M. Suchard, G. Baele and P. Lemey July 2017

Phylogeographic inference in discrete space

A hands-on practical

This chapter provides a step-by-step tutorial on reconstructing the spatial dispersal and cross-species dynamics of rabies virus (RABV) in North American bat populations based on a set of 372 nucleoprotein gene sequences (nucleotide positions: 594–1353). The data set comprises a total of 17 bat species sampled between 1997 and 2006 across 14 states in the United States (Streicker *et al*., Science, 2010, 329, 676-679). Following Faria et al. (Phil. Trans. Roy. Soc. B, 2013), two additional species that had been excluded from the original analysis owing to a limited amount of available sequences, Myotis austroriparius (Ma) and Parastrellus hesperus (Ph), are also included here. We also include a viral sequence with an unknown sampling date (accession no. TX5275, sampled in Texas from Lasiurus borealis), which will be adequately accommodated in our inference.

The aim of this tutorial is to estimate the ancestral locations of the virus using a Bayesian discrete phylogeographic approach and, at the same time, infer the history of host jumping using the same model approach. Using an extension of the discrete diffusion model, we will then test the factors that underly the host transition dynamics.

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 the following software packages in a format that is compatible with your computer system (all 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 (pre-)releases can also be found at https://github.com/beast-dev/beast-mcmc/releases.

- **BEAGLE** - this is a high-performance library that can perform the core calculations at the heart of most Bayesian and Maximum Likelihood phylogenetics packages. It can make use of highly-parallel processors such as those in graphics cards (GPUs) found in many PCs. Binary installers and installation instructions can be found at: https://github.com/beagle-dev/beagle-lib.

- **Tracer** - this program is used to explore the output of *BEAST* (and other Bayesian MCMC programs). It graphically and quantitively summarizes the 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://tree.bio.ed.ac.uk/software/tracer/ .

- **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

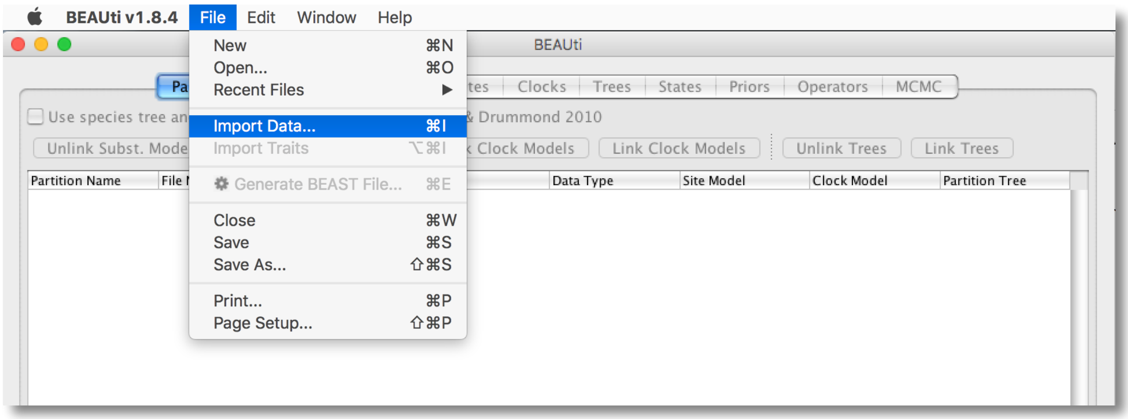
- **SpreaD3** - this is an application for the visualization of phylogeographic analyses performed with *BEAST*. At the time of writing, the current version is v0.9.7.1 It is available for download from https://rega.kuleuven.be/cev/ecv/software/SpreaD3.

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.

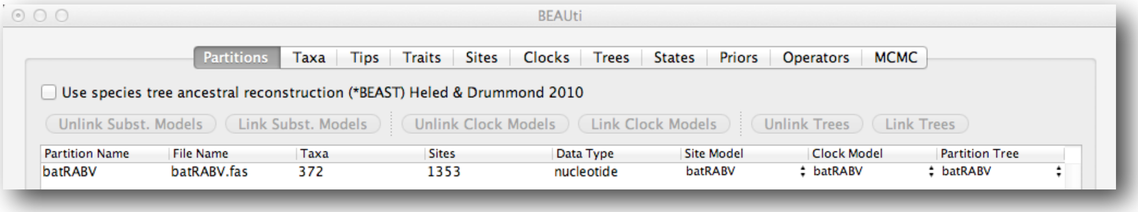
Loading the NEXUS file

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



The NEXUS alignment

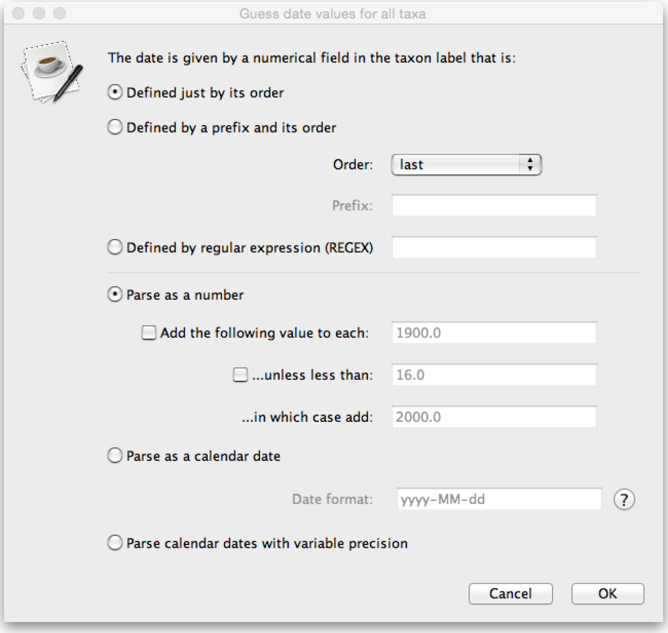
Select the file called `batRABV.fas`. This file contains an alignment of 372 nucleoprotein gene sequences of bat rabies viruses, 1353 nucleotides in length. Once loaded, the sequence data will be listed under **Partitions**:



Partition Name	File Name	Taxa	Sites	Data Type	Site Model	Clock Model	Partition Tree
batRABV	batRABV.fas	372	1353	nucleotide	batRABV	batRABV	batRABV

Specifying the sampling date information

To inform BEAUti/BEAST about the sampling dates of the sequences, go to the **Tips** tab 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). In this case, the RABV sequences have been sampled at various dates going back to 1997. 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 tab. Clicking this will make a dialog box appear:



The date is given by a numerical field in the taxon label that is:

● Defined just by its order

○ Defined by a prefix and its order

Order: last

Prefix:

○ Defined by regular expression (REGEX)

● Parse as a number

□ Add the following value to each: 1900.0

□ ...unless less than: 16.0

...in which case add: 2000.0

○ Parse as a calendar date

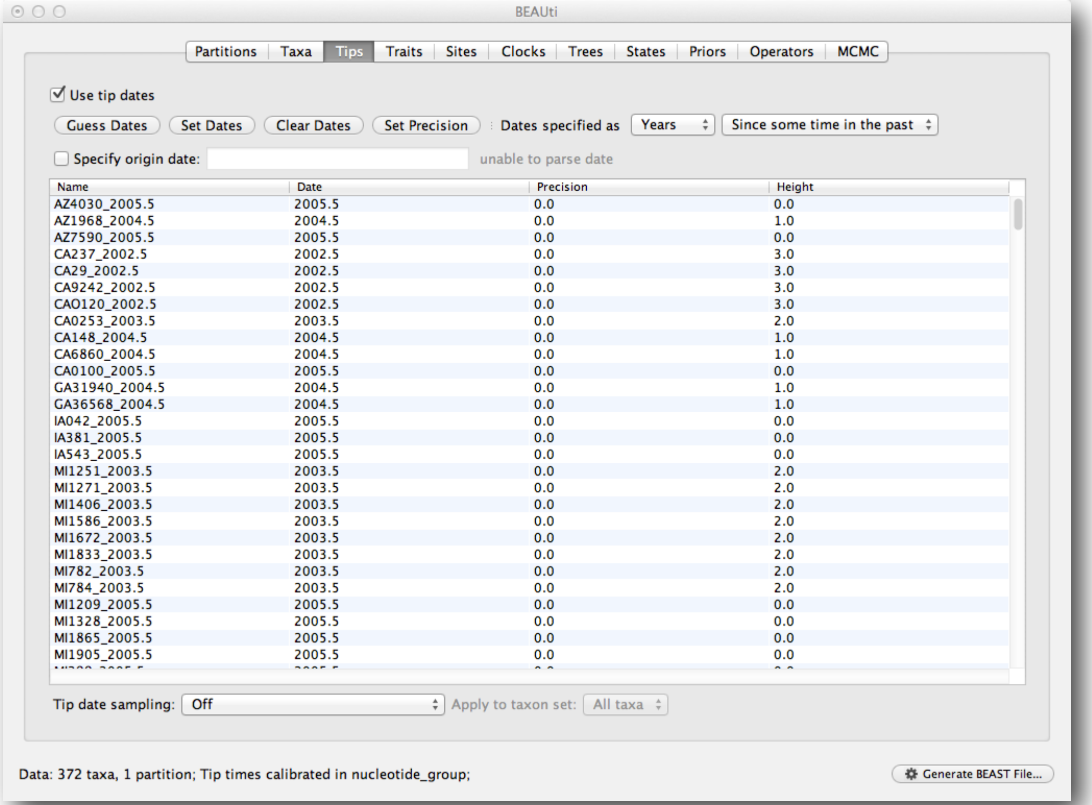
Date format: yyyy-MM-dd ?

○ Parse calendar dates with variable precision

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 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).

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. These operations are not necessary in our case since the dates are fully specified at the end of the sequence names. There is also an option to parse calendar dates and one for calendar dates with various precisions. For the H1N1/09 sequences you can keep the default 'Defined just by its order' and select 'last' from the drop-down menu for the order and press 'OK'. The dates will appear in the appropriate column of the main window. You can then check these and edit them manually as required. 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 2005) 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 2005.5). The Precision column allows specifying with what precision the sampling time is know. To include taxa only known up to the year of sampling for example (e.g., 2005), a precision of 1 year can be set and the age of those tips can be integrated over the time interval of 1 year using the Tip date sampling option at the bottom left of the Tips panel.



☑ Use tip dates

Dates specified as: Years; Since some time in the past

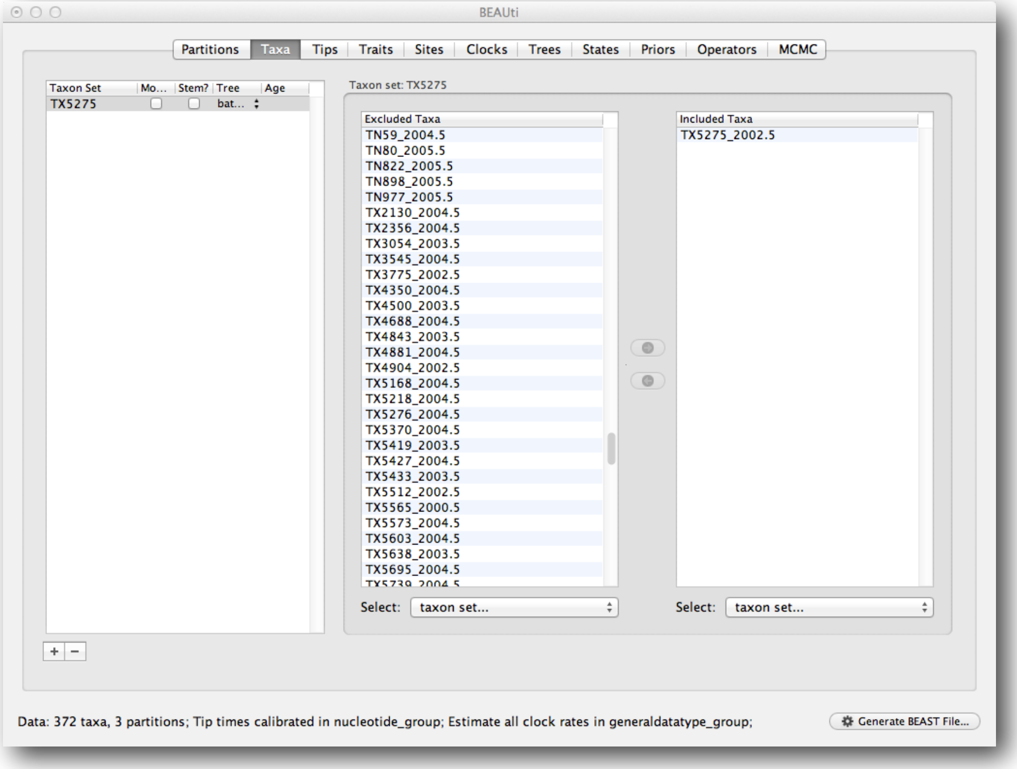
□ Specify origin date: unable to parse date

Name	Date	Precision	Height
AZ4030_2005.5	2005.5	0.0	0.0
AZ1968_2004.5	2004.5	0.0	1.0
AZ7590_2005.5	2005.5	0.0	0.0
CA237_2002.5	2002.5	0.0	3.0
CA29_2002.5	2002.5	0.0	3.0
CA9242_2002.5	2002.5	0.0	3.0
CAO120_2002.5	2002.5	0.0	3.0
CA0253_2003.5	2003.5	0.0	2.0
CA148_2004.5	2004.5	0.0	1.0
CA6860_2004.5	2004.5	0.0	1.0
CA0100_2005.5	2005.5	0.0	0.0
GA31940_2004.5	2004.5	0.0	1.0
GA36568_2004.5	2004.5	0.0	1.0
IA042_2005.5	2005.5	0.0	0.0
IA381_2005.5	2005.5	0.0	0.0
IA543_2005.5	2005.5	0.0	0.0
MI1251_2003.5	2003.5	0.0	2.0
MI1271_2003.5	2003.5	0.0	2.0
MI1406_2003.5	2003.5	0.0	2.0
MI1586_2003.5	2003.5	0.0	2.0
MI1672_2003.5	2003.5	0.0	2.0
MI1833_2003.5	2003.5	0.0	2.0
MI782_2003.5	2003.5	0.0	2.0
MI784_2003.5	2003.5	0.0	2.0
MI1209_2005.5	2005.5	0.0	0.0
MI1328_2005.5	2005.5	0.0	0.0
MI1865_2005.5	2005.5	0.0	0.0
MI1905_2005.5	2005.5	0.0	0.0

Tip date sampling: Off   Apply to taxon set: All taxa

Data: 372 taxa, 1 partition; Tip times calibrated in nucleotide_group;

In our data set, the sampling date is unknown for one particular sequence (TX5275_2002.5, the ‘2002.5’ is simply an arbitrary date that will be used as a starting value). To appropriately accommodate the uncertainty on the age of this tip, we will instruct *BEAST* to integrate over a particular sampling time interval for this tip. First, go back to the **Taxa** tab that we skipped, and make a taxon set for only that particular sequence. Press the small “plus” button at the bottom left of the panel ; this creates a new taxon set. Rename it by double-clicking on the entry that appears (it will initially be called `untitled1`). Call it TX5275 and keep the default settings. Move TX5275_2002.5 from the Excluded Taxa window to the Included Taxa window:



Taxon Set	Mo...	Stem?	Tree	Age
TX5275	□	□	bat...	

Taxon set: TX5275

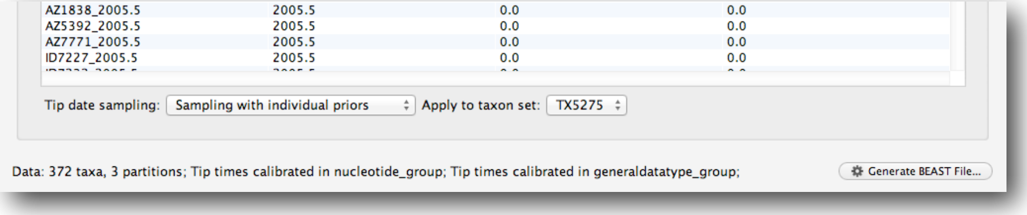
Excluded Taxa
TN59_2004.5
TN80_2005.5
TN822_2005.5
TN898_2005.5
TN977_2005.5
TX2130_2004.5
TX2356_2004.5
TX3054_2003.5
TX3545_2004.5
TX3775_2002.5
TX4350_2004.5
TX4500_2003.5
TX4688_2004.5
TX4843_2003.5
TX4881_2004.5
TX4904_2002.5
TX5168_2004.5
TX5218_2004.5
TX5276_2004.5
TX5370_2004.5
TX5419_2003.5
TX5427_2004.5
TX5433_2003.5
TX5512_2002.5
TX5565_2000.5
TX5573_2004.5
TX5603_2004.5
TX5638_2003.5
TX5695_2004.5
TX5739_2004.5

Included Taxa
TX5275_2002.5

Select: taxon set...   Select: taxon set...

Data: 372 taxa, 3 partitions; Tip times calibrated in nucleotide_group; Estimate all clock rates in generaldatatype_group;

Go back to the **Tips** tab, and in the bottom left, select the sampling with individual priors as Tip date sampling option. Apply this to the TX5275 taxa set instead of the default All taxa option. We will set a prior on its age when we get to the **Priors** tab.



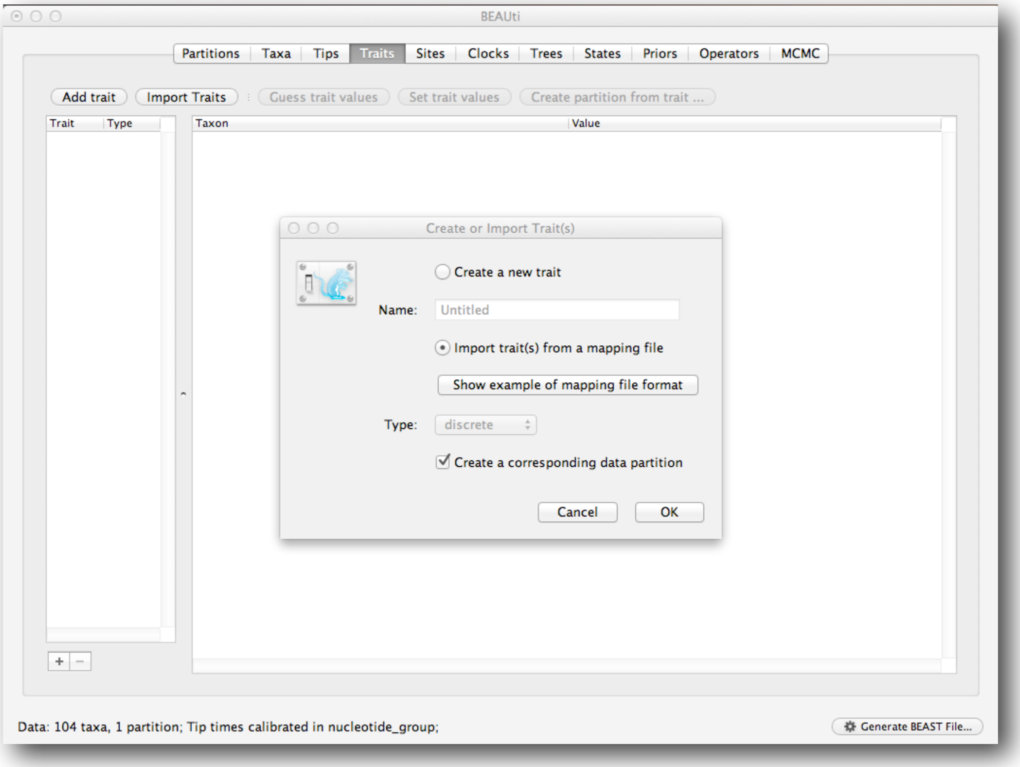
AZ1838_2005.5	2005.5	0.0	0.0
AZ5392_2005.5	2005.5	0.0	0.0
AZ7771_2005.5	2005.5	0.0	0.0
ID7227_2005.5	2005.5	0.0	0.0

Tip date sampling: Sampling with individual priors   Apply to taxon set: TX5275

Data: 372 taxa, 3 partitions; Tip times calibrated in nucleotide_group; Tip times calibrated in generaldatatype_group;

Specifying the trait information for the sequences

The next thing to do is to click on the **Traits** tab at the top of the main window. A trait can be any characteristic that is inherent to the specific taxon, for example, geographical location or species host. This step will assign a specific host and geographical location to each taxa. To associate the sequences with the these traits, we need to add a new trait under the **Traits** tab (click Add trait). This will open a new window to Create or Import Trait(s):


BEAUti
Partitions | Taxa | Tips | Traits | Sites | Clocks | Trees | States | Priors | Operators | MCMC
Add trait
Import Traits
Guess trait values
Set trait values
Create partition from trait ...
Trait
Type
Taxon
Value
Cancel
OK
Generate BEAST File...

Create or Import Trait(s)

○ Create a new trait

Name: Untitled

● Import trait(s) from a mapping file

Show example of mapping file format

Type: discrete

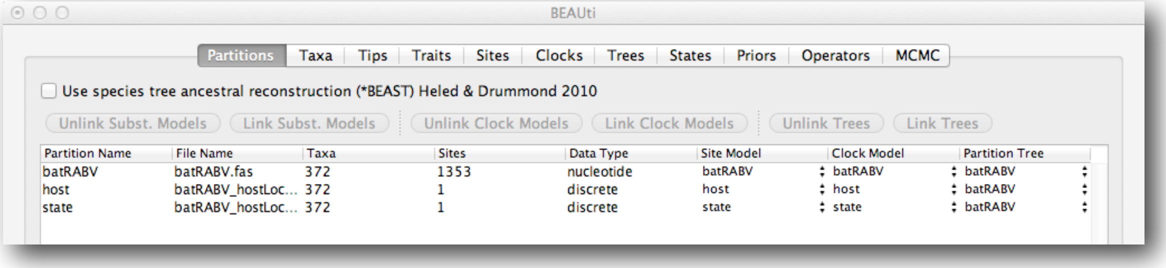
☑ Create a corresponding data partition

Data: 104 taxa, 1 partition; Tip times calibrated in nucleotide_group;

Select **Import trait(s) from a mapping file format** (the format of such a file can be shown). Browse to and load the **batRABV_hostLocation.txt** tab-delimited file which contains the discrete host and location for each sequence. Note that the host species is specified using a two-character abbreviation (e.g. Ef for Eptesicus fuscus, three characters for Lbl).

```
traits	host	state
AZ4030_2005.5Ap	Arizona
AZ1968_2004.5Ef	Arizona
AZ7590_2005.5Ef	Arizona
CA237_2002.5	Ef	California
CA29_2002.5	Ef	California
CA9242_2002.5		Ef	California
CAO120_2002.5		Ef	California
CA0253_2003.5		Ef	California
CA148_2004.5	Ef	California
CA6860_2004.5		Ef	California
CA0100_2005.5		Ef	California
GA31940_2004.5		Ef	Georgia
	....
TX3545_2004.5Tb		Texas
```

After clicking **OK**, select the host trait and click on **create partition from trait..'**. This new partition will be shown under the **Partitions** tab. Do the same for the location trait (state), resulting in three partitions in the **Partitions** tab:



□ Use species tree ancestral reconstruction (*BEAST) Heled & Drummond 2010

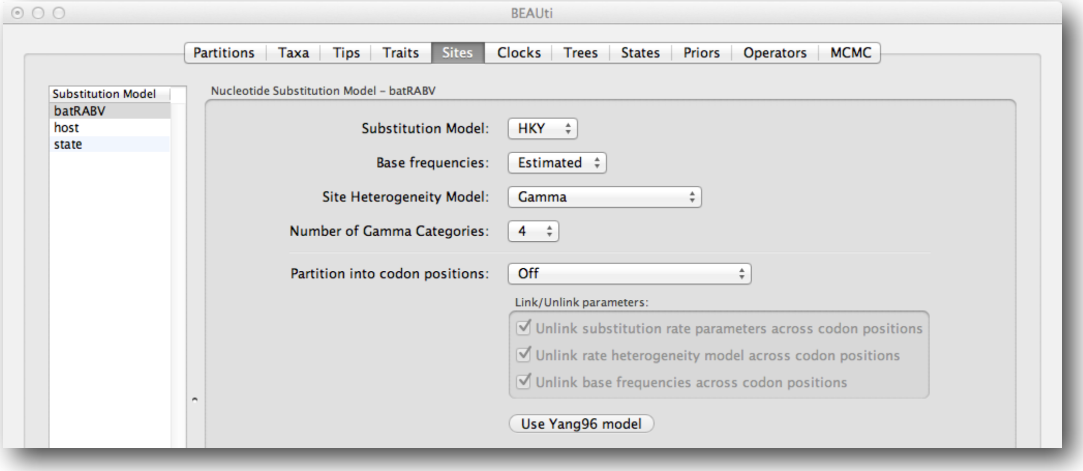
Partition Name	File Name	Taxa	Sites	Data Type	Site Model	Clock Model	Partition Tree
batRABV	batRABV.fas	372	1353	nucleotide	batRABV	batRABV	batRABV
host	batRABV_hostLoc...	372	1	discrete	host	host	batRABV
state	batRABV_hostLoc...	372	1	discrete	state	state	batRABV

Setting the evolutionary and diffusion models

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**:

- 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.

For the nucleotide model in this tutorial, keep the default HKY substitution model, set base frequencies to Empirical, and use Gamma-distributed rate variation among sites (with 4 discrete categories):



Nucleotide Substitution Model – batRABV

Substitution Model: HKY

Base frequencies: Estimated

Site Heterogeneity Model: Gamma

Number of Gamma Categories: 4

Partition into codon positions: Off

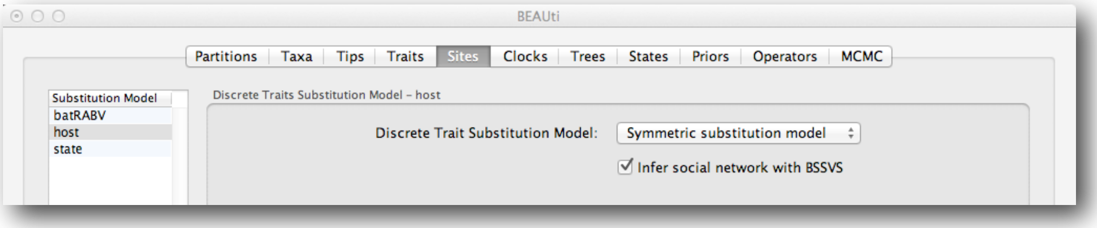
Link/Unlink parameters:

☑ Unlink substitution rate parameters across codon positions

☑ Unlink rate heterogeneity model across codon positions

☑ Unlink base frequencies across codon positions

Click on host in the Substitution model window and keep the Discrete Trait Substitution Model to Symmetric substitution model and select the option to perform BSSVS (Infer social network with BSSVS). The Symmetric substitution model specifies a discrete phylogeographic analysis using a standard continuous-time Markov chain (CTMC), in which the transition rates between locations are reversible. The alternative Asymmetric substitution model specifies a discrete phylogeographic analysis using a nonreversible CTMC. Selecting the BSSVS option enables the Bayesian Stochastic Search Variable Selection procedure. This procedure will attempt to invoke a limited number of rates (at least *k*-1) to adequately explain the phylogenetic diffusion process.



Discrete Traits Substitution Model – host

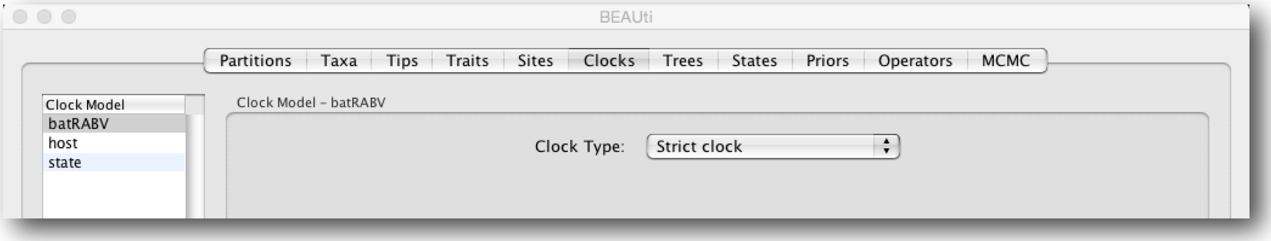
Discrete Trait Substitution Model: Symmetric substitution model

☑ Infer social network with BSSVS

Apply the same discrete diffusion model settings to the spatial ‘state’ trait.

Setting the clock model

Click on the **Clocks** tab at the top of the main window. We will perform our run using the default Strict clock model. We can also keep default settings for overall rate scalers in the host and location state transition processes.



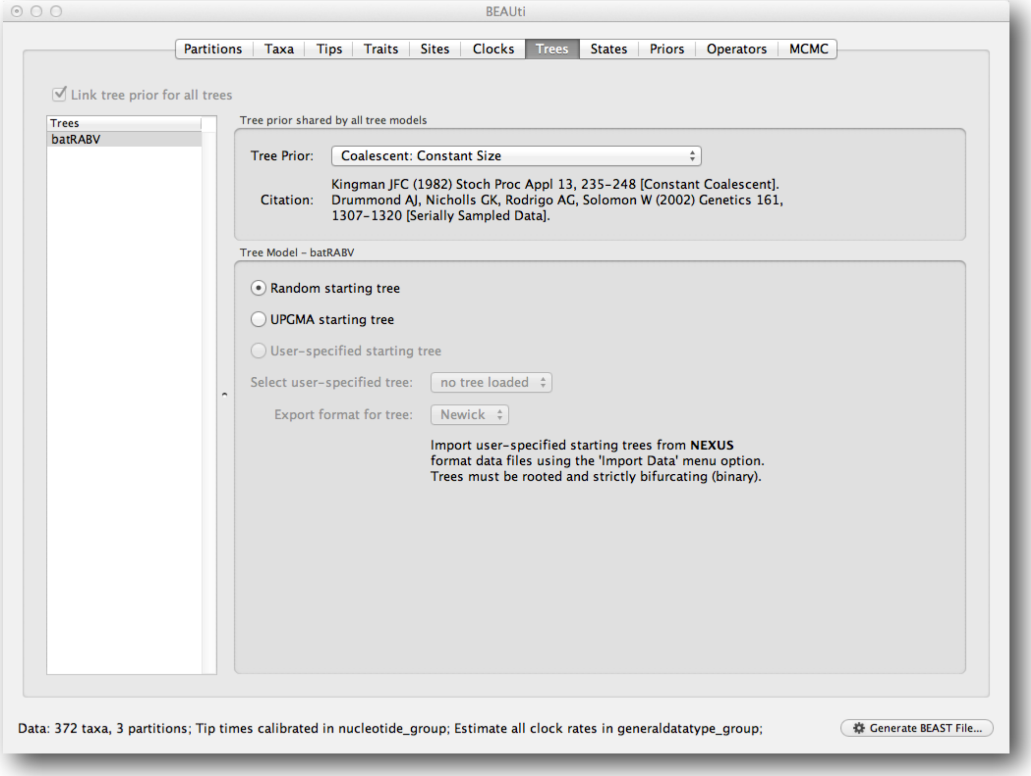
Clock Model: batRABV, host, state

Clock Model – batRABV

Clock Type: Strict clock

Setting the starting tree and tree prior

Click on the **Trees** tab at the top of the main window. We will select a simple constant size coalescent tree prior (Coalescent: Constant Size) and keep the default random starting tree.



☑ Link tree prior for all trees

Trees: batRABV

Tree prior shared by all tree models

Tree Prior: Coalescent: Constant Size

Citation: Kingman JFC (1982) Stoch Proc Appl 13, 235-248 [Constant Coalescent]. Drummond AJ, Nicholls GK, Rodrigo AG, Solomon W (2002) Genetics 161, 1307-1320 [Serially Sampled Data].

Tree Model – batRABV

● Random starting tree

○ UPGMA starting tree

○ User-specified starting tree

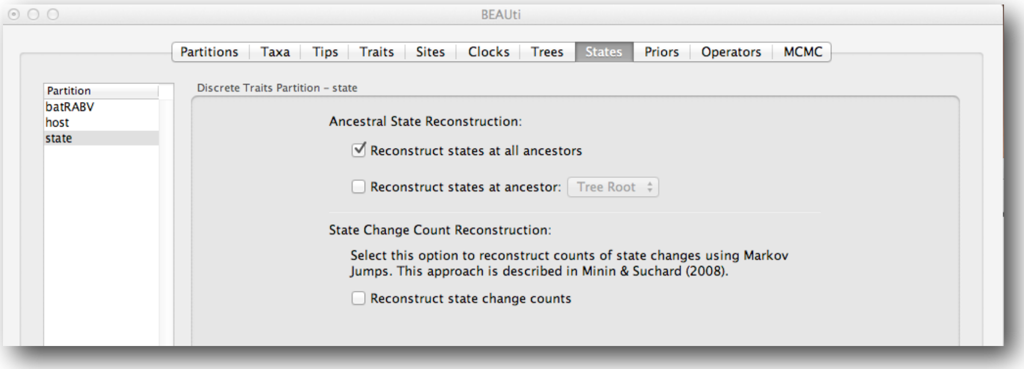
Select user-specified tree: no tree loaded

Export format for tree: Newick

Import user-specified starting trees from **NEXUS** format data files using the 'Import Data' menu option. Trees must be rooted and strictly bifurcating (binary).

Data: 372 taxa, 3 partitions; Tip times calibrated in nucleotide_group; Estimate all clock rates in generaldatatype_group;

In the States tab, check that for the host and state partition the option to Reconstruct states at all ancestors is selected (by default).



Partition: batRABV, host, state

Discrete Traits Partition – state

Ancestral State Reconstruction:

☑ Reconstruct states at all ancestors

☐ Reconstruct states at ancestor: Tree Root

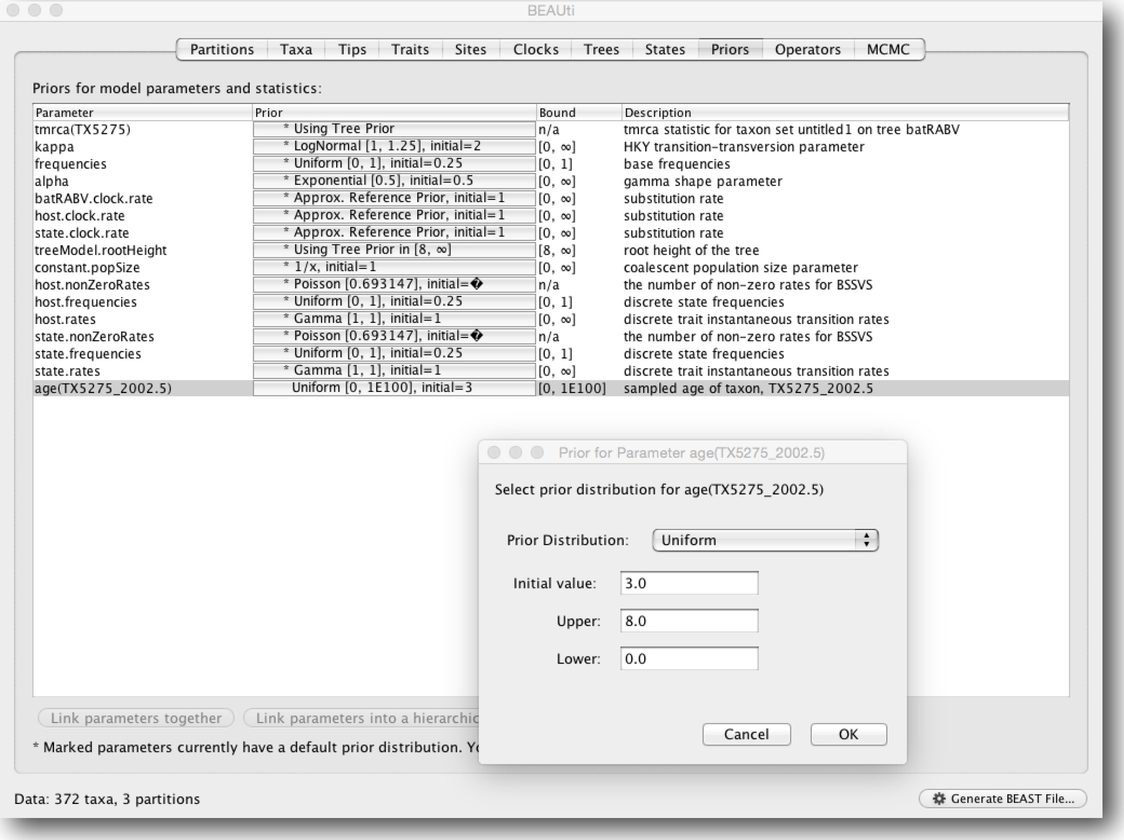
State Change Count Reconstruction:

Select this option to reconstruct counts of state changes using Markov Jumps. This approach is described in Minin & Suchard (2008).

☐ Reconstruct state change counts

Setting up the priors

Review the prior settings under the **Priors** tab. 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). The same is applied to the discrete host and location state rate. There is also a default uniform prior specification for the age of TX5275 (`age(TX5275_2002.5)`). We will assume that its sampling time for this tip is bounded by the sampling time distribution of the this data set, implying that it is sampled between 1997.5 and 2005.5. Click on the current uniform prior setting, set the `Upper` age to 8 years (reflecting the 1997.5 boundary) and click `OK`.


BEAUti
Partitions | Taxa | Tips | Traits | Sites | Clocks | Trees | States | Priors | Operators | MCMC

Priors for model parameters and statistics:

Parameter	Prior	Bound	Description
tmrca(TX5275)	* Using Tree Prior	n/a	tmrca statistic for taxon set untitled1 on tree batRABV
kappa	* LogNormal [1, 1.25], initial=2	[0, ∞]	HKY transition-transversion parameter
frequencies	* Uniform [0, 1], initial=0.25	[0, 1]	base frequencies
alpha	* Exponential [0.5], initial=0.5	[0, ∞]	gamma shape parameter
batRABV.clock.rate	* Approx. Reference Prior, initial=1	[0, ∞]	substitution rate
host.clock.rate	* Approx. Reference Prior, initial=1	[0, ∞]	substitution rate
state.clock.rate	* Approx. Reference Prior, initial=1	[0, ∞]	substitution rate
treeModel.rootHeight	* Using Tree Prior in [8, ∞]	[8, ∞]	root height of the tree
constant.popSize	* 1/x, initial=1	[0, ∞]	coalescent population size parameter
host.nonZeroRates	* Poisson [0.693147], initial=�	n/a	the number of non-zero rates for BSSVS
host.frequencies	* Uniform [0, 1], initial=0.25	[0, 1]	discrete state frequencies
host.rates	* Gamma [1, 1], initial=1	[0, ∞]	discrete trait instantaneous transition rates
state.nonZeroRates	* Poisson [0.693147], initial=�	n/a	the number of non-zero rates for BSSVS
state.frequencies	* Uniform [0, 1], initial=0.25	[0, 1]	discrete state frequencies
state.rates	* Gamma [1, 1], initial=1	[0, ∞]	discrete trait instantaneous transition rates
age(TX5275_2002.5)	Uniform [0, 1E100], initial=3	[0, 1E100]	sampled age of taxon, TX5275_2002.5

Select prior distribution for age(TX5275_2002.5)

Prior Distribution: Uniform

Initial value: 3.0

Upper: 8.0

Lower: 0.0

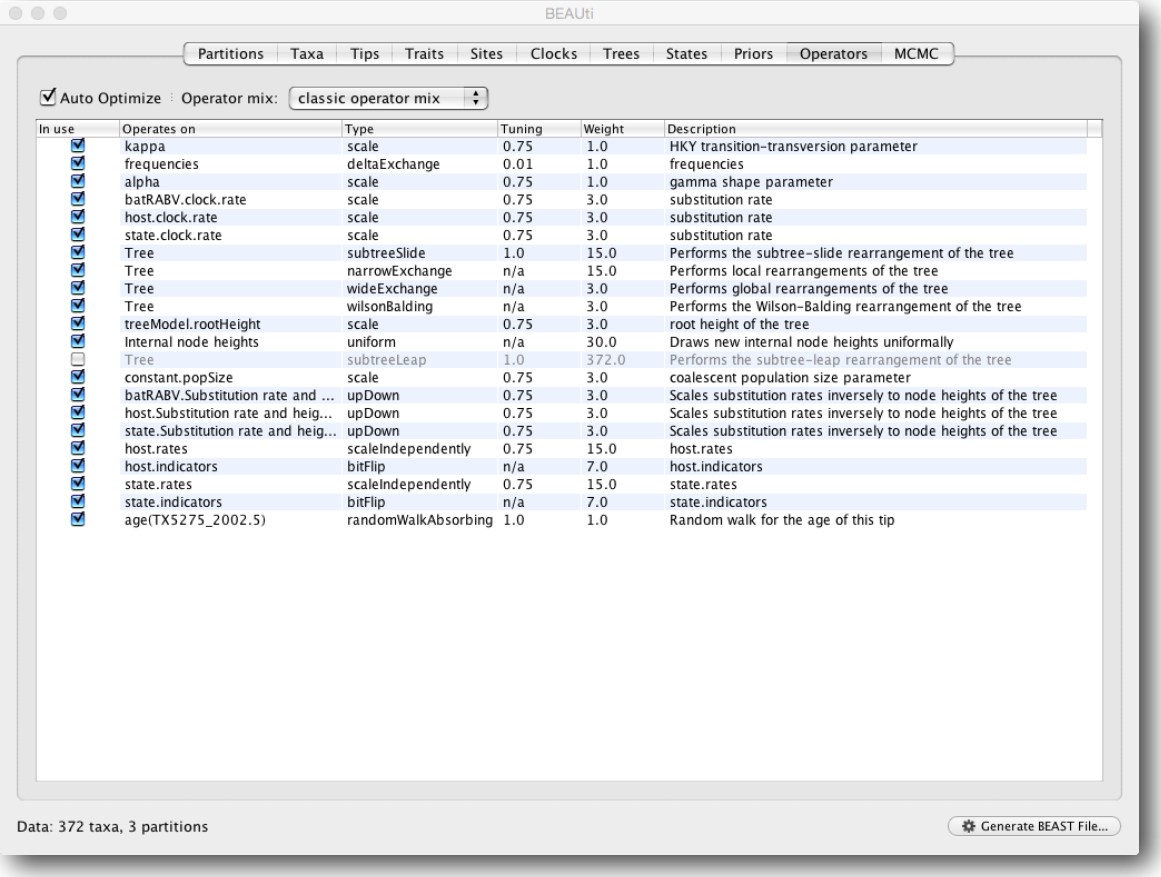
* Marked parameters currently have a default prior distribution. Yo

Data: 372 taxa, 3 partitions

Setting up the operators

Each parameter in the model has one or more ‘operators’ (these are variously called 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. 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 while 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 relate to the tree or to the divergence times of the nodes of the tree and these have special operators. 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 of 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 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.



☑ Auto Optimize Operator mix: classic operator mix

In use	Operates on	Type	Tuning	Weight	Description
☑	kappa	scale	0.75	1.0	HKY transition-transversion parameter
☑	frequencies	deltaExchange	0.01	1.0	frequencies
☑	alpha	scale	0.75	1.0	gamma shape parameter
☑	batRABV.clock.rate	scale	0.75	3.0	substitution rate
☑	host.clock.rate	scale	0.75	3.0	substitution rate
☑	state.clock.rate	scale	0.75	3.0	substitution rate
☑	Tree	subtreeSlide	1.0	15.0	Performs the subtree-slide rearrangement of the tree
☑	Tree	narrowExchange	n/a	15.0	Performs local rearrangements of the tree
☑	Tree	wideExchange	n/a	3.0	Performs global rearrangements of the tree
☑	Tree	wilsonBalding	n/a	3.0	Performs the Wilson-Balding rearrangement of the tree
☑	treeModel.rootHeight	scale	0.75	3.0	root height of the tree
☑	Internal node heights	uniform	n/a	30.0	Draws new internal node heights uniformally
□	Tree	subtreeLeap	1.0	372.0	Performs the subtree-leap rearrangement of the tree
☑	constant.popSize	scale	0.75	3.0	coalescent population size parameter
☑	batRABV.Substitution rate and ...	upDown	0.75	3.0	Scales substitution rates inversely to node heights of the tree
☑	host.Substitution rate and heig...	upDown	0.75	3.0	Scales substitution rates inversely to node heights of the tree
☑	state.Substitution rate and heig...	upDown	0.75	3.0	Scales substitution rates inversely to node heights of the tree
☑	host.rates	scaleIndependently	0.75	15.0	host.rates
☑	host.indicators	bitFlip	n/a	7.0	host.indicators
☑	state.rates	scaleIndependently	0.75	15.0	state.rates
☑	state.indicators	bitFlip	n/a	7.0	state.indicators
☑	age(TX5275_2002.5)	randomWalkAbsorbing	1.0	1.0	Random walk for the age of this tip

Data: 372 taxa, 3 partitions

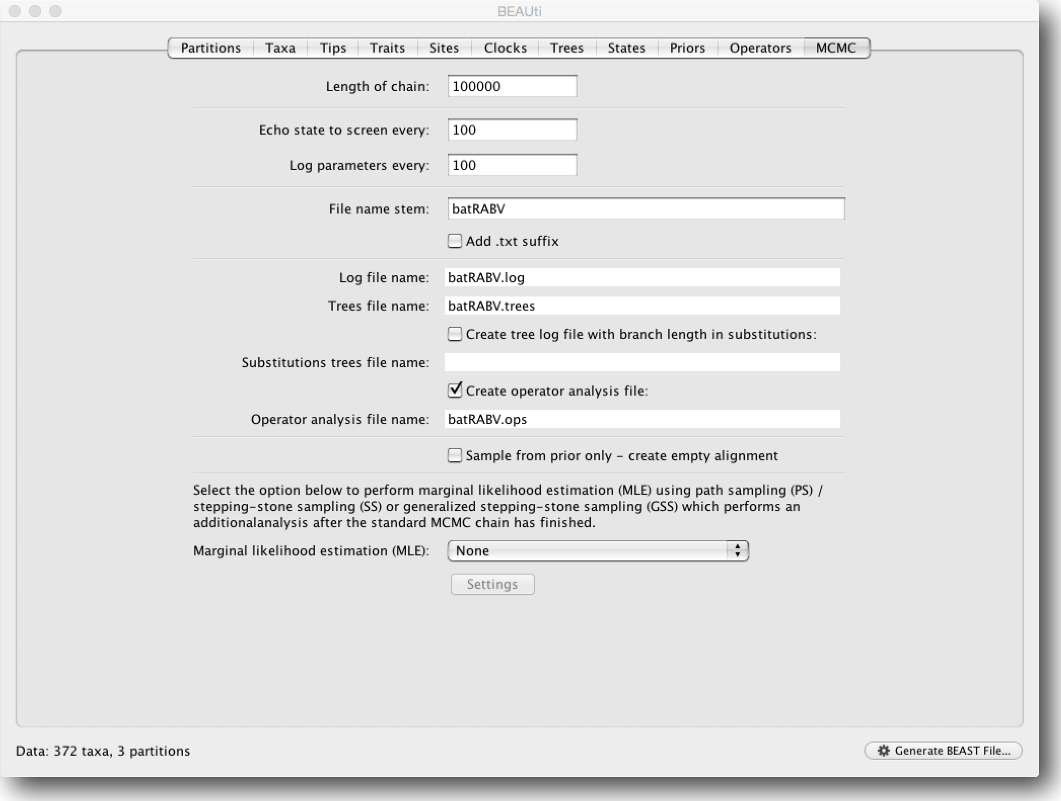
The next column, labelled **Weight**, specifies how often each operator is applied relative to the others. Some parameters tend to be sampled very efficiently - an example is the kappa parameter - these parameters can have their operators down-weighted so that they are not changed as often.

We can keep the default operator settings for the current analysis.

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 >= chain length / 10,000. For this dataset let's initially set the chain length to 100,000 as this will run 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.



Length of chain: 100000

Echo state to screen every: 100

Log parameters every: 100

File name stem: batRABV

□ Add .txt suffix

Log file name: batRABV.log

Trees file name: batRABV.trees

□ Create tree log file with branch length in substitutions:

Substitutions trees file name:

☑ Create operator analysis file:

Operator analysis file name: batRABV.ops

□ Sample from prior only - create empty alignment

Select the option below to perform marginal likelihood estimation (MLE) using path sampling (PS) / stepping-stone sampling (SS) or generalized stepping-stone sampling (GSS) which performs an additionalanalysis after the standard MCMC chain has finished.

Marginal likelihood estimation (MLE): None

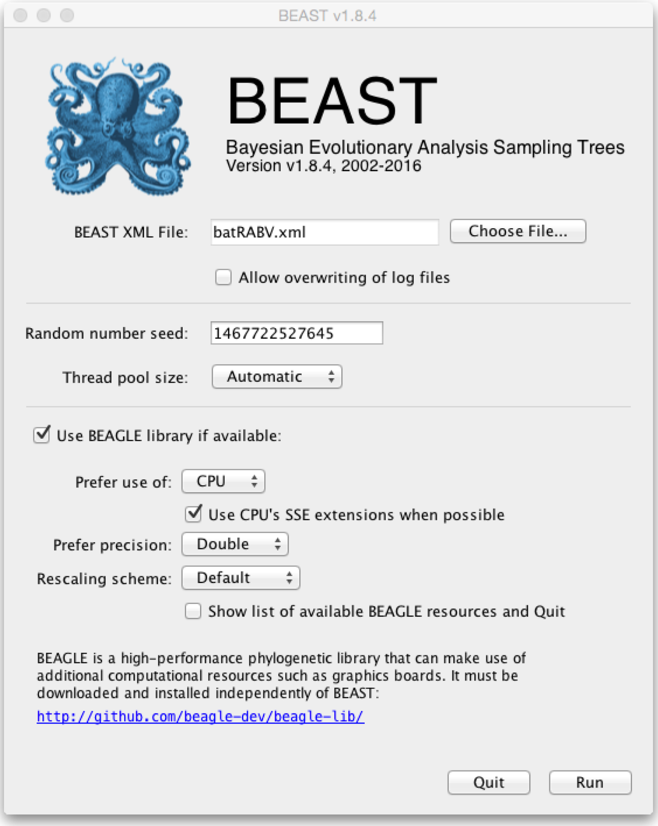
Data: 372 taxa, 3 partitions

The next option allows the user to set the File stem name, which is set to **batRABV**. 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. By default, an operator analysis file is also created. Finally, 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 the data. Here, we will not select this option, but analyze the actual data.

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. Continue and choose a name for the file (for example, **batRABV.xml** by adding the xml extension to the file name stem) and save the file. For convenience, you can leave the *BEAUti* window open so that you can change the values and re-generate the *BEAST* file if necessary.

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. If not selected by default, select the option **Use BEAGLE library if available (see** http://beagle-lib.googlecode.com/ **to install BEAGLE library),** which is required to run the discrete diffusion models, and use the default settings. 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.



BEAST v1.8.4

BEAST

Bayesian Evolutionary Analysis Sampling Trees
Version v1.8.4, 2002-2016

BEAST XML File: batRABV.xml

☐ Allow overwriting of log files

Random number seed: 1467722527645

Thread pool size: Automatic

☑ Use BEAGLE library if available:

Prefer use of: CPU

☑ Use CPU's SSE extensions when possible

Prefer precision: Double

Rescaling scheme: Default

☐ Show list of available BEAGLE resources and Quit

BEAGLE is a high-performance phylogenetic library that can make use of additional computational resources such as graphics boards. It must be downloaded and installed independently of BEAST:
http://github.com/beagle-dev/beagle-lib/

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.

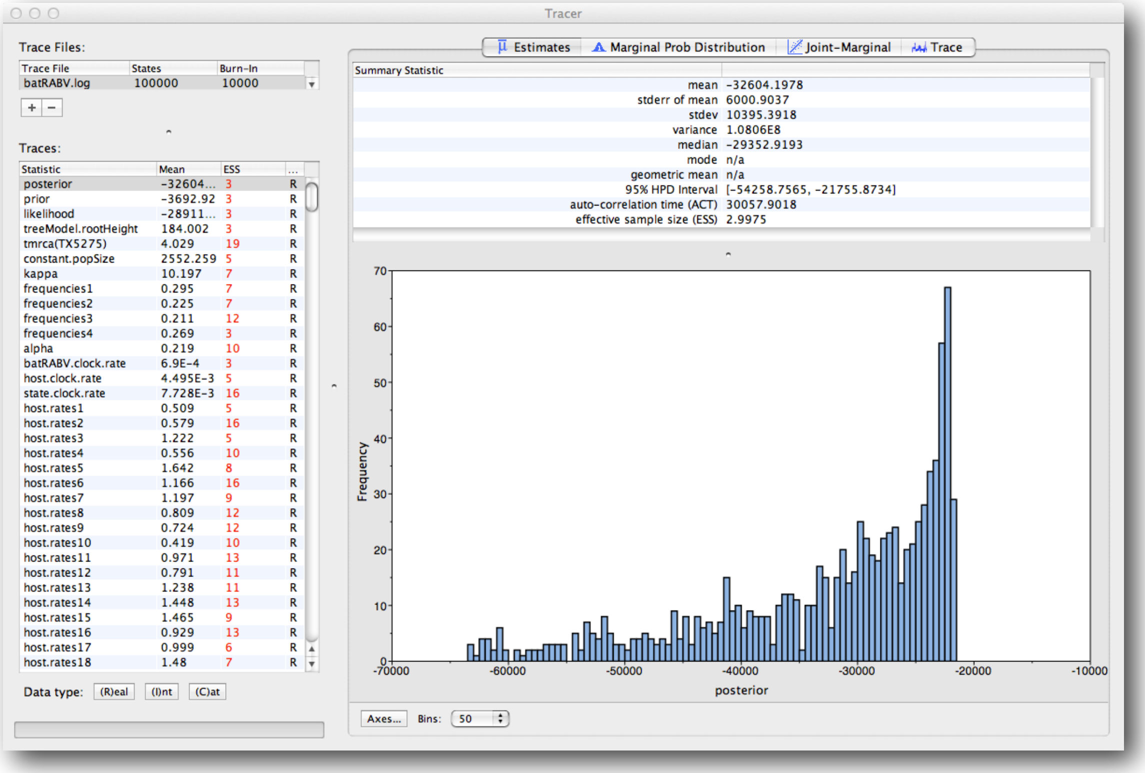
Select the Import Trace File... option from the File menu. If you have it available, select the log file that you created in the previous section (batRABV.log). The file will load and you will be presented with a window similar to the one below. Remember that MCMC is a stochastic algorithm so the actual 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.



Trace Files:

Trace File	States	Burn-In
batRABV.log	100000	10000

Traces:

Statistic	Mean	ESS	...
posterior	-32604...	3	R
prior	-3692.92	3	R
likelihood	-28911...	3	R
treeModel.rootHeight	184.002	3	R
tmrca(TX5275)	4.029	19	R
constant.popSize	2552.259	5	R
kappa	10.197	7	R
frequencies1	0.295	7	R
frequencies2	0.225	7	R
frequencies3	0.211	12	R
frequencies4	0.269	3	R
alpha	0.219	10	R
batRABV.clock.rate	6.9E-4	3	R
host.clock.rate	4.495E-3	5	R
state.clock.rate	7.728E-3	16	R
host.rates1	0.509	5	R
host.rates2	0.579	16	R
host.rates3	1.222	5	R
host.rates4	0.556	10	R
host.rates5	1.642	8	R
host.rates6	1.166	16	R
host.rates7	1.197	9	R
host.rates8	0.809	12	R
host.rates9	0.724	12	R
host.rates10	0.419	10	R
host.rates11	0.971	13	R
host.rates12	0.791	11	R
host.rates13	1.238	11	R
host.rates14	1.448	13	R
host.rates15	1.465	9	R
host.rates16	0.929	13	R
host.rates17	0.999	6	R
host.rates18	1.48	7	R

Summary Statistic	
mean	-32604.1978
stderr of mean	6000.9037
stdev	10395.3918
variance	1.0806E8
median	-29352.9193
mode	n/a
geometric mean	n/a
95% HPD Interval	[-54258.7565, -21755.8734]
auto-correlation time (ACT)	30057.9018
effective sample size (ESS)	2.9975

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.

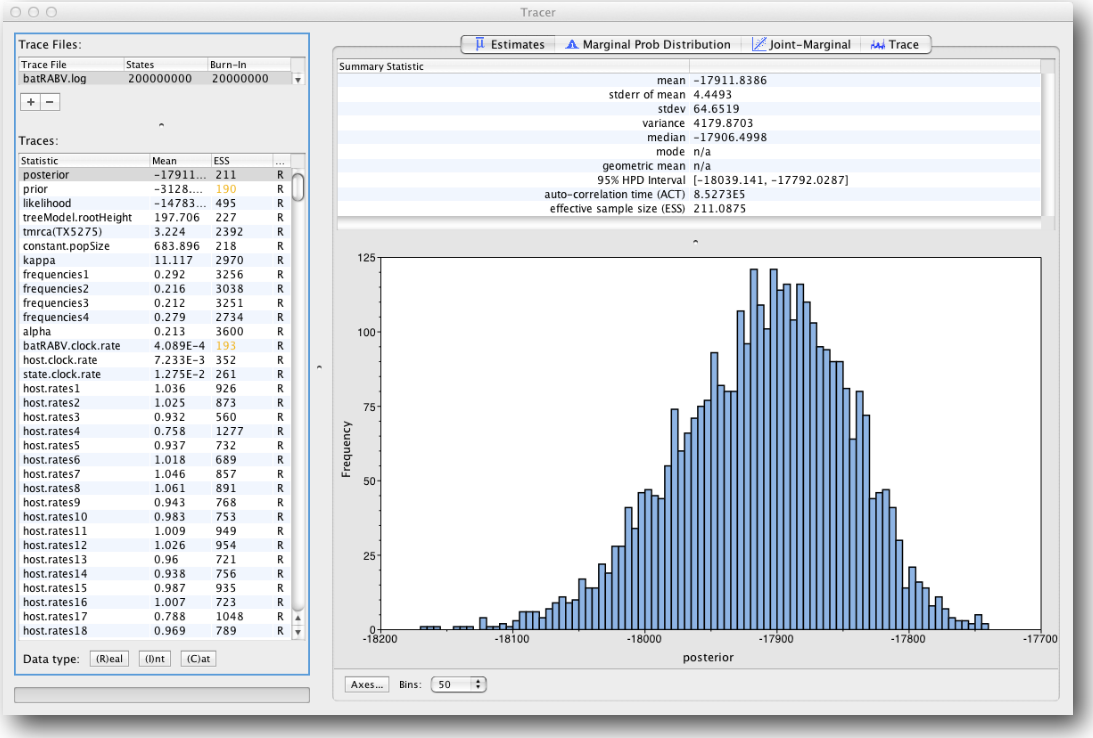
95% HPD Upper - The upper bound of the highest posterior density (HPD) interval.

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. Inspecting the **Trace** of many continuous parameters shows that the chain is still in the burn-in phase (the posterior values are still increasing over the entire chain), and this run does not allow us to summarize marginal posterior probability distributions for the parameters. The simple response to this situation is that we need to run the chain for longer. The example below was run for 200 million steps, sampling every 50,000th step, which means that 4,000 samples where stored in the log file. In this case, the MCMC run has reached stationarity, and almost all parameter traces still show satisfactory ESSs.

We can continue to summarize the annotated phylogeographic tree inferred with the BSSVS procedure and estimate the most significant rates of diffusion. If you are only interested in summarizing the Bayes Factor rates from the BSSVS analysis and not in summarizing the tree from your run, jump to the last section of this tutorial entitled **Visualizing tree and calculating Bayes factor support for rates using SpreadD3**. If you are also interested in summarizing the tree, continue to next section.



Trace Files:

Trace File	States	Burn-In
batRABV.log	200000000	20000000

Traces:

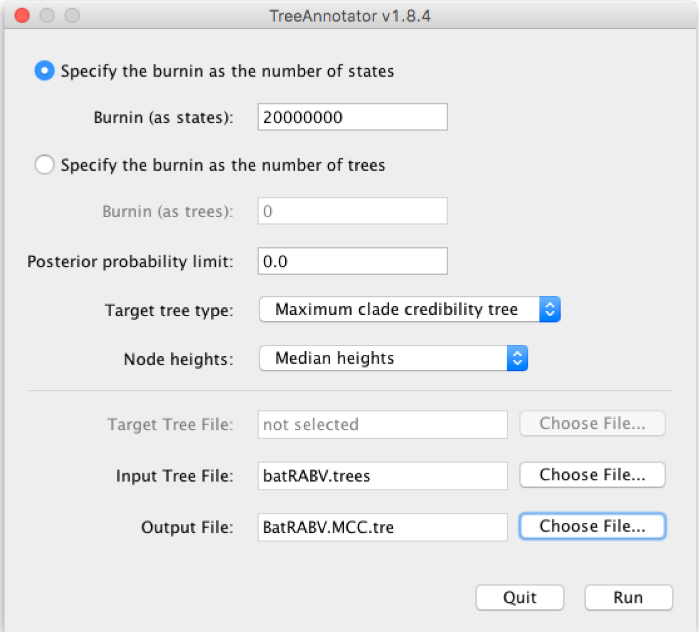
Statistic	Mean	ESS	...
posterior	-17911...	211	R
prior	-3128....	190	R
likelihood	-14783...	495	R
treeModel.rootHeight	197.706	227	R
tmrca(TX5275)	3.224	2392	R
constant.popSize	683.896	218	R
kappa	11.117	2970	R
frequencies1	0.292	3256	R
frequencies2	0.216	3038	R
frequencies3	0.212	3251	R
frequencies4	0.279	2734	R
alpha	0.213	3600	R
batRABV.clock.rate	4.089E-4	193	R
host.clock.rate	7.233E-3	352	R
state.clock.rate	1.275E-2	261	R
host.rates1	1.036	926	R
host.rates2	1.025	873	R
host.rates3	0.932	560	R
host.rates4	0.758	1277	R
host.rates5	0.937	732	R
host.rates6	1.018	689	R
host.rates7	1.046	857	R
host.rates8	1.061	891	R
host.rates9	0.943	768	R
host.rates10	0.983	753	R
host.rates11	1.009	949	R
host.rates12	1.026	954	R
host.rates13	0.96	721	R
host.rates14	0.938	756	R
host.rates15	0.987	935	R
host.rates16	1.007	723	R
host.rates17	0.788	1048	R
host.rates18	0.969	789	R

Summary Statistic	
mean	-17911.8386
stderr of mean	4.4493
stdev	64.6519
variance	4179.8703
median	-17906.4998
mode	n/a
geometric mean	n/a
95% HPD Interval	[-18039.141, -17792.0287]
auto-correlation time (ACT)	8.5273E5
effective sample size (ESS)	211.0875

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 MrBayes or 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.



● Specify the burnin as the number of states

Burnin (as states): 20000000

○ Specify the burnin as the number of trees

Burnin (as trees): 0

Posterior probability limit: 0.0

Target tree type: Maximum clade credibility tree

Node heights: Median heights

Target Tree File: not selected

Input Tree File: batRABV.trees

Output File: BatRABV.MCC.tre

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 the number of steps in the MCMC chain, Burnin (as states), or the number of trees, Burnin (as trees), that should be excluded from the summarization. For the example above, with a chain of 200,000,000 steps, a 10% burnin corresponds to 20,000,000 steps. Alternatively, sampling every 50,000 steps results in 4000 trees in the file, and to obtain at the same 10% burnin, the number of trees needs to be set to 400.

- Posterior probability limit - This is the minimum posterior probability for a node in order for TreeAnnotator to store the annotated information. The default is 0.0 so every node, no matter what its support, will have information summarized. Make sure this value remains 0.0 as every node will require location annotation for further visualization.

- 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 sum 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. Keep the default median node heights for the time being.

- 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., batRABV.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.

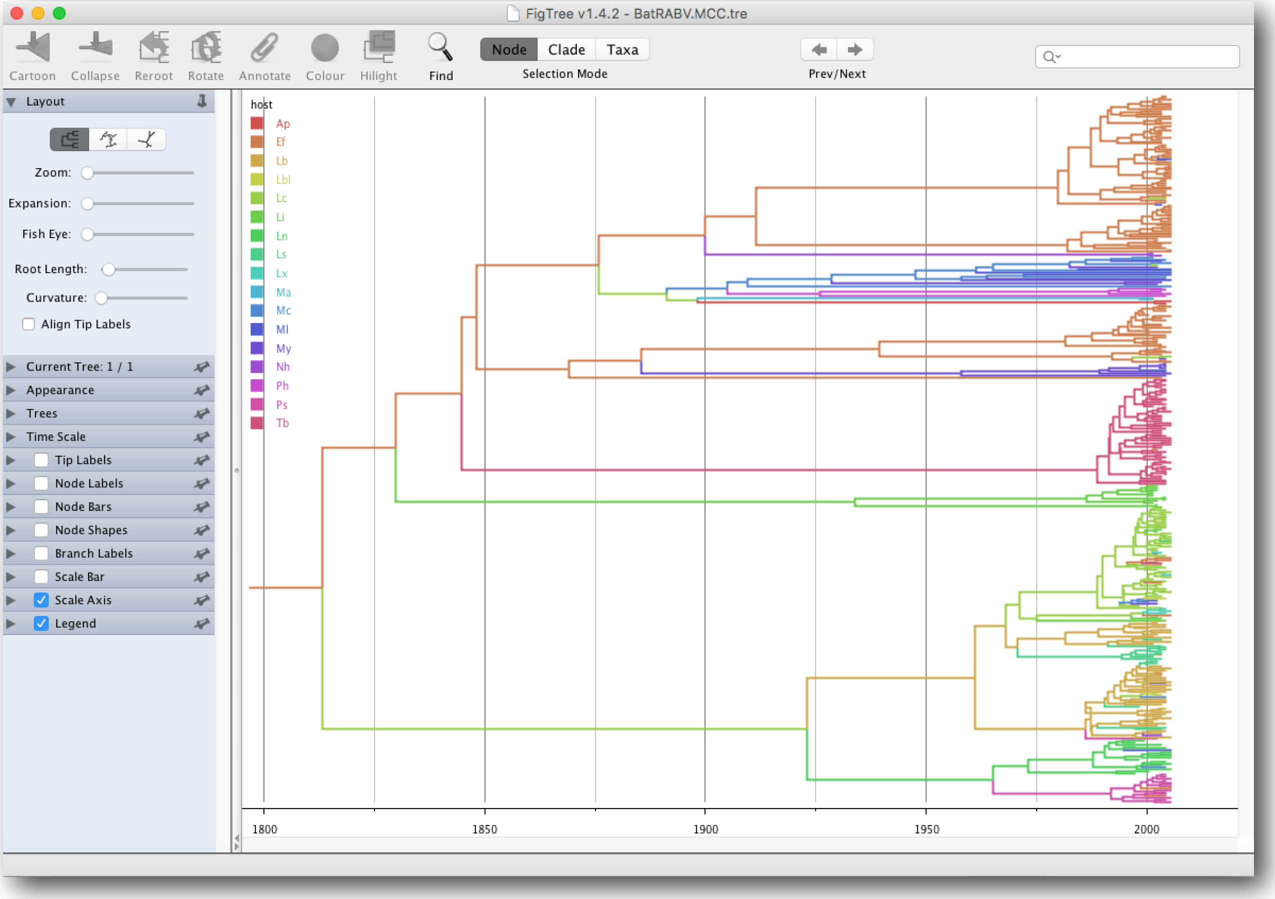
Viewing the annotated tree

Run FigTree 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. Then, open the **Branch Labels** section of the control panel on the left. Now select **posterior** from the **Display popup** menu. The posterior probabilities won't actually be displayed until you tick the check-box next to the **Branch Labels** title.

We now want to display bars on the tree to represent the estimated uncertainty in the date for each node. TreeAnnotator will have placed this information in the tree file in the shape of the 95% highest posterior density (HPD) intervals (see the description of HPDs, above). Open the Node Bars section of the control panel and you will notice that it is already set to display the 95% HPDs of the node heights so all you need to do is to select the check-box in order to turn the node bars on. 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 2005.5 (date of the most recent sample), the scale factor to -1.0. and 'Reverse Axis' under 'Scale Axis'.

Open the Appearance panel and alter the Line Weight to 2 in oder to draw the tree with thicker lines. Under the same panel, alter Colour by and select state. Alternatively, color the branches by host. Unselect the Tip Labels and the Scale Bar option. Finally, in the Legend panel select the state or host attribute depending on what you have used to color the branches with. None of the options actually alter the tree's topology or branch lengths in anyway so feel free to explore the options and settings. You can also save the tree and this will save all your settings so that when you load it into FigTree again it will be displayed exactly as you selected.



Layout
Zoom:
Expansion:
Fish Eye:
Root Length:
Curvature:
☐ Align Tip Labels
Current Tree: 1 / 1
Appearance
Trees
Time Scale
☐ Tip Labels
☐ Node Labels
☐ Node Bars
☐ Node Shapes
☐ Branch Labels
☐ Scale Bar
☑ Scale Axis
☑ Legend

Visualizing MCC trees and calculating Bayes factor support for rates using SpreaD3

SpreaD3, i.e. Spatial Phylogenetic Reconstruction of EvolutionAry Dynamics using Data-Driven Documents (D3), is a software to visualize the output from Bayesian phylogeographic analysis and constitutes a user-friendly application to analyze and visualize reconstructions resulting from Bayesian inference of sequence and trait evolutionary processes. **SpreaD3** allows to visualise spatial reconstructions on custom maps and generates HTML pages for display in modern-day browsers such as Firefox, Safari and Chrome. Some of the functionalities of **SpreaD3** that relate to the discrete phylogeographic analysis include visualizing location-annotated MCC trees and identification of well-supported rates using a Bayes Factor test. The latter option takes as input the rate matrix file (`batRABV.state.rates.log` for location states and `batRABV.host.rates.log` for host states) generated under the analysis using the Bayesian Stochastic Search Variable Selection (**BSSVS**) procedure. This test aims at identifying frequently invoked rates to explain the diffusion process and, in case of locations, visualize them on a circle and on a globe or a map, which needs to be provided to **SpreaD3.** A detailed tutorial for this particular step is available at https://rega.kuleuven.be/cev/ecv/software/SpreaD3_tutorial#sectionFourTwo. We have also provided a PDF version of the entire **SpreaD3** tutorial.

To visualize an MCC tree, start **SpreaD3** by double-clicking on the jar file and select **MCC tree with DISCRETE traits** in the Data panel. Load the MCC tree and set the location attribute to 'state'. Then, use *Setup location attribute coordinates* and load the states and their coordinates in the 'locationStates.txt' file, which should look like this:

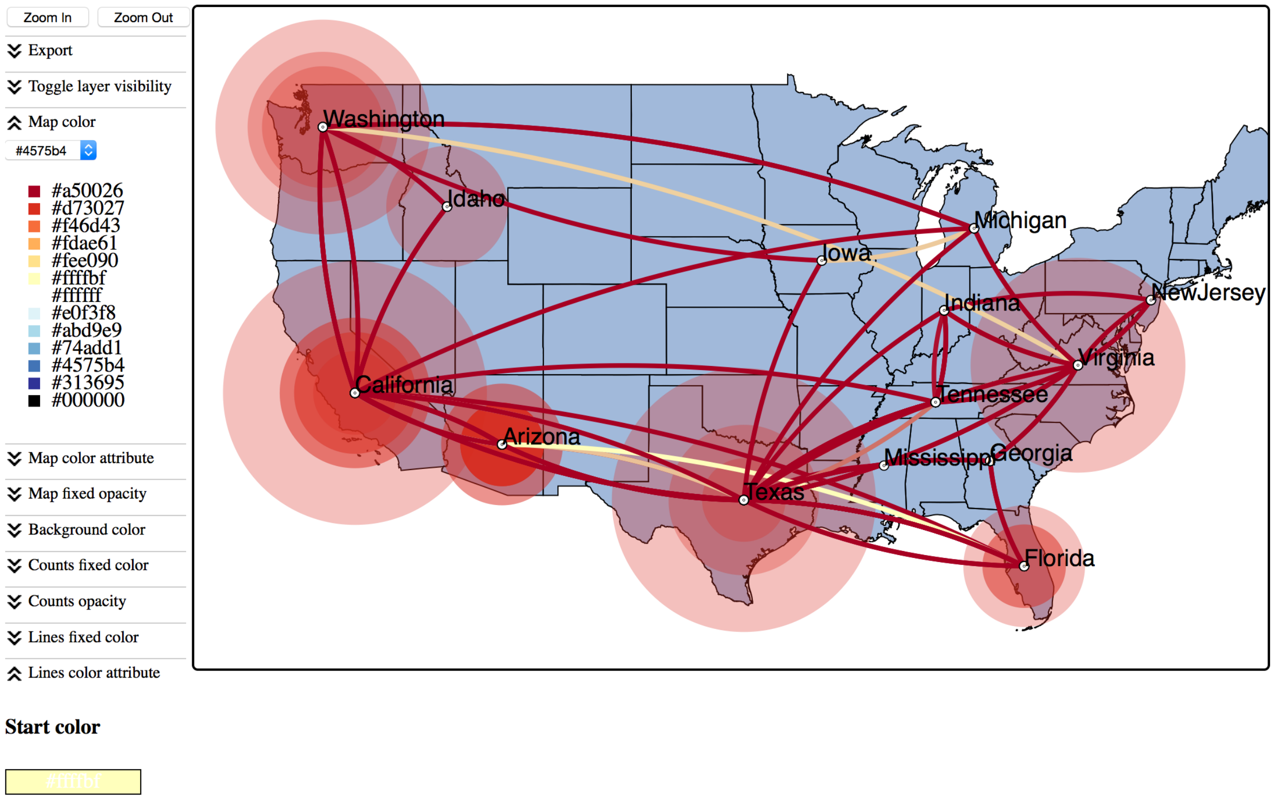
Arizona	33.7712	-111.3877
California	36.17	-119.7462
Georgia	32.9866	-83.6487
Iowa	42.0046	-93.214
Michigan	43.3504	-84.5603
NewJersey	40.314	-74.5089
Virginia	18.0001	-64.8199
Washington	47.3917	-121.5708
Florida	27.8333	-81.717
Tennessee	35.7449	-86.7489
Texas	31.106	-97.6475
Idaho	44.2394	-114.5103
Indiana	39.8647	-86.2604
Mississippi	32.7673	-89.6812

This will load the locations and their lat/long coordinates. Click done after uploading the locations and their coordinates. Set the most recent sampling date to 2005.5 and load a map of the United States in GeoJSON format. Once this is done, go to *Generate Output* and select a file name for the JSON file to be written. Finally, go to the Rendering panel in **SpreaD3** and load the JSON file you just saved. Click *Render to D3* and select a directory name which will contain the HMTL page that will automatically load in a browser (example below). Note that Google Chrome needs to be started with specific privileges for local file access in order to display the resulting visualisation (Firefox and Safari should work fine with default settings).

To summarise Bayes factor support for rates, select **Log file from BSSVS analysis** in the Data panel. Set an appropriate burn-in level and use *Load log file* upload the output BEAST file containing the spatial rates and rate indicators (`batRABV.state.rates.log`). Then, use *Setup location attribute coordinates* to visualise the Bayes Factors on a map of North America. Select *Load* and get the location file (`locationStates.txt`). Click 'Done'. In the same Data panel, you can also specify the Poisson prior mean and offset, which do not need to be changed in our case. If you do, load a map of the United States in GeoJSON format. Once this is done, go to *Generate Output* and select a file name for the JSON file to be written. Note that a plain text file will also be created with an additional .txt extension that will contain the actual Bayes Factor values. Finally, go to the Rendering panel in **SpreaD3** and load the JSON file you just saved. Click *Render to D3* and select a directory name. Examples of these two visualisations can be found below. Note that the visual aspects of the lines representing the rates can be modified and that the lines can also be filtered by a cut-off (under 'Lines cut-off')

We can obtain a similar summary for the host transition rates. Since these cannot be plotted on a map, we will organise them on a circle. Load the file containing the host rates and rate indicators (`batRABV.host.rates.log`). In setting up the locations, select *Generate* and enter the number of unique host states (17 in this case). If you want the names of the locations to be drawn rather than location1, location2, …, enter the names of each of the 17 locations (Ap, Ef, Lb, Lbl, Lc, Li, Ln, Ls, Lx, Ma, Mc, Ml, My, Nh, Ph, Ps, Tb). Click done when all the information has been entered and click on output under *Generate Output* and select a file name for the JSON file to be written. Finally, go to the Rendering panel in **SpreaD3**, load the JSON file you just saved, and click *Render to D3.*

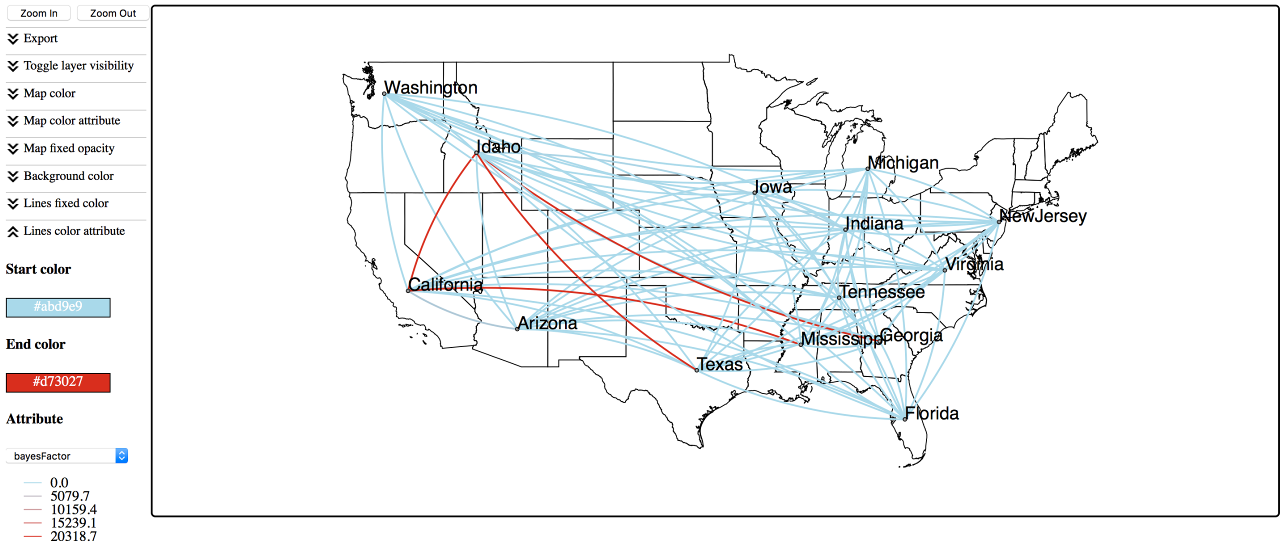
Which rates receive the highest Bayes factor support?



Map color #4575b4

Start color

#ffffbf



Start color

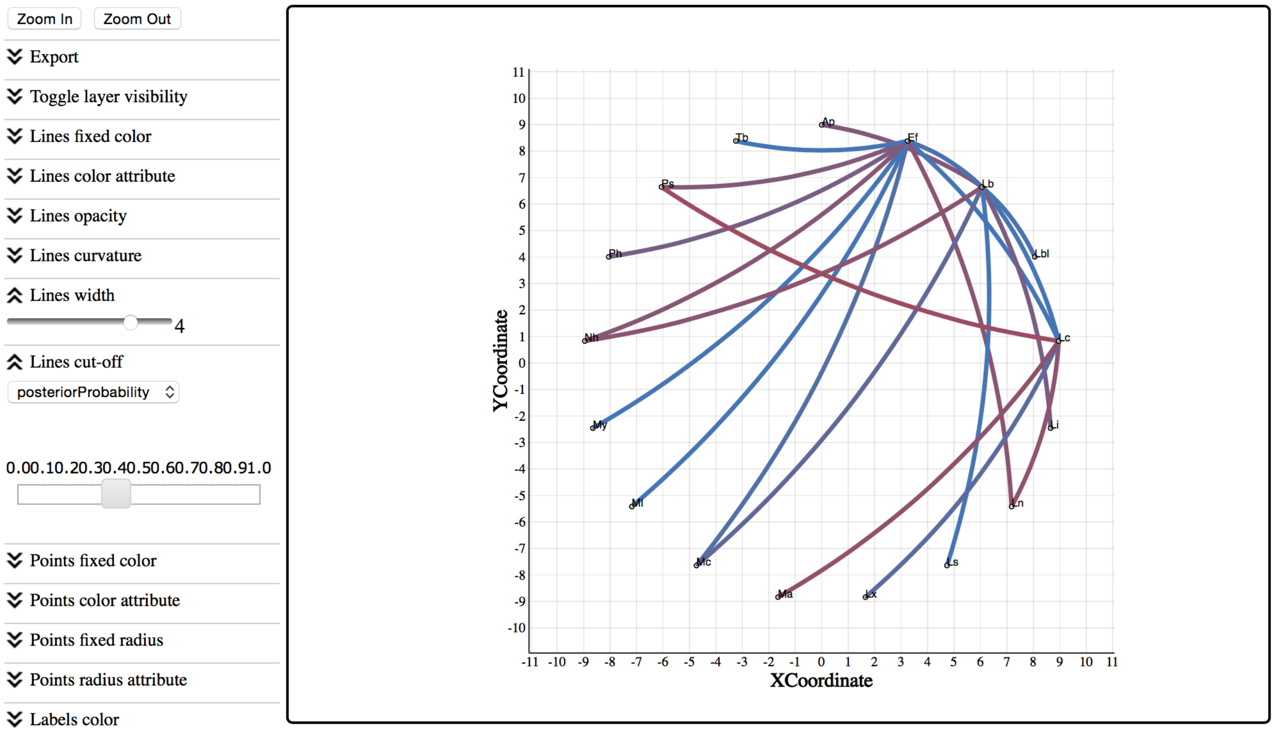
#abd9e9

End color

#d73027

Attribute

bayesFactor

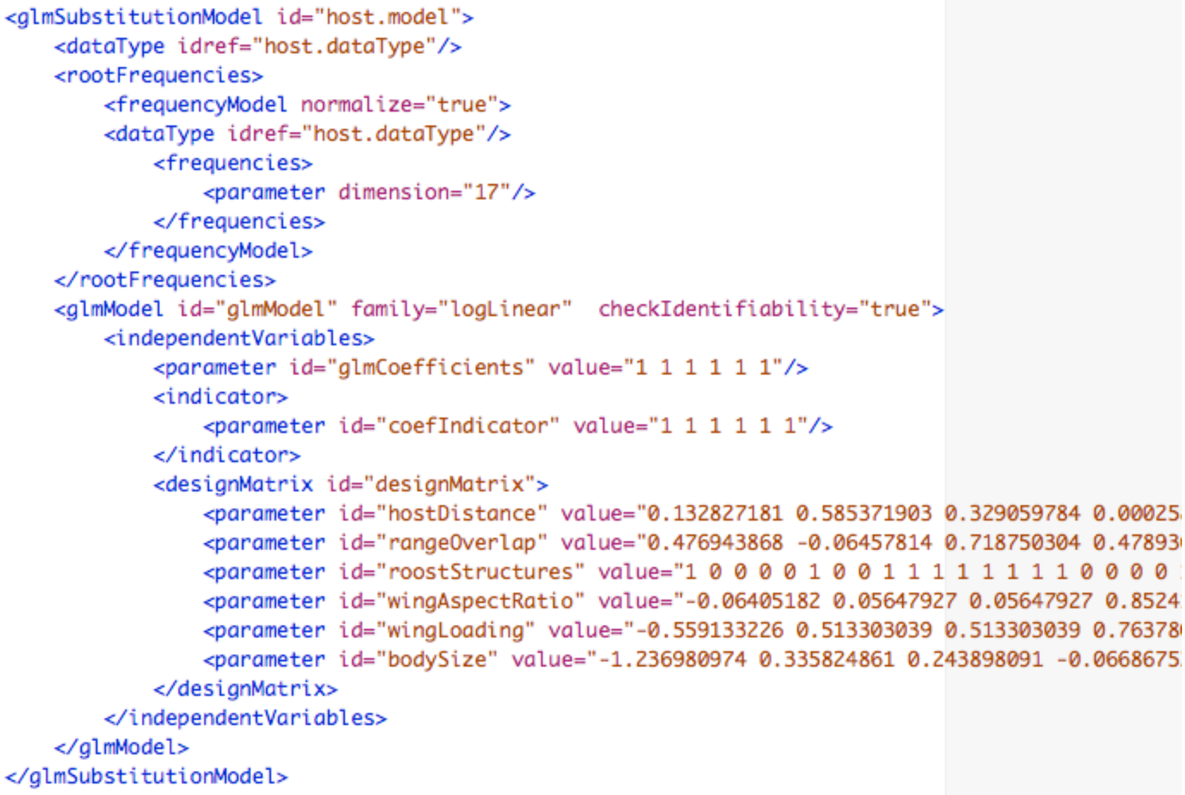

Zoom In
Zoom Out
Export
Toggle layer visibility
Lines fixed color
Lines color attribute
Lines opacity
Lines curvature
Lines width
4
Lines cut-off
posteriorProbability
0.00.10.20.30.40.50.60.70.80.91.0
Points fixed color
Points color attribute
Points fixed radius
Points radius attribute
Labels color
Ap
Tb
Ef
Ps
Lb
Ph
Lbl
Nh
Lc
My
Li
Ml
Ln
Mc
Ls
Ma
Lx
YCoordinate
11 10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10
-11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11
XCoordinate

	Ap	Ef	Lb	Lb	Lc	Li	Ln	Ls	Lx	M	M	Ml	My	Nh	Ph	Ps	Tb
Ap																	
Ef																	
Lb																	
Lb																	
Lc																	
Li																	
Ln																	
Ls																	
Lx																	
M																	
M																	
Ml																	
My																	
Nh																	
Ph																	
Ps																	
Tb																	

Exercise 2: identifying predictors for the host transition process

This exercise builds on the previous analysis and aims at testing the factors that drive the host transition process for bat rabies viruses in North America. The original analyses resorted a population genetic approach and post hoc statistical procedures to test such predictors (Streicker et al., 2010); here we adopt an extension of the discrete diffusion model as applied by Faria et al. (2013). This extension parameterizes the CTMC matrix as a generalized linear model (GLM), in which log CTMC rates are a log linear function of several potential predictors (most of the detail on the model can be found in Lemey et al., 2014). We use the predictors originally proposed by Streicker et al. (2010): host phylogenetic distance (based on host mitochondrial DNA), geographic range overlap, roost site overlap, and foraging niche overlap as approximated using three morphological measurements: wing aspect ratio, wing loading and body length, which are associated with foraging behavior in bats.

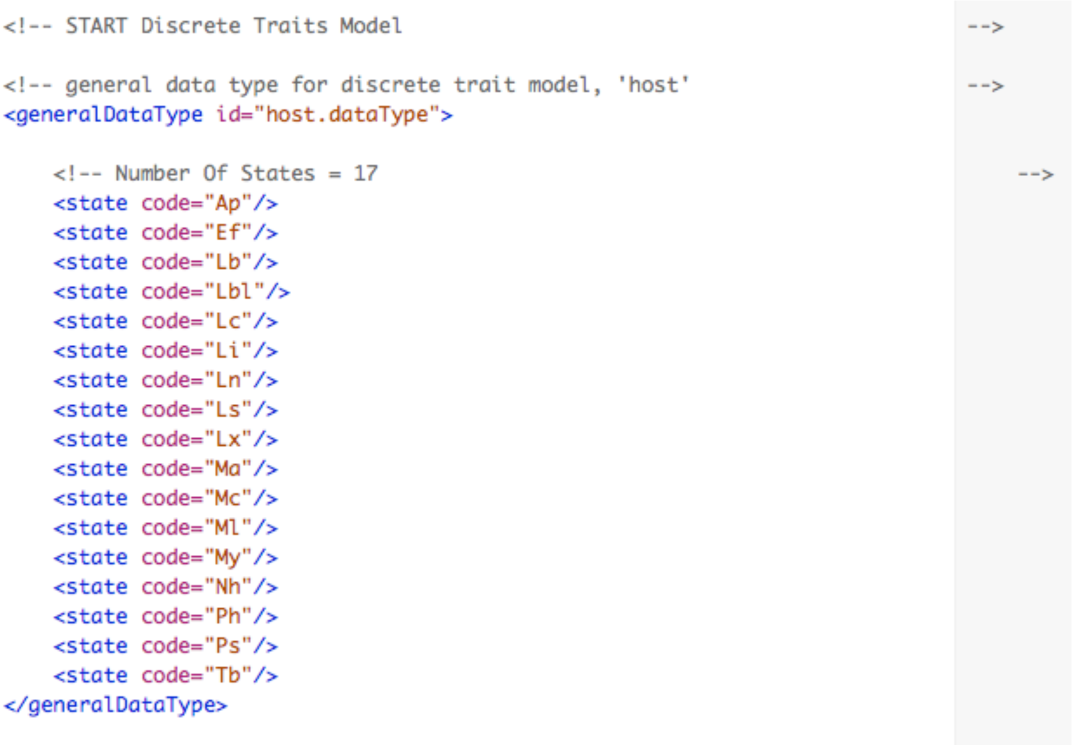
The GLM model allows us to estimate the support for an arbitrary number of predictors and quantify their contribution to the viral diffusion intensity while estimating the bat rabies evolutionary history. So, although they are also considered in the analysis, we will not focus anymore on the sequence evolutionary process or spatial diffusion across states in this part of the tutorial. At present, there is no BEAuti functionality to set up a GLM-discrete diffusion model, so we would have to manually edit our previous XML using a text editor (like TextWrangler). An xml (`batRABV_glm.xml`) is provided in which the model and analysis has been set up (with GLM specific edits indicated with comments) as well as output files resulting from this analysis.



```
<glmSubstitutionModel id="host.model">
    <dataType idref="host.dataType"/>
    <rootFrequencies>
        <frequencyModel normalize="true">
        <dataType idref="host.dataType"/>
            <frequencies>
                <parameter dimension="17"/>
            </frequencies>
        </frequencyModel>
    </rootFrequencies>
    <glmModel id="glmModel" family="logLinear"  checkIdentifiability="true">
        <independentVariables>
            <parameter id="glmCoefficients" value="1 1 1 1 1 1"/>
            <indicator>
                <parameter id="coefIndicator" value="1 1 1 1 1 1"/>
            </indicator>
            <designMatrix id="designMatrix">
                <parameter id="hostDistance" value="0.132827181 0.585371903 0.329059784 0.00025
                <parameter id="rangeOverlap" value="0.476943868 -0.06457814 0.718750304 0.47893
                <parameter id="roostStructures" value="1 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 0 0 0 0
                <parameter id="wingAspectRatio" value="-0.06405182 0.05647927 0.05647927 0.8524
                <parameter id="wingLoading" value="-0.559133226 0.513303039 0.513303039 0.76378
                <parameter id="bodySize" value="-1.236980974 0.335824861 0.243898091 -0.0668675
            </designMatrix>
        </independentVariables>
    </glmModel>
</glmSubstitutionModel>
```

The host state order

Because the GLM model requires us to specify predictors for the discrete diffusion process, we will need to get acquainted with the way the CTMC rates are associated with the discrete (host) states. Note that the order of the discrete states specified in the generalDataType will be important in this respect (which is alphabetical in our case):



```
<!-- START Discrete Traits Model                                              -->

<!-- general data type for discrete trait model, 'host'                       -->
<generalDataType id="host.dataType">

    <!-- Number Of States = 17                                                    -->
    <state code="Ap"/>
    <state code="Ef"/>
    <state code="Lb"/>
    <state code="Lbl"/>
    <state code="Lc"/>
    <state code="Li"/>
    <state code="Ln"/>
    <state code="Ls"/>
    <state code="Lx"/>
    <state code="Ma"/>
    <state code="Mc"/>
    <state code="Ml"/>
    <state code="My"/>
    <state code="Nh"/>
    <state code="Ph"/>
    <state code="Ps"/>
    <state code="Tb"/>
</generalDataType>
```

The CTMC matrix we will need to consider for the predictors follows this order:

The GLM substitution model specification

The GLM model is specified as a `glmSubstitutionModel` element, which replaces the `generalSubstitutionModel` in the XML. As in a `generalSubstitutionModel` model, this specification contains a `frequencyModel`, while the `glmModel` part contains the six predictors (in linearized format in the `designMatrix`) and coefficients and indicators associated with these predictors. The critical part involves the predictor specification which takes the format of a linearized vector of values corresponding to the pair-wise host entries in the CTMC matrix. For each pair of hosts (*i* and *j*), the rate of transition (λ_{ij}) between them is parameterized as:

$$\log \lambda_{ij} = \beta_1\delta_1\log(p_{1\{ij\}}) + \beta_2\delta_2\log(p_{2\{ij\}}) + \ldots + \beta_6\delta_6\log(p_{6\{ij\}}),$$

where the β*'s* represent the coefficients in log space and the δ*'s* represent the indicators that determine the inclusion or exclusion of the predictors from the model. $p_{1\{ij\}}$ is the ij^{th} element of p_1 (= host distance); p_2 = range overlap; ... ; p_6 = body size differences. The predictor specification involves ordering the values according to the CTMC matrix entries, transforming them to log space, and standardizing them (to grant the predictors equal variance *a priori*).

Take for example the host phylogenetic distances (first predictor), which are taken from the study by Streicker et al. (2010) (also provided in the `hostDistances.csv` file) and specified in our matrix format as follows (rounded to three decimals for representation):

	Ap	Ef	Lb	Lbl	Lc	Li	Ln	Ls	Lx	M	M	Ml	My	Nh	Ph	Ps	Tb
Ap		0.745	0.878	0.8	0.71	0.86	0.711	0.916	0.788	0.794	0.717	0.736	0.78	0.74	0.596	0.67	0.923
Ef	0.745		0.971	0.893	0.803	0.953	0.586	1.009	0.881	0.887	0.81	0.829	0.873	0.615	0.689	0.763	1.016
Lb	0.878	0.971		0.316	0.546	0.696	0.937	0.118	0.624	0.944	0.867	0.886	0.93	0.966	0.746	0.82	0.839
Lbl	0.8	0.893	0.316		0.468	0.618	0.859	0.354	0.546	0.866	0.789	0.808	0.852	0.888	0.668	0.742	0.761
Lc	0.71	0.803	0.546	0.468		0.65	0.769	0.584	0.578	0.776	0.699	0.718	0.762	0.798	0.578	0.652	0.671
Li	0.86	0.953	0.696	0.618	0.65		0.919	0.734	0.357	0.926	0.849	0.868	0.912	0.948	0.728	0.802	0.821
Ln	0.711	0.586	0.937	0.859	0.769	0.919		0.975	0.847	0.853	0.776	0.795	0.839	0.531	0.655	0.729	0.982
Ls	0.916	1.009	0.118	0.354	0.584	0.734	0.975		0.662	0.982	0.905	0.924	0.968	1.004	0.784	0.858	0.877
Lx	0.788	0.881	0.624	0.546	0.578	0.357	0.847	0.662		0.854	0.777	0.796	0.84	0.876	0.656	0.73	0.749
M	0.794	0.887	0.944	0.866	0.776	0.926	0.853	0.982	0.854		0.273	0.292	0.196	0.882	0.576	0.65	0.989
M	0.717	0.81	0.867	0.789	0.699	0.849	0.776	0.905	0.777	0.273		0.139	0.259	0.805	0.499	0.573	0.912
Ml	0.736	0.829	0.886	0.808	0.718	0.868	0.795	0.924	0.796	0.292	0.139		0.278	0.824	0.518	0.592	0.931
My	0.78	0.873	0.93	0.852	0.762	0.912	0.839	0.968	0.84	0.196	0.259	0.278		0.868	0.562	0.636	0.975
Nh	0.74	0.615	0.966	0.888	0.798	0.948	0.531	1.004	0.876	0.882	0.805	0.824	0.868		0.684	0.758	1.011
Ph	0.596	0.689	0.746	0.668	0.578	0.728	0.655	0.784	0.656	0.576	0.499	0.518	0.562	0.684		0.434	0.791
Ps	0.67	0.763	0.82	0.742	0.652	0.802	0.729	0.858	0.73	0.65	0.573	0.592	0.636	0.758	0.434		0.865
Tb	0.923	1.016	0.839	0.761	0.671	0.821	0.982	0.877	0.749	0.989	0.912	0.931	0.975	1.011	0.791	0.865	

After log-transformation and standardization, we get the following matrix (again rounded to three decimals for the sake of representation here):

	Ap	Ef	Lb	Lbl	Lc	Li	Ln	Ls	Lx	M	M	Ml	My	Nh	Ph	Ps	Tb
Ap		0.133	0.585	0.329	0.000	0.528	0.004	0.702	0.287	0.308	0.027	0.099	0.259	0.114	-0.482	-0.159	0.723
Ef	0.133		0.863	0.632	0.339	0.811	-0.529	0.969	0.595	0.613	0.363	0.427	0.570	-0.395	-0.082	0.199	0.988
Lb	0.585	0.863		-2.230	-0.723	-0.055	0.765	-4.944	-0.355	0.785	0.551	0.610	0.744	0.849	0.137	0.397	0.460
Lbl	0.329	0.632	-2.230		-1.148	-0.382	0.525	-1.917	-0.723	0.547	0.291	0.356	0.503	0.617	-0.168	0.122	0.191
Lc	0.000	0.339	-0.723	-1.148		-0.243	0.220	-0.538	-0.566	0.245	-0.043	0.031	0.195	0.322	-0.566	-0.235	-0.155
Li	0.528	0.811	-0.055	-0.382	-0.243		0.711	0.092	-1.894	0.732	0.493	0.554	0.690	0.797	0.069	0.336	0.400
Ln	0.004	-0.529	0.765	0.525	0.220	0.711		0.874	0.486	0.506	0.245	0.312	0.460	-0.800	-0.222	0.073	0.894
Ls	0.702	0.969	-4.944	-1.917	-0.538	0.092	0.874		-0.193	0.894	0.669	0.726	0.854	0.955	0.273	0.522	0.582
Lx	0.287	0.595	-0.355	-0.723	-0.566	-1.894	0.486	-0.193		0.509	0.249	0.315	0.463	0.579	-0.218	0.077	0.148
M	0.308	0.613	0.785	0.547	0.245	0.732	0.506	0.894	0.509		-2.633	-2.448	-3.546	0.598	-0.576	-0.243	0.913
M	0.027	0.363	0.551	0.291	-0.043	0.493	0.245	0.669	0.249	-2.633		-4.493	-2.778	0.346	-0.971	-0.590	0.690
Ml	0.099	0.427	0.610	0.356	0.031	0.554	0.312	0.726	0.315	-2.448	-4.493		-2.583	0.410	-0.868	-0.500	0.747
My	0.259	0.570	0.744	0.503	0.195	0.690	0.460	0.854	0.463	-3.546	-2.778	-2.583		0.554	-0.644	-0.303	0.874
Nh	0.114	-0.395	0.849	0.617	0.322	0.797	-0.800	0.955	0.579	0.598	0.346	0.410	0.554		-0.103	0.180	0.974
Ph	-0.482	-0.082	0.137	-0.168	-0.566	0.069	-0.222	0.273	-0.218	-0.576	-0.971	-0.868	-0.644	-0.103		-1.356	0.298
Ps	-0.159	0.199	0.397	0.122	-0.235	0.336	0.073	0.522	0.077	-0.243	-0.590	-0.500	-0.303	0.180	-1.356		0.544
Tb	0.723	0.988	0.460	0.191	-0.155	0.400	0.894	0.582	0.148	0.913	0.690	0.747	0.874	0.974	0.298	0.544	

The order in which these predictor values need to be linearized for BEAST xml vector representation is as follows (the entries now represent the order):

	A	Ef	L	L	L	Li	L	Ls	Lx	M	M	M	M	N	P	P	T
A		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Ef	137		17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
L	138	153		32	33	34	35	36	37	38	39	40	41	42	43	44	45
L	139	154	168		46	47	48	49	50	51	52	53	54	55	56	57	58
L	140	155	169	182		59	60	61	62	63	64	65	66	67	68	69	70
Li	141	156	170	183	195		71	72	73	74	75	76	77	78	79	80	81
L	142	157	171	184	196	207		82	83	84	85	86	87	88	89	90	91
Ls	143	158	172	185	197	208	218		92	93	94	95	96	97	98	99	100
Lx	144	159	173	186	198	209	219	228		101	102	103	104	105	106	107	108
M	145	160	174	187	199	210	220	229	237		109	110	111	112	113	114	115
M	146	161	175	188	200	211	221	230	238	245		116	117	118	119	120	121
M	147	162	176	189	201	212	222	231	239	246	252		122	123	124	125	126
M	148	163	177	190	202	213	223	232	240	247	253	258		127	128	129	130
N	149	164	178	191	203	214	224	233	241	248	254	259	263		131	132	133
P	150	165	179	192	204	215	225	234	242	249	255	260	264	267		134	135
P	151	166	180	193	205	216	226	235	243	250	256	261	265	268	270		136
T	152	167	181	194	206	217	227	236	244	251	257	262	266	269	271	272	

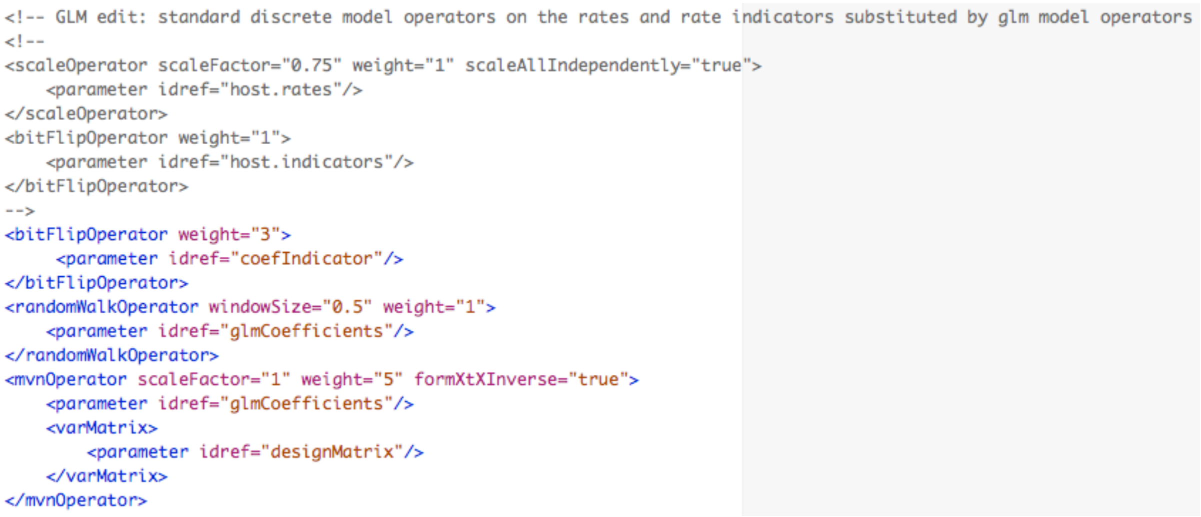
Note that the order in the linearized vector follows the upper matrix *row-by-row* and then the lower matrix *column-by-column*. The values are specified in this order under `value` in each predictor `parameter` in the `designMatrix`. Together with the `hostDistances.csv` file, we provide also an R script (convertToGLM.r) that can take a square matrix, log-transform and standardise the elements, and output the values in the appropriate BEAST xml predictor format.

Additional XML edits

A statistic is added that returns the product of the coefficients and the respective indicators for the predictors (below the `glmSubstitutionModel`). This is not essential for our analysis, but may be useful as a summary.

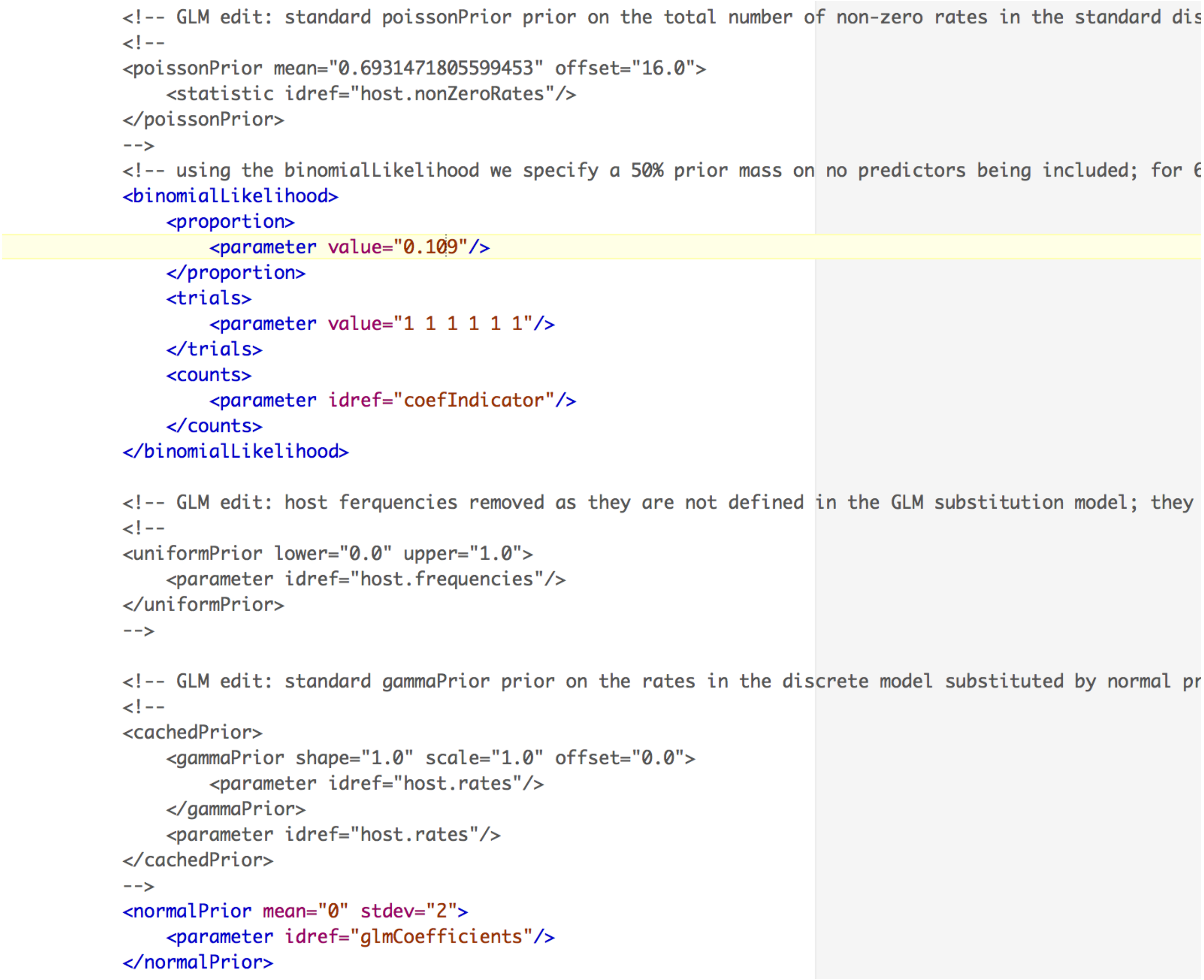
```
<!-- GLM edit: statistic that returns the product of the coefficients and the respective indicators for the predictors -->
<productStatistic id="coefficientsTimesIndicators" elementwise="false">
    <parameter idref="glmCoefficients"/>
    <parameter idref="coefIndicator"/>
</productStatistic>
```

In the operators, the standard discrete model operators on the rates and rate indicators are substituted by the GLM model operators on the predictor indicators (also `bitFlipOperator`) and predictor coefficients (both a `randomWalkOperator` and `mvnOperator`).



```
<!-- GLM edit: standard discrete model operators on the rates and rate indicators substituted by glm model operators
<!--
<scaleOperator scaleFactor="0.75" weight="1" scaleAllIndependently="true">
    <parameter idref="host.rates"/>
</scaleOperator>
<bitFlipOperator weight="1">
    <parameter idref="host.indicators"/>
</bitFlipOperator>
-->
<bitFlipOperator weight="3">
     <parameter idref="coefIndicator"/>
</bitFlipOperator>
<randomWalkOperator windowSize="0.5" weight="1">
    <parameter idref="glmCoefficients"/>
</randomWalkOperator>
<mvnOperator scaleFactor="1" weight="5" formXtXInverse="true">
    <parameter idref="glmCoefficients"/>
    <varMatrix>
        <parameter idref="designMatrix"/>
    </varMatrix>
</mvnOperator>
```

In the priors, three modifications have been made: (i) the Poisson prior on the total number of non-zero rates in the standard discrete model with BSSVS is substituted by binomial prior on the predictor indicators, (ii) the uniform prior on the host frequencies is commented out as no **id** is specified for those in the GLM substitution model and they are not estimated but fixed to equal frequencies, (iii) the standard gamma prior on the rates in the discrete model is substituted by a normal prior on the GLM coefficients.



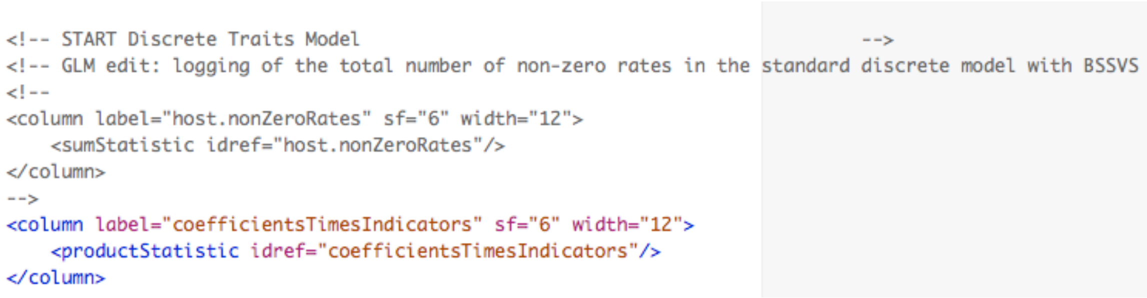
```
<!-- GLM edit: standard poissonPrior prior on the total number of non-zero rates in the standard dis
<!--
<poissonPrior mean="0.6931471805599453" offset="16.0">
    <statistic idref="host.nonZeroRates"/>
</poissonPrior>
-->
<!-- using the binomialLikelihood we specify a 50% prior mass on no predictors being included; for 6
<binomialLikelihood>
    <proportion>
        <parameter value="0.109"/>
    </proportion>
    <trials>
        <parameter value="1 1 1 1 1 1"/>
    </trials>
    <counts>
        <parameter idref="coefIndicator"/>
    </counts>
</binomialLikelihood>

<!-- GLM edit: host ferquencies removed as they are not defined in the GLM substitution model; they
<!--
<uniformPrior lower="0.0" upper="1.0">
    <parameter idref="host.frequencies"/>
</uniformPrior>
-->

<!-- GLM edit: standard gammaPrior prior on the rates in the discrete model substituted by normal pr
<!--
<cachedPrior>
    <gammaPrior shape="1.0" scale="1.0" offset="0.0">
        <parameter idref="host.rates"/>
    </gammaPrior>
    <parameter idref="host.rates"/>
</cachedPrior>
-->
<normalPrior mean="0" stdev="2">
    <parameter idref="glmCoefficients"/>
</normalPrior>
```

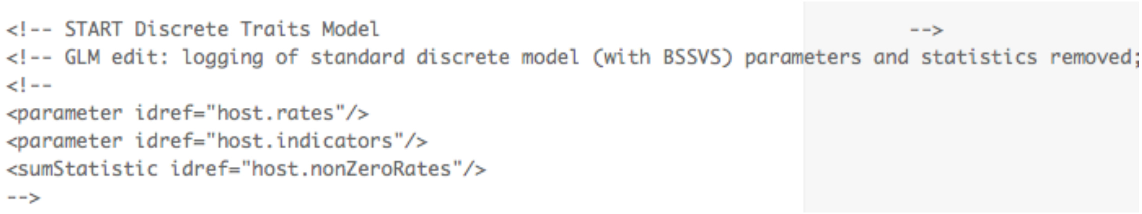
Using the binomialLikelihood we specify a 50% prior mass on no predictors being included. For none of the 6 predictors being included, the binomial distribution probability is 0.50 if the success probability for each predictor inclusion is set to 0.109.

In the screen log, the logging of the total number of non-zero rates in the standard discrete model with BSSVS is substituted by logging of the new product statistic (coefficients times indicators).



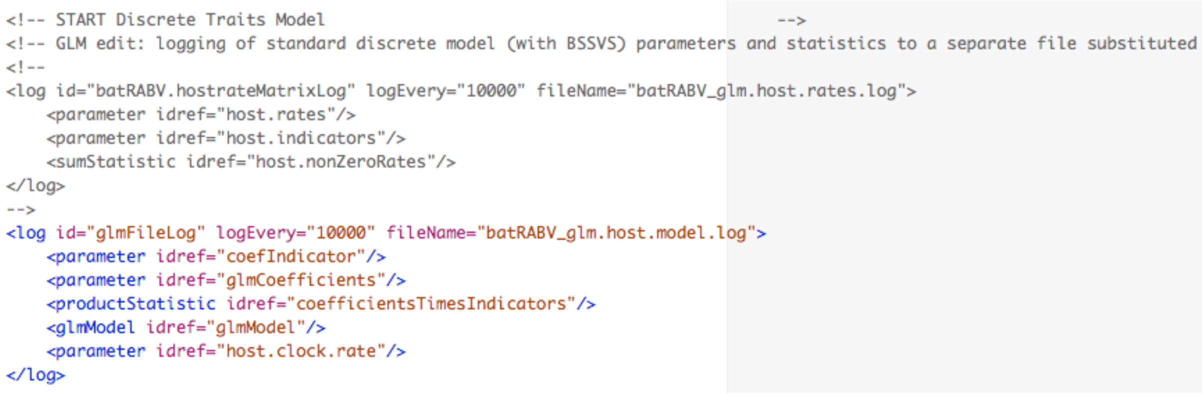
```
<!-- START Discrete Traits Model                                              -->
<!-- GLM edit: logging of the total number of non-zero rates in the standard discrete model with BSSVS
<!--
<column label="host.nonZeroRates" sf="6" width="12">
    <sumStatistic idref="host.nonZeroRates"/>
</column>
-->
<column label="coefficientsTimesIndicators" sf="6" width="12">
    <productStatistic idref="coefficientsTimesIndicators"/>
</column>
```

In the file log, the logging of the logging of the standard discrete model (with BSSVS) parameters and statistic is commented out.



```
<!-- START Discrete Traits Model                                              -->
<!-- GLM edit: logging of standard discrete model (with BSSVS) parameters and statistics removed;
<!--
<parameter idref="host.rates"/>
<parameter idref="host.indicators"/>
<sumStatistic idref="host.nonZeroRates"/>
-->
```

Finally, logging of standard discrete model (with BSSVS) parameters and statistic to a separate file is substituted by logging of the GLM model parameters and statistics to a separate log file (batRABV_glm.host.model.log).



```
<!-- START Discrete Traits Model                                              -->
<!-- GLM edit: logging of standard discrete model (with BSSVS) parameters and statistics to a separate file substituted
<!--
<log id="batRABV.hostrateMatrixLog" logEvery="10000" fileName="batRABV_glm.host.rates.log">
    <parameter idref="host.rates"/>
    <parameter idref="host.indicators"/>
    <sumStatistic idref="host.nonZeroRates"/>
</log>
-->
<log id="glmFileLog" logEvery="10000" fileName="batRABV_glm.host.model.log">
    <parameter idref="coefIndicator"/>
    <parameter idref="glmCoefficients"/>
    <productStatistic idref="coefficientsTimesIndicators"/>
    <glmModel idref="glmModel"/>
    <parameter idref="host.clock.rate"/>
</log>
```

Evaluating the support for the predictors of rabies host transitioning

The XML file (batRABV_glm.xml) can be run using BEASTv1.8.4, but will take a long time to complete. Output files for this analysis are therefore provided. Here, we focus on the GLM-diffusion parameters (in the batRABV_glm.host.model.log file), which are simultaneously estimated the sequence and location diffusion (other output files). Examine the batRABV_glm.host.model.log file in TRACER and summarize the predictor inclusion probabilities, their Bayes factors (based on posterior odds over prior odds for predictor inclusion) and their effect sizes.

Conclusion and Resources

This tutorial 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**/ (or **https://github.com/beast-dev/beast-mcmc**)
- Tutorials: **http://beast.bio.ed.ac.uk/Tutorials/**
- Phylogeography: **http://www.phylogeography.org** (includes **SPREAD** and tutorial)
- Frequently asked questions: **http://beast.bio.ed.ac.uk/FAQ/**

References

Streicker, D. G., A. S. Turmelle, M. J. Vonhof, I. V. Kuzmin, G. F. McCracken, and C. E. Rupprecht. 2010. Host phylogeny constrains cross-species emergence and establishment of rabies virus in bats. Science 329:676-679.

Faria, N. R., M. A. Suchard, A. Rambaut, D. G. Streicker, and P. Lemey. 2013. Simultaneously reconstructing viral cross-species transmission history and identifying the underlying constraints. Philosophical transactions of the Royal Society of London. Series B, Biological sciences 368:20120196.

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

Lemey, P., A. Rambaut, A. J. Drummond, and M. A. Suchard. 2009. Bayesian phylogeography finds its roots. PLoS computational biology 5:e1000520.

Lemey, P., A. Rambaut, T. Bedford, N. Faria, F. Bielejec, G. Baele, C. A. Russell, D. J. Smith, O. G. Pybus, D. Brockmann, and M. A. Suchard. 2014. Unifying Viral Genetics and Human Transportation Data to Predict the Global Transmission Dynamics of Human Influenza H3N2. PLoS pathogens 10:e1003932.

Bloomquist, E. W., P. Lemey, and M. A. Suchard. 2010. Three roads diverged? Routes to phylogeographic inference. Trends Ecol Evol 25:626-632.