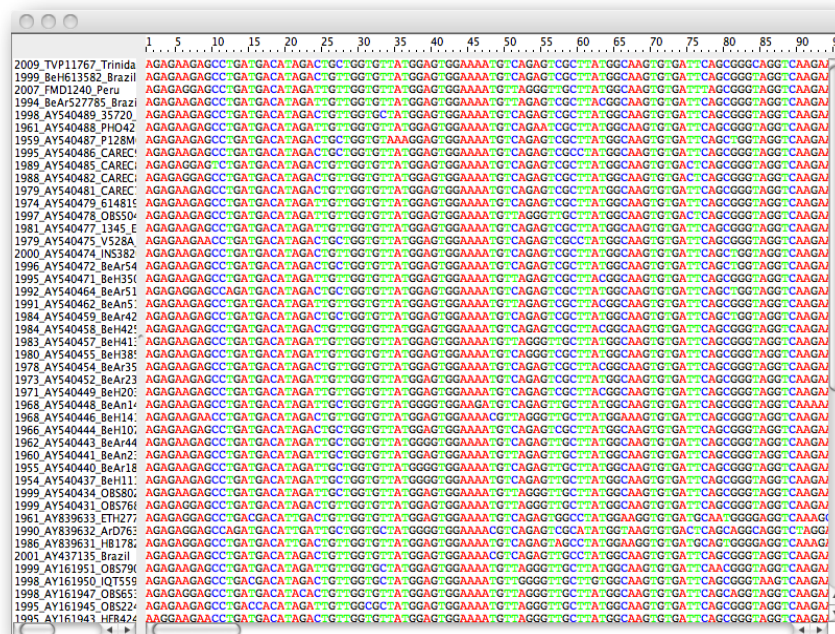


## The NEXUS alignment

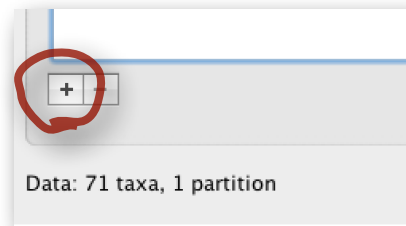
Select the file called **YFV.nex**. This file contains an alignment of 71 sequences from the *prME* gene of YFV, 654 nucleotides in length. Once loaded, the sequence data will be listed under **Data Partitions**:



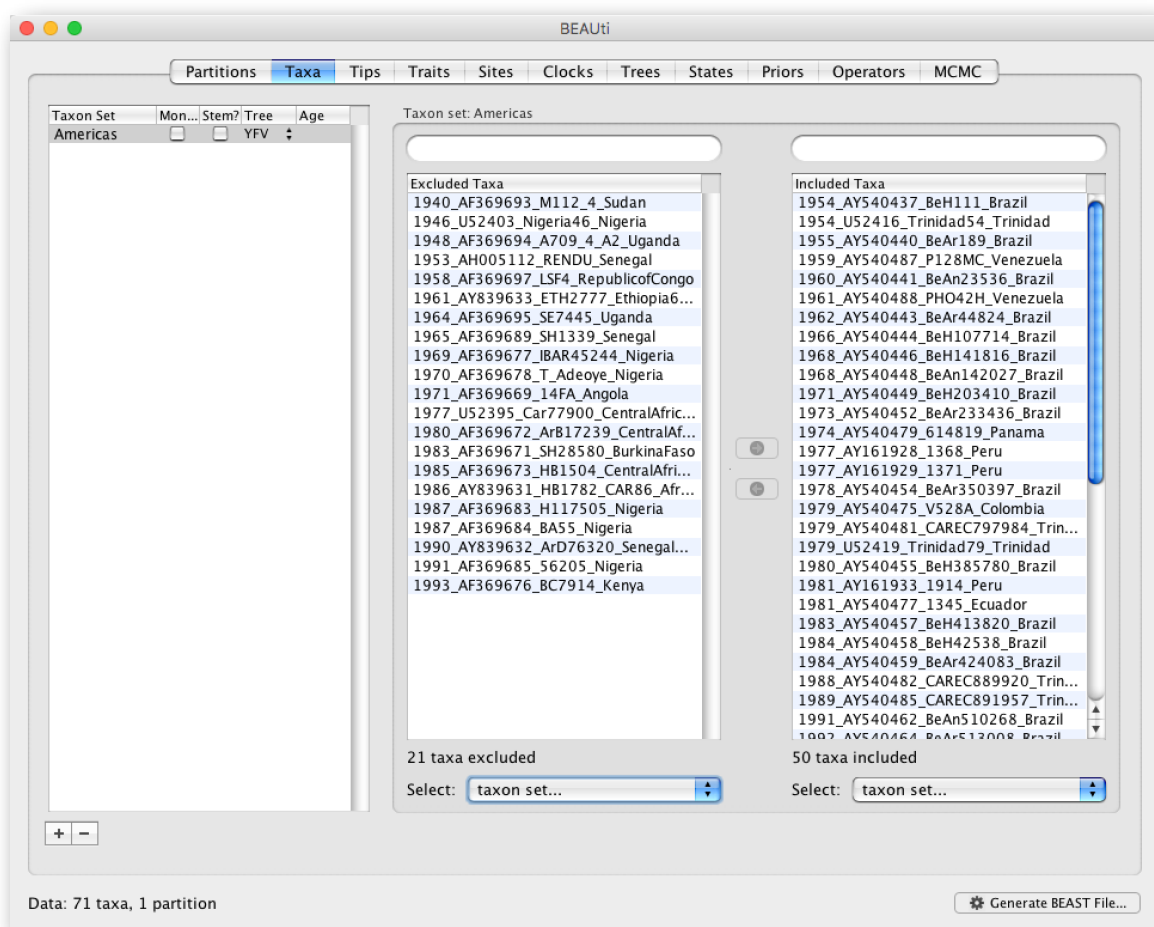
Double clicking on the YFV.nex File Name will display the alignment in a separate window:



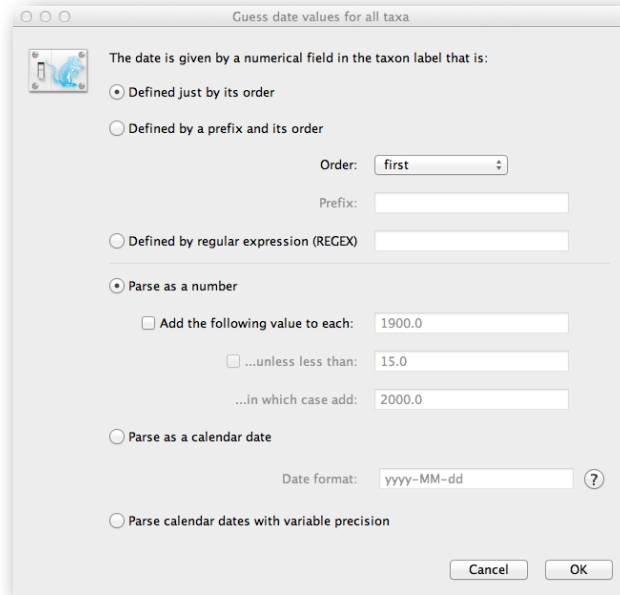
Under the **Taxa** panel, we can define sets of taxa for which we would like to obtain particular statistics, enforce a monophyletic constraint, or put calibration information on. Let's define an "Americas" taxon set by pressing the small "plus" button at the bottom left of the panel:



This will create a new taxon set. Rename it by double-clicking on the entry that appears (it will initially be called **untitled1**). Call it **Americas**. Do not enforce monophyly using the "monophyletic?" option because we will evaluate the support for this cluster. We do not opt for the "includeStem?" option either because we would like to estimate the TRMCA for the viruses from the Americas and not for the parent node leading to this clade. In the next table along you will see the available taxa. Taxa can be selected and moved to the 'Included taxa' set by pressing the green arrow button. Note that multiple taxa can be selected simultaneously holding down the cmd/ctrl button on a Mac/PC. Since most taxa are from the Americas, the most convenient is to simply select all taxa, move them to the 'Included taxa' set, and then move back the African taxa (the country of sampling is included at the end of the taxa names). After these operations, the screen should look like this:

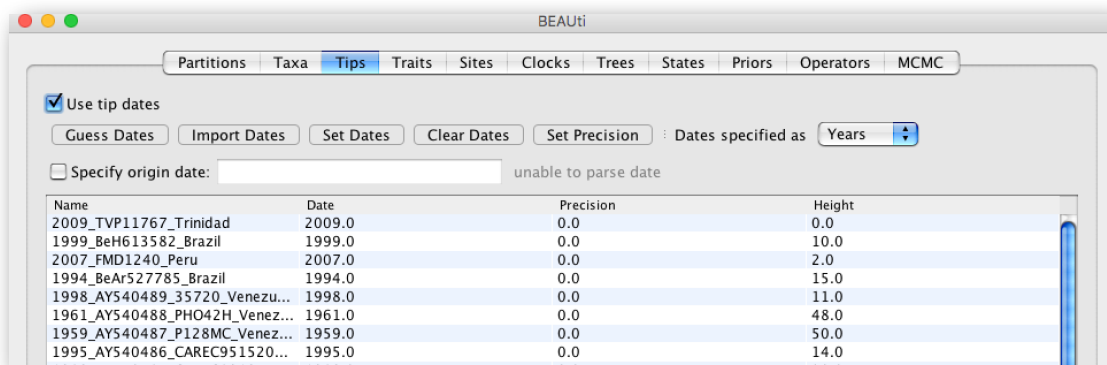


To inform BEAUti/BEAST about the sampling dates of the sequences, go to the **Tips** menu and select the "Use tip dates" option. By default all the taxa are assumed to have a date of zero (i.e. the sequences are assumed to be sampled at the same time; BEAST considers the present or most recent sampling time as time 0). In this case, the YFV sequences have been sampled at various dates going back to the 1940s. The actual year of sampling is given in the name of each taxon and we could simply edit the value in the Date column of the table to reflect these. However, if the taxa names contain the calibration information, then a convenient way to specify the dates of the sequences in BEAUti is to use the "Guess Dates" button at the top of the Data panel. Clicking this will make a dialog box appear:



This operation attempts to guess what the dates are from information contained within the taxon names. It works by trying to find a numerical field within each name. If the taxon names contain more than one numerical field (such as the some YFV sequences, above) then you can specify how to find the one that corresponds to the date of sampling. You can (1) specify the order that the date field comes (e.g., first, last or various positions in between) or (2) specify a prefix (some characters that come immediately before the date field in each name) and the order of the field, or (3) define a regular expression (REGEX). For the YFV sequences you can keep the default 'Defined just by its order' and 'Order: first' (but make sure that the 'Parse as a number' option is selected).

When parsing a number, you can ask BEAUti to add a fixed value to each guessed date. For example, the value "1900" can be added to turn the dates from 2 digit years to 4 digit. Any dates in the taxon names given as "00" would thus become "1900". However, if these "00" or "01", etc. represent sequences sampled in 2000, 2001, etc., "2000" needs to be added to those. This can be achieved by selecting the "unless less than: .." and "...in which case add:.." option adding for example 2000 to any date less than 10. There is also an option to parse calendar dates and one for calendar dates with various precisions. Because all dates are specified in a four digit format in this case, no additional settings are needed. So, we can press "OK".



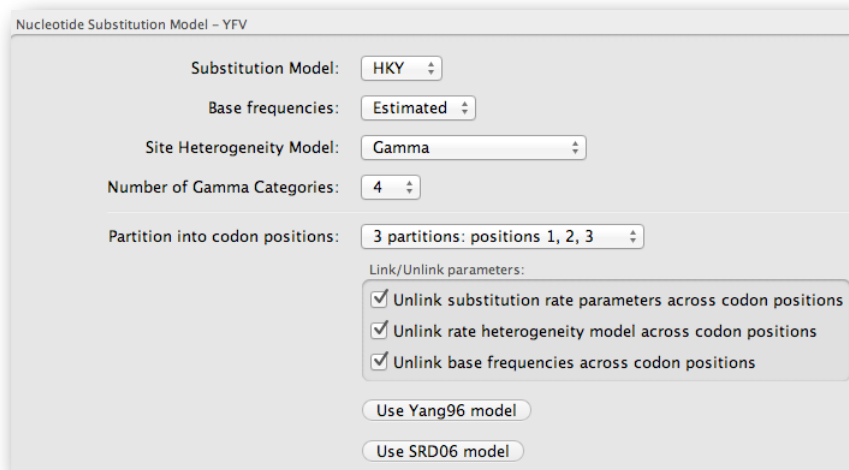
At the top of the window you can set the units that the dates are given in (years, months, days) and whether they are specified relative to a point in the past (as would be the case for years such as 1984) or backwards in time from the present (as in the case of radiocarbon ages). The “Height” column lists the ages of the tips relative to time 0 (in our case 2009). Note that there possibilities to accommodate the sampling time uncertainty (‘Tip date sampling’ at the bottom left); in this case, only the sampling years are provided and not the exact sampling dates. However, this uncertainty will negligible with respect to the relatively large evolutionary time scale of this example.

## Setting the evolutionary model

The next thing to do is to click on the **Sites** tab at the top of the main window. This will reveal the evolutionary model settings for BEAST. Exactly which options appear depend on whether the data are nucleotides or amino acids (or traits). This tutorial assumes that you are familiar with the evolutionary models available; however there are a couple of points to note about selecting a model in BEAUti:

- Selecting the **Partition into codon positions** option assumes that the data are aligned as codons. This option will then estimate a separate rate of substitution for each codon position, or for 1+2 versus 3, depending on the setting.
- Selecting the **Unlink substitution model across codon positions** will specify that BEAST should estimate a separate transition-transversion ratio or general time reversible rate matrix for each codon position.
- Selecting the **Unlink rate heterogeneity model across codon positions** will specify that BEAST should estimate set of rate heterogeneity parameters (gamma shape parameter and/or proportion of invariant sites) for each codon position.
- Selecting the **Unlink base frequencies across codon positions** will specify that BEAST should estimate a separate set of base frequencies for each codon position.

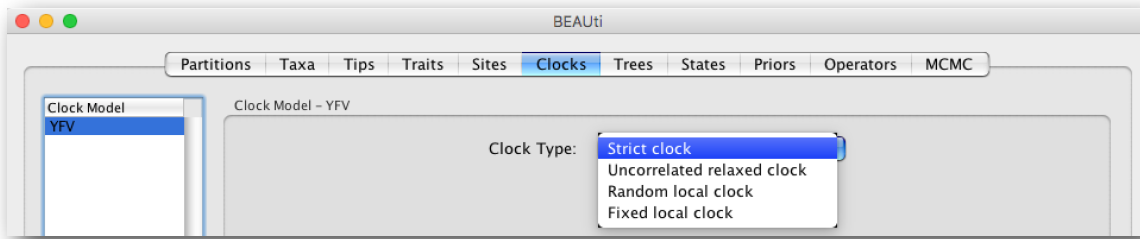
For this tutorial, select the **3 partitions: codon positions 1, 2 & 3** option so that each codon position has its own **HKY** substitution model, rate of evolution, **Estimated** base frequencies, and **Gamma**-distributed rate variation among sites:



## Setting the clock model

Click on the **Clocks** tab at the top of the main window. We will perform our initial run using the (default) strict molecular clock model:

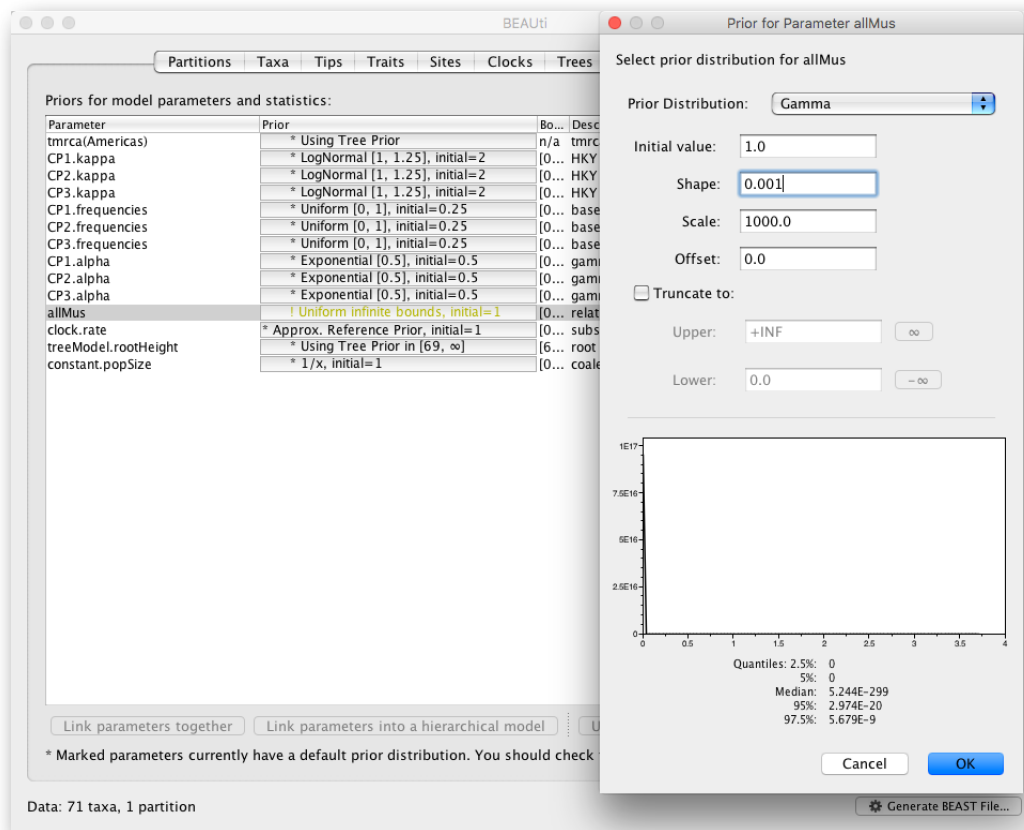
## Setting the starting tree and tree prior



Click on the **Trees** tab at the top of the main window. We keep a default random starting tree and a (simple) constant size coalescent prior. The tree priors (coalescent and other models) will be explained in other lectures.

### Setting up the priors

Review the prior settings under the **Priors** tab. Some of the default marginal priors may be improper (e.g. indicated in yellow); priors that would not have been set would appear in red. It's important to provide proper priors for all the parameters being estimated as improper priors lead to improper posteriors and improper marginal likelihoods (when performing Bayesian model selection, see further in this tutorial). To change the prior on the relative rates (allMus) for example, click on the corresponding prior and a prior selection window will appear. Set the prior to a gamma distribution with shape = 0.001 and scale = 1000. The graphical representation of this prior distribution indicates that most prior mass is put on small values, but the density remains sufficiently diffuse. Notice that the prior setting turns black after confirming this setting by clicking "OK".



Note that the default prior on the rate of evolution (clock.rate) is an approximation of a conditional reference prior (**Approx. Reference Prior**) (Ferreira and Suchard, 2008). If the sequences are not associated with different sampling dates (they are contemporaneous), or when the sampling time range is trivial for the evolutionary scale of the taxa, the substitution rate can be fixed to a value based on another source, or better, a prior distribution can be specified to also incorporate the uncertainty of this 'external' rate. Fixing the rate to 1.0 will result in the ages of the nodes of the tree being estimated in units of substitutions per site (i.e. the normal units of branch lengths in popular packages such as *MrBayes*). Note that when selecting to fix the rate to a value, the transition kernel(s) on this parameter ('Operators' panel, see next section) will be



automatically unselected.

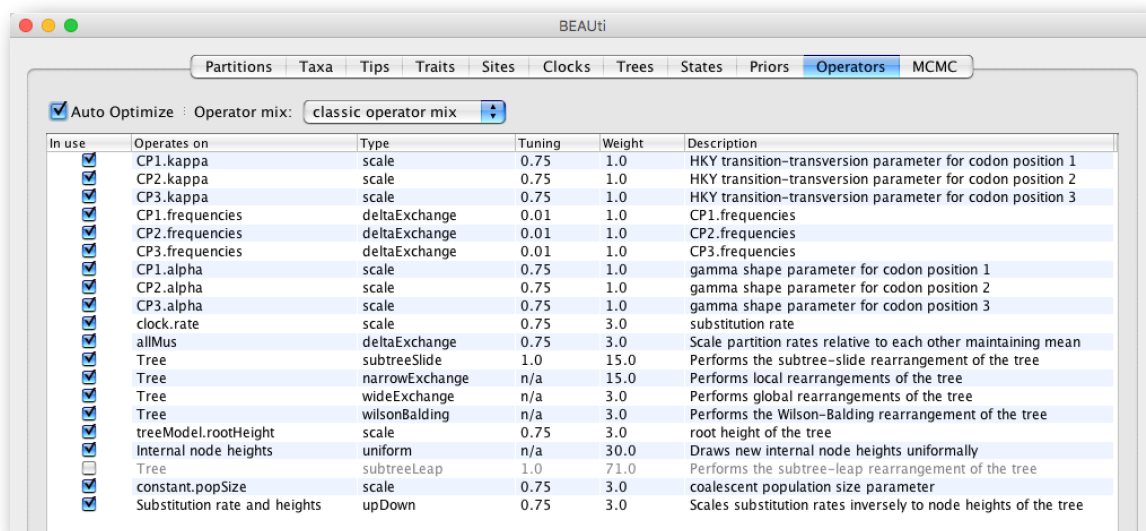
## Setting up the operators

Each parameter in the model has one or more “operators” (these are variously called moves, proposals or transition kernels by other MCMC software packages such as MrBayes and LAMARC). The operators specify how the parameters change as the MCMC runs. As of BEAST v1.8.4, different options are available w.r.t. exploring tree space. In this tutorial, we will use the ‘classic operator mix’, which consists of a set of tree transition kernels that propose changes to the tree. There is also an option to fix the tree topology as well as a ‘new experimental mix’, which is currently under development with the aim to improve mixing for large phylogenetic trees.

The operators tab in BEAUti has a table that lists the parameters, their operators and the tuning settings for these operators. In the first column are the parameter names. These will be called things like **CP1.kappa** which means the HKY model's kappa parameter (the transition-transversion bias) for the first codon position. The next column has the type of operators that are acting on each parameter. For example, the scale operator scales the parameter up or down by a random proportion and the uniform operator simply picks a new value uniformly within a range. Some parameters related to the tree or to the node ages in the tree are associated with specific operators.

The next column, labelled **Tuning**, gives a tuning setting to the operator. Some operators don't have any tuning settings so have **n/a** under this column. The tuning parameter will determine how large a move each operator will make which will affect how often that change is accepted by the MCMC which will affect the efficiency of the analysis. For most operators (like the subtree slide operator) a larger tuning parameter means larger moves. However for the scale operator a tuning parameter value closer to 0.0 means bigger moves. At the top of the window is an option called **Auto Optimize** which, when selected, will automatically adjust the tuning setting as the MCMC runs to try to achieve maximum efficiency. At the end of the run a table of the operators, their performance and the final values of these tuning settings can be written to standard output.

The next column, labelled **Weight**, specifies how often each operator is applied relative to the others. Some parameters tend to be sampled very efficiently from their target distribution (e.g. the kappa parameter); these parameters can have their operators down-weighted so that they are not changed as often. We will start by using the default settings for this analysis.



The screenshot shows the BEAUti interface with the 'Operators' tab selected. The 'Auto Optimize' checkbox is checked, and the 'Operator mix' is set to 'classic operator mix'. Below this is a table with the following columns: 'In use', 'Operates on', 'Type', 'Tuning', 'Weight', and 'Description'.

In use	Operates on	Type	Tuning	Weight	Description
<input checked="" type="checkbox"/>	CP1.kappa	scale	0.75	1.0	HKY transition-transversion parameter for codon position 1
<input checked="" type="checkbox"/>	CP2.kappa	scale	0.75	1.0	HKY transition-transversion parameter for codon position 2
<input checked="" type="checkbox"/>	CP3.kappa	scale	0.75	1.0	HKY transition-transversion parameter for codon position 3
<input checked="" type="checkbox"/>	CP1.frequencies	deltaExchange	0.01	1.0	CP1.frequencies
<input checked="" type="checkbox"/>	CP2.frequencies	deltaExchange	0.01	1.0	CP2.frequencies
<input checked="" type="checkbox"/>	CP3.frequencies	deltaExchange	0.01	1.0	CP3.frequencies
<input checked="" type="checkbox"/>	CP1.alpha	scale	0.75	1.0	gamma shape parameter for codon position 1
<input checked="" type="checkbox"/>	CP2.alpha	scale	0.75	1.0	gamma shape parameter for codon position 2
<input checked="" type="checkbox"/>	CP3.alpha	scale	0.75	1.0	gamma shape parameter for codon position 3
<input checked="" type="checkbox"/>	clock.rate	scale	0.75	3.0	substitution rate
<input checked="" type="checkbox"/>	allMus	deltaExchange	0.75	3.0	Scale partition rates relative to each other maintaining mean
<input checked="" type="checkbox"/>	Tree	subtreeSlide	1.0	15.0	Performs the subtree-slide rearrangement of the tree
<input checked="" type="checkbox"/>	Tree	narrowExchange	n/a	15.0	Performs local rearrangements of the tree
<input checked="" type="checkbox"/>	Tree	wideExchange	n/a	3.0	Performs global rearrangements of the tree
<input checked="" type="checkbox"/>	Tree	wilsonBalding	n/a	3.0	Performs the Wilson-Balding rearrangement of the tree
<input checked="" type="checkbox"/>	treeModel.rootHeight	scale	0.75	3.0	root height of the tree
<input checked="" type="checkbox"/>	Internal node heights	uniform	n/a	30.0	Draws new internal node heights uniformly
<input type="checkbox"/>	Tree	subtreeLeap	1.0	71.0	Performs the subtree-leap rearrangement of the tree
<input checked="" type="checkbox"/>	constant.popSize	scale	0.75	3.0	coalescent population size parameter
<input checked="" type="checkbox"/>	Substitution rate and heights	upDown	0.75	3.0	Scales substitution rates inversely to node heights of the tree

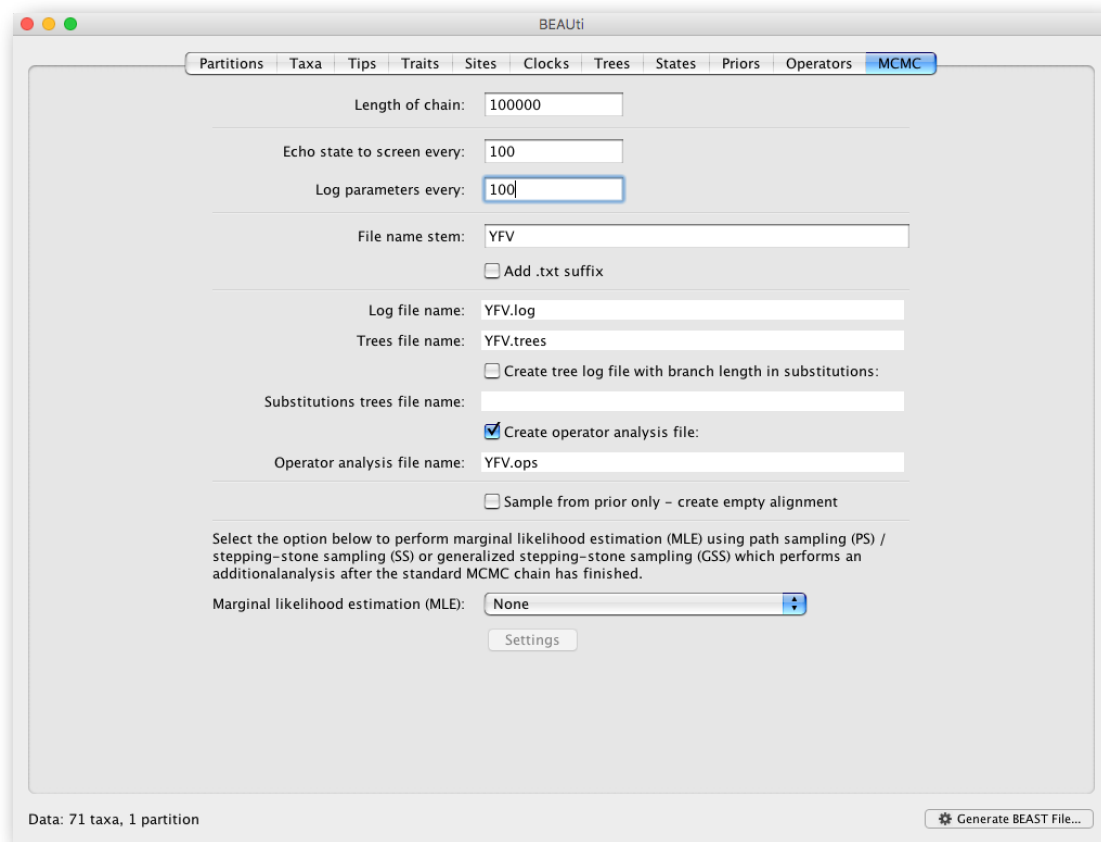
## Setting the MCMC options

The **MCMC** tab in BEAUti provides settings to control the MCMC chain. Firstly we have the **Length of chain**. This is the number of steps the MCMC will make in the chain before finishing. How long this should depend 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 entirely arbitrary

and should be adjusted according to the size of your dataset. We will see later how the resulting log file can be analyzed using Tracer in order to examine whether a particular chain length is adequate.

The next couple of options specify how often the current parameter values should be displayed on the screen and recorded in the log file. The screen output is simply for monitoring the program's progress so can be set to any value (although if set too small, the sheer quantity of information being displayed on the screen will slow the program down). For the log file, the value 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 precision of the estimates. Sample too infrequently and the log file will not contain much information about the distributions of the parameters. You probably want to aim to store no more than 10,000 samples so this should be set to something  $\geq$  chain length / 10,000.

For this dataset let's initially set the chain length to 100,000 as this will run reasonably quickly on most modern computers. Although the suggestion above would indicate a lower sampling frequency, in this case set both the sampling frequencies to 100.



The next option allows the user to set the File stem name; if not set to 'YFV' by default, you can type this in here. The next two options give the file names of the log files for the parameters and the trees. These will be set to a default based on the file stem name. Let's also create an operator analysis file by selecting the relevant option. An option is also available to sample from the prior only, which can be useful to evaluate how divergent our posterior estimates are when information is drawn from the data. Here, we will not select this option, but analyze the actual data. Finally, one can select to perform marginal likelihood estimation to assess model fit; we will return to this later in this tutorial.

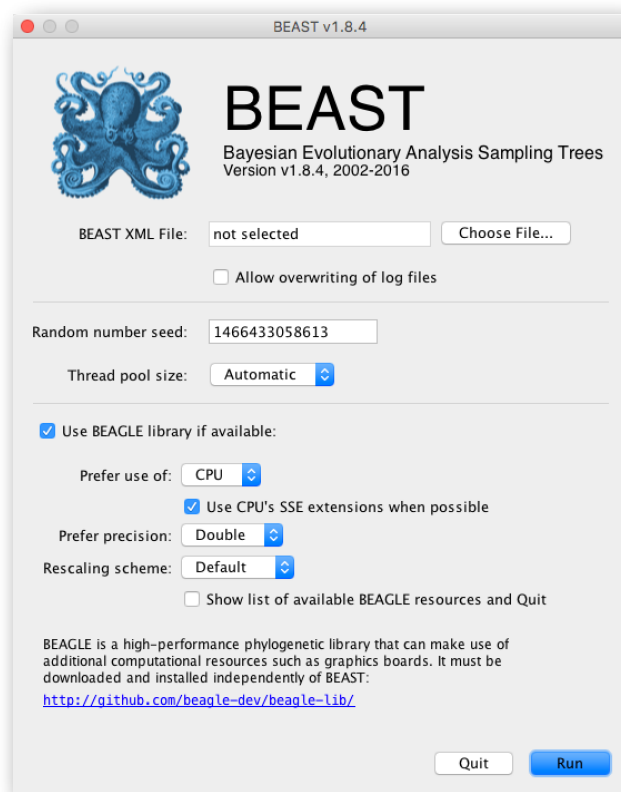
At this point we are ready to generate a BEAST XML file and to use this to run the Bayesian evolutionary analysis. To do this, either select the **Generate BEAST File...** option from the File menu or click the similarly labelled button at the bottom of the window. BEAUti will ask you to review the prior settings one more time before saving the file (and indicate that some are improper). Continue and choose a name for the file (for example, YFV.xml) and save the file.

**For convenience, leave the BEAUti window open so that you can change the values and re-generate the BEAST file as required later in this tutorial.**



## Running BEAST

Once the BEAST XML file has been created the analysis itself can be performed using BEAST. The exact instructions for running BEAST depends on the computer you are using, but in most cases a standard file dialog box will appear in which you select the XML file: If the command line version is being used then the name of the XML file is given after the name of the BEAST executable. When you have installed the BEAGLE library ([code.google.com/p/beagle-lib/](http://code.google.com/p/beagle-lib/)), you can use this in conjunction with BEAST to speed up the calculations. If not installed, unselect the use of the BEAGLE library. When pressing “Run”, the analysis will be performed with detailed information about the progress of the run being written to the screen. When it has finished, the log file and the trees file will have been created in the same location as your XML file.

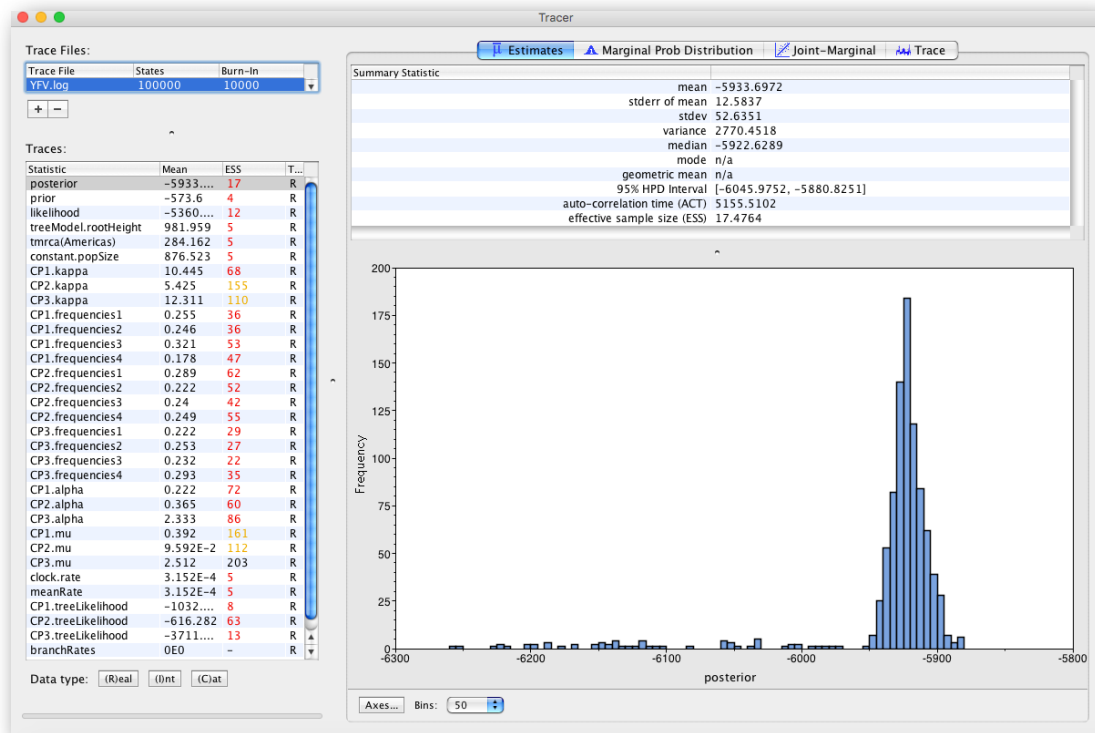


## Analysing the BEAST output

To analyze the results of running BEAST we are going to use the program **Tracer**. The exact instructions for running Tracer differs depending on which computer you are using. Please see the README text file that was distributed with the version you downloaded. Once running, Tracer will look similar irrespective of which computer system it is running on.

Select the **Import Trace File...** option from the **File** menu. If you have it available, select the log file that you created in the previous section. The file will load and you will be presented with a window similar to the one below. Remember that MCMC is a stochastic algorithm so the actual numbers will not be exactly the same.

On the left hand side is the name of the log file loaded and the traces that it contains. There are traces for a quantity proportional to posterior (this is the product of the data likelihood and the prior probabilities, on the log-scale), and the continuous parameters. Selecting a trace on the left brings up analyses for this trace on the right hand side depending on tab that is selected. When first opened, the **posterior** trace is selected and various statistics of this trace are shown under the **Estimates** tab.



In the top right of the window is a table of calculated statistics for the selected trace. The statistics and their meaning are described in the table below.

**Mean** - The mean value of the samples (excluding the burn-in).

**Stdev of mean** - The standard error of the mean. This takes into account the effective sample size so a small ESS will give a large standard error.

**Median** - The median value of the samples (excluding the burn-in).

**Geometric mean** - The central tendency or typical value of the set of samples (excluding the burn-in).

**95% HPD Lower** - The lower bound of the highest posterior density (HPD) interval. The HPD is the shortest interval that contains 95% of the sampled values.

**95% HPD Upper** - The upper bound of the highest posterior density (HPD) interval.

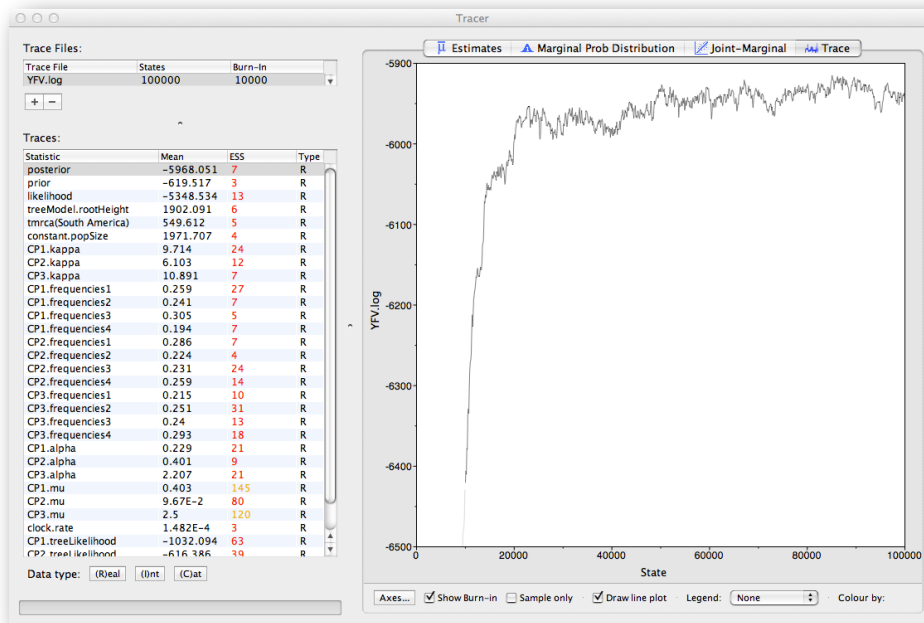
**Auto-Correlation Time (ACT)** - The average number of states in the MCMC chain that two samples have to be separated by for them to be uncorrelated (i.e. independent samples from the posterior). The ACT is estimated from the samples in the trace (excluding the burn-in).

**Effective Sample Size (ESS)** - The effective sample size (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.

Note that the effective sample sizes (ESSs) for all the traces are small (ESSs less than 100 are highlighted in red by Tracer and values > 100 but < 200 are in yellow). This is not good. A low ESS means that the trace contained a lot of correlated samples and thus may not represent the posterior distribution well. In the bottom right of the window is a frequency plot of the samples which is expected given the low ESSs is extremely rough.

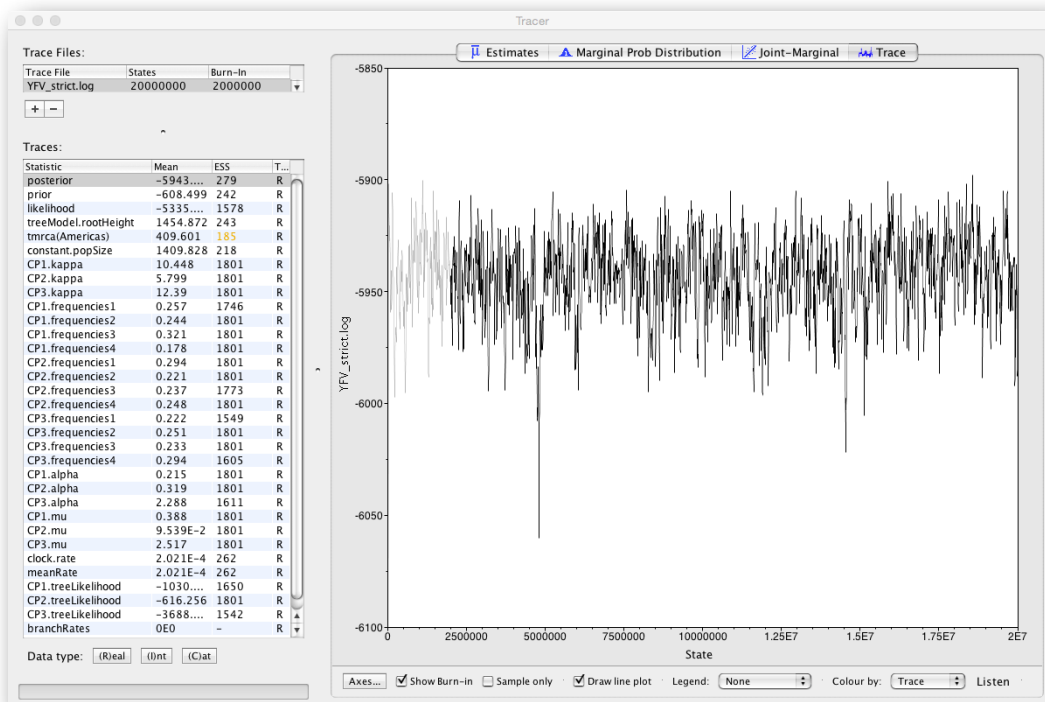
If we select the tab on the right-hand-side labelled 'Trace' we can view the raw trace, that is, the sampled values against the step in the MCMC chain.

Here you can see how the samples are correlated. There are 1000 samples in the trace (we ran the MCMC for 100,000 steps sampling every 100) but it is clear that adjacent samples often tend to have similar values. The ESS for the age of the



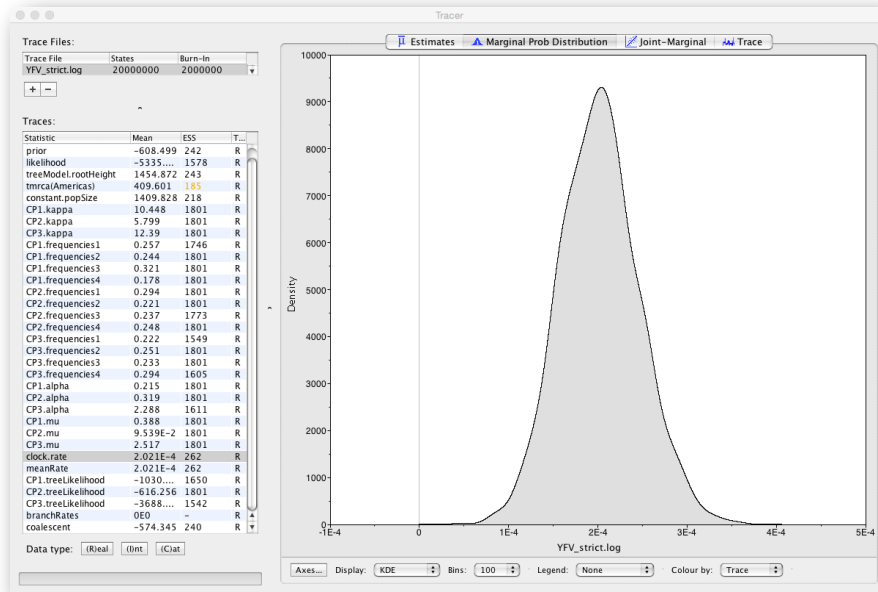
root (**treeModel.rootHeight** is about 6 so we are only getting 1 independent sample to every 167 actual samples). It also seems that the default burn-in of 10% of the chain length is inadequate (the posterior values are still increasing over most of the chain). Not excluding enough of the start of the chain as burn-in will bias the results and render estimates of ESS unreliable.

The simple response to this situation is that we need to run the chain for longer. Go back to the **MCMC Options** section, above, and create a new **BEAST XML** file with a longer chain length (e.g. 10.000.000). To continue the tutorial without having to wait for a long run to complete, you can make use of the log files provided with this tutorial (chain length of 20.000.000 and logged every 10,000 sample). Import the log file for the strict clock analysis and click on the **Trace** tab and look at the raw trace plot.

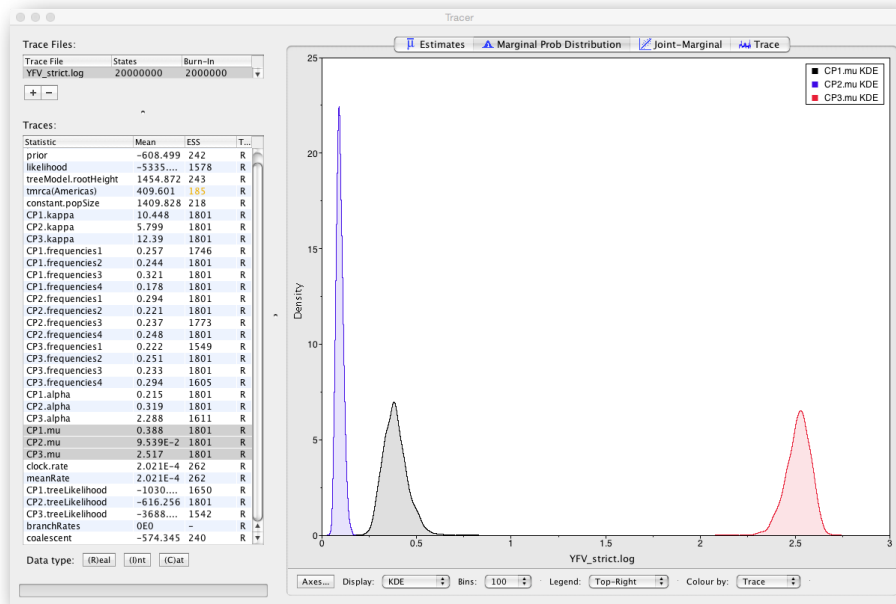


The log file provided contains 2000 samples and with an ESS of 262 for the **clock.rate** there is still some degree of auto-correlation between the samples but 262 effectively independent samples will now provide a reasonable estimate of the posterior distribution. There are no obvious trends in the plot which would suggest that the MCMC has not yet converged, and there are no large-scale fluctuations in the trace which would suggest poor mixing.

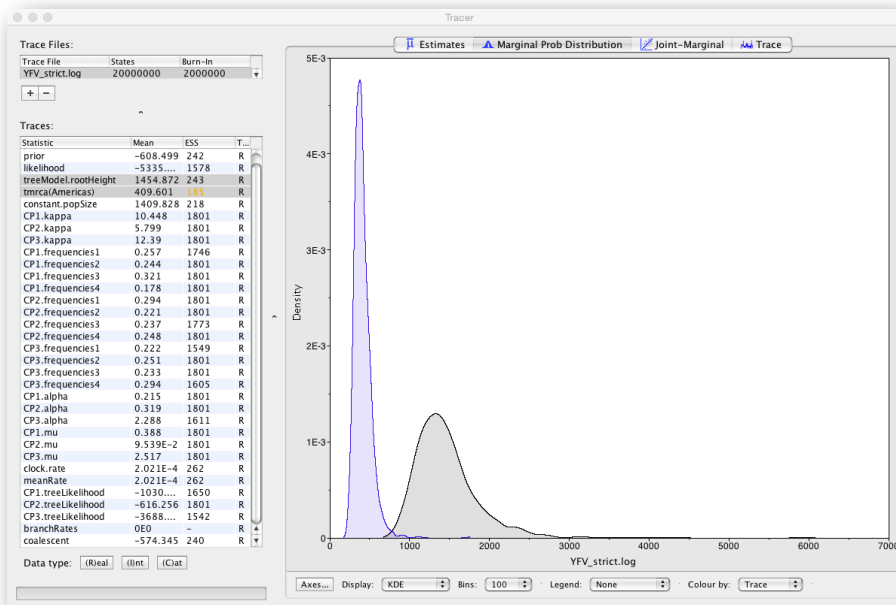
As we are happy with the behavior of posterior probability we can now move on to one of the parameters of interest: substitution rate. Select **clock.rate** in the left-hand table. This is the average substitution rate across all sites in the alignment. Now choose the density plot by selecting the tab labeled Marginal Prob Distribution. This plot shows a kernel density estimate (KDE) of the posterior probability density of this parameter. You should see a plot similar to this:



As you can see the posterior probability density is nicely bell-shaped. When looking at the equivalent histogram in the Estimates panel, there is some sampling noise which is smoothed by the KDE; this would be reduced if we ran the chain for longer but we already have a reasonable estimate of the mean and HPD interval. You can overlay the density plots of multiple traces in order to compare them (it is up to the user to determine whether they are comparable on the the same axis or not). Select the relative substitution rates for all three codon positions in the table to the left (labelled CP1.mu, CP2.mu and CP3.mu) and select 'Top-Right' under Legend. This will show the posterior probability densities for the relative substitution rate at all three codon positions overlaid:



Note that the three rates are markedly different, what does this tell us about the selective pressure on this gene? Now, let's have a look at the time to the most recent common ancestor (tmrca) for the strains from the Americas relative to the general tmrca:



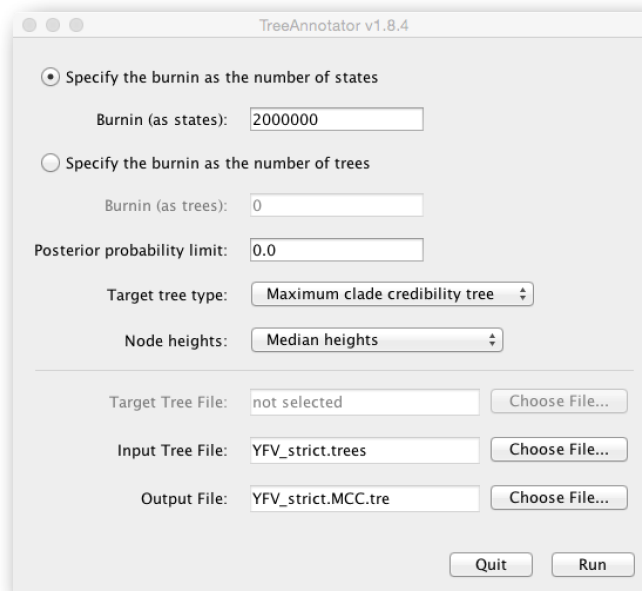
This indicates that the tmrca for the Americas is significantly younger than the root height and argues for more recent origin of YFV in the Americas.

## Summarizing the trees

We have seen how we can diagnose our MCMC run using Tracer and produce estimates of the marginal posterior distributions of parameters of our model. However, BEAST also samples trees (either phylogenies or genealogies) at the same time as the other parameters of the model. These are written to a separate file called the 'trees' file. This file is a standard NEXUS format file. As such it can easily be loaded into other software in order to examine the trees it contains. One possibility is to load the trees into a program such as PAUP\* and construct a consensus tree in a similar manner to

summarizing a set of bootstrap trees. In this case, the support values reported for the resolved nodes in the consensus tree will be the posterior probability of those clades.

In this tutorial, however, we are going to use a tool that is provided as part of the BEAST package to summarize the information contained within our sampled trees. The tool is called **TreeAnnotator** and once running, you will be presented with a window like the one below.



TreeAnnotator takes a single 'target' tree and annotates it with the summarized information from the entire sample of trees. The summarized information includes the average node ages (along with the HPD intervals), the posterior support and the average rate of evolution on each branch (for models where this can vary). The program calculates these values for each node or clade observed in the specified 'target' tree.

- **Burnin** - This is either the number of states or the number of trees in the input file that should be excluded from the summarization. This value is given as the number of trees rather than the number of steps in the MCMC chain. Thus for the example above, with a chain of 20,000,000 steps, a burn-in of 10% as the number of states can be specified as 2,000,000 states. Alternatively, with a chain of 20,000,000 steps, sampling every 10000 steps, there are 2000 trees in the file. To obtain a 10% burnin, set the burnin as the number of trees value to 200.
- **Posterior probability limit** - This is the minimum posterior probability for a node in order for TreeAnnotator to store the annotated information. Keep this value to the default of 0.0 to summarize all nodes in the target tree.
- **Target tree type** - This has two options "Maximum clade credibility" or "User target tree". For the latter option, a NEXUS tree file can be specified as the Target Tree File, below. For the former option, TreeAnnotator will examine every tree in the Input Tree File and select the tree that has the *highest product of the posterior probabilities* of all its nodes.
- **Node heights** - This option specifies what node heights (times) should be used for the output tree. If the 'Keep target heights' is selected, then the node heights will be the same as the target tree. Node heights can also be summarised as a Mean or a Median over the sample of trees. Sometimes a mean or median height for a node may actually be higher than the mean or median height of its parental node (because particular ancestral-descendent relationships in the MCC tree may still be different compared to a large number of other tree sampled). This will result in artifactual negative branch lengths, but can be avoided by the 'Common Ancestor heights' option. Let's use the default Median heights for our summary tree.
- **Target Tree File** - If the "User target tree" option is selected then you can use "Choose File..." to select a NEXUS file containing the target tree.

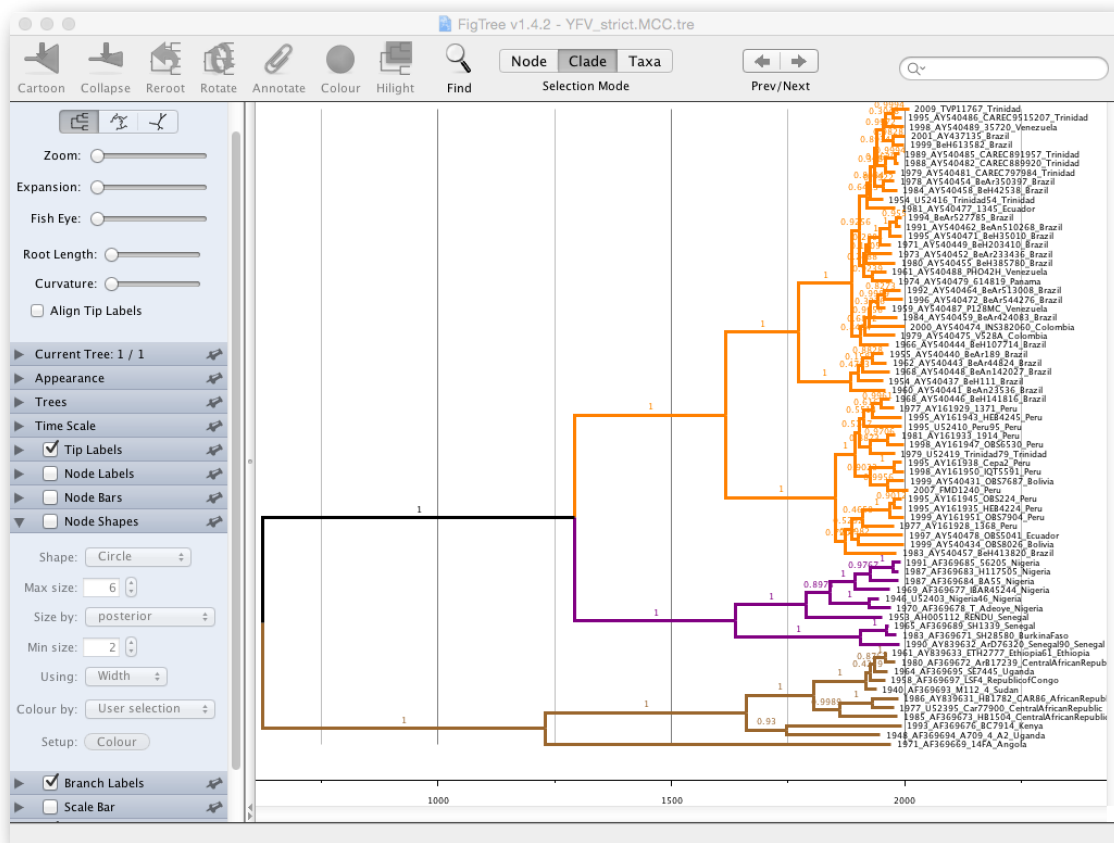


- **Input Tree File** - Use the "Choose File..." button to select an input trees file. This will be the trees file produced by BEAST.
- **Output File** - Select a name for the output tree file (e.g., YFV.MCC.tre).

Once you have selected all the options above, press the "Run" button. TreeAnnotator will analyze the input tree file and write the summary tree to the file you specified. This tree is in standard NEXUS tree file format so may be loaded into any tree drawing package that supports this. However, it also contains additional information that can only be displayed using the FigTree program.

## Viewing the annotated tree

Run **FigTree** now and select the **Open...** command from the **File** menu. Select the tree file you created using TreeAnnotator in the previous section. The tree will be displayed in the **FigTree** window. On the left hand side of the window are the options and settings which control how the tree is displayed. In this case we want to display the posterior probabilities of each of the clades present in the tree and estimates of the age of each node. In order to do this you need to change some of the settings.



First, re-order the node order by Increasing Node Order under the Tree Menu. Click on **Branch Labels** in the control panel on the left and open its section by clicking on the arrow on the left. Now select **posterior** under the **Display** option. The relative magnitude of such annotations can also be represented by node shapes.

We can also plot a time scale axis for this evolutionary history (select 'Scale Axis' and deselect 'Scale bar'). For appropriate scaling, open the 'Time Scale' section of the control panel, set the 'Offset' to 2009.0, the scale factor to -1.0, and 'Reverse Axis' under 'Scale Axis'.

Finally, open the **Appearance** panel and alter the **Line Weight** to draw the tree with thicker lines. You can also color clades by selecting a branch, select the 'Clade' selection mode and choose a color. None of the options actually alter the tree's topology or branch lengths in anyway so feel free to explore the options and settings (Highlight or collapse for example the

Americas clade). You can also save the tree and this will save most of your settings so that when you load it into FigTree again it will be displayed almost exactly as you selected. The tree can also be exported to a graphics file (pdf, eps, etc.).

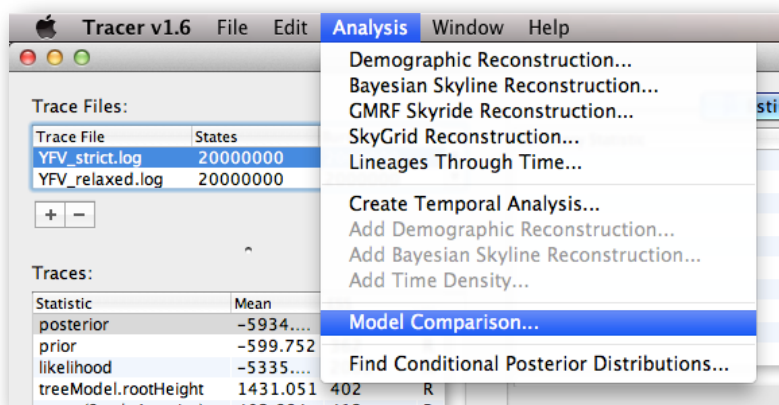
How do the viruses from the Americas cluster relative to the African viruses and what conclusions can we draw from the inferred time scale?

## Evaluating rate variation (using model selection)

To investigate lineage-specific rate heterogeneity in this data set and its impact on divergence date estimates, a log and trees file is available for an analysis using an uncorrelated lognormal relaxed clock. Import this log file in **Tracer** in addition to the previously imported strict clock log file. Investigate the posterior density for the lognormal standard deviation; if this density excludes zero (= no rate variation), it would suggest that the strict clock model can be rejected in favor of the relaxed clock model.

A more formal test can be performed using a marginal likelihood estimator (MLE, Suchard et al., 2001, MBE 18: 1001-1013), which employs a mixture of model prior and posterior samples (Newton and Raftery 1994). The ratio of marginal likelihoods defines a Bayes factor, which measures the relative fit for two different models given the data at hand. Accurate estimation of the marginal likelihood is however not possible using Tracer, which has been shown on many occasions (Baele et al., 2012, 2013, 2016).

The harmonic mean estimator (HME) unfortunately remains a frequently used method to obtain marginal likelihood estimates, in large part because it's so easily computed but it is more and more being disregarded as a reliable marginal likelihood estimator. Should you still choose to use the HME, select both trace files in Tracer and choose "Model Comparison..." from the "Analysis" menu.



Keep the default likelihood traces, select 'harmonic mean' as Analysis type and set the bootstrap replicates to '0' (as this does not provide a good estimate of the uncertainty of the MLE estimate anyway) and press 'OK'. Note that only the likelihood trace can be used in these calculations, any other trace will produce meaningless results. After a few seconds, log marginal likelihood estimates and log Bayes factors will appear in a Bayes Factors window. Which model is favored according to the log marginal likelihood estimates using the HME? The log Bayes factors are relatively convincing in this case, but it should be noted that the HME does not yield accurate MLEs and is therefore deemed to be obsolete.

Note that a different estimator of model fit is also available in Tracer: the AICM (Akaike Information Criterion through MCMC). This estimator does not yield an estimate of the marginal likelihood and can hence not be used to compute the Bayes Factor of one model compared to another. Instead, it computes an 'AICM score', with a higher AICM score indicating better performance of the model. Which model does the AICM favor? Again, note that only the likelihood trace should be used in the computation of the AICM. Simulations have so far shown that the AICM is only a slightly better choice to compare models than the HME (Baele et al., 2012), when choosing among demographic models. To compare molecular clock models, both HME and AICM have been shown to be unreliable (Baele et al., 2013).

While not available directly in Tracer, **BEAST** allows to compute the stabilized/smoothed harmonic mean estimator (sHME, Redelings and Suchard 2005). We here use its calculation to illustrate how to create/use a simple BEAST XML file that reads the output log file of a standard MCMC analysis and produces estimate of the HME, sHME and AICM. In both folders containing the large output files provided, there is a BEAST XML file: calculate-HME-sHME-AICM.xml. Note that the content of these files is different in both folders (depending on the name of the log file and the burn-in). Note that the HME and AICM could be calculated by loading the log files into Tracer. In some cases, for example when working on a terminal to use a cluster, this might not be possible and the provided XML file can then be used to compute the different estimates.

Load the xml files into **BEAST**. Note that all the necessary samples are already present in the log files, so no more MCMC chains need to be run. The samples will now merely be used to compute the model fit estimators, which only takes a small amount of time. Does the sHME provide a different outcome than the HME? You can try to adapt these XML files to obtain an sHME estimate of the short runs comprising 100,000 iterations earlier in this practical.

More accurate/reliable MLE estimates can be obtained using computationally more demanding approaches, such as:

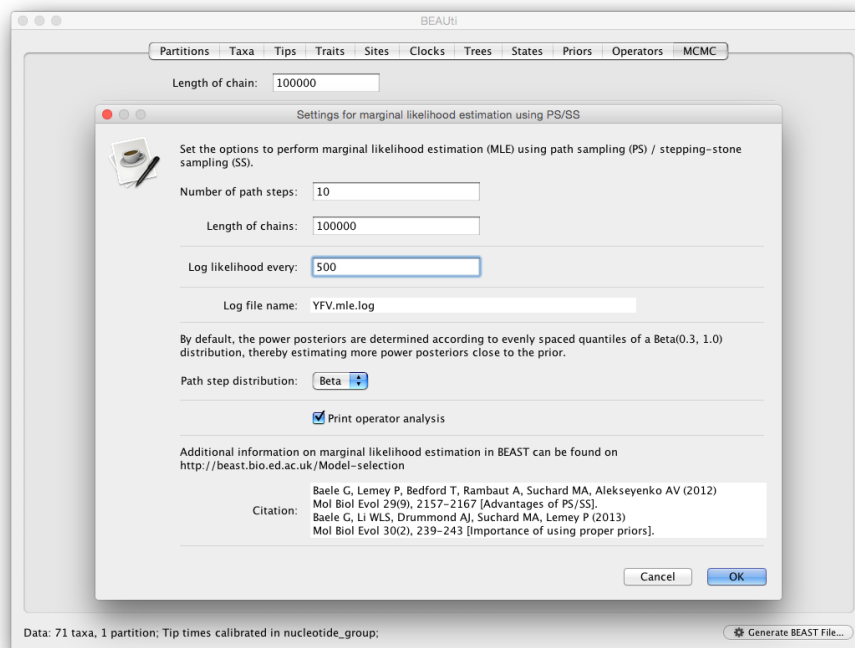
- path sampling (PS): proposed in the statistics literature over 2 decades ago (sometimes also referred to as 'thermodynamic integration'), this approach has been introduced in phylogenetics in 2006 (Lartillot and Philippe 2006). Rather than only using samples from the posterior distribution, samples are required from a series of power posteriors between posterior and prior. As the power posterior gets closer to the prior, there is less and less data available for the posterior, which may lead to improper results when improper priors have been specified. The use of proper priors is hence of primary importance for PS (Baele et al., 2013, MBE, doi:10.1093/molbev/mss243).
- stepping-stone sampling (SS): following essentially the same approach as PS and using the same collection of samples from a series of power posteriors, stepping-stone sampling (SS) yields faster-converging results compared to PS. As was the case for PS, the use of proper priors is critical for SS.
- generalised stepping-stone sampling (GSS): in a way, GSS combines the advantages of PS/SS - in that they yield reliable estimates of the (log) marginal likelihood - with what has been an important reason for the popularity of the HME/AICM, i.e. that these approaches make use of the samples collected during the exploration of the posterior. GSS makes use of working distributions to shorten the path of the PS/SS integration and as such constructs a path between posterior and a product of working distributions, thereby avoiding exploration of the priors and the numerical issues associated with this (Baele et al., 2016, Syst. Biol., doi:10.1093/sysbio/syv083).

Both PS and SS approaches have been implemented in **BEAST** (Baele et al., 2012, MBE, doi:10.1093/molbev/mss084). Typically, PS/SS model selection is performed after doing a standard MCMC analysis. PS and SS sampling can then start where the MCMC analysis has stopped (i.e. you should have run the MCMC analysis long enough so that it converged towards the posterior), thereby eliminating the need for PS and SS to first converge towards the posterior.

As discussed earlier in the tutorial and in the theory on Bayesian model selection, obtaining reliable model comparison results requires the use of proper priors for all parameters being estimated. When performing model comparison, improper priors will lead to improper Bayes Factors. Further, for the PS/SS procedure, we need to sample from the prior at the end of the series of the power posteriors, which will be problematic without proper priors and lead to numerical instabilities (Baele et al., 2013, MBE, doi:10.1093/molbev/mss243).

To set up the PS/SS analyses, we can return to the MCMC panel in BEAUti and select "path sampling / stepping-stone sampling" as the technique we will use to perform "Marginal likelihood estimation (MLE)". Click on "settings" to specify the PS/SS settings. Because of time constraints, we will keep the length of the standard MCMC chain set to 100,000 and we will collect samples from 11 power posteriors (i.e. 10 path steps between 1.0 and 0.0). The length of the chain for the power posteriors can differ from the length of the standard MCMC chain, but we keep it here set to 100,000 as well. The powers for the different power posteriors are defined using evenly spaced quantiles of a Beta( $\alpha$ ,1.0) distribution, with  $\alpha$  here equal to 0.30, as suggested in the stepping-stone sampling paper (Xie et al. 2011) since this approach is shown to outperform a uniform spreading suggested in the path sampling paper (Lartillot and Philippe 2006).

Note that there is an additional option available in the MLE panel: 'Print operator analysis'. When selected, this option will print an operator analysis for each power posterior to the screen, which can then be used to spot potential problems with the operators' performance across the path from posterior to prior. This option is useful when employing highly complex models and when having obtained improbable results.



Having set the PS/SS settings and proper priors, we can write to xml and run the analysis in **BEAST**. Use the same settings for the strict clock and the uncorrelated relaxed clock and run the analyses. What can we conclude from these (much too) short PS/SS analyses?

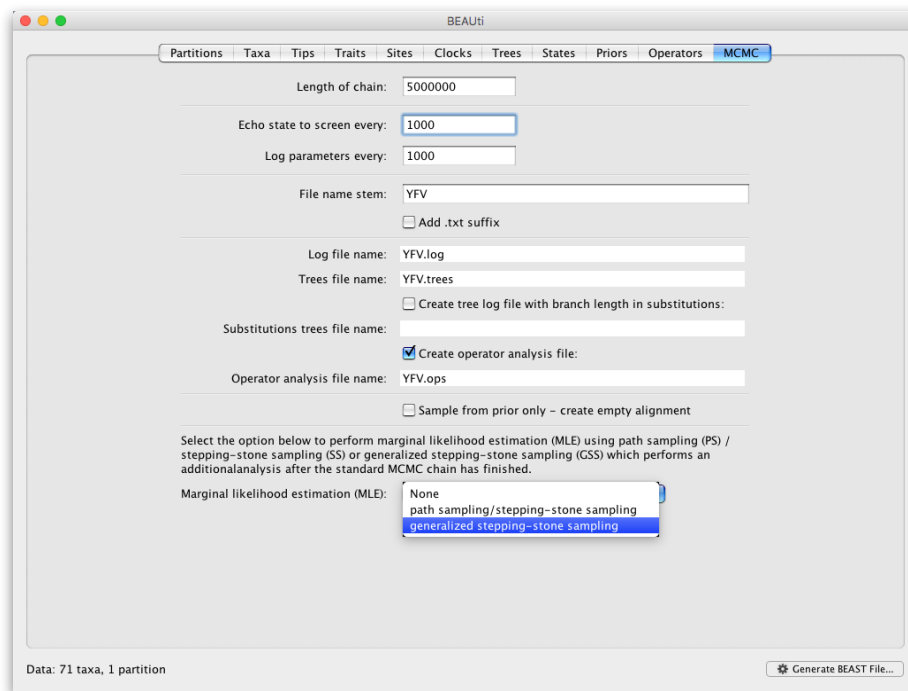
In order to obtain reliable estimates for the marginal likelihoods using PS/SS, we need to rerun these analyses using much more demanding computational settings. For example, by setting the number of path steps to 50 and the length of the MCMC chain for each power posterior to 500,000 (the logging frequency could also be increased). The length of the initial standard MCMC chain should also be increased to ensure convergence towards the posterior before the PS/SS calculations are initiated. An initial chain length of 5,000,000 iterations should be sufficient here. Note that using these settings, the marginal likelihood estimation will take approximately the time it takes to complete a standard MCMC run of 25,000,000 generations for this data (+ 5,000,000 iterations for the initial chain). Due to time constraints, the output files of these analyses have been made available.

**IMPORTANT:** as opposed to when calculating the simple model fit estimators (HME and AICM), it makes no sense to load the output files from a PS/SS/GSS analysis into **Tracer** as it is the output of a series of MCMC analyses!

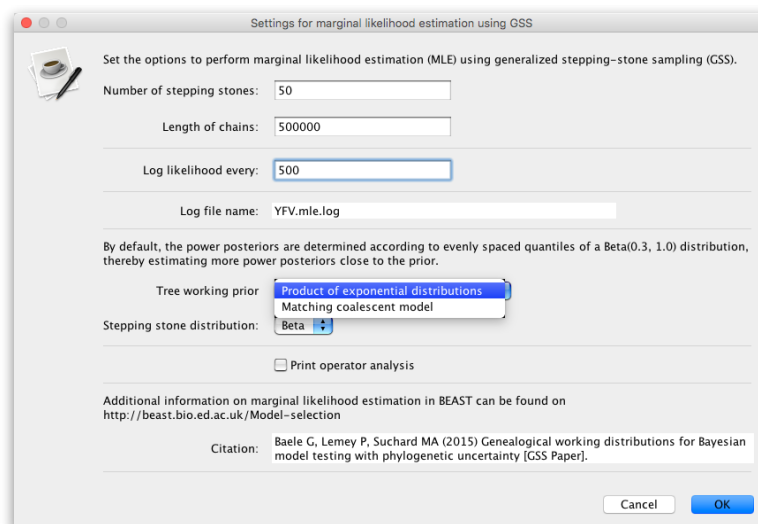
Note that in both folders containing the large output files for PS/SS/GSS provided, there is a **BEAST** XML file available to compute the MLE: calculate-PS-SS.xml or calculate-GSS.xml. As was the case when calculating the HME, sHME and AICM for the longer MCMC runs provided, these XML files will merely read in the samples from the power posteriors collected and will hence take only a short time to compute the actual estimates using these samples. Again, these simple XML files can be used when the actual MLE was lost but the log files are still available (which is a better alternative than rerunning the whole analysis).

For the strict clock analysis, we arrive at -5676.04 and -5676.10 for PS and SS respectively. For the uncorrelated relaxed clock analysis, we get -5656.8 and -5658.14 for the same estimators. How do the PS/SS MLEs compare to those obtained by the HME, and the Bayes factors resulting from these different estimators? What can be observed concerning the difference between the PS estimate and the SS estimate of the log marginal likelihood (also compared to the shorter runs)?

To set up the GSS analyses, we can return to the MCMC panel in BEAUti and select “generalized stepping-stone sampling” as the technique we will use to perform “Marginal likelihood estimation (MLE)”. in the MCMC panel. Click on “settings” to specify the GSS settings. Because of time constraints, we will keep the length of the standard MCMC chain set to 100,000 and we will collect samples from 11 power posteriors (i.e. 10 path steps between 1.0 and 0.0). The length of the chain for the power posteriors can differ from the length of the standard MCMC chain, but we set it here to 100,000 as well.



We again define the powers for the different power posteriors using evenly spaced quantiles of a Beta(0.3,1.0) distribution, since this has been shown to outperform a uniform spreading for generalised stepping-stone sampling (Baele et al., 2016).



The working priors/distribution for the models' parameters are automatically generated depending on their domain, but we need to make a choice when it comes to selecting a working prior for the tree topology. For parametric coalescent model, such as the constant population size model, the fastest approach is to use a “matching coalescent model”; whereas for non-parametric coalescent models, such as the Bayesian Skygrid model, the general-purpose “product of exponential distributions” is the only appropriate option.

For the strict clock analysis, we arrive at -5696.32 using GSS whereas for the uncorrelated relaxed clock analysis, we get -5680.06 using GSS. How do these MLEs compare to those obtained using PS/SS, and the Bayes factors resulting from these different estimators?

The number of power posteriors needed as well as the chain length per power posterior are important settings to achieve a reliable estimate of the (log) marginal likelihood. However, these settings can depend on the data set being analysed and hence different PS/SS/GSS analyses (with differing computational settings) are required when performing model comparison. It is advised to estimate the MLE using PS/SS/GSS with increasing computational settings, until these values no longer change significantly, to ensure convergence to the correct MLE.

As discussed earlier, the model fit estimators used in this tutorial are not limited to specific types of models that can be compared. So apart from comparing different clock models to one another, we can use the same approaches to compare different evolutionary substitution models to one another, or different demographic models, ... For example, try to compare the model fit of the HKY model of nucleotide substitution we've been using so far to a GTR model of nucleotide substitution.

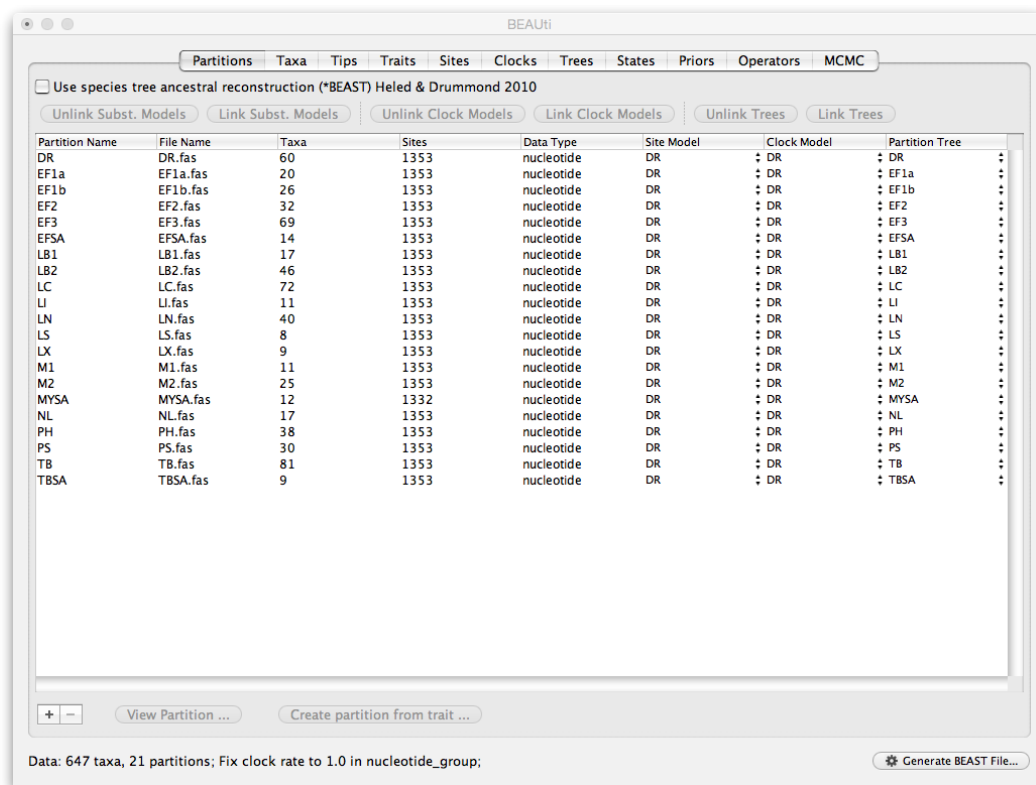


## 2. Testing evolutionary rate hypotheses in bat rabies viruses

This part briefly describes how to set up an BEAST analysis aimed at identifying the factors responsible for evolutionary rate variation in bat rabies viruses (RABV) based on a data set previously analysed by Streicker et al. (2012). Bat rabies viruses in the Americas have established host-associated lineages through sequential host jumping followed by successful transmission in the new bat species (see Streicker et al., 2010). This provides a rare occasion to test the impact of host factors (physiological, environmental or ecological) on virus evolutionary rates. We will test these factors by setting up a mixed effects model for the evolutionary rate, which requires manual xml editing. We will first set up an xml using BEAUti that specifies hierarchical prior distributions for the clock rates and substitution model parameters (which models the random effects) and then manually introduce the fixed effects by editing the xml.

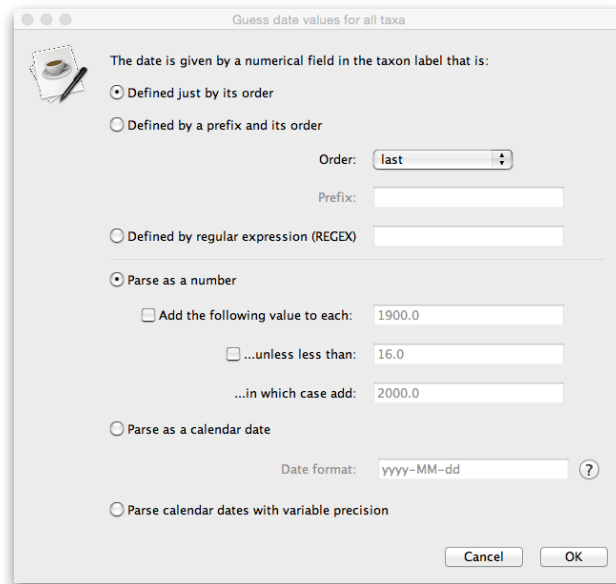
### A hierarchical phylogenetic model

Separate alignments in fasta format are available for each host-associated lineage (DR.fas, EF1a.fas, EF1b.fas, EF2.fas, EF3.fas, EFSA.fas, LB1.fas, LB2.fas, LC.fas, LI.fas, LN.fas, LS.fas, LX.fas, M1.fas, M2.fas, MYSA.fas, NL.fas, PH.fas, PS.fas, TB.fas, TBSA.fas). Start BEAUti, select all the fasta files and drag them into the Partitions panel:

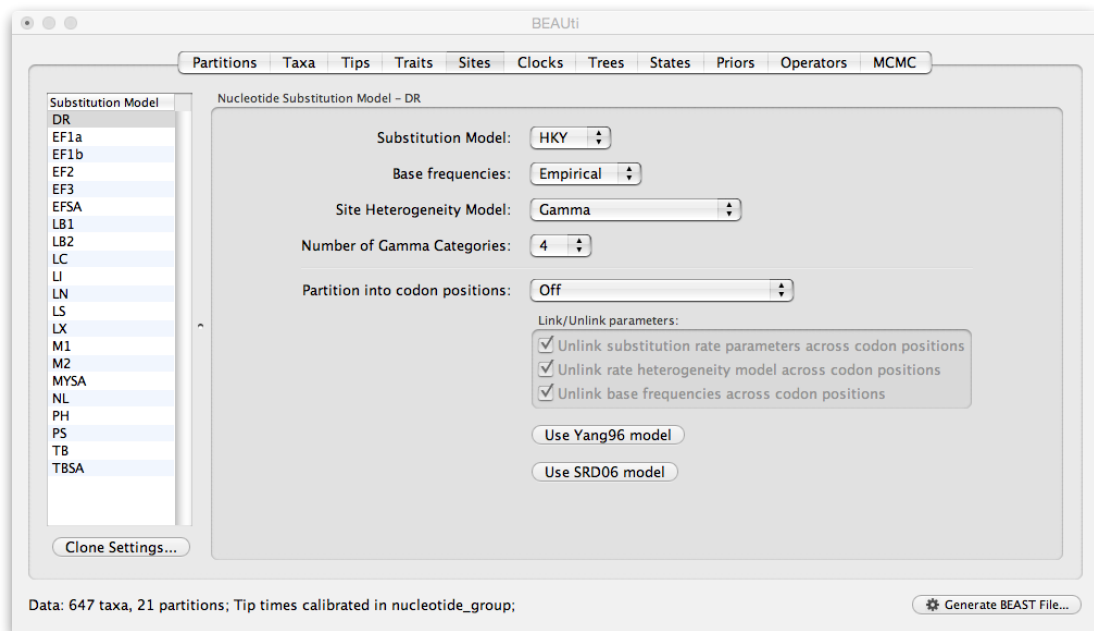


Select all partitions, and click on 'Unlink Subst. Models' to allow each lineage to evolve according to different substitution parameters (check that the Site Model has changed and become specific to each partition). Also select 'Unlink Clock Models' to allow each lineage to evolve under different rates of evolution (check that the Clock Model has become specific to each partition).

In the Tips panel, select 'Use tip dates' and click on 'Guess Dates'. In the new 'Guess date values for all taxa', set the Order to 'last' and click OK.

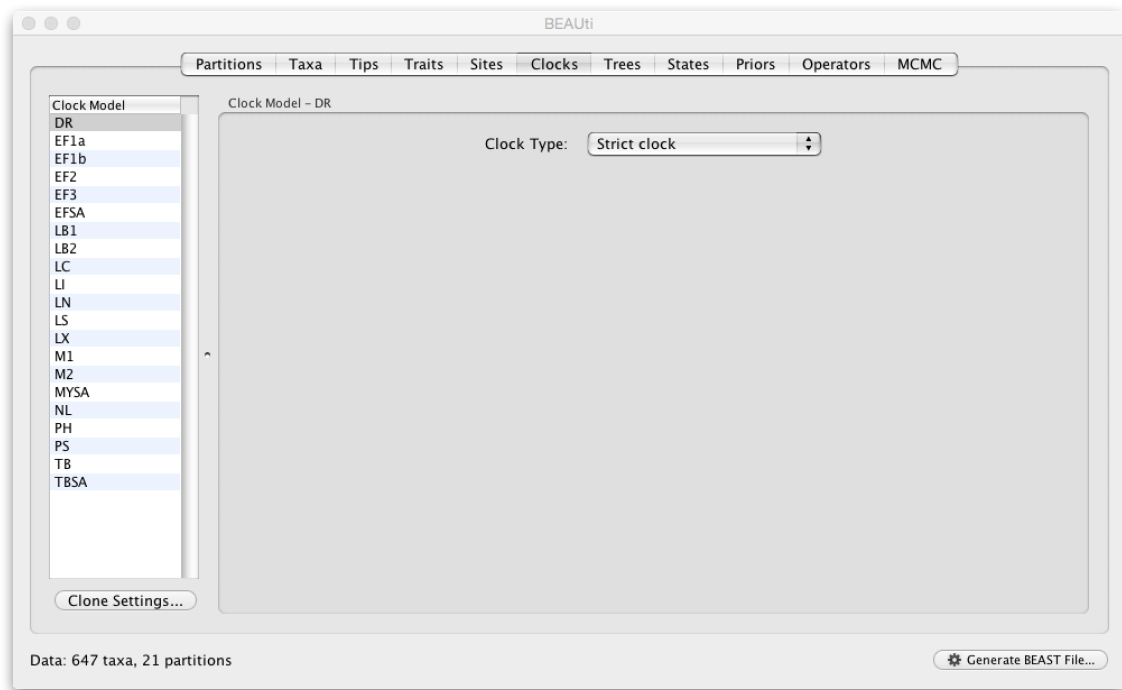


In the Sites panel, keep the default HKY model, but set the 'Base Frequencies' to 'Empirical' and 'Site heterogeneity Model' to 'Gamma' for the DR partition:

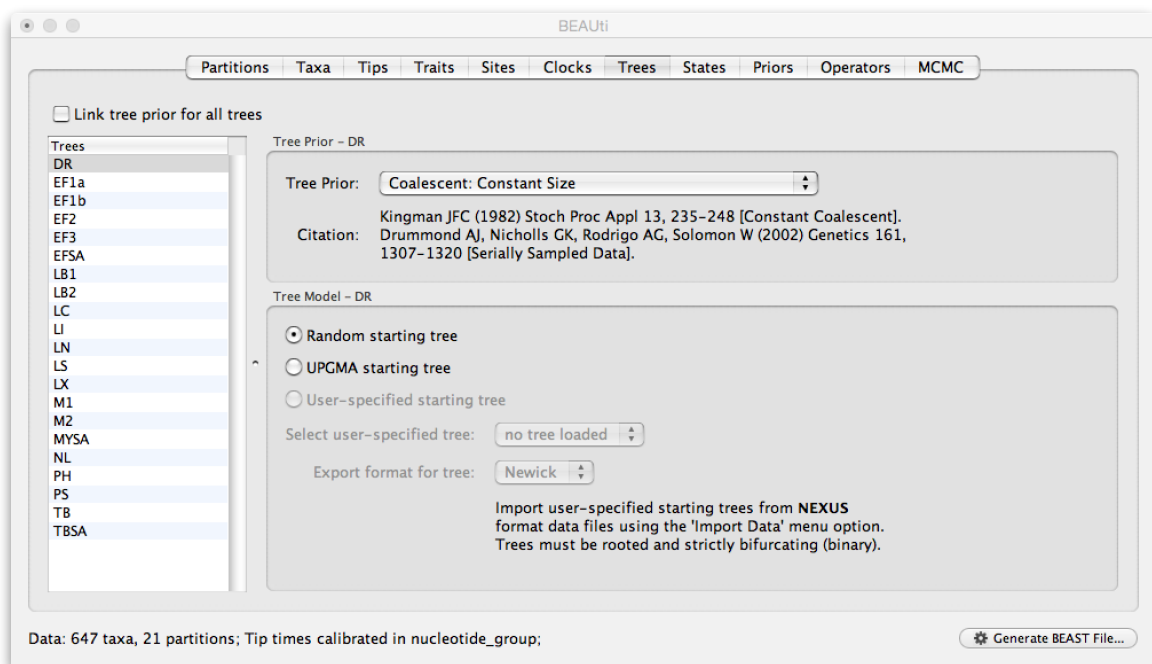


To apply the same settings to all other partitions, select all partitions and click on the 'Clone Settings' button and keep the default 'DR' partition as source model for the cloning.

Keep the default 'Strict Clock' model in the Clocks panel:

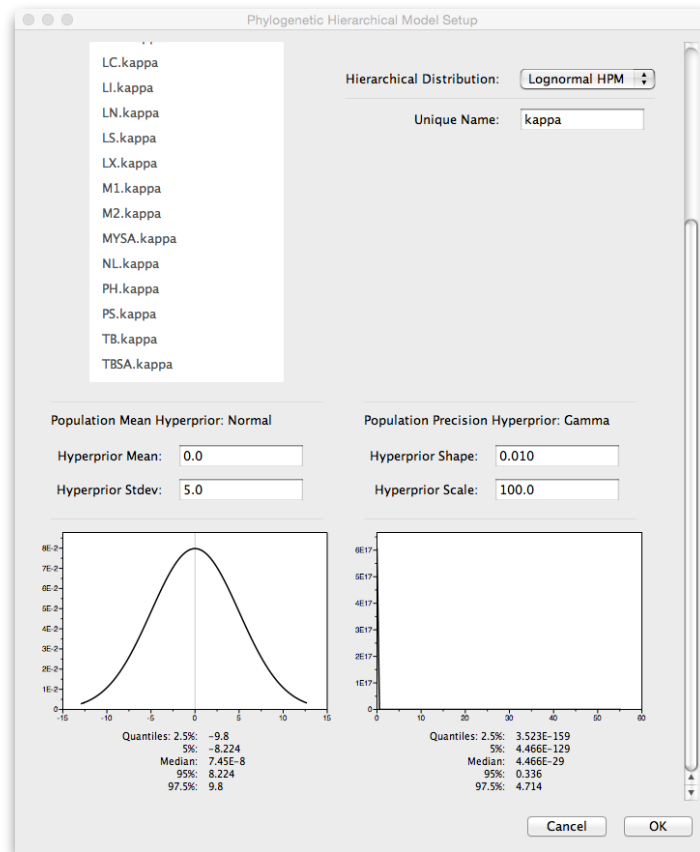


In the 'Trees' panel, make sure to unlink the tree prior (in the top left), and keep the default constant population size model as a simple tree prior:

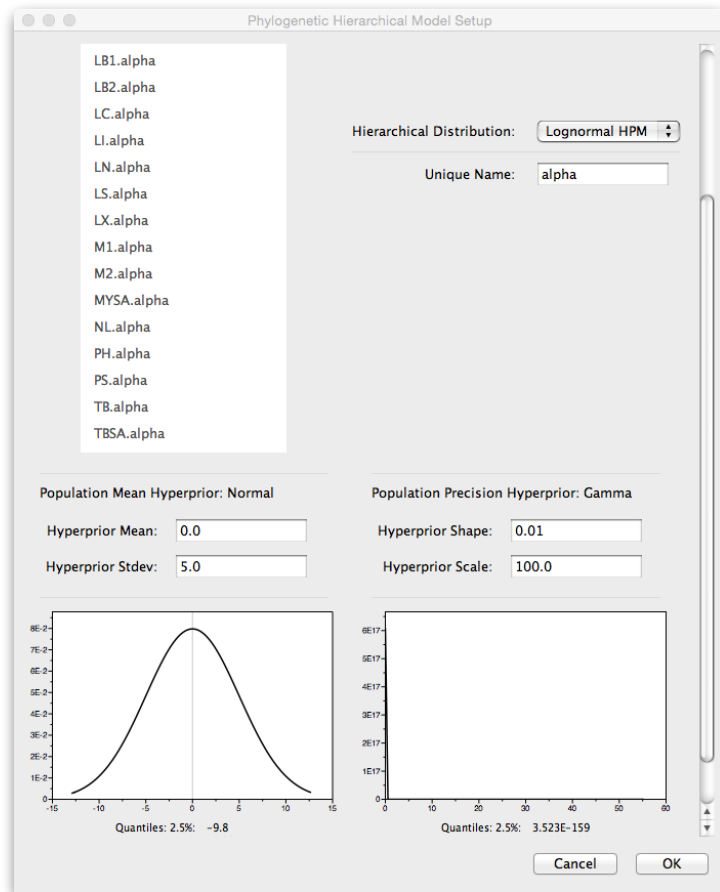


In the 'Priors' panel, we will specify the hierarchical priors for the substitution model parameters and the clock rates. We can achieve this by cmd-selecting all the equivalent parameters across the partitions and using the 'Link parameters into a hierarchical model' button at the bottom of the window. First cmd-select all kappa parameters, and select 'Link parameters into a hierarchical model'. In the new Phylogenetic Hierarchical Model Setup window, enter a Unique Name (e.g. kappa), set

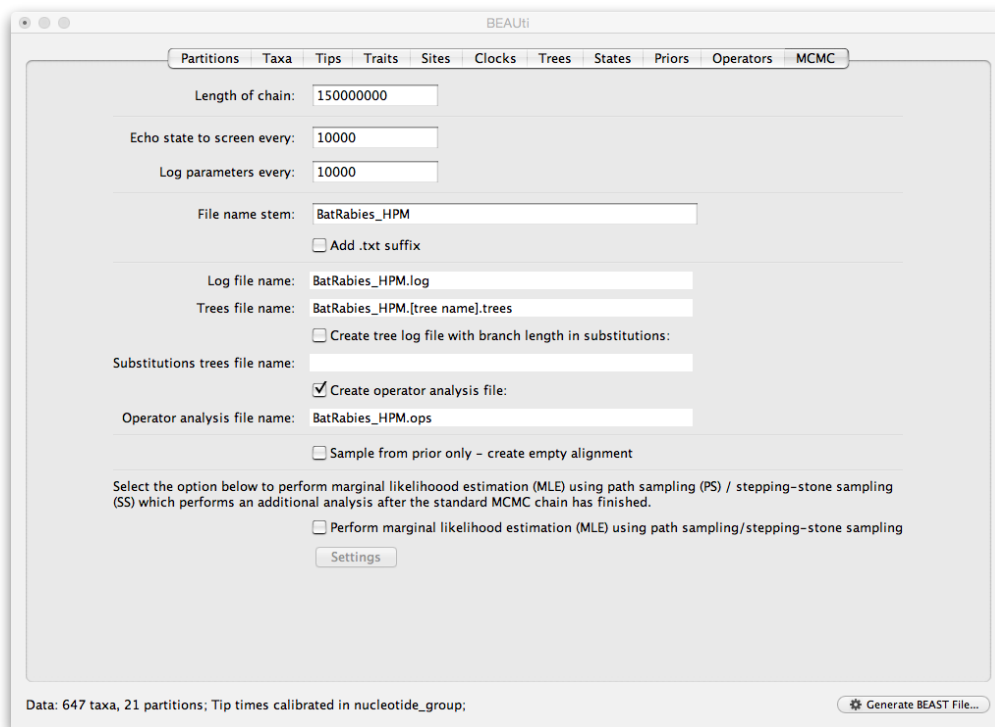
the Normal Hyperprior Stdev to 5.0 and the Gamma Hyperprior Shape and Scale to 0.01 and 100.0 respectively and click OK.



Repeat this operation for the alpha's with the same hyperprior settings and 'alpha' as Unique Name:



Finally, shift select the first and the last clock.rate to select all clock.rate parameters and 'Link parameters into a hierarchical model'. Provide clock.rate as unique name and set the Normal Hyperprior Stdev to 100.0 and the Gamma Hyperprior Shape and Scale to 0.001 and 1000.0 respectively and click OK. Keep the default pop size priors. In the mcmc panel, set the chain frequency to 150,000,000 and the logging frequency to 10,000 and provide 'BatRabies\_HPM' as File stem name:



Check that the xml file runs in BEAST, but there is no real need to run this to completion.

## Introducing fixed effects

In the next section, the xml edits will be highlighted that are required for introducing fixed effects. We will use this to test several potential predictors of evolutionary rate variation among bat RABV lineages. An xml that includes these edits is provided for guidance (look for start and end mixed effect modelling edit comments).

First comment out the clock.rate.hpm distributionLikelihood and clock.rate.prior.mean normalPrior, but keep the clock.rate.prior.precision gammaPrior:

```

<!--
  <distributionLikelihood id="clock.rate.hpm">
    <data>
      <parameter idref="DR.clock.rate"/>
      <parameter idref="EF1a.clock.rate"/>
      <parameter idref="EF1b.clock.rate"/>
      <parameter idref="EF2.clock.rate"/>
      <parameter idref="EF3.clock.rate"/>
      <parameter idref="EFSA.clock.rate"/>
      <parameter idref="LB1.clock.rate"/>
      <parameter idref="LB2.clock.rate"/>
      <parameter idref="LC.clock.rate"/>
      <parameter idref="LI.clock.rate"/>
      <parameter idref="LN.clock.rate"/>
      <parameter idref="LS.clock.rate"/>
      <parameter idref="LX.clock.rate"/>
      <parameter idref="M1.clock.rate"/>
      <parameter idref="M2.clock.rate"/>
      <parameter idref="MYSA.clock.rate"/>
      <parameter idref="NL.clock.rate"/>
      <parameter idref="PH.clock.rate"/>
      <parameter idref="PS.clock.rate"/>
      <parameter idref="TB.clock.rate"/>
      <parameter idref="TBSA.clock.rate"/>
    </data>
  </distributionLikelihood>
  <logNormalDistributionModel id="clock.rate.model" meanInRealSpace="false">
    <mean>
      <parameter id="clock.rate.mean" value="0.0" lower="-Infinity" upper="Infinity"/>
    </mean>
    <precision>
      <parameter id="clock.rate.precision" value="1.0" lower="0.0" upper="Infinity"/>
    </precision>
  </logNormalDistributionModel>
</distribution>
</distributionLikelihood>
<normalPrior id="clock.rate.prior.mean" mean="0.0" stdev="100.0">
  <parameter idref="clock.rate.mean"/>
</normalPrior>
-->
<gammaPrior id="clock.rate.prior.precision" shape="0.0010" scale="1000.0" offset="0.0">
  <parameter idref="clock.rate.precision"/>
</gammaPrior>

```

Before this clock.rate.prior.precision gammaPrior, add a designMatrix, a scaleDesign and a compoundParameter including all the rate parameter. The designMatrix includes the offset (the same grand mean for all lineages) and the different predictors. The latter includes boolean indicators (for climatic regions, colonial aggregation, winter activity, long-distance migration) and log-transformed, standardised continuous predictors (for physiological traits: basal metabolic rate and torpid metabolic rate, and for sampling characteristics: number of samples and sampling time interval for each viral lineage).





```

<glmModel id="clock.rate.hpm" family="logNormal">
  <dependentVariables>
    <parameter idref="clock.rate.all"/>
  </dependentVariables>
  <independentVariables>
    <parameter id="clock.rate.effects" value="1 0 0 0 0 0 0 0 0"/>
    <designMatrix idref="clock.rate.designMatrix"/>
    <indicator>
      <parameter id="clock.rate.effectIndicators" value="1 1 1 1 1 1 1 1 1"/>
    </indicator>
  </independentVariables>
  <scaleVariables>
    <parameter id="clock.rate.precision" value="1"/>
    <indicator>
      <parameter idref="scaleDesign"/>
    </indicator>
  </scaleVariables>
</glmModel>

<productStatistic id="clock.rate.effectsTimesIndicators" elementWise="false">
  <parameter idref="clock.rate.effects"/>
  <parameter idref="clock.rate.effectIndicators"/>
</productStatistic>

<multivariateNormalPrior id="clock.rate.prior.effects">
  <data>
    <parameter idref="clock.rate.effects"/>
  </data>
  <meanParameter>
    <parameter value="0 0 0 0 0 0 0 0 0"/>
  </meanParameter>
  <precisionParameter>
    <matrixParameter>
      <parameter value="0.001 0 0 0 0 0 0 0 0"/>
      <parameter value="0 2.0 0 0 0 0 0 0 0"/>
      <parameter value="0 0 2.0 0 0 0 0 0 0"/>
      <parameter value="0 0 0 2.0 0 0 0 0 0"/>
      <parameter value="0 0 0 0 2.0 0 0 0 0"/>
      <parameter value="0 0 0 0 0 2.0 0 0 0"/>
      <parameter value="0 0 0 0 0 0 2.0 0 0"/>
      <parameter value="0 0 0 0 0 0 0 2.0 0"/>
      <parameter value="0 0 0 0 0 0 0 0 2.0"/>
    </matrixParameter>
  </precisionParameter>
</multivariateNormalPrior>

```

In the operator block, comment out the normalNormalMeanGibbsOperator on the clock.rate.prior.mean and the normalGammaPrecisionGibbsOperator on the normalGammaPrecisionGibbsOperator:

```

<!-- start mixed effect modelling edits-->
<!--
  <normalNormalMeanGibbsOperator weight="1.0">
    <likelihood>
      <distribution idref="clock.rate.hpm"/>
    </likelihood>
    <prior>
      <normalPrior idref="clock.rate.prior.mean"/>
    </prior>
  </normalNormalMeanGibbsOperator>
  <normalGammaPrecisionGibbsOperator weight="1.0">
    <likelihood>
      <distribution idref="clock.rate.hpm"/>
    </likelihood>
    <prior>
      <gammaPrior idref="clock.rate.prior.precision"/>
    </prior>
  </normalGammaPrecisionGibbsOperator>
-->

```

And add in a `regressionGibbsEffectOperator`, a `regressionGibbsPrecisionOperator`, a `bitFlipOperator` and a `regressionMetropolizedIndicatorOperator`:

```
<regressionGibbsEffectOperator weight="2">
  <glmModel idref="clock.rate.hpm"/>
  <parameter idref="clock.rate.effects"/>
  <indicator>
    <parameter idref="clock.rate.effectIndicators"/>
  </indicator>
  <multivariateNormalPrior idref="clock.rate.prior.effects"/>
</regressionGibbsEffectOperator>

<regressionGibbsPrecisionOperator weight="2">
  <glmModel idref="clock.rate.hpm"/>
  <parameter idref="clock.rate.precision"/>
  <gammaPrior idref="clock.rate.prior.precision"/>
</regressionGibbsPrecisionOperator>

<!-- Model selection Operator -->

<bitFlipOperator weight="2" usesPriorOnSum="false">
  <maskedParameter>
    <parameter idref="clock.rate.effectIndicators"/>
    <mask>
      <parameter value="0 1 1 1 1 1 1 1"/>
    </mask>
  </maskedParameter>
</bitFlipOperator>

<regressionMetropolizedIndicatorOperator weight="5">
  <glmModel idref="clock.rate.hpm"/>
  <parameter idref="clock.rate.effects"/>
  <indicator>
    <parameter idref="clock.rate.effectIndicators"/>
  </indicator>
  <mask>
    <parameter value="0 1 1 1 1 1 1 1"/>
  </mask>
  <multivariateNormalPrior idref="clock.rate.prior.effects"/>
</regressionMetropolizedIndicatorOperator>

<!-- end mixed effect modelling edits-->
```

Note that we use a mask on the indicator vector when operating on the indicators to avoid operating on the grand mean (which should always be included).

In the prior block, comment out the `distributionLikelihood` on the `clock.rate.hpm` and the `normalPrior` on the `clock.rate.prior.mean` and refer to the `glmModel` and `multivariateNormalPrior` on the `clock.rate.prior.effects` instead. Keep the `gammaPrior` on the `clock.rate.prior.precision`:

```

        <!-- START Hierarchical phylogenetic models
        <distributionLikelihood idref="kappa.hpm"/>
        <normalPrior idref="kappa.prior.mean"/>
        <gammaPrior idref="kappa.prior.precision"/>
        <distributionLikelihood idref="alpha.hpm"/>
        <normalPrior idref="alpha.prior.mean"/>
        <gammaPrior idref="alpha.prior.precision"/>
    <!-- start mixed effect modelling edits-->
    <!--
        <distributionLikelihood idref="clock.rate.hpm"/>
        <normalPrior idref="clock.rate.prior.mean"/>
    -->
        <glmModel idref="clock.rate.hpm"/>
        <multivariateNormalPrior idref="clock.rate.prior.effects"/>
        <gammaPrior idref="clock.rate.prior.precision"/>
    <!-- end mixed effect modelling edits-->
    <!-- END Hierarchical phylogenetic models

```

Finally, in the fileLog, comment out the clock.rate.mean parameter, keep the clock.rate.precision and add in the clock.rate.effects and clock.rate.effectIndicators:

```

        <!-- START Hierarchical phylogenetic models
        <parameter idref="kappa.mean"/>
        <parameter idref="kappa.precision"/>
        <parameter idref="alpha.mean"/>
        <parameter idref="alpha.precision"/>
    <!-- start mixed effect modelling edits-->
    <!--
        <parameter idref="clock.rate.mean"/>
    -->
        <parameter idref="clock.rate.effects"/>
        <parameter idref="clock.rate.effectIndicators"/>
        <parameter idref="clock.rate.precision"/>
    <!-- end mixed effect modelling edits-->

```

Check that the xml file runs in BEAST. Output files are provided for the complete analysis. Are there any predictors that help to explain the rate variation among the host-associated bat RABV lineages. What support do they get? Note that we assumed implicitly a 0.5 prior success probability for each predictor. It may be useful to explore a different prior that prefers a smaller inclusion probability a priori. For example, we could assign independent Bernoulli prior probability distributions on the indicators and use a small prior probability on each predictor's inclusion that reflects a 50% prior probability on no predictors being included, e.g.:

```

<!-- start mixed effect modelling edits-->
<!--
      <distributionLikelihood idref="clock.rate.hpm"/>
      <normalPrior idref="clock.rate.prior.mean"/>
-->
      <glmModel idref="clock.rate.hpm"/>
      <multivariateNormalPrior idref="clock.rate.prior.effects"/>
      <gammaPrior idref="clock.rate.prior.precision"/>
      <binomialLikelihood>
        <proportion>
          <parameter value="0.083"/>
        </proportion>
        <trials>
          <parameter value="1 1 1 1 1 1 1 1"/>
        </trials>
        <counts>
          <maskedParameter>
            <mask>
              <parameter value="0 1 1 1 1 1 1 1"/>
            </mask>
            <parameter idref="clock.rate.effectIndicators"/>
          </maskedParameter>
        </counts>
      </binomialLikelihood>
<!-- end mixed effect modelling edits-->

```

What effect will this have on the support for the predictors?

## Conclusion and Resources

This chapter only scratches the surface of the analyses that are possible to undertake using BEAST. It has hopefully provided a relatively gentle introduction to the fundamental steps that will be common to all BEAST analyses and provide a basis for more challenging investigations. BEAST is an ongoing development project with new models and techniques being added on a regular basis. The BEAST website provides details of the mailing list that is used to announce new features and to discuss the use of the package. The website also contains a list of tutorials and recipes to answer particular evolutionary questions using BEAST as well as a description of the XML input format, common questions and error messages.

- The BEAST website: <http://beast.bio.ed.ac.uk/>
- Tutorials: <http://beast.bio.ed.ac.uk/Tutorials/>
- Model selection tutorial: <http://beast.bio.ed.ac.uk/Model-selection>
- More model selection tutorials: <https://rega.kuleuven.be/cev/ecv/tutorials>
- Frequently asked questions: <http://beast.bio.ed.ac.uk/FAQ/>

## References

- Drummond, A. J., S. Y. W. Ho, M. J. Phillips, and A. Rambaut. 2006. Relaxed phylogenetics and dating with confidence. *PLoS Biol* 4.
- Bryant, J. E., E. C. Holmes, and A. D. Barrett. 2007. Out of Africa: a molecular perspective on the introduction of yellow fever virus into the Americas. *PLoS pathogens* 3:e75.
- Drummond, A. J., M. A. Suchard, D. Xie, and A. Rambaut. 2012. Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Molecular biology and evolution* 29:1969-1973.
- Rambaut, A., T.T. Lam, L. de Carvalho, and O.G. Pybus. 2016. Exploring the temporal structure of heterochronous sequences using TempEst. *Virus Evolution* 2: vew007 DOI: <http://dx.doi.org/10.1093/ve/vew007>.

Ferreira, M. A. R. and M. A. Suchard. 2008. Bayesian analysis of elapsed times in continuous-time Markov chains. *Can J Statistics*, 36: 355–368. doi: 10.1002/cjs.5550360302

Baele, G., P. Lemey, T. Bedford, A. Rambaut, M. A. Suchard, and A. V. Alekseyenko. 2012. Improving the accuracy of demographic and molecular clock model comparison while accommodating phylogenetic uncertainty. *Molecular biology and evolution* 29:2157-2167.

Baele, G., W. L. Li, A. J. Drummond, M. A. Suchard, and P. Lemey. 2013. Accurate model selection of relaxed molecular clocks in bayesian phylogenetics. *Molecular biology and evolution* 30:239-243.

Baele, G., P. Lemey, M. A. Suchard. 2016. Genealogical working distributions for Bayesian model testing with phylogenetic uncertainty. *Systematic biology* 65:250-264.