Bayesian Inference of Species Trees from Multilocus Data using *BEAST

Alexei J Drummond, Walter Xie and Joseph Heled

April 13, 2012

Introduction

We describe a full Bayesian framework for species tree estimation. We have attempted to combine the best aspects of previous methods to provide joint inference of a species tree topology, divergence times, population sizes, and gene trees from multiple genes sampled from multiple individuals across a set of closely related species. We have achieved this by extending BEAST to *BEAST (pronounced ”star beast”), which is published below:

Joseph Heled and Alexei J. Drummond Bayesian Inference of Species Trees from Multilocus Data Mol. Biol. Evol. 2010 27: 570-580.

You will need the following software at your disposal:

- **BEAST** - this package contains the BEAST program, BEAUti, TreeAnnotator and other utility programs. This tutorial is written for BEAST v1.7.x, which is available for download from `http://beast.bio.ed.ac.uk/`.
- **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 v1.5. It is available for download from `http://beast.bio.ed.ac.uk/`.
- **FigTree** - this is an application for displaying and printing molecular phylogenies, in particular those obtained using BEAST. At the time of writing, the current version is v1.3.1. It is available for download from `http://tree.bio.ed.ac.uk/`.

*BEAST

This tutorial will guide you through the analysis of three loci sampled from 26 individuals representing nine species of pocket gophers. This is a subset of previous published data [1]. The objective of this tutorial is to estimate the species tree that is most probable given the multi-individual multi-locus sequence data. The species tree has 9 taxa, whereas each gene tree has 26 taxa. *BEAST will co-estimate three gene trees embedded in a shared species tree (see Heled and Drummond, 2010 for details).

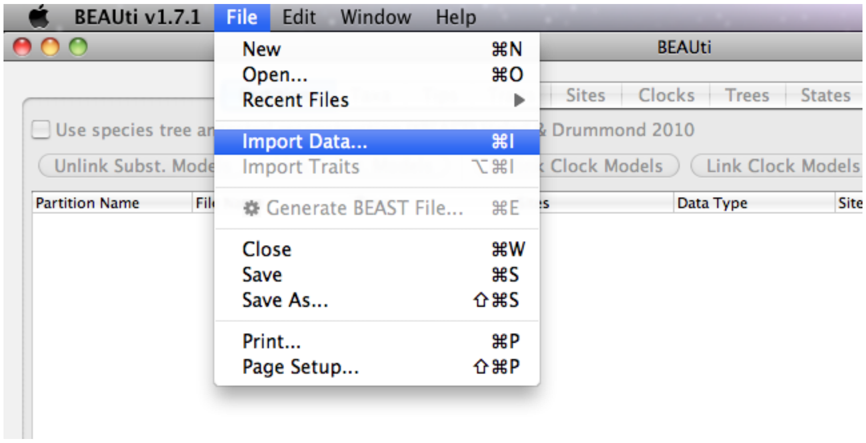
The first step will be to convert a NEXUS file with a DATA or CHARACTERS block into a BEAST XML input file. This is done using the program BEAUti (Bayesian Evolutionary Analysis Utility). This is a user-friendly program for setting the evolutionary model and options for the MCMC analysis. The second 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.

BEAUti

Run BEAUti by double clicking on its icon.

Loading the NEXUS file

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

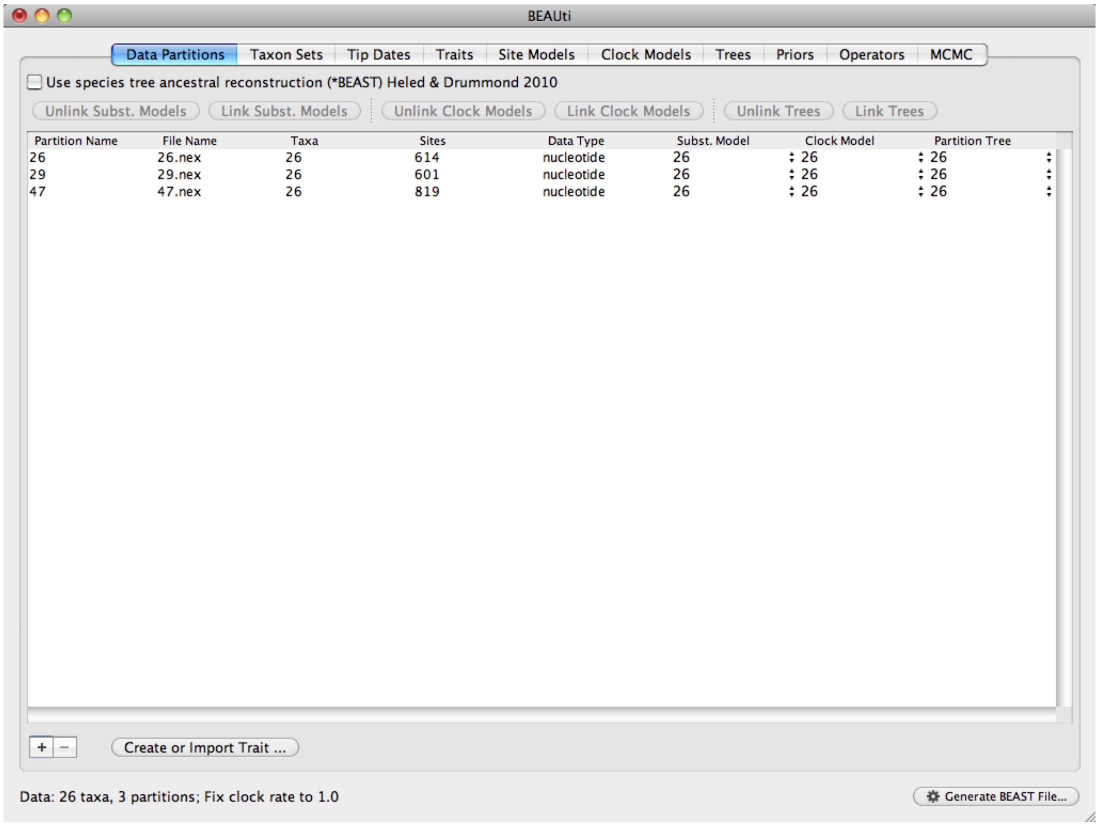


Select three files called `26.nex, 29.nex, 47.nex` by holding `shift` key. Each file contains an alignment of sequences of from an independent locus. The `26.nex` looks like this (content has been truncated):

```
#NEXUS
[TBO26oLong]
BEGIN DATA;
DIMENSIONS  NTAX =26 NCHAR=614;
FORMAT DATATYPE = DNA GAP = - MISSING = ?;
MATRIX
Orthogeomys_heterodus        ATTCTAGGCAAAAAGAGCAATGC ...
```

```
Thomomys_bottae_awahnee_a    ????????????????????ATGCTG ...
Thomomys_bottae_awahnee_b    ????????????????????ATGCTG ...
Thomomys_bottae_xerophilus   ????????????????????ATGCTG ...
Thomomys_bottae_cactophilus  ????????????????AGCAATGCT ...


         ... ...


;
END;
```

Once loaded, the three partitions are displayed in the main panel:

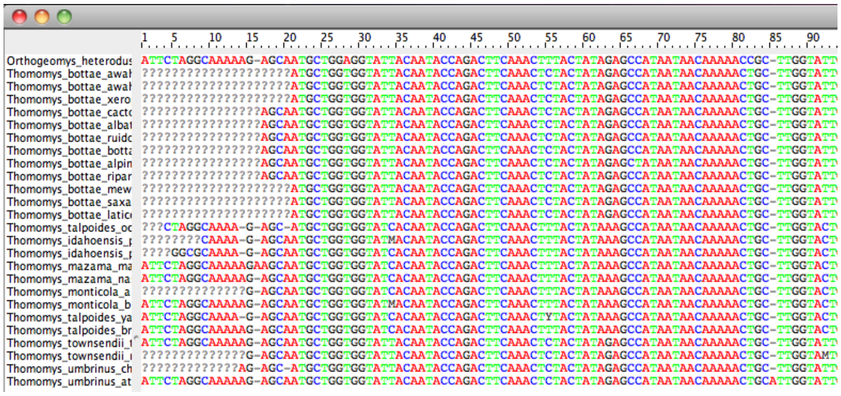


☐ Use species tree ancestral reconstruction (*BEAST) Heled & Drummond 2010

Partition Name	File Name	Taxa	Sites	Data Type	Subst. Model	Clock Model	Partition Tree
26	26.nex	26	614	nucleotide	26	26	26
29	29.nex	26	601	nucleotide	26	26	26
47	47.nex	26	819	nucleotide	26	26	26

Data: 26 taxa, 3 partitions; Fix clock rate to 1.0

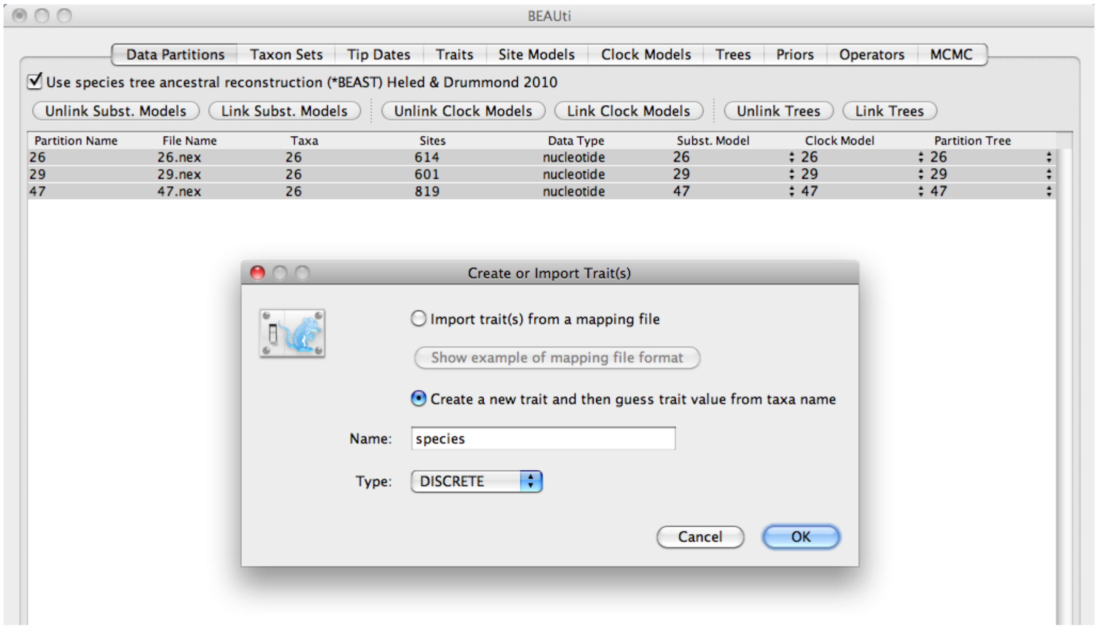
Double click any alignment (partition) to show its detail:



```
                      1   5    10   15   20   25   30   35   40   45   50   55   60   65   70   75   80   85   90
Orthogeomys_heterodus ATTCTAGGCAAAAAG-AGCAATGCTGGAGGTATTACAATACCAGACTTCAAACTTTACTATAGAGCCATAATAACAAAAACCGC-TTGGTATT
Thomomys_bottae_awah  ????????????????????ATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_awah  ????????????????????ATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_xero  ????????????????????ATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_cactc ????????????????AGCAATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_albat ????????????????AGCAATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_ruidc ????????????????AGCAATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_botta ????????????????AGCAATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_alpin ????????????????AGCAATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCTATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_ripar ????????????????AGCAATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_mew   ????????????????????ATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_saxa  ????????????????????ATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_bottae_latic ????????????????????ATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_talpoides_oc ???CTAGGCAAAA-G-AGC-ATGCTGGTGGTATCACAATACCAGACTTCAAACTTTACTATAAAGCCATAATAACAAAAACTGC-TTGGTACT
Thomomys_idahoensis_p ????????CAAAA-G-AGCAATGCTGGTGGTATMACAATACCAGACTTCAAACTTTACTATAAAGCCATAATAACAAAAACTGC-TTGGTACT
Thomomys_idahoensis_p ????GGCGCAAAA-G-AGCAATGCTGGTGGTATCACAATACCAGACTTCAAACTTTACTATAAAGCCATAATAACAAAAACTGC-TTGGTACT
Thomomys_mazama_ma    ATTCTAGGCAAAAAGAAGCAATGCTGGTGGTATCACAATACCAGACTTCAAACTTTACTATAAAGCCATAATAACAAAAACTGC-TTGGTACT
Thomomys_mazama_na    ATTCTAGGCAAAAAG-AGCAATGCTGGTGGTATCACAATACCAGACTTCAAACTTTACTATAAAGCCATAATAACAAAAACTGC-TTGGTACT
Thomomys_monticola_a  ??????????????G-AGCAATGCTGGTGGTATCACAATACCAGACTTCAAACTTTACTATAAAGCCATAATAACAAAAACTGC-TTGGTACT
Thomomys_monticola_b  ATTCTAGGCAAAAAG-AGCAATGCTGGTGGTATMACAATACCAGACTTCAAACTTTACTATAAAGCCATAATAACAAAAACTGC-TTGGTACT
Thomomys_talpoides_ya ATTCTAGGCAAAA-G-AGCAATGCTGGTGGTATCACAATACCAGACTTCAAACTYTACTATAAAGCCATAATAACAAAAACTGC-TTGGTACT
Thomomys_talpoides_br ATTCTAGGCAAAAAG-AGCAATGCTGGTGGTATCACAATACCAGACTTCAAACTTTACTATAAAGCCATAATAACAAAAACTGC-TTGGTACT
Thomomys_townsendii_t ATTCTAGGCAAAAAG-AGCAATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_townsendii_  ??????????????G-AGCAATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTAMT
Thomomys_umbrinus_ch  ?????????????AG-AGC-ATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGC-TTGGTATT
Thomomys_umbrinus_at  ATTCTAGGCAAAAAG-AGCAATGCTGGTGGTATTACAATACCAGACTTCAAACTCTACTATAGAGCCATAATAACAAAAACTGCATTGGTATT
```

Import trait(s) from a mapping file to fire *BEAST

To enable *BEAST in BEAST v1.7.x, simply click the check box labelled `Use species tree ancestral reconstruction (*BEAST) Heled & Drummond 2010` on the top of **Data Partitions** panel. Then, a **Create or Import Trait(s)** dialog will pop up.



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

Partition Name	File Name	Taxa	Sites	Data Type	Subst. Model	Clock Model	Partition Tree
26	26.nex	26	614	nucleotide	26	26	26
29	29.nex	26	601	nucleotide	29	29	29
47	47.nex	26	819	nucleotide	47	47	47

○ Import trait(s) from a mapping file

● Create a new trait and then guess trait value from taxa name

Name: species

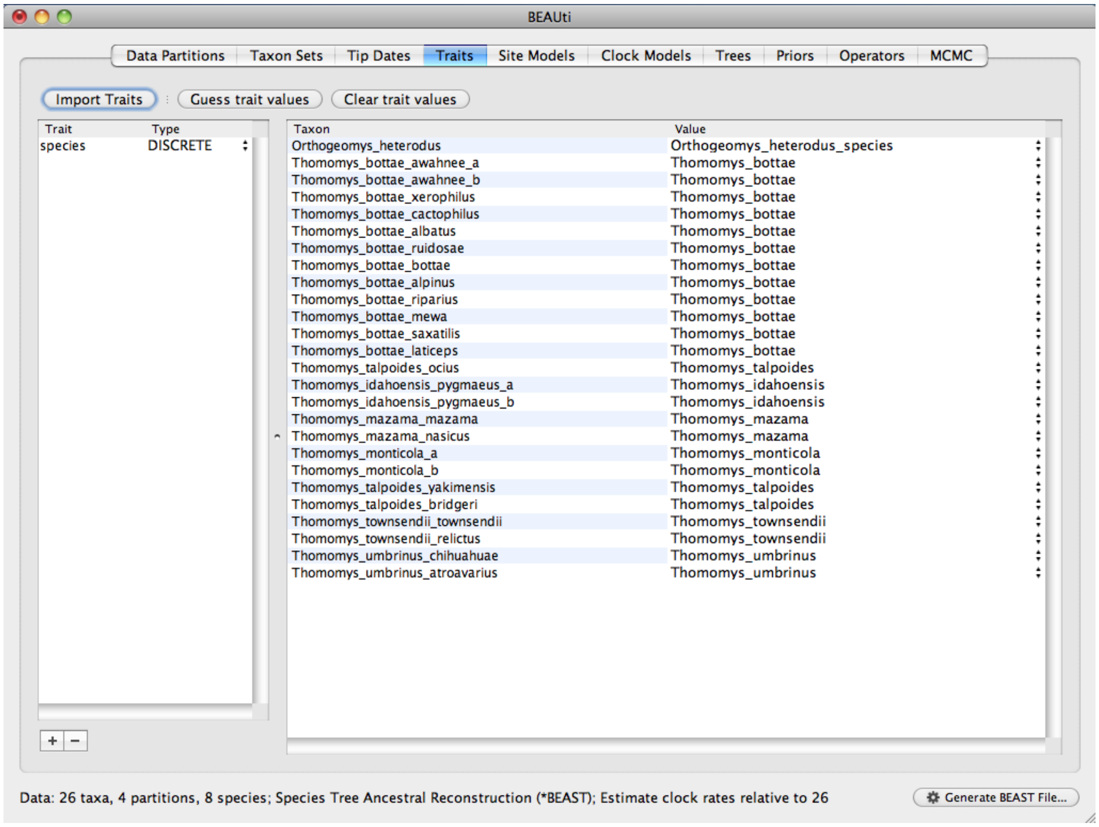
Type: DISCRETE

There are two options to be selected:

1. Import trait(s) from a mapping file;
2. Create a new trait and then guess trait value from taxa name `species`.

Choose the first option and click **OK** to load the mapping file, named `gopher_mapping.txt`.

Once loaded, a message indicating the use of *BEAST will be displayed in the status at the bottom of the window, and a trait named `species` is created in the trait table in the **Traits** tab. Click it to show trait values.



Trait	Type
species	DISCRETE

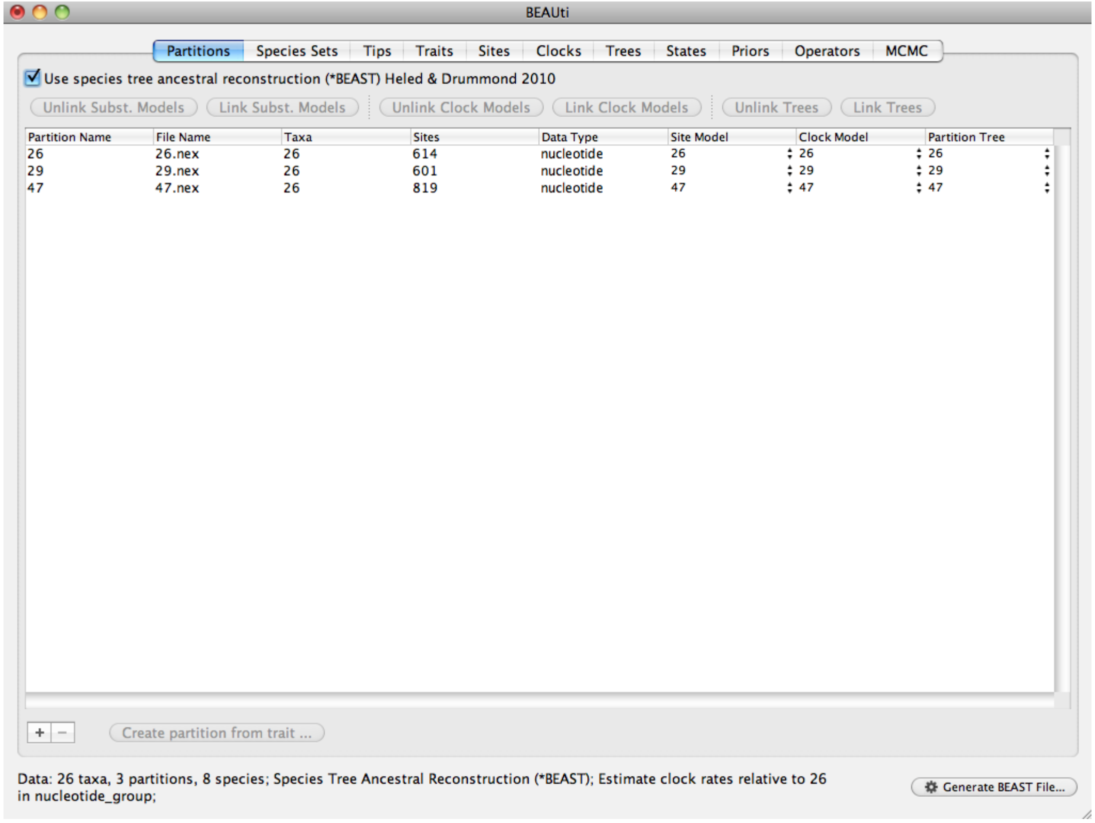
Taxon	Value
Orthogeomys_heterodus	Orthogeomys_heterodus_species
Thomomys_bottae_awahnee_a	Thomomys_bottae
Thomomys_bottae_awahnee_b	Thomomys_bottae
Thomomys_bottae_xerophilus	Thomomys_bottae
Thomomys_bottae_cactophilus	Thomomys_bottae
Thomomys_bottae_albatus	Thomomys_bottae
Thomomys_bottae_ruidosae	Thomomys_bottae
Thomomys_bottae_bottae	Thomomys_bottae
Thomomys_bottae_alpinus	Thomomys_bottae
Thomomys_bottae_riparius	Thomomys_bottae
Thomomys_bottae_mewa	Thomomys_bottae
Thomomys_bottae_saxatilis	Thomomys_bottae
Thomomys_bottae_laticeps	Thomomys_bottae
Thomomys_talpoides_ocius	Thomomys_talpoides
Thomomys_idahoensis_pygmaeus_a	Thomomys_idahoensis
Thomomys_idahoensis_pygmaeus_b	Thomomys_idahoensis
Thomomys_mazama_mazama	Thomomys_mazama
Thomomys_mazama_nasicus	Thomomys_mazama
Thomomys_monticola_a	Thomomys_monticola
Thomomys_monticola_b	Thomomys_monticola
Thomomys_talpoides_yakimensis	Thomomys_talpoides
Thomomys_talpoides_bridgeri	Thomomys_talpoides
Thomomys_townsendii_townsendii	Thomomys_townsendii
Thomomys_townsendii_relictus	Thomomys_townsendii
Thomomys_umbrinus_chihuahuae	Thomomys_umbrinus
Thomomys_umbrinus_atroavarius	Thomomys_umbrinus

Data: 26 taxa, 4 partitions, 8 species; Species Tree Ancestral Reconstruction (*BEAST); Estimate clock rates relative to 26

A proper trait file is tab delimited. The first row is always `traits` followed by the keyword `species` in the second column and separated by tab. The rest of the rows map each individual taxon name to a species name: the taxon name in the first column and species name in the second column separated by tab. For example:

```
traits species
taxon1 speciesA
taxon2 speciesA
taxon3 speciesB
...  ...
```

For multi-locus analyses, BEAST can link or unlink substitutions models across the loci by clicking buttons on the top of **Data Partitions** panel. The default of *BEAST is unlinking all models: substitution model, clock model, and tree models. Note that you should only unlink the tree model across data partitions that are actually genetically unlinked. For example, in most organisms all the mitochondrial genes are effectively linked due to a lack of recombination and they should be set up to use the same tree model in a *BEAST analysis.



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

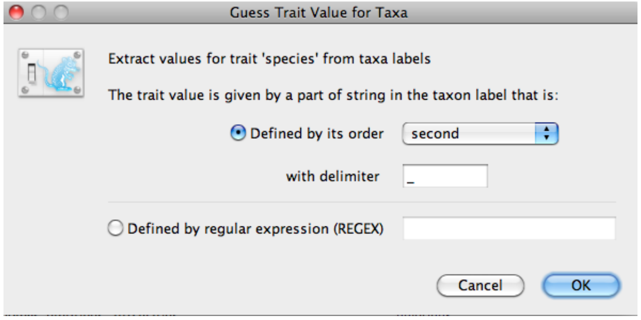
Partition Name	File Name	Taxa	Sites	Data Type	Site Model	Clock Model	Partition Tree
26	26.nex	26	614	nucleotide	26	26	26
29	29.nex	26	601	nucleotide	29	29	29
47	47.nex	26	819	nucleotide	47	47	47

Data: 26 taxa, 3 partitions, 8 species; Species Tree Ancestral Reconstruction (*BEAST); Estimate clock rates relative to 26 in nucleotide_group;

Alternatively: Create a species trait from taxa names

The advantage of using the **Traits** panel is that we can extract species names from the taxa names if they already contain that information. Let's go to **Data Partitions** panel and unselect the check box labelled `Use species tree ancestral reconstruction (*BEAST) Heled & Drummond 2010`. As we can see in the status bar on the bottom, the analysis has been reverted to a standard BEAST analysis.

To enable *BEAST again, click the `Use species tree ancestral reconstruction (*BEAST) Heled & Drummond 2010` on the top of **Data Partitions** panel, and then choose the second option in **Create or Import Trait(s)** dialog this time. Click **OK** to continue, and then we will get to **Traits** panel and click on the **Guess trait values** at the top to pop out **Guess Trait Value for Taxa** dialog. Choose **second** in the drop list of **Defined by its order**, and input _ as separator. Click **OK**, and *BEAST is applied again.



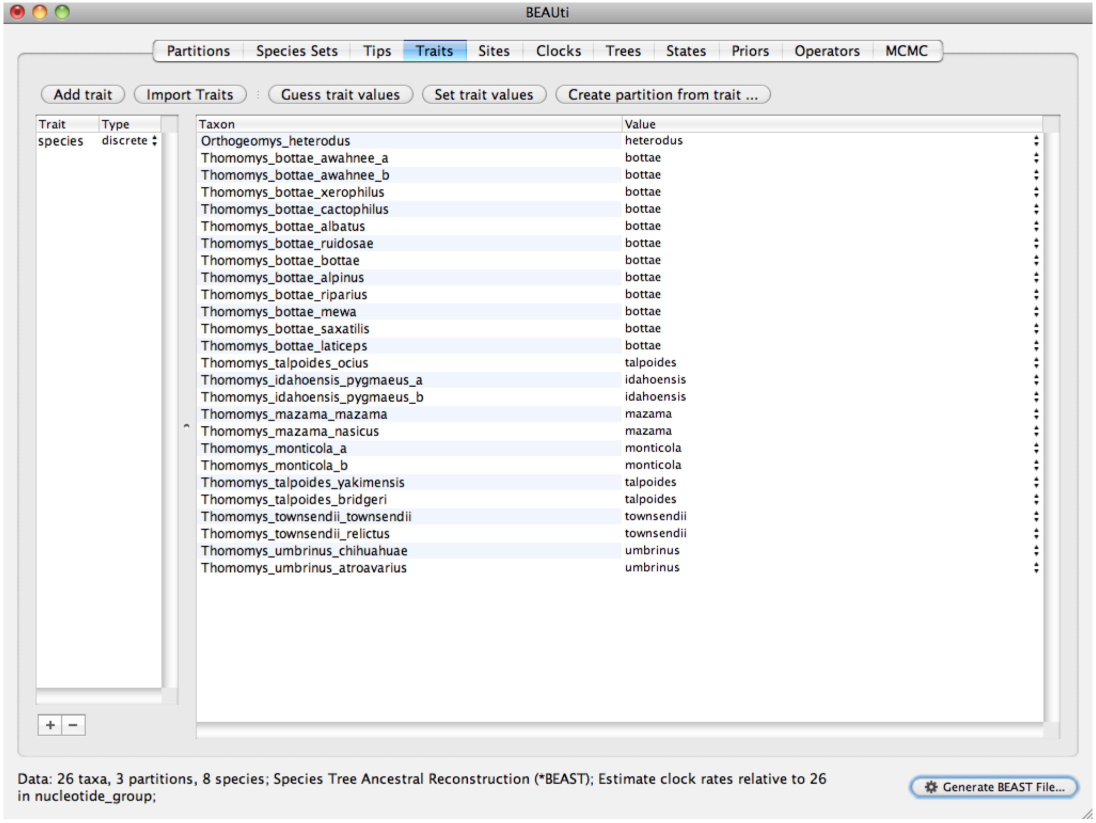
Extract values for trait 'species' from taxa labels

The trait value is given by a part of string in the taxon label that is:

● Defined by its order: second

with delimiter: _

○ Defined by regular expression (REGEX)



Trait	Type
species	discrete

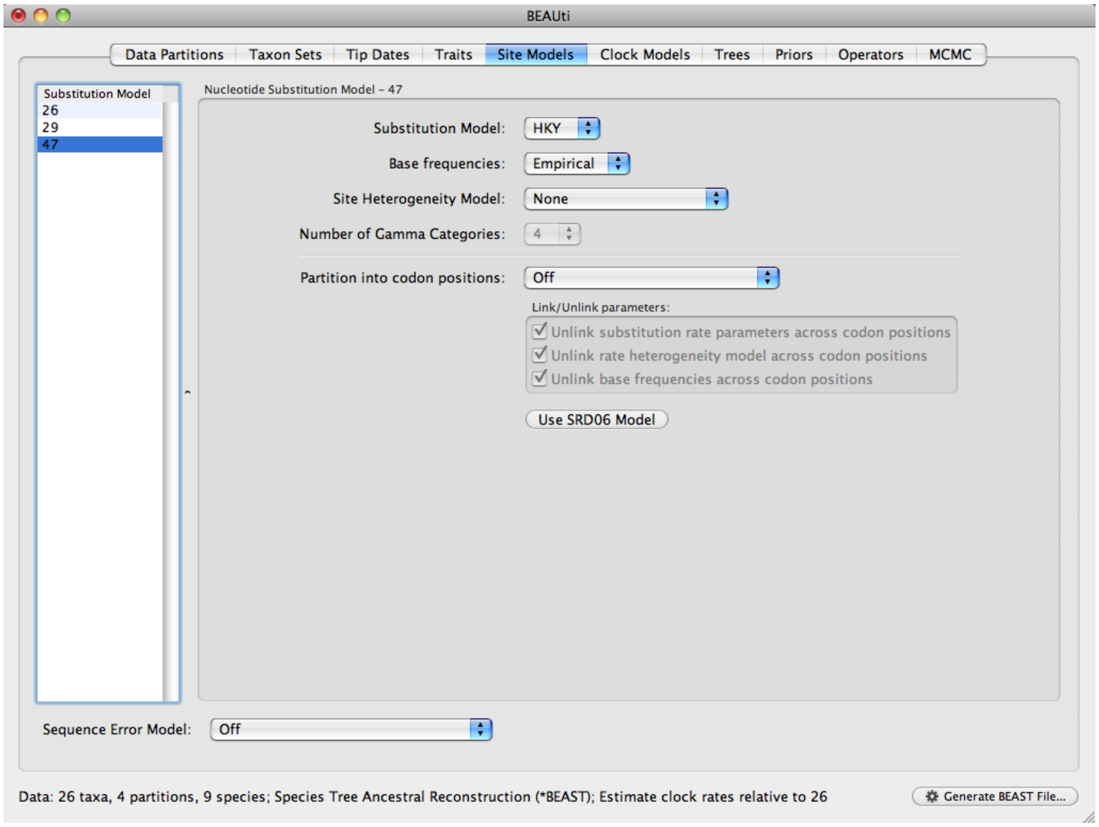
Taxon	Value
Orthogeomys_heterodus	heterodus
Thomomys_bottae_awahnee_a	bottae
Thomomys_bottae_awahnee_b	bottae
Thomomys_bottae_xerophilus	bottae
Thomomys_bottae_cactophilus	bottae
Thomomys_bottae_albatus	bottae
Thomomys_bottae_ruidosae	bottae
Thomomys_bottae_bottae	bottae
Thomomys_bottae_alpinus	bottae
Thomomys_bottae_riparius	bottae
Thomomys_bottae_mewa	bottae
Thomomys_bottae_saxatilis	bottae
Thomomys_bottae_laticeps	bottae
Thomomys_talpoides_ocius	talpoides
Thomomys_idahoensis_pygmaeus_a	idahoensis
Thomomys_idahoensis_pygmaeus_b	idahoensis
Thomomys_mazama_mazama	mazama
Thomomys_mazama_nasicus	mazama
Thomomys_monticola_a	monticola
Thomomys_monticola_b	monticola
Thomomys_talpoides_yakimensis	talpoides
Thomomys_talpoides_bridgeri	talpoides
Thomomys_townsendii_townsendii	townsendii
Thomomys_townsendii_relictus	townsendii
Thomomys_umbrinus_chihuahuae	umbrinus
Thomomys_umbrinus_atroavarius	umbrinus

Data: 26 taxa, 3 partitions, 8 species; Species Tree Ancestral Reconstruction (*BEAST); Estimate clock rates relative to 26 in nucleotide_group;

Setting the substitution model

The next thing to do is to click on the **Site Models** tab at the top of the main window. This will reveal the evolutionary model settings for BEAST. Exactly which options appear depend on whether the data are nucleotides, or amino acids, or binary data, or general data. The settings that will appear after loading the data set will be the default values so we need to make some changes.

Most of the models should be familiar to you. For this analysis, we will select each substitution model listed on the left side in turn to make the following change: select **Empirical** for the **Base frequencies**. *Remember to do this for all site models.*



Substitution Model
26
29
47

Nucleotide Substitution Model - 47

Substitution Model: HKY

Base frequencies: Empirical

Site Heterogeneity Model: None

Number of Gamma Categories: 4

Partition into codon positions: Off

Link/Unlink parameters:

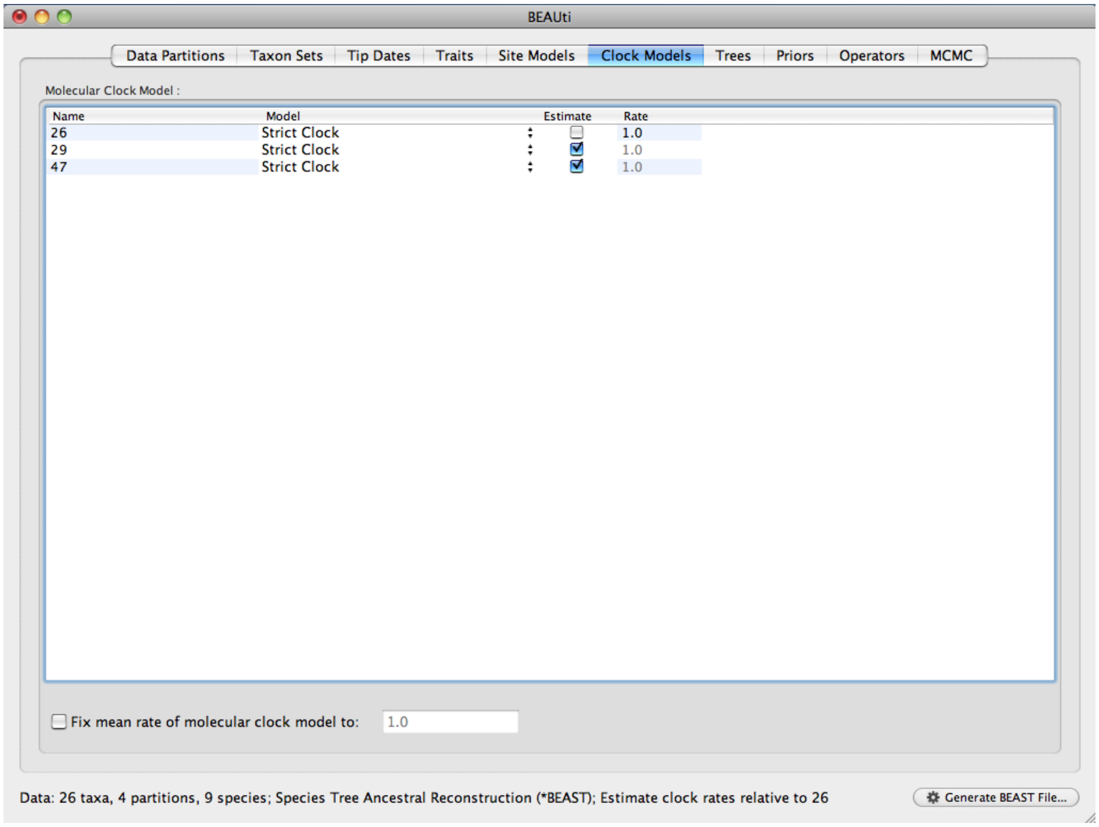
☑ Unlink substitution rate parameters across codon positions
☑ Unlink rate heterogeneity model across codon positions
☑ Unlink base frequencies across codon positions

Sequence Error Model: Off

Data: 26 taxa, 4 partitions, 9 species; Species Tree Ancestral Reconstruction (*BEAST); Estimate clock rates relative to 26

Setting the clock model

Second, click on the **Clock Models** tab at the top of the main window. In this analysis, we use the **Strict Clock** molecular clock model as default. Your model options should now look like this:



Molecular Clock Model :

Name	Model	Estimate	Rate
26	Strict Clock	□	1.0
29	Strict Clock	☑	1.0
47	Strict Clock	☑	1.0

□ Fix mean rate of molecular clock model to: 1.0

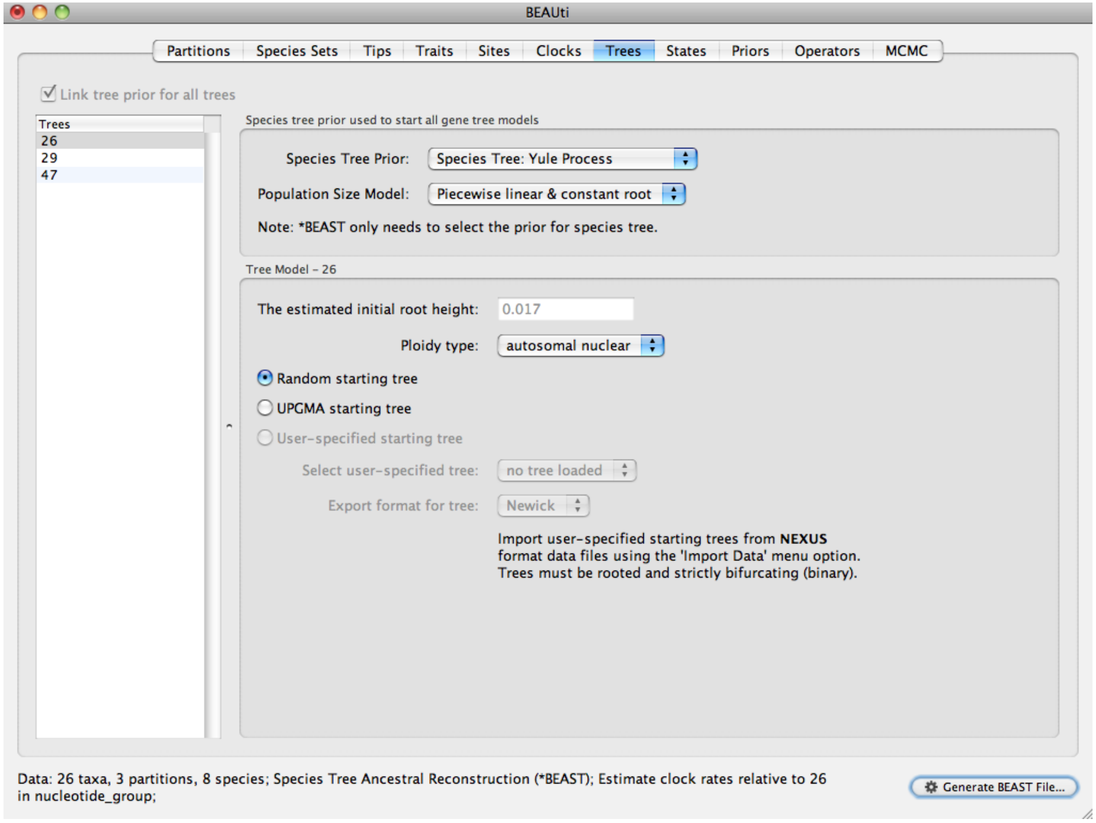
Data: 26 taxa, 4 partitions, 9 species; Species Tree Ancestral Reconstruction (*BEAST); Estimate clock rates relative to 26

The **Estimate** check box is unchecked for the first clock model and checked for the rest clock models, because we wish to estimate the mutation rate of each subsequent locus relative to the first locus whose rate is fixed to 1.0.

Trees

The **Trees** panel allows priors to be specified for each parameter in the model, which can be defined on the top of the panel. *BEAST has a different tree prior panel where users can only configure the species tree prior not gene tree priors (which are automatically specified by the multispecies coalescent). Currently, we have two species tree priors: **Yule Process** and **Birth-Death Process**; and three population size models: **Piecewise linear and constant root**, **Piecewise linear**, and **Piecewise constant**. In this analysis, we use the default options.

The bottom right panel is used to configure the corresponding starting trees. The **Ploidy Type** menu determines the type of sequence (mitochondrial, nuclear, X, Y). This matters since different modes of inheritance give rise to different effective population sizes. The **Starting Tree** menu provides three options, where the **user-specified** starting tree has to be loaded from the data file (e.g. NEXUS file) in advance. In this analysis, we simply use a random starting tree.



☑ Link tree prior for all trees

Trees
26
29
47

Species tree prior used to start all gene tree models

Species Tree Prior: Species Tree: Yule Process

Population Size Model: Piecewise linear & constant root

Note: *BEAST only needs to select the prior for species tree.

Tree Model - 26

The estimated initial root height: 0.017

Ploidy type: autosomal nuclear

● 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: 26 taxa, 3 partitions, 8 species; Species Tree Ancestral Reconstruction (*BEAST); Estimate clock rates relative to 26 in nucleotide_group;

Priors and Operators

The **Priors** panel allows priors to be specified for each parameter in the model. The **Operators** panel is used to configure technical settings that affect the efficiency of the MCMC program (see Notes for details). We leave these two panels unchanged in this analysis.

Setting the MCMC options

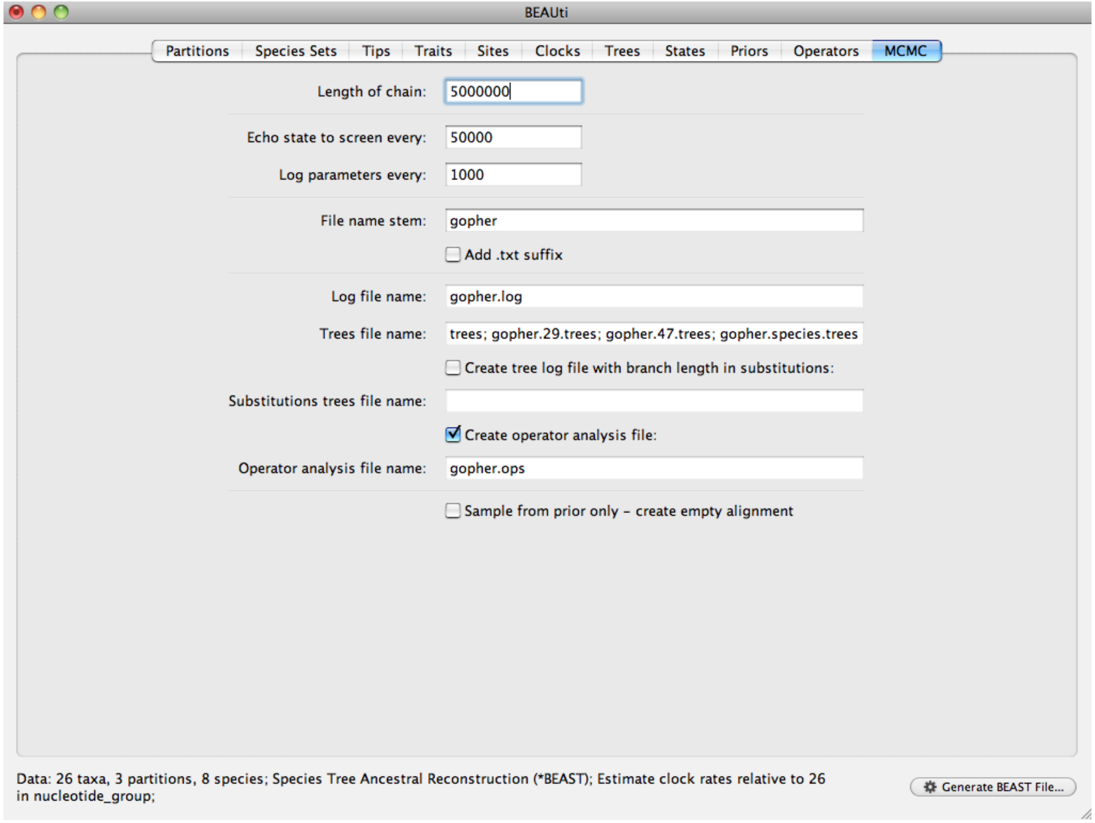
The next tab, **MCMC**, provides more general settings to control the length of the MCMC and the file names.

Firstly we have the **Length of chain**. This is the number of steps the MCMC will make in the chain before finishing. The appropriate length of the chain depends on the size of the data set, the complexity of the model and the accuracy of the answer required. The default value of 10,000,000 is entirely arbitrary and should be adjusted according to the size of your data set. For this data set let's initially set the chain length to 5,000,000 as this will run reasonably quickly on most modern computers (less than 20 minutes).

The next options specify how often the parameter values in the Markov chain should be displayed on the screen and recorded in the log file. The screen output is simply for monitoring the programs progress so can be set to any value (although if set too small, the sheer quantity of information being displayed on the screen will actually 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 analysis. 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 no less than chain length / 10,000.

For this exercise we will set the screen log to 10000 and the file log to 1000. The final two options give the file names of the log files for the sampled parameters and the trees. These will be set to a default based on the name of the imported NEXUS file.

If you would like to save the operator analysis into a file, you need to check **Create operator analysis file** which will generate a file with the suffix `.ops`.



Length of chain: 5000000

Echo state to screen every: 50000

Log parameters every: 1000

File name stem: gopher

□ Add .txt suffix

Log file name: gopher.log

Trees file name: trees; gopher.29.trees; gopher.47.trees; gopher.species.trees

□ Create tree log file with branch length in substitutions:

Substitutions trees file name:

☑ Create operator analysis file:

Operator analysis file name: gopher.ops

□ Sample from prior only – create empty alignment

Data: 26 taxa, 3 partitions, 8 species; Species Tree Ancestral Reconstruction (*BEAST); Estimate clock rates relative to 26 in nucleotide_group;

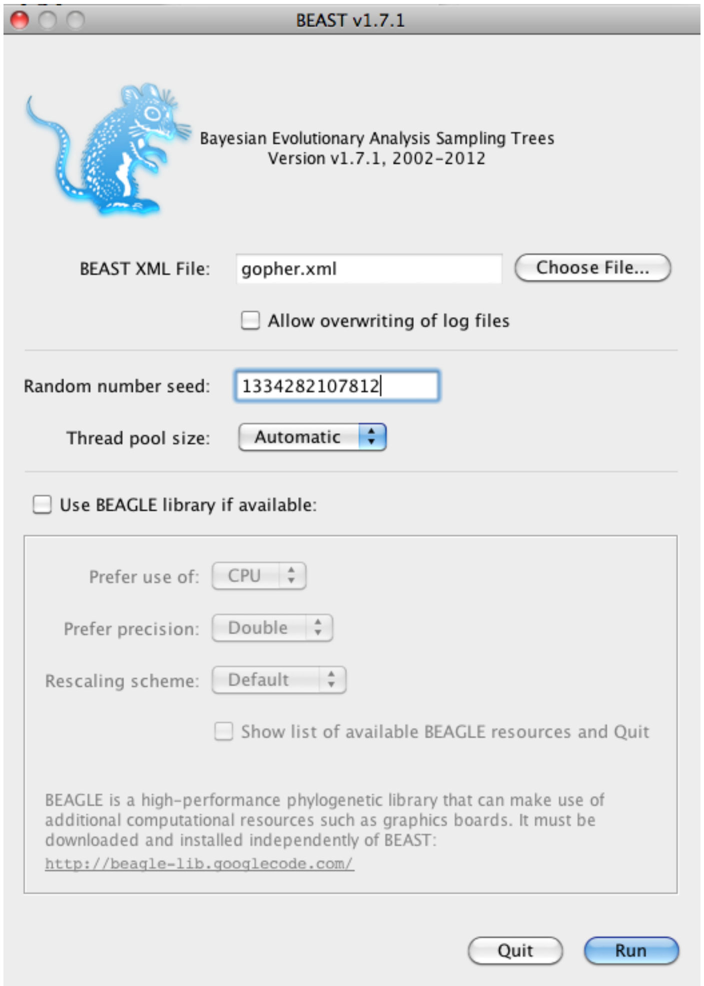
- If you are using windows then we suggest you add the suffix `.txt` to both of these (so, `gopher.log.txt` and `gopher.trees.txt`) so that Windows recognizes these as text files.

Generating the BEAST XML file

We are now ready to create the BEAST XML file. 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. Check the default priors setting and click **Continue**. Save the file with an appropriate name (we usually end the filename with `.xml`, i.e., `gopher.xml`). We are now ready to run the file through BEAST.

Running BEAST

Now run BEAST and when it asks for an input file, provide your newly created XML file as input by click **Choose File ...**, and then click **Run**.



Bayesian Evolutionary Analysis Sampling Trees
Version v1.7.1, 2002-2012

BEAST XML File: gopher.xml

□ Allow overwriting of log files

Random number seed: 1334282107812

Thread pool size: Automatic

□ Use BEAGLE library if available:

Prefer use of: CPU

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://beagle-lib.googlecode.com/

BEAST will then run until it has finished reporting information to the screen. The actual results files are saved to the disk in the same location as your input file. The output to the screen will look something like this:

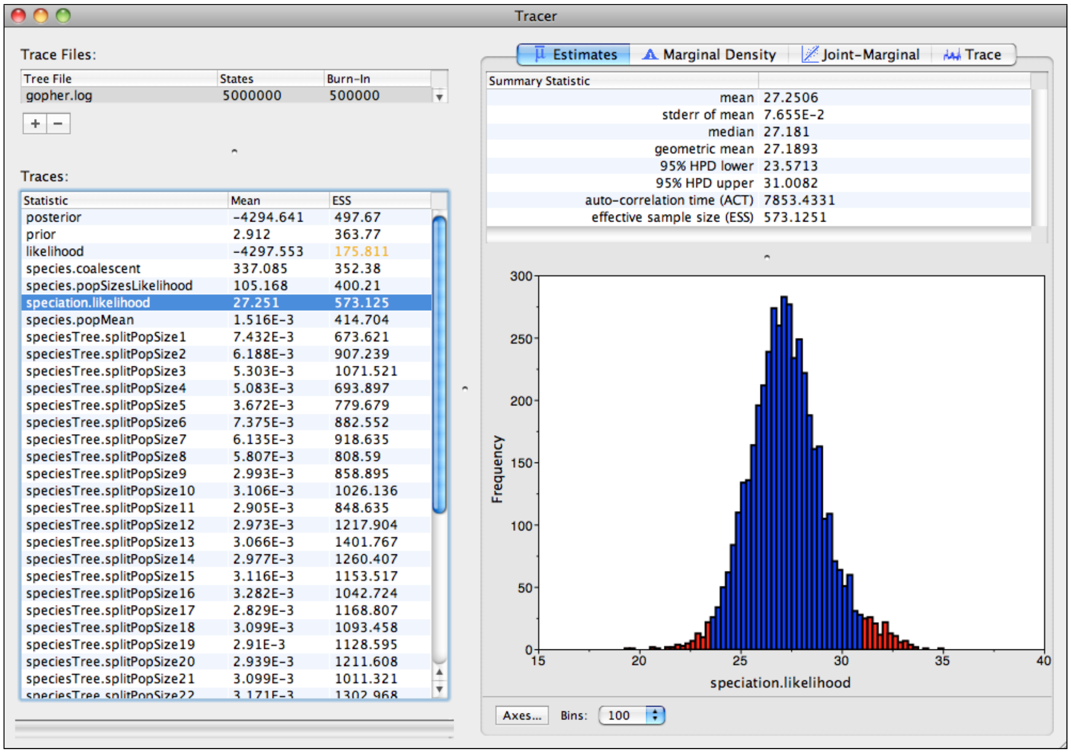
```
                BEAST v1.7.1, 2002-2012
    Bayesian Evolutionary Analysis Sampling Trees
              Designed and developed by
Alexei J. Drummond, Andrew Rambaut and Marc A. Suchard
```

```
                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.bio.ed.ac.uk

Source code distributed under the GNU Lesser General Public License:
             http://code.google.com/p/beast-mcmc

                      BEAST developers:
Alex Alekseyenko, Trevor Bedford, Erik Bloomquist, Joseph Heled,
Sebastian Hoehna, Denise Kuehnert, Philippe Lemey, Wai Lok Sibon Li,
Gerton Lunter, Sidney Markowitz, Vladimir Minin, Michael Defoin Platel,
           Oliver Pybus, Chieh-Hsi Wu, Walter Xie

                          Thanks to:
     Roald Forsberg, Beth Shapiro and Korbinian Strimmer


Random number seed: 1334282107812


Parsing XML file: gopher.xml
  File encoding: MacRoman
Read alignment: alignment1
  Sequences = 26
      Sites = 614
   Datatype = nucleotide
Read alignment: alignment2
  Sequences = 26
      Sites = 601
   Datatype = nucleotide
Read alignment: alignment3
  Sequences = 26
      Sites = 819
   Datatype = nucleotide
Site patterns '26.patterns' created from positions 1-614 of alignment 'alignment1'
  pattern count = 144
Site patterns '29.patterns' created from positions 1-601 of alignment 'alignment2'
  pattern count = 71
Site patterns '47.patterns' created from positions 1-819 of alignment 'alignment3'
  pattern count = 153
Creating the tree model, '26.treeModel'
  initial tree topology = ((((((((((Thomomys_bottae_bottae,Thomomys_monticola_b),Orthogeomys_heterodus),(Thomomys_talpoides_y
  tree height = 0.017
Creating the tree model, '29.treeModel'
  initial tree topology = (((((((((Thomomys_bottae_albatus,Thomomys_bottae_xerophilus),Thomomys_bottae_saxatilis),Thomomys_mo
  tree height = 0.016
Creating the tree model, '47.treeModel'
  initial tree topology = (((((((((Thomomys_bottae_cactophilus,Thomomys_bottae_saxatilis),(Thomomys_bottae_mewa,Thomomys_town
  tree height = 0.017
Using strict molecular clock model.
Using strict molecular clock model.
Using strict molecular clock model.
Creating state frequencies model '26.frequencies': Using empirical frequencies from data = {0.40772, 0.20916, 0.19046, 0.1926
Creating HKY substitution model. Initial kappa = 2.0
Creating site model.
Creating state frequencies model '29.frequencies': Using empirical frequencies from data = {0.24545, 0.21384, 0.23701, 0.3037
Creating HKY substitution model. Initial kappa = 2.0
Creating site model.
Creating state frequencies model '47.frequencies': Using empirical frequencies from data = {0.2017, 0.21368, 0.21208, 0.37254
Creating HKY substitution model. Initial kappa = 2.0
Creating site model.
Loading native NucleotideLikelihoodCore successfully
TreeLikelihood(26.treeModel) using native nucleotide likelihood core
  Ignoring ambiguities in tree likelihood.
  With 144 unique site patterns.
Branch rate model used: strictClockBranchRates
TreeLikelihood(29.treeModel) using native nucleotide likelihood core
  Ignoring ambiguities in tree likelihood.
  With 71 unique site patterns.
Branch rate model used: strictClockBranchRates
TreeLikelihood(47.treeModel) using native nucleotide likelihood core
  Ignoring ambiguities in tree likelihood.
  With 153 unique site patterns.
Branch rate model used: strictClockBranchRates
Using Yule prior on tree
Likelihood is using -1 threads.
Creating the MCMC chain:
  chainLength=5000000
  autoOptimize=true
  autoOptimize delayed for 50000 steps
# BEAST v1.7.1, r4860
# Generated Fri Apr 13 16:02:42 NZST 2012 [seed=1334282107812]
state Posterior    Prior        Likelihood   PopMean      26.rootHeight 29.rootHeight 47.rootHeight 26.clock.rate
29.clock.rate 47.clock.rate
0 -8271.7599   -408.9912    -7862.7688   1.0000       1.7E-2       1.6E-2       1.7E-2       1.00000      1.00000      1.0000
50000 -4402.6081   -110.7170    -4291.8911   0.0093       2.85227E-2   2.66503E-2   1.70572E-2   1.00000      0.83166      1.
100000 -4282.8572   5.5473       -4288.4044   0.0009       2.49661E-2   3.13122E-2   1.68536E-2   1.00000      0.80206      1
150000 -4321.6989   -32.7608     -4288.9382   0.0025       3.17837E-2   2.81154E-2   1.88322E-2   1.00000      0.83429      1
200000 -4301.2881   -12.5667     -4288.7214   0.0013       2.32344E-2   4.17748E-2   1.82301E-2   1.00000      0.65006      1
250000 -4312.7097   -9.6236      -4303.0861   0.0020       3.12738E-2   2.28171E-2   1.84486E-2   1.00000      0.82719      1
300000 -4278.9322   35.3165      -4314.2487   0.0008       2.27743E-2   2.26837E-2   1.82648E-2   1.00000      1.19420      1

... ...

4900000 -4276.9360   17.6994      -4294.6354   0.0018       2.23113E-2   1.66667E-2   1.36587E-2   1.00000      0.91509
4950000 -4238.0976   49.0728      -4287.1704   0.0008       2.45626E-2   1.28858E-2   1.29766E-2   1.00000      1.24784
5000000 -4304.9506   -17.2098     -4287.7408   0.0017       2.47898E-2   3.78072E-2   2.25424E-2   1.00000      0.74967

Operator analysis
Operator                                          Tuning   Count      Time     Time/Op  Pr(accept)
scale(26.kappa)                                   0.36    1106       122      0.11     0.3698
scale(29.kappa)                                   0.383   1097       91       0.08     0.381
scale(47.kappa)                                   0.438   1143       131      0.11     0.3596
scale(29.clock.rate)                              0.485   33005      2668     0.08     0.2684
scale(47.clock.rate)                              0.568   33415      3546     0.11     0.2804
up:29.clock.rate 47.clock.rate species.yule.birthRate down:speciesTree species.popMean speciesTree.splitPopSize nodeHeights(2
subtreeSlide(26.treeModel)                        0.003   165916     7583     0.05     0.2326
Narrow Exchange(26.treeModel)                             165585     6047     0.04     0.2971
Wide Exchange(26.treeModel)                               33216      988      0.03     0.0263
wilsonBalding(26.treeModel)                               33151      1648     0.05     0.0379
scale(26.treeModel.rootHeight)                    0.511   33209      1527     0.05     0.2864
uniform(nodeHeights(26.treeModel))                        331028     17469    0.05     0.5605
subtreeSlide(29.treeModel)                        0.003   166248     7404     0.04     0.2201
Narrow Exchange(29.treeModel)                             165728     5610     0.03     0.3173
Wide Exchange(29.treeModel)                               33262      887      0.03     0.0428
wilsonBalding(29.treeModel)                               33206      1468     0.04     0.0473
scale(29.treeModel.rootHeight)                    0.447   33466      1530     0.05     0.2679
uniform(nodeHeights(29.treeModel))                        330344     16735    0.05     0.5782
```

```
subtreeSlide(47.treeModel)                         0.002   165265    7854    0.05    0.2449
Narrow Exchange(47.treeModel)                              165918    6217    0.04    0.2445
Wide Exchange(47.treeModel)                                33011     998     0.03    0.0141
wilsonBalding(47.treeModel)                                33105     1795    0.05    0.0186
scale(47.treeModel.rootHeight)                     0.61    33012     1317    0.04    0.2409
uniform(nodeHeights(47.treeModel))                         332093    17991   0.05    0.5415
up:down:nodeHeights(26.treeModel)                  0.782   32762     3383    0.1     0.1912
up:29.clock.rate down:nodeHeights(29.treeModel)    0.766   33152     2502    0.08    0.1841
up:47.clock.rate down:nodeHeights(47.treeModel)    0.713   32871     3484    0.11    0.1732
scale(species.popMean)                             0.495   54949     1784    0.03    0.2719
scale(species.yule.birthRate)                      0.24    33178     1132    0.03    0.3156
scale(speciesTree.splitPopSize)                    0.171   1037737   40459   0.04    0.2572
nodeReHeight(sptree,species)                               1036242   39954   0.04    0.295

6.349283333333333 minutes
```

Analyzing the results

Run the program called **Tracer** to analyze the output of BEAST. When the main window has opened, choose **Import Trace File...** from the **File** menu and select the file that BEAST has created called `gopher.log`. You should now see a window like the following:



Trace Files:

Tree File	States	Burn-In
gopher.log	5000000	500000

Traces:

Statistic	Mean	ESS
posterior	-4294.641	497.67
prior	2.912	363.77
likelihood	-4297.553	175.811
species.coalescent	337.085	352.38
species.popSizesLikelihood	105.168	400.21
speciation.likelihood	27.251	573.125
species.popMean	1.516E-3	414.704
speciesTree.splitPopSize1	7.432E-3	673.621
speciesTree.splitPopSize2	6.188E-3	907.239
speciesTree.splitPopSize3	5.303E-3	1071.521
speciesTree.splitPopSize4	5.083E-3	693.897
speciesTree.splitPopSize5	3.672E-3	779.679
speciesTree.splitPopSize6	7.375E-3	882.552
speciesTree.splitPopSize7	6.135E-3	918.635
speciesTree.splitPopSize8	5.807E-3	808.59
speciesTree.splitPopSize9	2.993E-3	858.895
speciesTree.splitPopSize10	3.106E-3	1026.136
speciesTree.splitPopSize11	2.905E-3	848.635
speciesTree.splitPopSize12	2.973E-3	1217.904
speciesTree.splitPopSize13	3.066E-3	1401.767
speciesTree.splitPopSize14	2.977E-3	1260.407
speciesTree.splitPopSize15	3.116E-3	1153.517
speciesTree.splitPopSize16	3.282E-3	1042.724
speciesTree.splitPopSize17	2.829E-3	1168.807
speciesTree.splitPopSize18	3.099E-3	1093.458
speciesTree.splitPopSize19	2.91E-3	1128.595
speciesTree.splitPopSize20	2.939E-3	1211.608
speciesTree.splitPopSize21	3.099E-3	1011.321
speciesTree.splitPopSize22	3.171E-3	1302.968

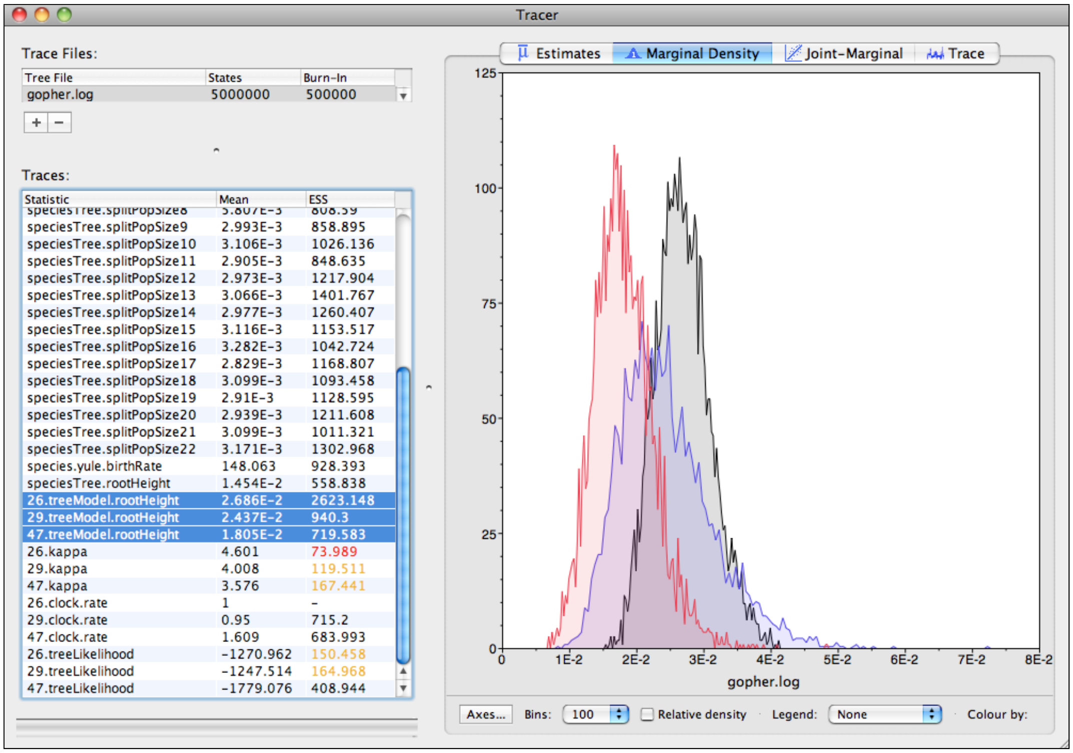
Summary Statistic	
mean	27.2506
stderr of mean	7.655E-2
median	27.181
geometric mean	27.1893
95% HPD lower	23.5713
95% HPD upper	31.0082
auto-correlation time (ACT)	7853.4331
effective sample size (ESS)	573.1251

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

On the left hand side is a list of the different quantities that BEAST has logged. There are traces for the posterior (this is the log of the product of the tree likelihood and the prior probabilities), 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.

Tracer will plot a (marginal posterior) distribution for the selected parameter and also give you statistics such as the mean and median. The `95% HPD lower` or `upper` stands for *highest posterior density interval* and represents the most compact interval on the selected parameter that contains 95% of the posterior probability. It can be thought of as a Bayesian analog to a confidence interval.

Select the `treeModel.rootHeight` parameter and the next three (hold shift whilst selecting). This will show a display of the age of the root and the three gene trees. If you switch the tab at the top of the window to **Marginal Density** then you will get a plot of the marginal posterior densities of each of these date estimates overlayed:



Trace Files:

Tree File	States	Burn-In
gopher.log	5000000	500000

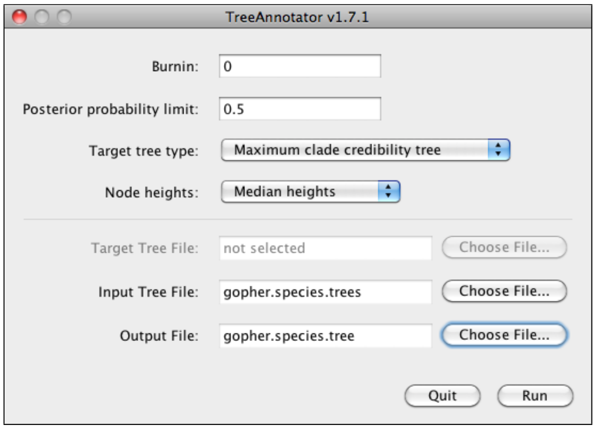
Traces:

Statistic	Mean	ESS
speciesTree.splitPopSize8	3.807E-3	808.59
speciesTree.splitPopSize9	2.993E-3	858.895
speciesTree.splitPopSize10	3.106E-3	1026.136
speciesTree.splitPopSize11	2.905E-3	848.635
speciesTree.splitPopSize12	2.973E-3	1217.904
speciesTree.splitPopSize13	3.066E-3	1401.767
speciesTree.splitPopSize14	2.977E-3	1260.407
speciesTree.splitPopSize15	3.116E-3	1153.517
speciesTree.splitPopSize16	3.282E-3	1042.724
speciesTree.splitPopSize17	2.829E-3	1168.807
speciesTree.splitPopSize18	3.099E-3	1093.458
speciesTree.splitPopSize19	2.91E-3	1128.595
speciesTree.splitPopSize20	2.939E-3	1211.608
speciesTree.splitPopSize21	3.099E-3	1011.321
speciesTree.splitPopSize22	3.171E-3	1302.968
species.yule.birthRate	148.063	928.393
speciesTree.rootHeight	1.454E-2	558.838
26.treeModel.rootHeight	2.686E-2	2623.148
29.treeModel.rootHeight	2.437E-2	940.3
47.treeModel.rootHeight	1.805E-2	719.583
26.kappa	4.601	73.989
29.kappa	4.008	119.511
47.kappa	3.576	167.441
26.clock.rate	1	-
29.clock.rate	0.95	715.2
47.clock.rate	1.609	683.993
26.treeLikelihood	-1270.962	150.458
29.treeLikelihood	-1247.514	164.968
47.treeLikelihood	-1779.076	408.944

Bins: 100 □ Relative density Legend: None Colour by:

Obtaining an estimate of the phylogenetic tree

BEAST also produces a sample of plausible trees. These need to be summarized using the program **TreeAnnotator** (see Notes for details). This will take the set of trees and identify a single tree that best represents the posterior distribution. It will then annotate this selected tree topology with the mean ages of all the nodes as well as the 95% HPD interval of divergence times for each clade in the selected tree. It will also calculate the posterior clade probability for each node. Run the **TreeAnnotator** program and set it up to look like this:



TreeAnnotator v1.7.1

Burnin: 0

Posterior probability limit: 0.5

Target tree type: Maximum clade credibility tree

Node heights: Median heights

Target Tree File: not selected

Input Tree File: gopher.species.trees

Output File: gopher.species.tree

The burnin is the number of trees to remove from the start of the sample. Unlike **Tracer** which specifies the number of steps as a burnin, in **TreeAnnotator** you need to specify the actual number of trees. For this run, we use the default setting.

The **Posterior probability limit** option specifies a limit such that if a node is found at less than this frequency in the sample of trees (i.e., has a posterior probability less than this limit), it will not be annotated. The default of 0.5 means that only nodes seen in the majority of trees will be annotated. Set this to zero to annotate all nodes.

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

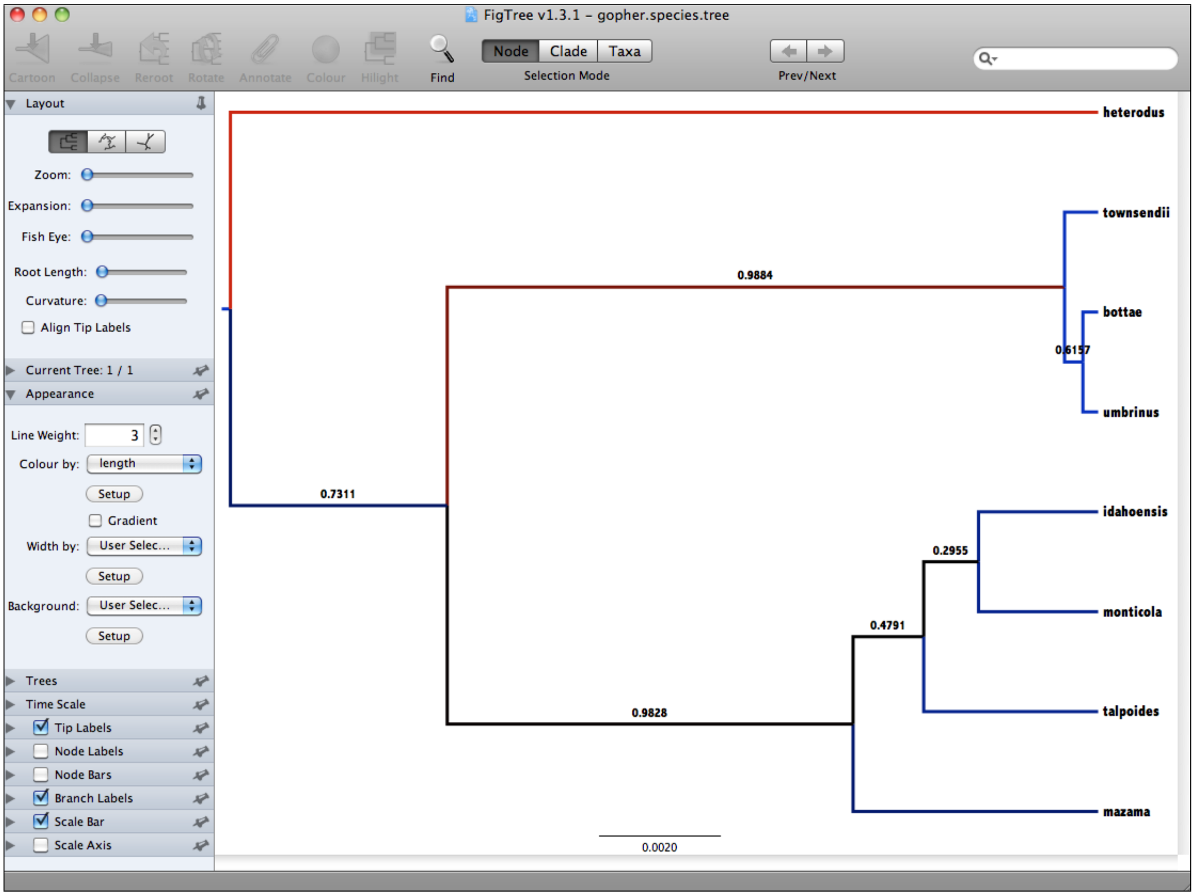
Choose **Mean heights** for node heights. This sets the heights (ages) of each node in the tree to the mean height across the entire sample of trees for that clade.

For the input file, select the trees file that BEAST created (by default this will be called `gopher.species.trees`) and select a file for the output (here we called it `gopher.species.tree`).

Now press `Run` and wait for the program to finish.

Viewing the Tree

Finally, we can look at the tree in another program called **FigTree**. Run this program, and open the `gopher.species.tree` file by using the Open command in the File menu. The tree should appear. You can now try selecting some of the options in the control panel on the left. Try selecting **Node Bars** to get node age error bars. Also turn on **Branch Labels** and select **posterior** to get it to display the posterior probability for each node. Under **Appearance** you can also tell FigTree to colour the branches by the length. You should end up with something like this:



Layout
Zoom:
Expansion:
Fish Eye:
Root Length:
Curvature:
□ Align Tip Labels
Current Tree: 1 / 1
Appearance
Line Weight: 3
Colour by: length
Setup
□ Gradient
Width by: User Selec...
Setup
Background: User Selec...
Setup
Trees
Time Scale
☑ Tip Labels
□ Node Labels
□ Node Bars
☑ Branch Labels
☑ Scale Bar
□ Scale Axis

Comparing your results to the prior

Using BEAUti, set up the same analysis but under the MCMC options, select the **Sample from prior only** option. This will allow you to visualize the full prior distribution in the absence of your sequence data. Summarize the trees from the full prior distribution and compare the summary to the posterior summary tree.

References

[1] N.M. Belfiore, L. Liu, and C. Moritz, *Multilocus phylogenetics of a rapid radiation in the genus Thomomys (Rodentia: Geomyidae)*, Systematic Biology **57** (2008), no. 2, 294.