Estimating rates and dates from time-stamped sequences

Summary: This tutorial provides a step-by-step tutorial for analyzing a set of virus sequences which have been isolated at different points in time (heterochronc data). 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 ***Aede aegypti*** mosquitoes, in the bilges of sailing vessels during the slave trade. Although the hypothesis of a slave trade introduction had often been suggested, pric 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 ances and to infer the phylogenetic relationships with appropriate measures of statistical support.

Table of Contents

Introduction

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 us 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 exp 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 syste


(beast) BEAST - this package contains the BEAST (beast) program, BEAUti (beauti) and a couple of utility programs. At the time of writing, the current version is v1.10.4. BEAST releases are available for downlo from https://github.com/beast-dev/beast-mcmc/releases.


(tracer) Tracer - this program is used to explore the output of BEAST (and other Bayesian MCMC programs). It graphically and quantitively summarizes the empirical distributions of continuous parameters and provides diagnostic information. At the time of writing, the current version is v1.7.1. It is available for download from https://github.com/beast-dev/tracer/.


(figtree) 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 v1.4.3. It is available for download from http://tree.bio.ed.ac.uk/software/figtree/.

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 E 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).

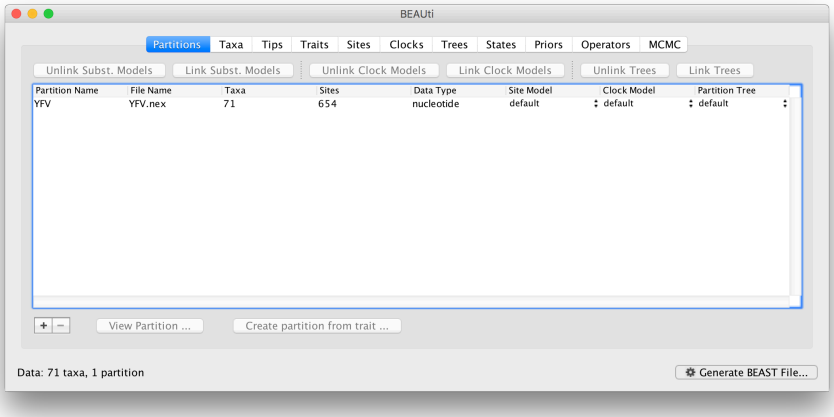
All the files needed for this tutorial can be downloaded from here (/tutorials/workshop_rates_and_dates/files/YFVtutorialFiles.zip). If you download this zipped folder, there is no need to download other files/fold linked further in the tutorial.

Loading the data into BEAUti

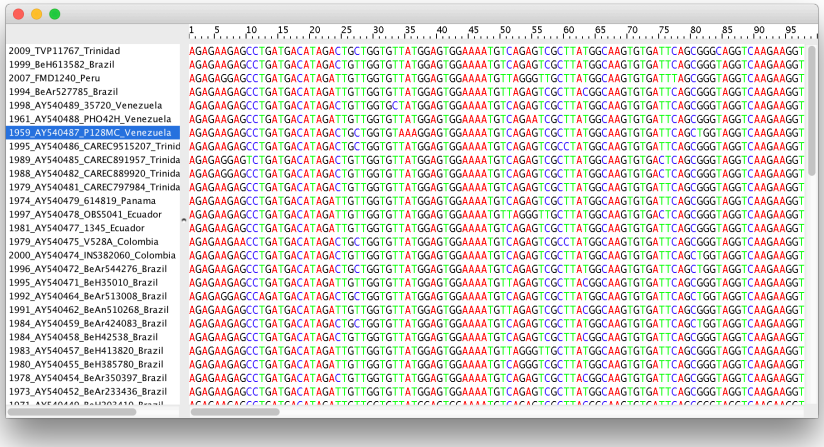
The data file is called '`YFV.nex`' and can be downloaded from here (/tutorials/workshop_rates_and_dates/files/YFV.nex).


Run BEAUti (beauti) by double clicking on its icon. BEAUti is an interactive graphical application for designing your analysis and generating the control file (a BEAST XML file) which BEAST will use to run the analysis.

To load a NEXUS format alignment, simply select the `Import Data...` option from the `File` menu and select the file called `YFV.nex` . This file contains an alignment of 71 sequences from the prM/E gene of YF nucleotides in length. Once loaded, the sequence data will be listed under Data Partitions:

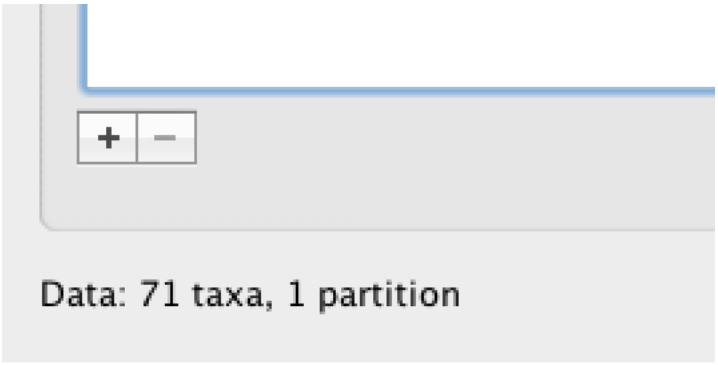


Double-click on the File Name in the table (but not on Partition Name) to display the actual sequence alignment:



Specifying a taxon set

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 p the small "plus" button at the bottom left of the panel:

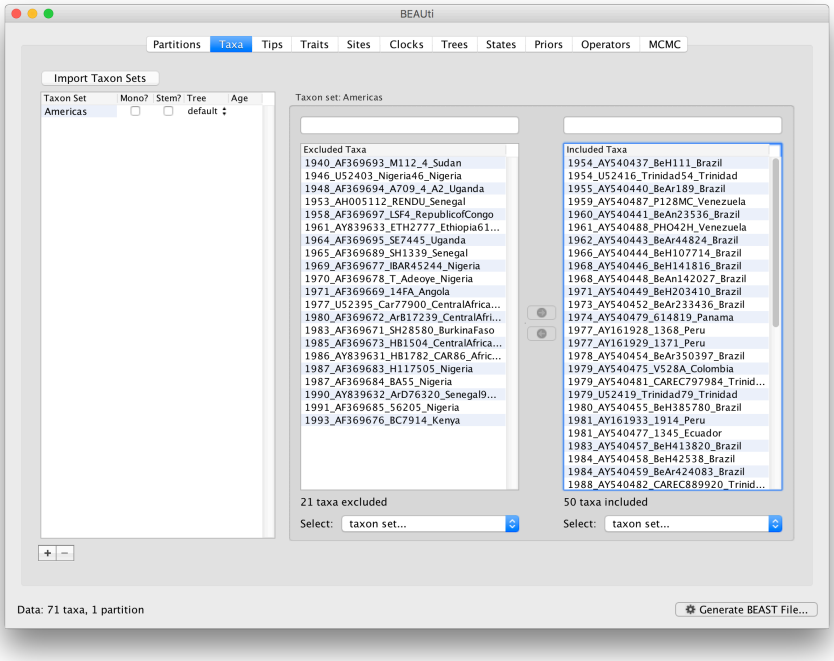


This will create a new taxon set. Rename it by double-clicking on the entry that appears (it will initially be called `untitled1`). Call it `Americas` . Do not enforce monophyly using the `monophyletic?` option becau will evaluate the support for this cluster. We do not opt for the `includeStem?` option either because we would like to estimate the TRMCA for the viruses from the Americas and not for the parent node leading to th 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 holdin the Command or Control button on a Mac or PC, respectively. 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 t African taxa (the country of sampling is included at the end of the taxa names). Check there are only African countries on the left (there should be 21) and only American countries on the right (there should be 50).

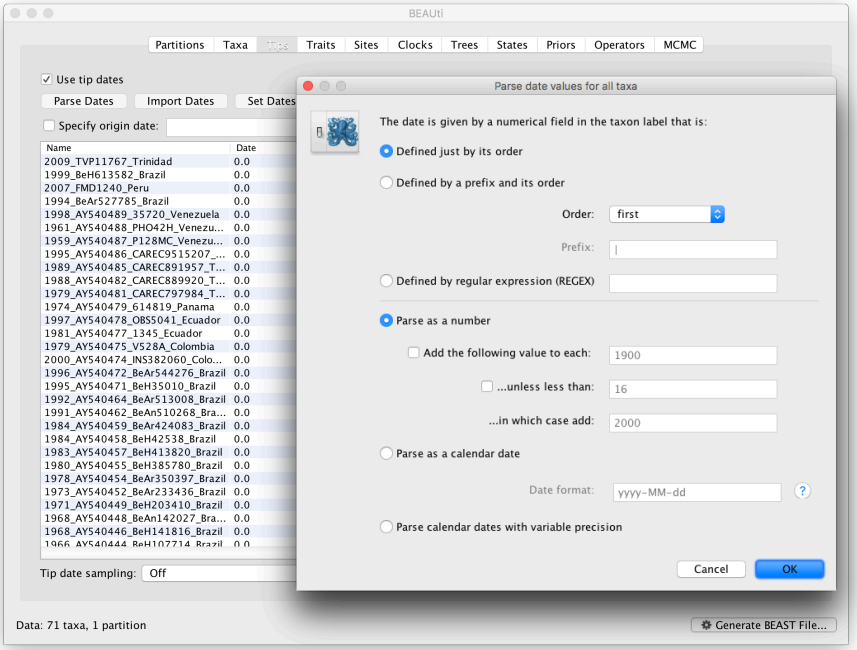
For more information on creating taxon sets, see this page (taxon_sets).

After these operations, the screen should look like this:



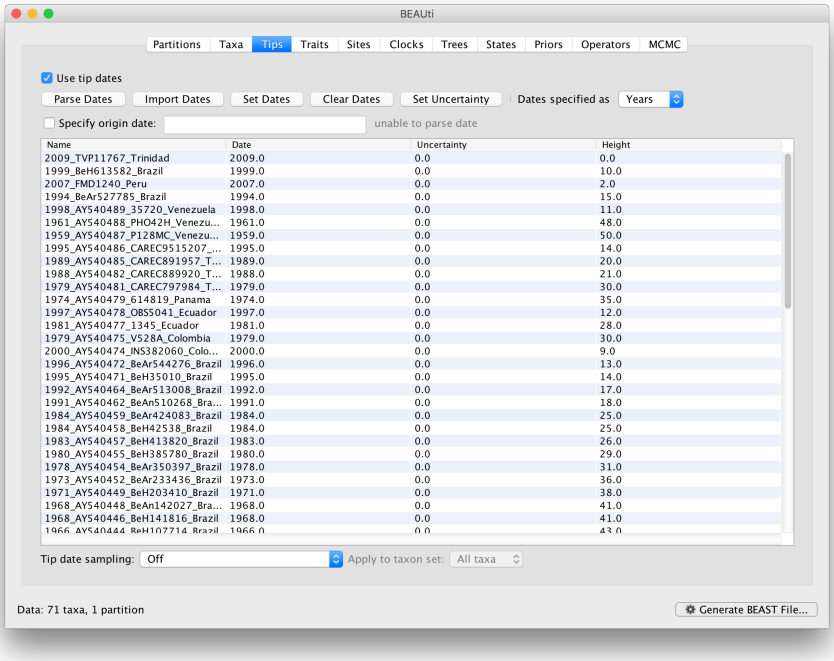
Setting the tip dates

To inform BEAUti/BEAST about the sampling dates of the sequences, go to the `Tips` panel and select the `Use tip dates` option. By default all the taxa are assumed to have a date of zero (i.e. the sequences ar 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 a 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 convenien to specify the dates of the sequences in BEAUti is to use the `Parse Dates` button at the top of the `Tips` 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 numeri field (such as the some YFV sequences, above) then you can specify how to find the one that corresponds to the date of sampling. See this page for details about the various options for setting dates in this panel (tip_dates). 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 date which can be useful for transforming a 2 digit year into a 4 digit year (tip_dates). Because all dates are specified in a four digit format in t case, no additional settings are needed. So, we can press `OK` .



The `Height` column lists the ages of the tips relative to time 0 (in our case 2009).

> ☑ **Tip:** There are many other options for reading and parsing tip dates in different formats. See this page for a more detailed description of these options. (tip_dates.html)

For these sequences, only the sampling year is provided and not the exact sampling dates. This uncertainty will negligible with respect to the relatively large evolutionary time scale of this example, however, it is possi accommodate the sampling time uncertainty — see here (sampling_tipdates).

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 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:

Substitution Model:

For nucleotide data, this is a choice of `JC`, `HKY`, `GTR` or `TN93`. Other substitution models are possible by constraining one of these model. See this page for more details (custom_substitution_models).

Base frequencies:

The nucleotide base frequencies can be `Estimated` (estimated as a parameter in the model), `Empirical` (estimated emprically from the data and then fixed) or `All equal` (fixed to be 0.25 each).

Site Heterogeneity Model:

A choice of the discrete gamma distribution model, the invariant site model or both.

Partition into codon positions:

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 depending on the setting.

Unlink substitution model across codon positions:

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.

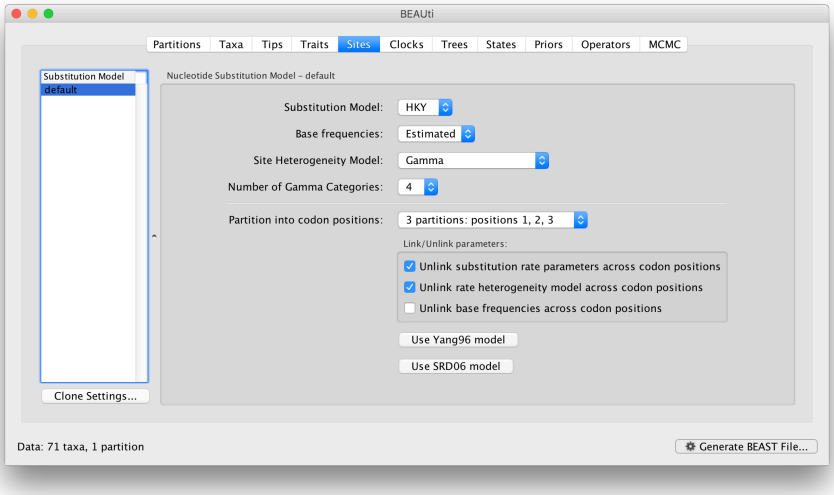
Unlink rate heterogeneity model across codon positions:

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.

Unlink base frequencies across codon positions:

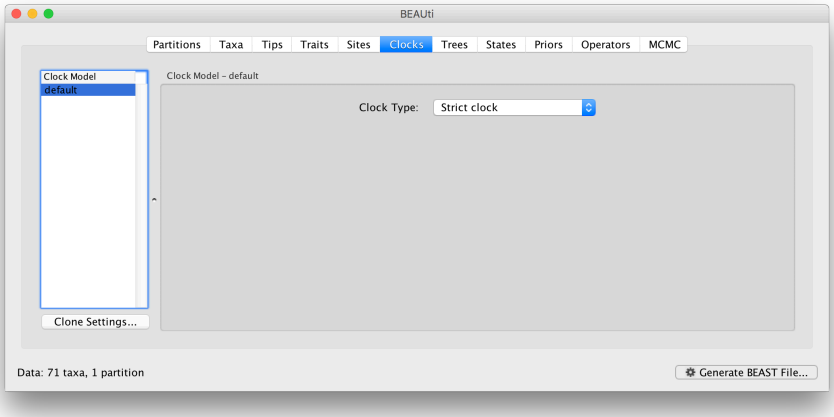
Selecting the `Unlink base frequencies across codon positions` will specify that BEAST should estimate a separate set of base frequencies for each codon position.

For this tutorial, select the `3 partitions: 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 rat variation among sites:



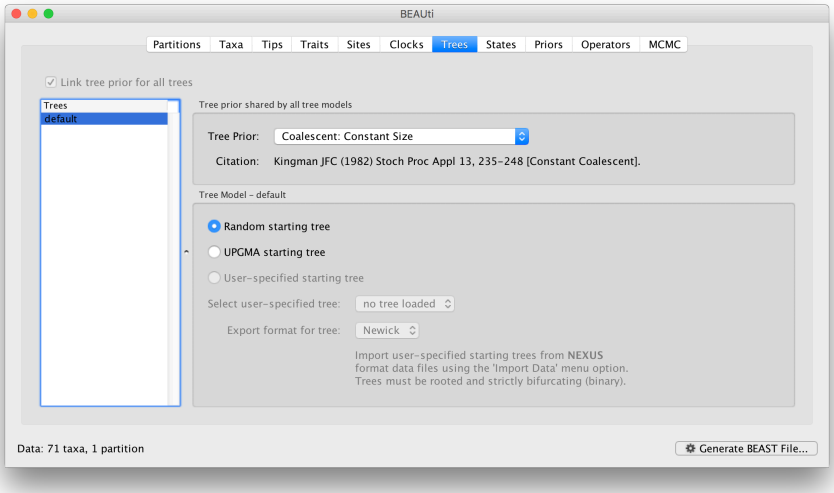
Setting the clock model

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:



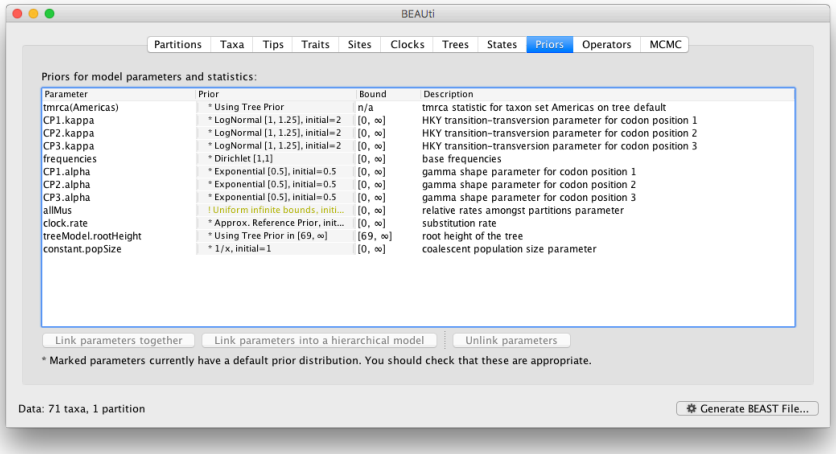
Setting the starting tree and tree prior

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) are described on this page (tree_priors).



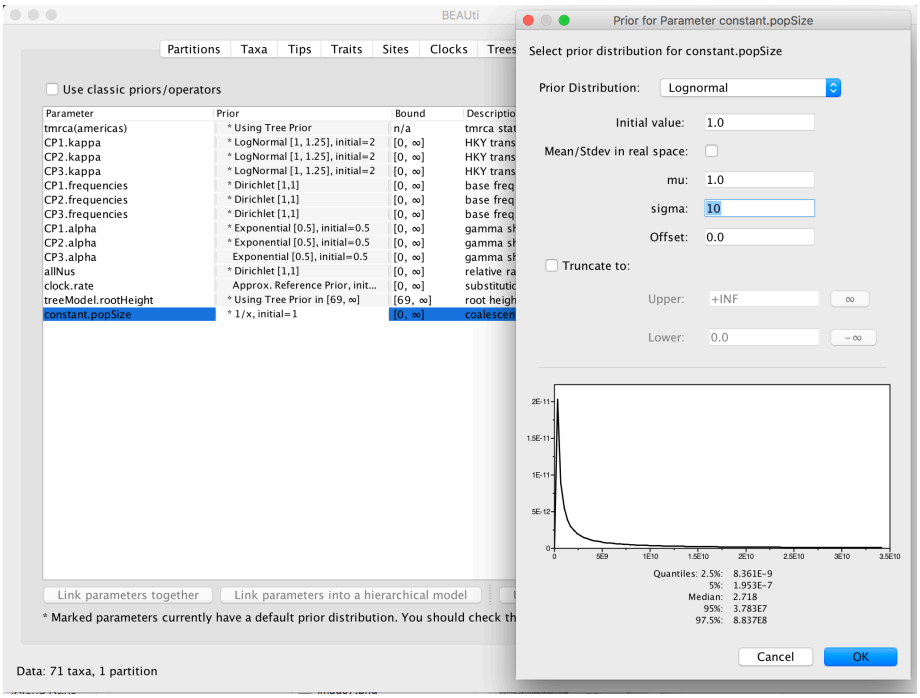
Setting up the priors

Review the prior settings under the `Priors` panel:



Some of the default marginal priors may be improper — this means that the probability distribution does not integrate to a finite value. In our current default settings, the `1/x` prior on the `constant.popSize` is an example of an improper prior.

It is 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, cfr a differe workshop tutorial). To change the prior on the `constant.popSize` for example, click on the corresponding prior and a prior selection window will appear. Set the prior to a lognormal distribution with `mu = 1` and `sigma = 10` . The graphical representation of this prior distribution indicates that most prior mass is put on relatively small values, but the density remains sufficiently diffuse over larger values. <Notice that the prior turns black after confirming this setting by clicking "OK".>

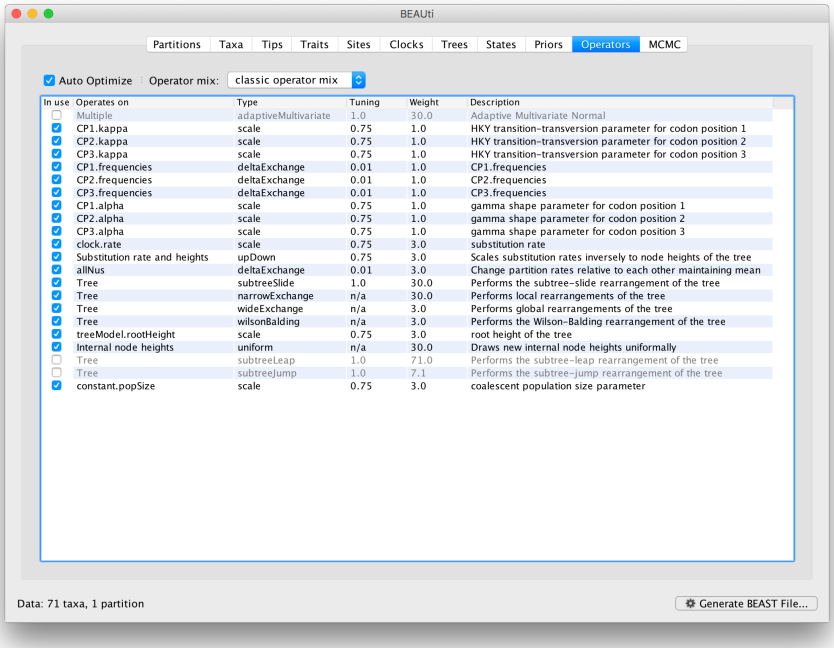


Note that the default prior on the rate of evolution (clock.rate) is an approximation of a conditional reference prior (Approx. Reference Prior) (Ferreira and Suchard, 2008). If the sequences are not associated with differ sampling dates (they are contemporaneous), or when the sampling time range is 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 norma 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 unsele

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 specif the parameters change as the MCMC runs. As of BEAST v1.8.4, different options are available with respect to exploring tree space. In this tutorial, we will use the 'classic operator mix', which consists of of set of tre 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` panel 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 colum the type of operators that are acting on each parameter. For example, the scale operator scales the parameter up or down by a proportion, the random walk operator adds or subtracts an amount to the parameter a uniform operator simply picks a new value uniformly within a range. Some parameters relate to the tree or to the divergence times of the nodes of the tree and these have special operators.

Each operator also has a check box (the `In use` column) which can be used to turn individual operators on and off. For example, deselecting the operators on the rate of evolution (`clock.rate` and `Substitut rates and heights`) will fix the rate to the initial value. The initial value for a parameter is set in the `Priors` table.

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 in turn affect the efficiency of the analysis. For most operators (like random walk and subtree slide operators) a larger tu 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 will be writ standard output. In general, the auto-optimization of the operators works well end and nothing needs to be changed.

The next column, labelled `Weight` , specifies how often each operator is applied relative to the others. Some parameters tend to be sampled very efficiently - an example is the kappa parameter - these parameters h their operators down-weighted so that they are not changed as often.

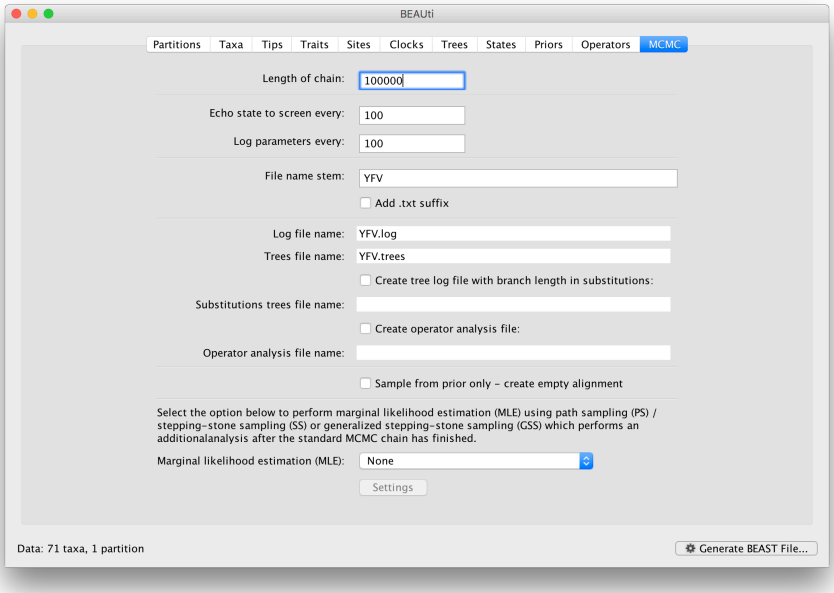
As of BEAST v1.8.4, different options are available with respect to exploring parameter space using the `Operator mix` option. The default is the `classic operator mix` , which is a mix of operators and weights correspond to previous versions of BEAST. There is also an option to fix the tree topology as well as a `new experimental mix` , which is currently under development with the aim to improve mixing for large phylo trees. These options turn operators on and off so can be overridden using the `In use` switches.

In most cases, no changes are required to this table but operators can be 'turned-off' which has the effect of fixing the parameter to its initial value.

For this analysis, no changes are required to this table.

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 depen 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 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 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. Samplin 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. Yo probably want to aim to store no more than 10,000 samples so this should be set to something >= 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 t the sampling frequencies to 100.

> ☑ **Tip:** You can set the screen sampling frequency something different from the main log files. Here, the analysis is going to run very fast so printing to the screen every 100 steps will cause a large amount of information to scroll up the screen. Try setting the `Echo state to screen` option to **`10000`** resulting in only 100 updates to the screen as the analysis runs.

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 (or add more detail about the analysis). 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.

> ℹ **Note:** On Windows machines the operating system patronisingly hides the extensions of files from you. It is sometimes easier to add an additional extension **`.txt`** to the log and the trees file — Windows will hi the **`.txt`** but still show you the **`.log`** and **`.trees`** extensions so you can distinguish the files.

The remaining options can be left unselected this time. An option is available to sample from the prior only, which can be useful to evaluate how divergent our posterior estimates are when information is drawn from t data. Also, one can chose to perform marginal likelihood estimation to assess model fit; we will return to this in a later tutorial.

Saving and Loading BEAUti files

If you select the `Save` option from the `File` menu this will save a document in BEAUti's own format. Note that is not in the format that BEAST understands — it can only be reopened by BEAUti. The idea is that th settings and data in BEAUti can be saved and loaded at a later time. We suggest you save BEAUti files with the extension '.beauti'.

> ℹ **Note:** Just as BEAUti files cannot be read and understood by BEAST, BEAST XML files cannot be reloaded back into BEAUti. They can however be 'Imported' just like NEXUS or FASTA files. The sequence dat contained within will be imported as will all the tipdates and certain other information.

Generating the BEAST XML file

We are now ready to create the BEAST XML file. Select `Generate XML...` from the `File` menu (or the button at the bottom of the window). BEAUti will ask you to review the prior settings one more time before s the file (and will indicate if any are improper). Continue and choose a name for the file — it will offer the name you gave it in the MCMC panel and we usually end the filename with '.xml' (although see the note, above, extensions on Windows machines – you may want to give the file the extension '.xml.txt').

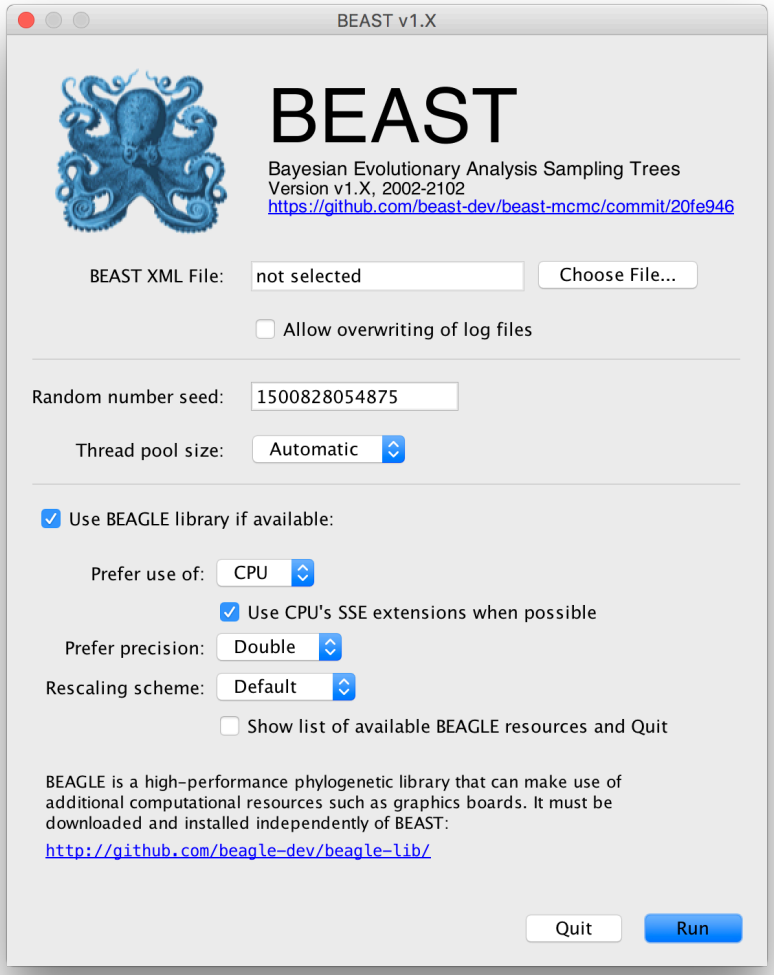
> ☑ **Tip:** For convenience, leave the BEAUti window open so that you can change the values and re-generate the BEAST file as required later in this tutorial.

Running BEAST

We are now ready to run the file through BEAST.

> Run BEAST by double clicking on the BEAST icon.

The following dialog box will appear:



All you need to do is to click the `Choose File...` button, select the XML file you created in BEAUti, above, and press `Run` . For information about the other options see the page on the BEAST program (beast).

When you press `Run` BEAST will load the XML file, setup the analysis and then run it with no further interaction. In the output window you will see lots of information appearing. It starts by printing the title and credits

```
      BEAST v1.10.0 Prerelease #VEME2017, 2002-2017
      Bayesian Evolutionary Analysis Sampling Trees
                Designed and developed by
  Alexei J. Drummond, Andrew Rambaut and Marc A. Suchard

             Department of Computer Science
                 University of Auckland
                alexei@cs.auckland.ac.nz

           Institute of Evolutionary Biology
                University of Edinburgh
                   a.rambaut@ed.ac.uk

             David Geffen School of Medicine
         University of California, Los Angeles
                   msuchard@ucla.edu

              Downloads, Help & Resources:
                     http://beast.community

Source code distributed under the GNU Lesser General Public License:
               http://github.com/beast-dev/beast-mcmc
```

Then it gives some details about the data it loaded and the models you have specified (some lines have been omitted, below, for brevity). You should see that it has repeated all of the choices you made in BEAUti.

```
Random number seed: 1503410333443

Parsing XML file: YFV.xml

Read alignment: alignment
  Sequences = 71
      Sites = 654
   Datatype = nucleotide
Site patterns 'CP1.patterns' created from positions 1-654 of alignment 'alignment'
  only using every 3 site
  unique pattern count = 59
Site patterns 'CP2.patterns' created from positions 2-654 of alignment 'alignment'
  only using every 3 site
  unique pattern count = 33
Site patterns 'CP3.patterns' created from positions 3-654 of alignment 'alignment'
  only using every 3 site
  unique pattern count = 201

Creating the tree model, 'treeModel'
  initial tree topology = (...)
  tree height = 69.38328510064179

Using strict molecular clock model.

Creating state frequencies model 'frequencies': Initial frequencies = {0.25, 0.25, 0.25, 0.25}

Creating HKY substitution model. Initial kappa = 2.0

Creating state frequencies model 'frequencies': Initial frequencies = {0.25, 0.25, 0.25, 0.25}

Creating HKY substitution model. Initial kappa = 2.0

Creating state frequencies model 'frequencies': Initial frequencies = {0.25, 0.25, 0.25, 0.25}

Creating HKY substitution model. Initial kappa = 2.0

Creating site rate model:
  with initial relative rate = 1.0
  4 category discrete gamma with initial shape = 0.5

Creating site rate model:
  with initial relative rate = 1.0
  4 category discrete gamma with initial shape = 0.5

Creating site rate model:
  with initial relative rate = 1.0
  4 category discrete gamma with initial shape = 0.5
.
.
.
Creating the MCMC chain:
  chainLength=100000
  autoOptimize=true
  autoOptimize delayed for 1000 steps
```

Next it prints out a block of citations for BEAST and for the individual models and components selected. This is intended to help you write up the analysis, specifying and citing the models used:

```
Citations for this analysis:

FRAMEWORK
BEAST primary citation:
        Drummond AJ, Suchard MA, Xie Dong, Rambaut A (2012) Bayesian phylogenetics with BEAUti and the BEAST 1.7. Mol Biol Evol. 29, 1969-19
. DOI:10.1093/molbev/mss075

SUBSTITUTION MODELS
HKY nucleotide substitution model:
        Hasegawa M, Kishino H, Yano T (1985) Dating the human-ape splitting by a molecular clock of mitochondrial DNA. J. Mol. Evol.. 22, 16
174
Discrete gamma-distributed rate heterogeneity model:
        Yang Z (1994) Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. J.
l. Evol.. 39, 306-314

PRIOR MODELS
CTMC Scale Reference Prior model:
        Ferreira MAR, Suchard MA (2008) Bayesian analysis of elapsed times in continuous-time Markov chains. Canadian Journal of Statistics.
6, 355-368
```

Finally BEAST starts to run. It prints up various pieces of information that is useful for keeping track of what is happening. The first column is the 'state' number — in this case it is incrementing by 1000 so between ea
these lines it has made 1000 operations. The screen log shows only a few of the metrics and parameters but it is also recording a log file to disk with all of the results in it (along with a '.trees' file containing the sampl
trees for these states).

After a few thousand states it will start to report the number of hours per million states (or if it is running very fast, per billions states). This is useful to allow you to predict how long the run is going to take and whether
have time to go and get a cup of coffee, or lunch, or a two week vacation in the Caribbean.

```
# BEAST v1.10.0 Prerelease #VEME2017
# Generated Tue Aug 22 14:59:04 BST 2017 [seed=1503410333443]
# -beagle
state   Posterior     Prior       Likelihood    rootAge   clock.rate
0       -17832.0434   -209.3400   -17622.7035   1939.62   1.00000     -
100     -16759.9962   -201.5887   -16558.4075   1939.63   0.79663     -
200     -15788.1880   -195.6302   -15592.5578   1939.63   0.74501     -
300     -15339.8944   -201.7870   -15138.1074   1939.65   0.74501     -
400     -14704.4588   -192.4580   -14512.0008   1939.65   0.61861     -
.
.
```

After waiting the expected amount of time, BEAST will finish.

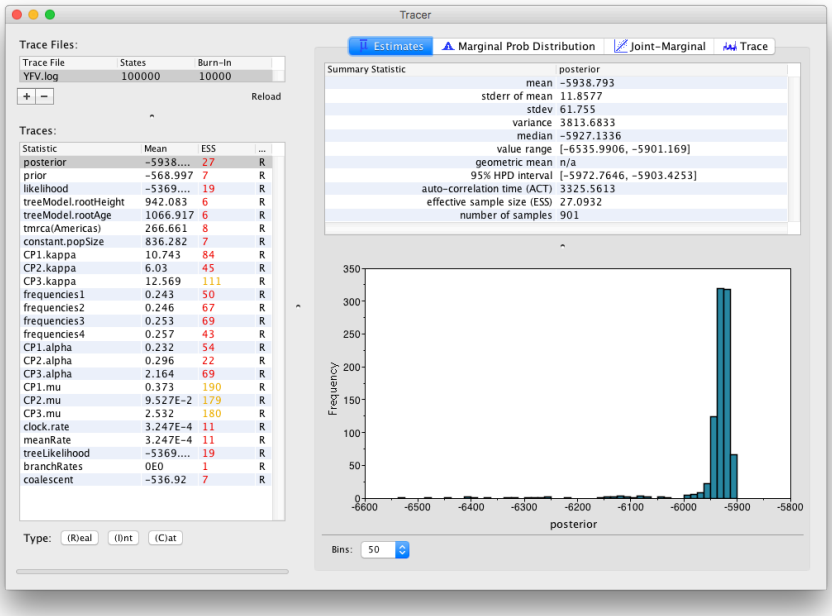
```
.
.
99500   -5947.7998   -606.3356   -5341.4642   535.003   2.06631E-4   43.95 hours/billion states
99600   -5944.2435   -605.9852   -5338.2583   464.495   2.06631E-4   43.95 hours/billion states
99700   -5943.2009   -606.0432   -5337.1577   471.835   1.88318E-4   43.94 hours/billion states
99800   -5952.6018   -610.7744   -5341.8274   549.930   2.10672E-4   43.95 hours/billion states
99900   -5944.0227   -603.4808   -5340.5419   730.490   2.08943E-4   43.95 hours/billion states
100000  -5944.2243   -600.5219   -5343.7025   598.543   2.08943E-4   43.95 hours/billion states

Operator analysis
Operator                                     Tuning   Count      Time     Time/Op  Pr(accept)
scale(CP1.kappa)                             0.357   1153       187      0.16     0.2402
scale(CP2.kappa)                             0.219   1049       149      0.14     0.2479
scale(CP3.kappa)                             0.55    1116       410      0.37     0.2348
frequencies                                  0.07    1105       438      0.4      0.2471
scale(CP1.alpha)                             0.385   1109       224      0.2      0.2435
scale(CP2.alpha)                             0.245   1161       189      0.16     0.2343
scale(CP3.alpha)                             0.46    1158       397      0.34     0.2383
scale(clock.rate)                            0.747   3406       1345     0.39     0.2372
up:clock.rate down:nodeHeights(treeModel)    0.905   3471       979      0.28     0.2264
allMus                                       0.129   3430       957      0.28     0.2423
subtreeSlide(treeModel)                      55.702  16910      1668     0.1      0.2367
Narrow Exchange(treeModel)                           17007      1449     0.09     0.1364
Wide Exchange(treeModel)                             3373       184      0.05     0.0044
wilsonBalding(treeModel)                             3342       466      0.14     0.0093
scale(treeModel.rootHeight)                  0.304   3441       324      0.09     0.2479
uniform(nodeHeights(treeModel))                      34356      4387     0.13     0.4551
scale(constant.popSize)                      0.514   3413       132      0.04     0.2429

17.445 seconds
```

The table at the end lists each of the operators, how many times each was used, how much time they took and some other details. This information may be useful for optimising the performance of runs but generally be ignored.

Analysing the BEAST output


To analyze the results of running BEAST we are going to use the program Tracer (tracer). Run Tracer by double clicking on the Tracer icon.

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 one below. Remember that MCMC is a stochastic algorithm so the actual numbers will not be exactly the same.



On the left hand side at the top is the name of the log file loaded (in this case `YFB.log`), the number of states in the log and the burn-in (10% by default). Below this is a table containing all the columns in the log file including various statistics and continuous parameters. Selecting a trace on the left brings up analyses for this trace on the right hand side depending on the tab that is selected at the top. When first opened, the `posterior` trace (a quantity proportional to posterior — the product of the data likelihood and the prior probabilities, on the log-scale) is selected and various statistics of this trace are shown under the `Estimates`

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.

stderr 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.

stdev

The standard deviation of the samples.

variance

The variance of the samples.

median

The median value of the samples.

value range

The full range of values sampled.

geometric mean

The central tendency or typical value of the set of samples.

95% HPD Interval

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

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 t trace (excluding the burn-in).

effective sample size (ESS)

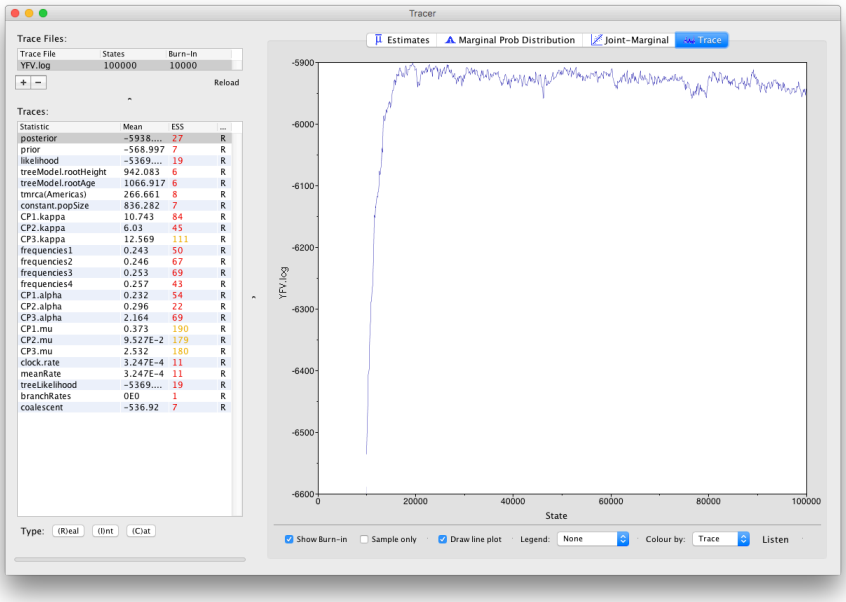
The effective sample size (ESS) is the number of independent samples that the trace is equivalent to. This is calculated as the chain length (excluding the burn-in) divided by the ACT.

number of samples

The number of samples (values) used to compute the above statistics. This will be the total number of samples minus the burn-in.

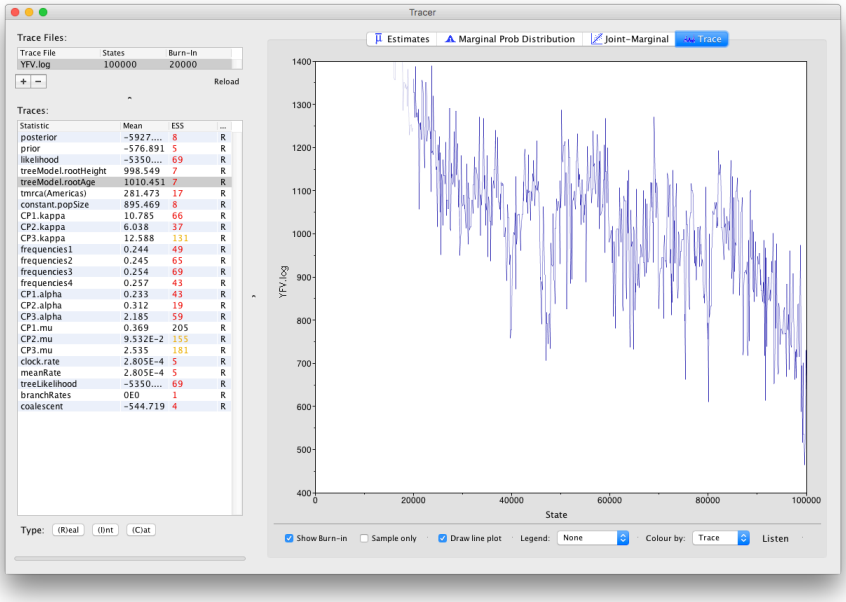
Note that the effective sample sizes (ESSs) for all the traces are small (ESSs less than 100 are highlighted in red by Tracer and values > 100 but < 200 are in yellow). This is not good. A low ESS means that the trace contained a lot of correlated samples and thus may not represent the posterior distribution 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 autocorrelated — 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.rootAge`) is only 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 resu and render estimates of ESS unreliable. Set the `burn-in` to 20,000 (doule-click on the number in the trace file table to edit it). You should see something like this:

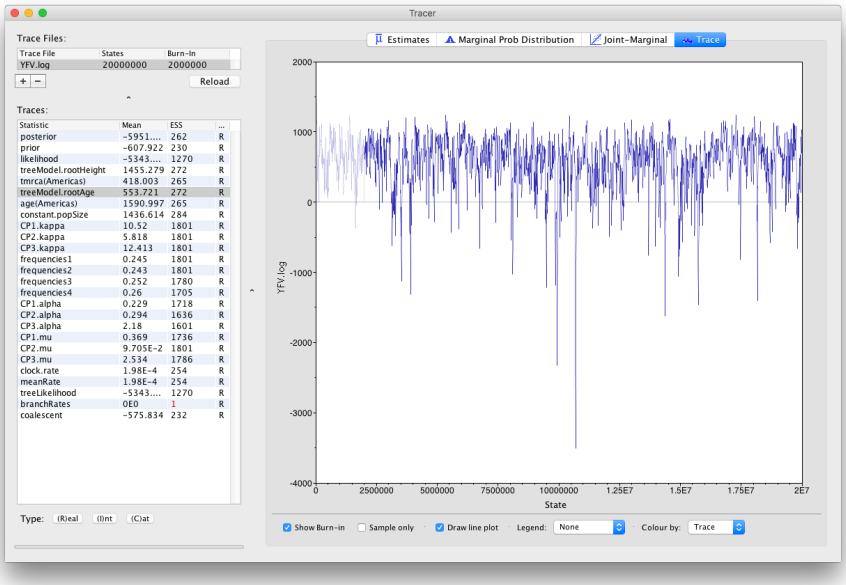


You may see some improvement in the ESSs but they will still be very low. In the image above, the root age looks to have a trend downwards which may mean that it hasn't burnt-in yet or it may be part of a longer c autocorrelated values.

The simple response to this situation is that we need to run the chain for longer. Go back to BEAUti (which hopefully you left open), go to the `MCMC` options section, above, and create a new BEAST XML file with a lo chain length (e.g. 10,000,000).

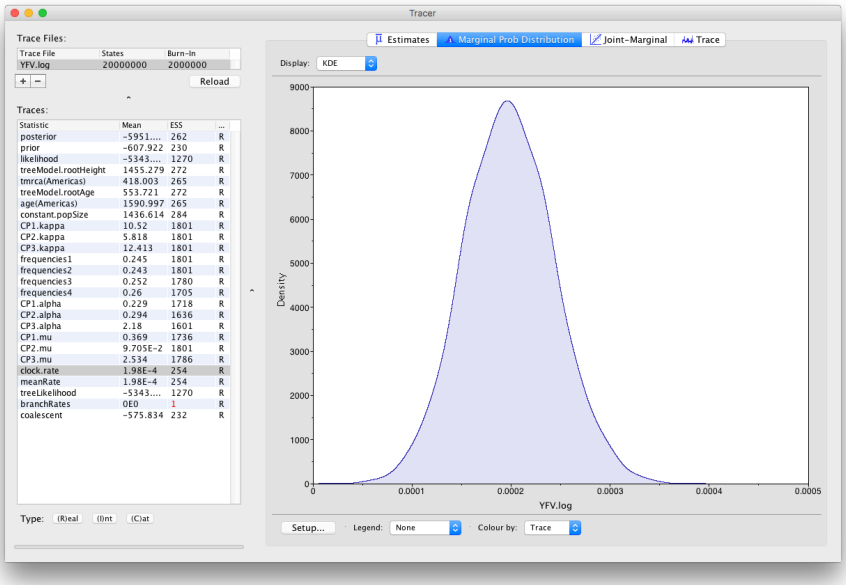
To continue the tutorial without having to wait for a long run to complete, you can make use of the BEAST output files provided with this tutorial (a chain length of 20,000,000 and logged every 10,000 sample). The files, `YFV.log` and `YFV.trees`, YFVLongRuns.zip, can be downloaded from here (/tutorials/workshop_rates_and_dates/files/YFVLongRuns.zip).

Import the new longer log file for the strict clock run, select the `treeModel.rootAge` statistic and click on the `Trace` tab to look at the raw trace plot.

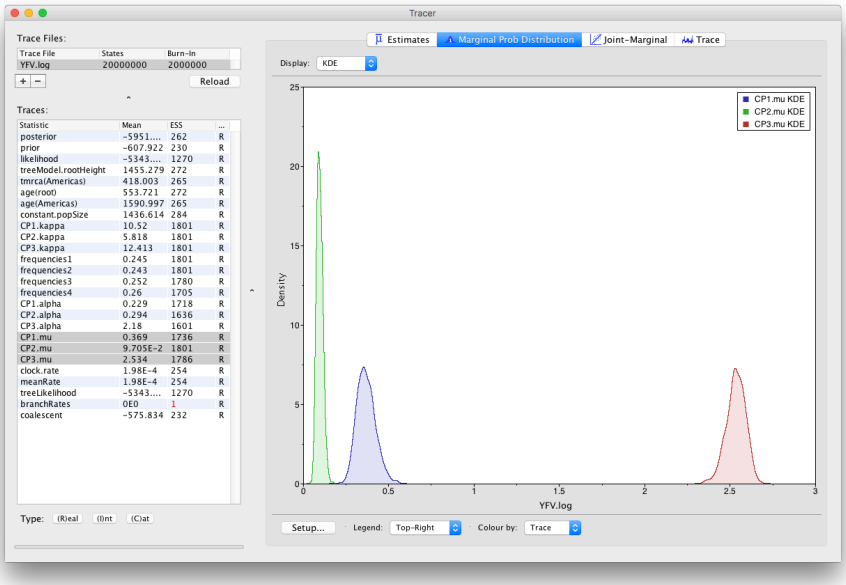


The log file provided contains 2000 samples and with an ESS of 262 for the clock.rate there is still some degree of auto-correlation between the samples but 262 effectively independent samples will now provide an reasonable estimate of the posterior distribution. There are no obvious trends in the plot 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 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 param You should see a plot similar to this:

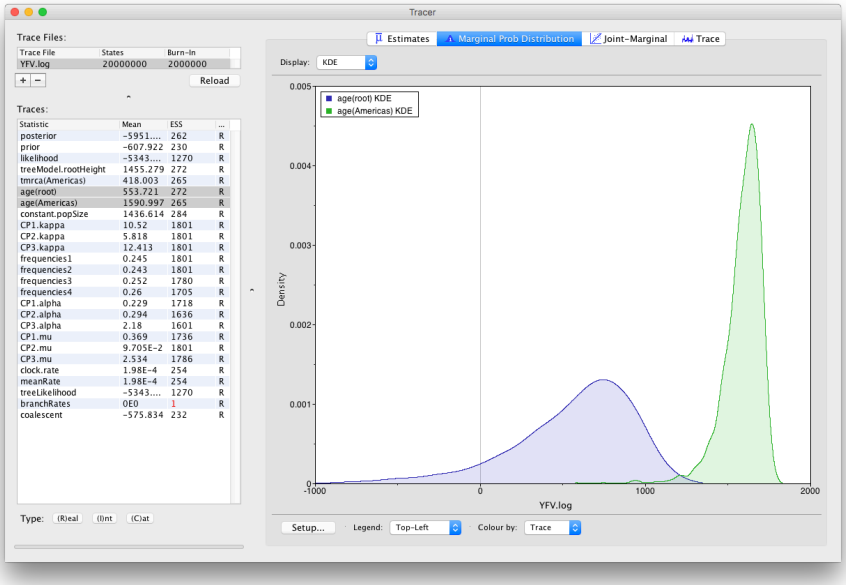


As you can see the posterior probability density is nicely bell-shaped. When looking at the equivalent histogram in the `Estimates` panel, there is some sampling noise which is smoothened by the KDE; this would b reduced if we ran the chain for longer but we already have a reasonable estimate of the 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 det 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` u `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 timescale of the tree. Select the statistics called `age(root)` and `age(Americas)`. These are the dates of the root of the tree and the clade we defined for all of the American sequence
Because we have dated the tips of the tree, these statistics are given as the calendar year with the present day being on the right hand side (the most recently sampled sequence is from 2009).



> **Note:** Negative numbers denote years as Before the Common Era (BCE) but technically the calendar goes from 1 BCE to 1 CE — there was no year zero. So if you want to report BCE years, you should take th absolute value and add 1

This indicates that the TMRCA for the Americas is significantly more recent than the entire tree and argues for a relatively recent introduction of yellow fever virus into the Americas. Note, however, that there is conside uncertainty in these estimates. Switching to the `Estimates` panel shows that the mean date of of the TMRCA into the Americas is the year 1590 but the 95% HPD credible interval spans 1377 to 1755. Bryant et al (2007) suggest that the introduction of YFV into the Americas is likely the result of the Atlantic slave trade which occurred from the 16th to 19th Centuries.

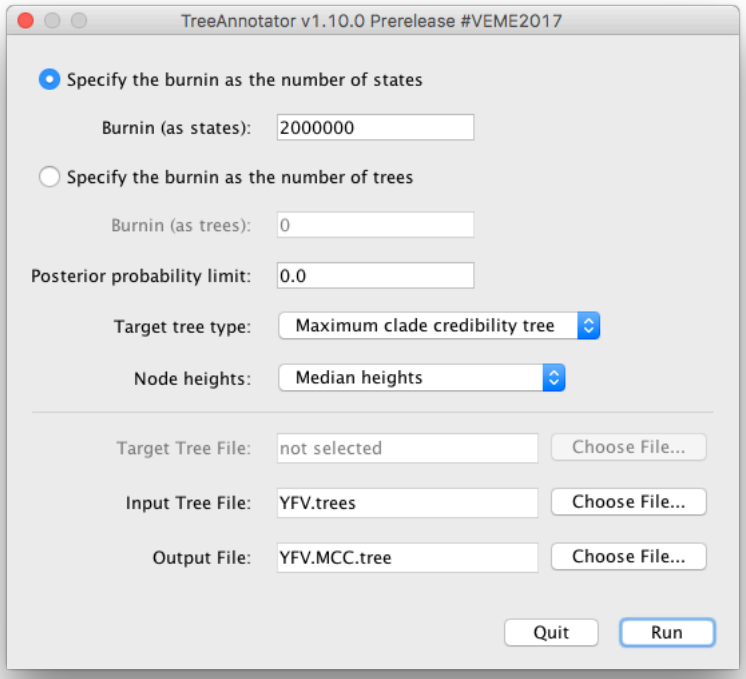
Summarizing the trees

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 `YFV.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 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.

To summarize the trees from BEAST into a single representitive tree that we can view we are going to use the BEAST utility called TreeAnnotator (treeannotator). Run TreeAnnotator by double clicking on the icon.

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 interval 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 For the example, 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, if sampling every 10000 steps, there are 2000 trees in the file. To 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. I.e., a value of **`0.5`** will mean that only nodes in the MCC tree that are present in at least 50% of t posterior distribution of trees will have posterior probabilities, height HPDs etc.

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 eve 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 ance 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 Ance heights' option.

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.

Select a `Burnin (as states)` of **`2,000,000`**, `Posterior probability limit` of **`0.0`**, `Target tree type: Maximum clade credibility tree`, and `Node heights: Median heights`. Then s `YFV.trees` as the `Input Tree File` .

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 s 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

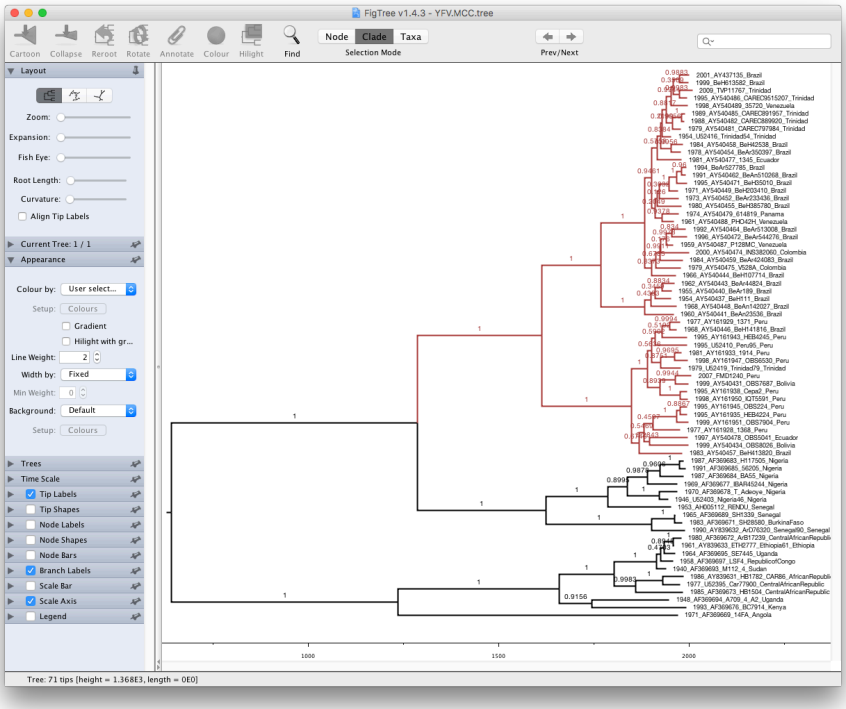
FigTree (figtree) is a user-friendly, graphical program for viewing trees and the associated information provided by BEAST. Double-click on the FigTree icon to run it.

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 ha 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 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.

We can also plot a time scale axis for this evolutionary history: select `Scale Axis` and deselect `Scale bar`. By default the timescale is going forwards in time from the root of the tree. For appropriate scaling, set `Reverse Axis` option in the `Scale Axis` panel and open the `Time Scale` section of the control panel setting the `Offset` to `2009.0`.

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 options actually alter the tree's topology or branch lengths in anyway so feel free to explore the options and settings (`Highlight` or `Collapse` the Americas clade, for example).



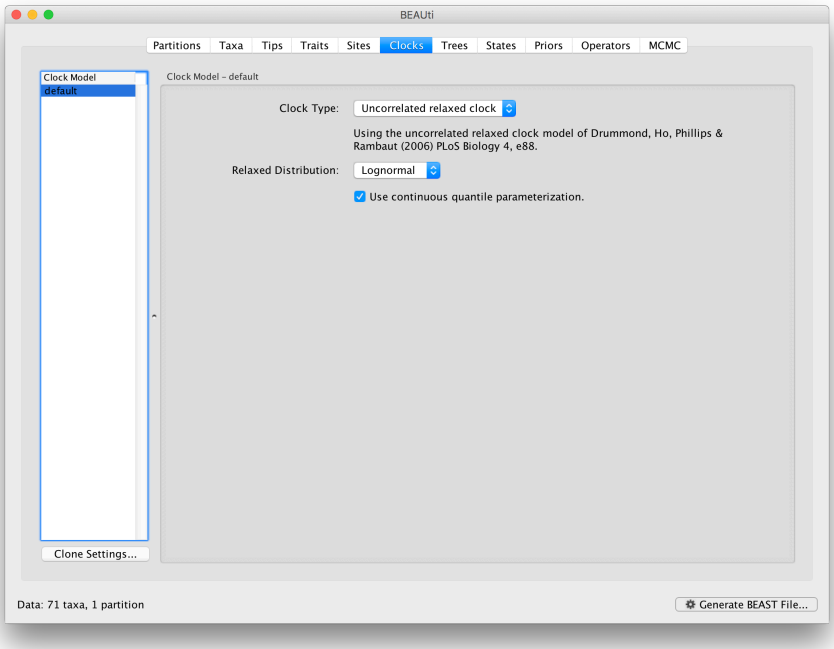
You can also save the tree and this will save most of your settings so that when you load it into FigTree again it will be displayed almost exactly as you selected.

Finally, the tree can also be exported to a graphics file (pdf, svg, etc.) using the options in the `File` menu.

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

Relaxed molecular clock analysis

A relaxed clock analysis using a lognormal distribution can also be set up for this data set, by selecting the relevant clock model in the `Clocks` tab:



Also for this analysis, we made available the output files for a longer run:

> You can make use of the BEAST output files provided with this tutorial (a chain length of 50,000,000 and logged every 50,000 sample). The files, `YFV.log` and `YFV.trees`, `YFVLongRuns.zip`, can be downloaded from here (/tutorials/workshop_rates_and_dates/files/YFVLongRuns.zip).

Is the default 10% burn-in sufficient for this sample?

Does the standard deviation estimate for the lognormal distribution in the relaxed clock suggest significant rate variation among lineages? How can we formally test this?

Does the hypothesis of YFV introduction in the Americas during the slave trade also hold under this model?

References

Bryant JE, Holmes EC and Barrett ADT (2007) Out of Africa: A Molecular Perspective on the Introduction of Yellow Fever Virus into the Americas. ***PLoS Pathogens***, **3**: e75. doi: 10.1371/journal.ppat.0030075

Help and documentation

The BEAST website: http://beast.community

Tutorials: http://beast.community/tutorials

Frequently asked questions: http://beast.community/faq

Tags: tutorial (tag_tutorial.html) workshop (tag_workshop.html)

©2019 BEAST Developers. All rights r
Site uses a modified Documentation Theme for Jekyll by Tom
Page last updated: August
Site last generated: Jun