# The Phylogenetics Handbook

September 14, 2007

# Contents

[Alexei J. Drummond and Andrew Rambaut] Alexei J. Drummond
Department of Computer Science, University of Auckland, Auckland, NZ Andrew Rambaut
Institute of Evolutionary Biology, University of Edinburgh, Edinburgh, UK

# Chapter 18

# Bayesian evolutionary analysis by sampling trees

Alexei J. Drummond[1] and Andrew Rambaut[2]

[1]Department of Computer Science
The University of Auckland, Private Bag 92019
Auckland, New Zealand

[2]Institute of Evolutionary Biology
University of Edinburgh
Edinburgh, United Kingdom
a.rambaut@ed.ac.uk

## 18.1 Theory

### 18.1.1 Background

The BEAST software package is an ambitious attempt to provide a general framework for parameter estimation and hypothesis testing of evolutionary models from molecular sequence data. BEAST is a Bayesian statistical framework and thus provides a role for prior knowledge in combination with the information provided by the data. Bayesian ***Markov chain Monte Carlo (MCMC)*** has already been enthusiastically embraced as the state-of-the-art method for phylogenetic reconstruction, largely driven by the rapid and widespread adoption of MrBayes [1]. This enthusiasm can be attributed to a number of factors. Firstly, Bayesian methods allow the relatively straightforward implementation of extremely complex ***evolutionary models***. Secondly, there is an often erroneous perception that Bayesian estimation is "faster" than heuristic optimization based on the ***maximum likelihood*** criterion.

BEAST can be compared to a number of other software packages with similar goals, such as MrBayes [1], which currently focuses on phylogenetic inference and LAMARC [8] (discussed in the next chapter) and BATWING [3], which focus predominantly on ***coalescent***-based population genetics. Like these software packages, the core algorithm implemented in BEAST is Metropolis-Hastings MCMC [12, 11]. MCMC is a stochastic algorithm that produces sample-based estimates of a target distribution of choice. For our purposes the target distribution is the ***posterior distribution*** of a set of evolutionary parameters given an alignment of molecular sequences.

Possibly the most distinguishing feature of BEAST is its firm focus on calibrated phylogenies and genealogies, that is, rooted trees incorporating a timescale. This is achieved by explicitly modeling the ***rate of molecular evolution*** on each branch in the tree. On the simplest level this can be a uniform rate over the entire tree (i.e. the ***molecular clock model*** [13]) with this rate known in advance or estimated from calibration information. However, one of the most promising recent advances in molecular phylogenetics has been the introduction of ***relaxed molecular clock*** models that do not assume a constant rate across lineages[18, 19, 15, 16, 17, 14]. BEAST was the first software package that allows phylogenetic inference under such models [32].

In the context of ***genealogy***-based population genetics (see previous chapter), the target distribution of interest is the posterior probability of the the population genetic parameters ($\phi$) given a multiple sequence alignment ($D$):

$$p(\phi|D) = \frac{1}{Z}\int_{g,\omega} Pr\{D|g,\omega\}p(g|\phi)p(\phi)p(\omega)dgd\omega \tag{18.1}$$

In order to estimate the posterior probability distribution of $\phi$ it is necessary to average over all possible genealogies ($g$) and substitution model parameters ($\omega$) proportional to their probabilities. This integration is achieved by MCMC. In the above equation $Pr\{D|g,\omega\}$ is the ***likelihood*** of genealogy $g$ given the sequence data and the substitution model [5] and $p(g|\phi)$ is the coalescent prior of the genealogy given the population parameters $\phi$. In the original formulation of the Kingman coalescent [29] (see also previous chapter), there is a single population size, $\phi = \{\theta\}$ and the coalescent prior takes the form:

$$p(g|\phi) = \frac{1}{\theta^{n-1}}\prod_{i=2}^{n}\exp\frac{-i(i-1)u_i}{2\theta} \tag{18.2}$$

where $u_i$ is the length of time over which the genealogy $g$ has exactly $i$ lineages. This formulation assumes that the units of time are mutations per site and that all sequences are sampled from the same time. Both of these assumptions can be relaxed [2]. It is also possible to devise more complex coalescent models so that the population size is a function of time. BEAST supports a number of demographic models including constant size, exponential growth, logistic growth, expansion and the highly parameteric Bayesian skyline plot [30]. Currently BEAST does not include coalescent models of migration or recombination but these processes will be included in a future version. For the case of

contemporaneously sampled sequences these processes can be investigated using LAMARC (see next chapter).

The purpose behind the development of BEAST is to bring a large number of complementary evolutionary models (e.g. substitution models, demographic tree priors, relaxed clock models, node calibration models) into a single coherent framework for evolutionary inference. This building-block principle of constructing a complex evolutionary model out of a number of simpler model components provides powerful new possibilities for molecular sequence analysis. The motivation for doing this is (1) to avoid the unnecessary simplifying assumptions that currently exist in many evolutionary analysis packages and (2) to provide new model combinations and a flexible system for model specification so that researchers can tailor their evolutionary analyses to their specific set of questions.

### 18.1.2 Bayesian MCMC for genealogy-based population genetics

The integration in equation 18.1 is achieved by constructing a chain of parameter/genealogy combinations in such a way that they form a (correlated) sample of states from the full posterior distribution:

$$p(g, \omega, \phi|D) = \frac{1}{Z} Pr\{D|g, \omega\} p(g|\phi) p(\phi) p(\omega) \tag{18.3}$$

We summarize the marginal density $p(\phi|D)$ by using samples $(g, \omega, \phi) \sim p(g, \omega, \phi|D)$. The sampled genealogies and substitution model parameters can be thought of as uninteresting ***nuisance parameters***.

To construct the ***Markov chain*** we begin with an initial state $x_0 = (g^{(0)}, \omega^{(0)}, \phi^{(0)})$. At each step $i$ in the chain we begin by proposing a new state $y$. An operator $(m)$ proposes this state by copying the previous state $x_{i-1}$ and making a small alteration (to the genealogy, or the parameter values, or both). The probability of the previous state and the newly proposed state are then compared in an accept/reject step. The proposed state is accepted as the new state in the Markov chain with probability:

$$\alpha = \min\left(1, \frac{p(y|D)}{p(x_{i-1}|D)}\right) \tag{18.4}$$

If the proposed state $y$ is accepted, then state $x_i = y$ otherwise the previous state is kept $(x_i = x_{i-1})$. Notice that if the posterior probability of $y$ is greater than $x_{i-1}$ then $y$ will definitely be accepted. Whereas when $y$ has lower probability than $x_{i-1}$ it will only be accepted with a probability proportional to the ratio of their posterior probabilities. The above acceptance probability assumes that the operator is symmetric, so that the probability of proposing state $y$ from state $x$, $q(y|x)$, is the same as proposing state $x$ from state $y$, $q(x|y)$. BEAST uses a mixture of symmetric and asymmetric operators. At each step in the chain an operator $(m)$ is chosen at random (with weights). When operator $m$ is not symmetric then $q_m(y|x) \neq q_m(x|y)$ and the acceptance probability becomes

$$\alpha = \min\left(1, \frac{p(y|D)}{p(x_{i-1}|D)} \frac{q(x_{i-1}|y)}{q(y|x_{i-1})}\right) \tag{18.5}$$

The additional ratio of proposal probabilities is called the Hastings ratio [11].

### Implementation

The overall architecture of the `BEAST` software package is a file-mediated pipeline. The core program takes, as input, an XML file describing the data to be analyzed, the models to be used and technical details of the MCMC algorithm such as the proposal distribution (defined by the operators), the length of the Markov chain (chain length) and the output options. The output of a `BEAST` analysis is a set of tab-delimited plain text files that summarize the estimated posterior distribution of parameter values and trees.

A number of additional software programs assist in generating the input and analyzing the output:

- `BEAUti` is a software package written in Java and distributed with `BEAST` that provides a graphical user interface for generating `BEAST` XML input files for a number of simple model combinations.
- `Tracer` is a software package written in Java and distributed separately from `BEAST` that provides a graphical tool for MCMC output analysis. It can be used for the analysis of the output of `BEAST` as well as the output of other common MCMC packages such as `MrBayes` [1] and `Bali-Phy` [20].

Because of the combinatorial nature of the `BEAST` XML input format, not all models can be specified through the graphical interface of `BEAUti`. Indeed, the sheer number of possible combinations of models mean that, inevitably, some combinations will be untested. It is also possible to create models that are inappropriate or meaningless for the data being analysed. `BEAUti` is therefore intended as a way of generating commonly used and well-understood analyses. For the more adventurous researcher, and with the above warnings in mind, the XML file can be directly edited. A number of online tutorials are available to guide users on how to do this.

### Input Format

One of the primary motivations for providing a highly structured XML input format is to facilitate reproducibility of complex evolutionary analyses. While an interactive graphical user interface provides a pleasant user experience, it can be time-consuming and error-prone for a user to record and reproduce the full sequence of choices that are made, especially with the large array of options typically available for MCMC analysis. By separating the graphical user interface (`BEAUti`) from the analysis (`BEAST`) we accommodate an XML layer that captures the exact details of the MCMC analysis being performed. We

strongly encourage the routine publication of XML input files as supplementary information with publication of the results of a BEAST analysis. Because of the non-trivial nature of MCMC analyses and the need to promote reproducibility, it is our view that the publication of the exact details of any Bayesian MCMC analysis should be made a pre-requisite for publication of all MCMC analysis results.

#### Output and results

The output from BEAST is a simple tab-delimited plain text file format with one a row for each sample. When accumulated into frequency distributions, this file provides an estimate of the marginal posterior probability distribution of each parameter. This can be done using any standard statistics package or using the specially written package, Tracer [21]. Tracer provides a number of graphical and statistical ways of analyzing the output of BEAST to check performance and accuracy. It also provides specialized functions for summarizing the posterior distribution of population size through time when a coalescent model is used.

The phylogenetic tree of each sample state is written to a separate file as either NEWICK or NEXUS format. This can be used to investigate the posterior probability of various phylogenetic questions such as the ***monophyly*** of a particular group of organisms or to obtain a consensus phylogeny.

#### Computational Performance

Although there is always a trade-off between a program's flexibility and its computational performance, BEAST performs well on large analyses (e.g. [22]). A Bayesian MCMC algorithm needs to evaluate the likelihood of each state in the chain and thus performance is dictated by the speed at which these likelihood evaluations can be made. BEAST attempts to minimize the time taken to evaluate a state by only recalculating the likelihood for parts of the model that have changed from the previous state. Furthermore, the core computational functions have been implemented in the C programming language. This can be compiled into a highly optimized library for a given platform providing an improvement in speed. If this library is not found, BEAST will use its Java version of these functions, thereby retaining its platform-independence.

### 18.1.3 Results and Discussion

BEAST provides considerable flexibility in the specification of an evolutionary model. For example, consider the analysis of a multiple sequence alignment of protein-coding DNA. In a BEAST analysis, it is possible to allow each codon position to have a different rate, a different amount of rate heterogeneity among sites, and a different amount of rate heterogeneity among branches, while, at the same time, sharing the same intrinsic ratio of ***transitions*** to ***transversions*** with the other codon positions. In fact, all parameters can be shared or made independent among partitions of the sequence data.

An unavoidable feature of Bayesian statistical analysis is the specification of a prior distribution over parameter values. This requirement is both an advantage and a burden. It is an advantage because relevant knowledge such as palaeontological calibration of phylogenies is readily incorporated into an analysis. However, when no obvious prior distribution for a parameter exists, a burden is placed on the researcher to ensure that the prior selected is not inadvertently influencing the posterior distribution of parameters of interest.

In `BEAST`, all parameters (whether they be substitutional, demographic or genealogical) can be given informative priors (e.g. exponential, normal, lognormal or uniform with bounds, or combinations of these). For example, the age of the root of the tree can be given an exponential prior with a pre-specified mean.

The five components of an evolutionary model for a set of aligned nucleotides in `BEAST` are:

- *Substitution model* - The substitution model is a homogeneous ***Markov process*** that defines the relative rates at which different substitutions occur along a branch in the tree.
- *Rate model among sites* - The rate model among sites defines the distribution of relative rates of evolutionary change among sites.
- *Rate model among branches* - The rate model among branches defines the distribution of rates among branches and is used to convert the tree, which is in units of time, to units of substitutions. These models are important for divergence time estimation procedures and producing time scales on demographic reconstructions.
- *Tree* - a model of the phylogenetic or genealogical relationships of the sequences.
- *Tree prior* - The tree prior provides a parameterized prior distribution for the node heights (in units of time) and tree topology.

### Substitution models and rate models among sites

For nucleotide data, all of the models that are nested in the general time-reversible (GTR) model [23] - including the well known HKY85 model [24] - can be specified. For the analysis of amino acid sequence alignments all of the following replacement models can be used: Blosum62, CPREV, Dayhoff, JTT, MTREV and WAG. When nucleotide data represents a coding sequence (i.e. an in-frame protein-coding sequence) the Goldman and Yang model [25] can be used to model codon evolution.

In addition, both Γ-distributed rates among sites [26] and a proportion of invariant sites can be used to describe rate heterogeneity among sites.

### Rate models among branches, divergence time estimation and time-stamped data

Without calibration information, ***mutation rate*** ($\mu$) and time ($t$) are confounded and thus branches must be estimated in units of mutations per site, $\mu t$. However when a strong prior is available for (1) the time of one or more nodes, or (2) the overall mutation rate, then the genealogy can be estimated in units of time.

The basic model for rates among branches supported by `BEAST` is the strict molecular clock model [13], calibrated by specifying either a substitution rate or the date of a node or set of nodes. In this context, dates of divergence for particular clades can be estimated. The clades can be defined either by a monophyletic grouping of taxa or as the most recent common ancestor of a set of taxa of interest. The second alternative does not require monophyly of the selected taxa with respect to the rest of the tree. Furthermore, when the differences in the dates associated with the tips of the tree comprise a significant proportion of the age of the entire tree, these dates can be incorporated into the model providing a source of information about the overall rate of evolutionary change [2, 27].

In `BEAST`, divergence time estimation has also been extended to include *relaxed phylogenetics* models, in which the rate of evolution is allowed to vary among the branches of the tree. In particular we support a class of ***uncorrelated relaxed clock*** branch rate models, in which the rate at each branch is drawn from an underlying distribution such as exponential or lognormal [32].

If the sequence data are all from one time point, then the overall evolutionary rate must be specified with a strong prior. The units implied by the prior on the evolutionary rate will determine the units of the node heights in the tree (including the age of the most recent common ancestor) as well as the units of the demographic parameters such as the population size parameter and the growth rate. For example, if the evolutionary rate is set to 1.0, then the node heights (and root height) will be in units of mutations per site (i.e. the units of branch lengths produced by common software packages such as `MrBayes` 3.0). Similarly, for a ***haploid*** population, the coalescent parameter will be an estimate of $N_e\mu$. However, if, for example, the evolutionary rate is expressed in mutations per site per year, then the branches in the tree will be in units of years. Furthermore the population size parameter of the demographic model will then be equal to $N_e\tau$, where $N_e$ is the ***effective population size*** and $\tau$ is the generation length in years. Finally, if the evolutionary rate is expressed in units of mutations per site per generation then the resulting tree will be in units of generations and the population parameter of the demographic model will be in natural units (i.e. will be equal to the effective number of reproducing individuals, $N_e$).

### Tree Priors

When sequence data has been collected from a homogenous population, various coalescent [29, 28] models of demographic history can be used in `BEAST` to model population size changes through time. At present the simple parametric models available include constant size $N(t) = N_e$ (1 parameter), exponential growth $N(t) = N_e e^{-gt}$ (2 parameters), expansion or logistic growth (3 parameters).

In addition, the highly parametric Bayesian skyline plot [30] is also available, but this model can only be used when the data are strongly informative about population history. All of these demographic models are parametric priors on the ages of nodes in the tree, in which the hyperparameters (e.g., population size, $N_e$, and growth rate, $g$) can be sampled and estimated. As well as performing single locus coalescent-based inference, two or more unlinked gene trees can be simultaneously analyzed under the same demographic model. Sophisticated multi-locus coalescent inference can be achieved by allocating a separate overall rate and substitution process for each locus, thereby accommodating loci with heterogeneous evolutionary processes.

At present there are only a limited number of options for non-coalescent priors on tree shape and branching rate. Currently a simple Yule prior on birth rate of new lineages (1 parameter) can be employed. However, generalized birth-death tree priors are currently under development.

In addition to general models of branching times such as the coalescent and Yule priors, the tree prior may also include specific distributions and/or constraints on certain node heights and topological features. These additional priors may represent other sources of knowledge such as expert interpretation of the fossil record. For example, as briefly noted above, each node in the tree can have a prior distribution representing knowledge of its date. A recent paper on "relaxed phylogenetics" contains more information on calibration priors [32].

### Multiple data partitions and linking and unlinking parameters

`BEAST` provides the ability to analyze multiple data partitions simultaneously. This is useful when combining multiple genes in a single multi-locus coalescent analysis (e.g. [34]) or to allocate different evolutionary processes to different regions of a sequence alignment, such as the codon positions; e.g. [7]). The parameters of the substitution model, the rate model among sites, the rate model among branches, the tree, and the tree prior can all be 'linked' or 'unlinked' in an analysis involving multiple partitions. For example in an analysis of HIV-1 group O by Lemey *et al* [34], three loci (*gag*, *int*, *env*) were assumed to share the same substitution model parameters (GTR), as well as sharing the same demographic history of exponential growth. However they were assumed to have different shape parameters for Γ-distributed rate heterogeneity among sites, different rate parameters for the strict molecular clock and the three tree topologies and sets of divergence times were also assumed to be independent and unlinked.

### Definitions and units of the standard parameters and variables

Crucial to the interpretation of all BEAST parameters is an understanding of the units that the tree is measured in. The simplest situation occurs when no calibration information is available, either from knowledge of the rate of evolution of the gene region, or from knowledge of the age of any of the nodes in the tree. If this is the case the rate of evolution is set to 1.0 (via the clock.rate or ucld.mean parameters) and the branch lengths in the tree are then in substitutions per site. However if the rate of evolution is known in substitutions per site per unit time, then the genealogy will be expressed in the relevant time units. Likewise, if the age of one or more nodes (internal or external) are known then this will also provide the units for the rest of the branch lengths and the rate of evolution. With this in mind, the following table lists the parameters that are used in the models that can be generated by BEAUti, with their interpretation and units.

- `clock.rate` - The rate of the strict molecular clock. This parameter only appears when you have selected the strict molecular clock in the model panel. The units of this parameter are in substitutions per site per unit time. If this parameter is fixed to 1.0 then the branch lengths in the tree will be in units of substitutions per site. However, if, for example, the tree is being calibrated by using fossil calibrations on internal nodes and those fossil dates are expressed in millions of years ago (Mya), then the `clock.rate` parameter will be an estimate of the evolutionary rate in units of substitutions per site per million years (Myr).

- `constant.popSize` - This is the coalescent parameter under the assumption of a constant population size. This parameter only appears if you select a constant size coalescent tree prior. This parameter represents the product of effective population size ($N_e$) and the generation length in units of time ($\tau$). If time is measured in generations this parameter a direct estimate of $N_e$. Otherwise it is a composite parameter and an estimate of $N_e$ can be computed from this parameter by dividing it by the generation length in the units of time that your calibrations (or `clock.rate`) are defined in. Finally, if `clock.rate` is set to 1.0 then `constant.popSize` is an estimate of $N_e\mu$ for haploid data such as mitochondrial sequences and $2N_e\mu$ for ***diploid*** data, where $\mu$ is the substitution rate per site per generation.

- `covariance` - If this value is significantly positive, it means that within your phylogeny, branches with fast rates are followed by branches with fast rates. This statistic measures the covariance between parent and child branch rates in your tree in a relaxed molecular clock analysis. If this value spans zero, then branches with fast rates and slow rates are next to each other. It also means that there is no strong evidence of autocorrelation of rates in the phylogeny.

- `exponential.growthRate` - This is the coalescent parameter representing the rate of growth of the population assuming exponential growth. The population size at time $t$ is determined by $N(t) = N_e \exp(-gt)$ where $t$ is in the same units as the branch lengths and $g$ is the `exponential.growthRate` parameter. This parameter only appears if you have selected a exponential growth coalescent tree prior.

- `exponential.popSize` - This is the parameter representing the modern day population size assuming exponential growth. Like `constant.popSize`, it is a composite parameter unless the time scale of the genealogy is in generations. This parameter only appears if you have selected a exponential growth coalescent tree prior.

- `gtr.{ac,ag,at,cg,gt}` - These five parameters are the relative rates of substitutions for $A{\leftrightarrow}C$, $A{\leftrightarrow}G$, $A{\leftrightarrow}T$, $C{\leftrightarrow}G$ and $G{\leftrightarrow}T$ in the general time-reversible model of nucleotide substitution [23]. In the default set up these parameters are relative to $r_{C\leftrightarrow T} = 1.0$. These parameters only appear if you have selected the GTR substitution model.

- `hky.kappa` - This parameter is the transition/transversion ratio ($\kappa$) parameter of the HKY85 model of nucleotide substitution [24]. This parameter only appears if you have selected the HKY substitution model.

- `siteModel.alpha` - This parameter is the shape ($\alpha$) parameter of the $\Gamma$ distribution of rate heterogeneity among sites [26]. This parameter only appears when you have selected Gamma or Gamma+Invariant Sites in the site heterogeneity model.

- `siteModel.pInv` - This parameter is the proportion of invariant sites ($p_{inv}$) and has a range between 0 and 1. This parameter only appears when you have selected "Invariant sites" or "Gamma+Invariant Sites" in the site heterogeneity model. The starting value must be less than 1.0.

- `treeModel.rootHeight` - This parameter represents the total height of the tree (often known as the $t_{MRCA}$). The units of this parameter are the same as the units for the branch lengths in the tree.

- `ucld.mean` - This is the mean molecular clock rate under the uncorrelated lognormal relaxed molecular clock. This parameter can be in "real" space or in log space depending on the `BEAST` XML. However,under default `BEAUti` options for the uncorrelated log-normal relaxed clock this parameter has the same units as `clock.rate`.

- `ucld.stdev` -This is the standard deviation ($\sigma$) of the uncorrelated lognormal relaxed clock (in log-space). If this parameter is 0 there is no variation in rates among branches. If this parameter is greater than 1 then the standard deviation in branch rates is greater than the mean rate. This is also the case for the coefficient of variation. When viewed in `Tracer`, if the coefficient of variation frequency histogram is abutting against zero, then your data can't reject a strict molecular clock. If the frequency histogram is not abutting against zero then there is among branch rate heterogeneity within your data, and we recommend the use of a relaxed molecular clock.



- `yule.birthRate` - This parameter is the rate of lineage birth in the Yule model of speciation. If `clock.rate` is 1.0 then this parameter estimates the number of lineages born from a parent lineage per substitution per site. If the tree is instead measured in, for example, years, then this parameter would be the number of new lineages born from a single parent lineage per year.

- `tmrca(`*taxon group*`)` - This is the parameter for the $t_{MRCA}$ of the specified taxa subset. The units of this variable are the same as the units for the branch lengths in the tree and will depend on the calibration information for the rate and/or dates of calibrated nodes. Setting priors on these parameters and/or `treeModel.rootHeight` parameter will act as calibration information.

**Model comparison**

Considering the large number of models available in a Bayesian inference package like BEAST, a common question is "Which model should I use?". This is especially the case for the parts of the evolutionary model that are not of direct interest to the researcher (and responsible for the so-called nuisance parameters). If the research question is a question of demographic inference then the researcher may not be interested in the substitution model parameters, but nevertheless some substitution model must be chosen. It is in these situations that it often makes sense to chose the substitution model which best fits the data.

In a Bayesian setting, the most theoretically sound method of determining which of two models is better is to calculate the ***Bayes Factor (BF)***, which is the ratio of their marginal likelihoods. Generally speaking calculating the BF involves a Bayesian MCMC that averages over both models (using a technique called ***reversible jump MCMC***), and this is not something that can currently be done in BEAST. However there are a couple of ways of approximately calculating the marginal likelihood of each model (and therefore the Bayes factor between them) that can be achieved by processing the output of two BEAST analyses. For exampled, a simple method first described by Newton and Raftery (1994) computes the Bayes factor via importance sampling (with the posterior as the importance distribution). With this importance distribution it turns out that the harmonic mean of the sampled likelihoods is an estimator of the marginal likelihood. So by calculating the harmonic mean of the likelihood from the posterior output of each of the models and then taking the difference (in log space) you get the log BF and you can look up this number in a table to decide if the BF is large enough to strongly favour the better model. This method of calculating the BF is only approximate and in certain situations it is not very stable, so model comparison is an area of Bayesian evolutionary analysis that could certainly be improved.

### 18.1.4 Conclusions

BEAST is a flexible analysis package for evolutionary parameter estimation and hypothesis testing. The component-based nature of model specification in BEAST means that the number of different evolutionary models possible is very large and therefore difficult to summarize. However a number of published uses of the BEAST software already serve to highlight the breadth of application the software enjoys [7, 34, 22, 30, 10].

BEAST is an actively developed package and enhancements for the next version include (1) birth-death priors for tree shape (2) faster and more flexible codon-based substitution models (3) the structured coalescent to model subdivided populations with migration (4) models of continuous character evolution and (5) new relaxed clock models based on random local molecular clocks.

## 18.2 Practice

### 18.2.1 The BEAST software package

This chapter provides a step-by-step tutorial for analysing a set of virus sequences which have been isolated at different points in time (heterochronous data). The data are 35 sequences from the $G$ (attachment protein) gene of human respiratory syncytial virus subgroup A (RSVA) from various parts of the world with isolation dates ranging from 1956-2002 [35]. The input file required for this exercise is available for download at http://www.thephylogenetichandbook.org. The aim is to obtain an estimate of the rate of molecular evolution, an estimate of the date of the most recent common ancestor and to infer the phylogenetic relationships with appropriate measures of statistical support.

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

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

- BEAST - this package contains the BEAST program, BEAUti and a couple of utility programs. At the time of writing, the current version is v1.4.4. It 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 v1.4. 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 v1.0. It is available for download from http://tree.bio.ed.ac.uk/.

### 18.2.2 Running BEAUti

The exact instructions for running BEAUti differs depending on which computer you are using. Please see the README text file that was distributed with the version you downloaded. Once running, BEAUti will look similar irrespective of which computer system it is running on. For this tutorial, the Mac OS X version will be used in the Figures but the Linux and Windows versions will have exactly the same layout and functionality.

### 18.2.3 Loading the NEXUS file

To load a NEXUS format alignment, simply select the *Import NEXUS...* option from the File menu. The example file, called `RSVA.nex`, is available from http://www.thephylogenetichandbook.org/. This file contains an alignment of 35 sequences from the $G$ gene of RSVA virus, 629 nucleotides in length. Once loaded, the list of taxa and the actual alignment will be displayed in the main window (Figure 18.1).

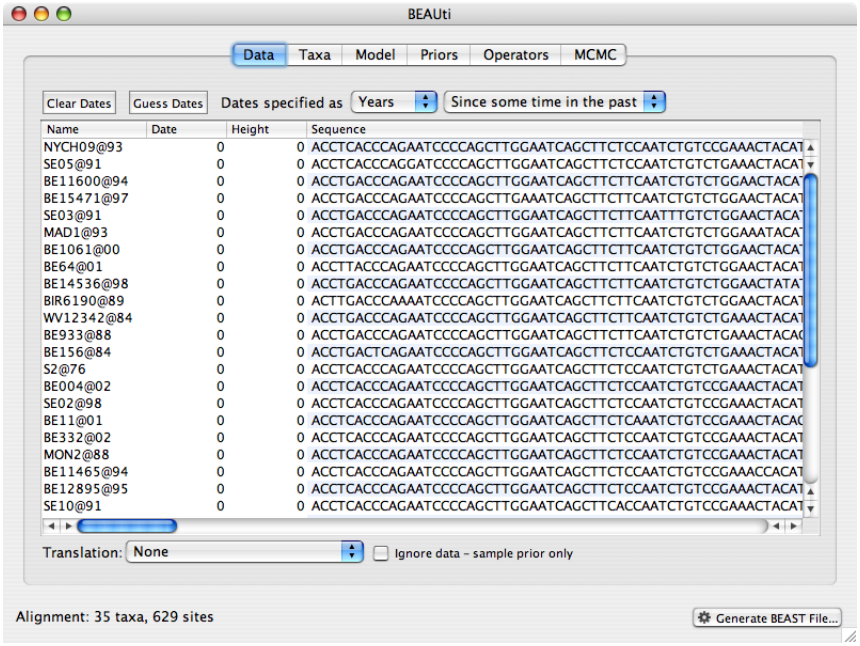


Figure 18.1: The data panel in `BEAUti`

### 18.2.4 Setting the dates of the taxa

If the NEXUS file contains a calibrations block then the dates will automatically be loaded. Otherwise, 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 RSVA sequences have been sampled at various dates going back to the 1950s. The actual year of sampling is given in the name of each taxon and we could simply edit the value in the Date column of the table to reflect these. However, if the taxa names contain the calibration information, then a convenient way to specify the dates of the sequences in `BEAUti` is to use the "Guess Dates" button at the top of the Data panel. Clicking this will make a dialog box appear (Figure 18.2).

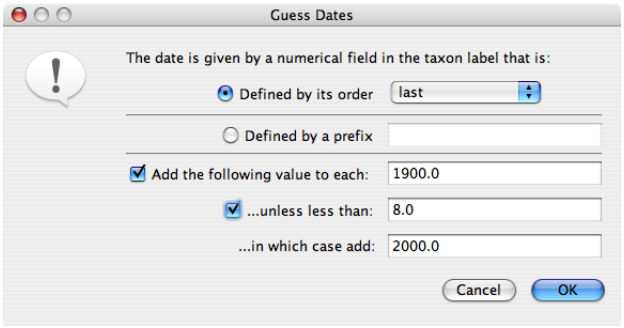


Figure 18.2: The Guess Dates dialog

This operation attempts to guess what the dates are from information contained within the taxon names. It works by trying to find a numerical field within each name. If the taxon names contain more than one numerical field (such as the RSVA sequences, above) then you can specify how to find the one that corresponds to the date of sampling. You can either specify the order that the date field comes (e.g., first, last or various positions in between) or specify a prefix (some characters that come immediately before the date field in each name). For the RSVA sequences you can select 'last' from the drop-down menu for the order or use the prefix option and specify '@' as the prefix ('@' is the prefix used for dates by PAML, see Chapter 11).

In this dialog box, you can also get `BEAUti` to add a fixed value to each guessed date. In this case the value "1900" has been 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". Some of the sequences in the example file actually have dates after the year 2000 so selecting the will option would convert them correctly, adding 2000 to any date less than 08. When you 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 1984) or backwards in time from the present (as in the case of radiocarbon ages).

### Translating the data in amino acid sequences

At the bottom of the main window is the option to translate the data into amino acid sequences. This will be done using the genetic code specified in the associated drop down menu. If the loaded sequence are not nucleotides then this option will be disabled.

### 18.2.5 Setting the evolutionary model

The next thing to do is to click on the Model 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 nucleotides translated into amino acids). Figure 18.3 shows the settings that will appear after loading the RSVA data and selecting a codon partitioning.

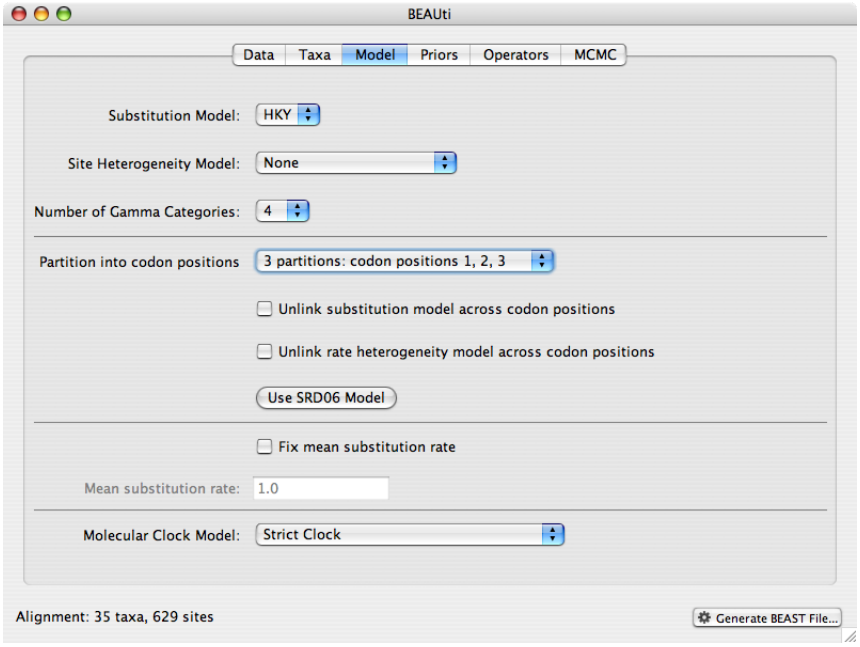


Figure 18.3: The evolutionary model settings in BEAUti

This chapter 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.

- If there are no dates for the sequences (they are contemporaneous) then you can specify a fixed *mean substitution rate* obtained from another source. Setting this to 1.0 will result in the ages of the nodes of the tree being estimated in units of substitutions per site (i.e. the normal units of branch lengths in popular packages such as `MrBayes`).

For this tutorial, select the '3 partitions: codon positions 1, 2 & 3' option so that each codon position has its own rate of evolution.

### 18.2.6 Setting up the operators

Each parameter in the model has one or more"operators" (these are variously called moves and proposals 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. These will be called things like `hky.kappa` which means the HKY model's kappa parameter (the transition-transversion bias). 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 proportion, the random walk operator adds or subtracts an amount to the parameter 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.

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 efficency of the analysis. For most operators (like random walk and subtree slide operators) 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 will be written to standard ouput. These can then be used to set the starting tuning settings in order to minimize the amount of time taken to reach optimum performance in subsequent runs.

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 (this may mean upweighting other operators since the weights must be integers).

### 18.2.7 Setting the MCMC options

The *MCMC* tab in `BEAUti` provides settings to control the MCMC chain (Figure 18.4). 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 be depends on the size of the dataset, the complexity of the model and the precision of the answer required. The default value of 10,000,000 is entirely arbitrary and should be adjusted according to the size of your dataset. We will see later how the resulting log file can be analysed 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 the chain length / 10,000.

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

The final 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 name of the imported NEXUS file but feel free to change these.

### 18.2.8 Running `BEAST`

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. Choose a name for the file (for example, `RSVA.xml`) and save the file. For convenience, leave the `BEAUti` window open so that you can change the values and re-generate the `BEAST` file as required later in this tutorial.

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. The analysis will then 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.

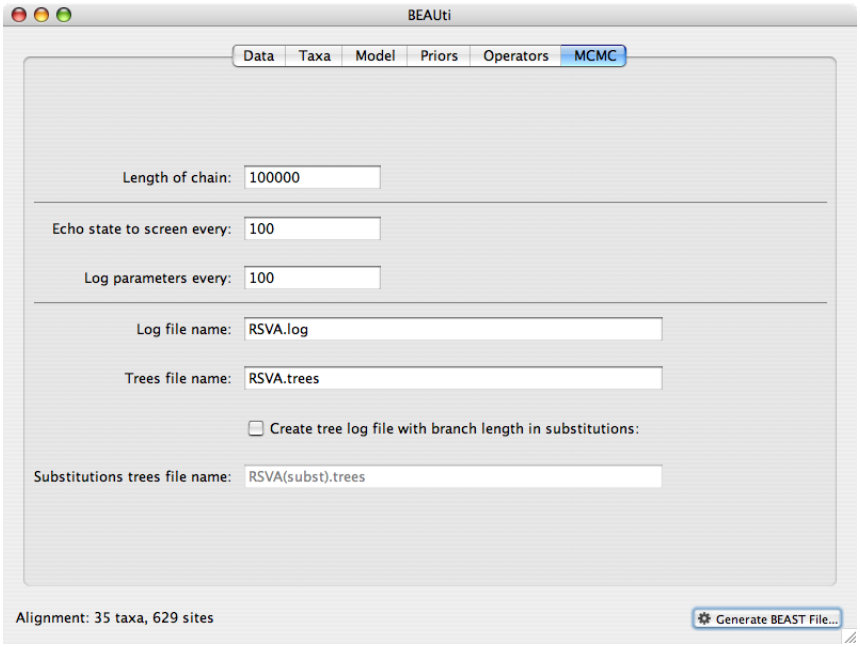


Figure 18.4: The MCMC settings in `BEAUti`

### 18.2.9 Analysing the `BEAST` output

To analyse 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 Open option from the File menu. If you have it available, select the log file that you created in the previous section. The file will load and you will be presented with a window similar to the one below (Figure 18.5). 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 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 (Figure 18.5), 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

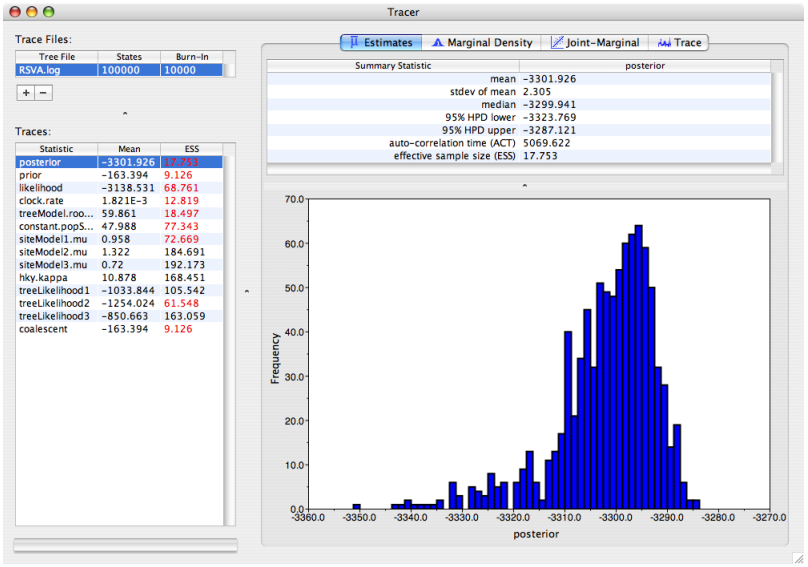


Figure 18.5: The main `Tracer` window with a `BEAST` log file loaded.

below.

- *Mean* - The mean value of the samples (excluding the burn-in).
- *Stdev* - The standard error of the mean. This takes into account the effective sample size so a small ESS will give a large standard error.
- *Median* - The median value of the samples (excluding the burn-in).
- *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`). 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 (Figure 18.5).

If we select the tab on the right-hand-side labelled 'Trace' we can view the raw trace, that is, the sampled values against the step in the MCMC chain (Figure 18.6).

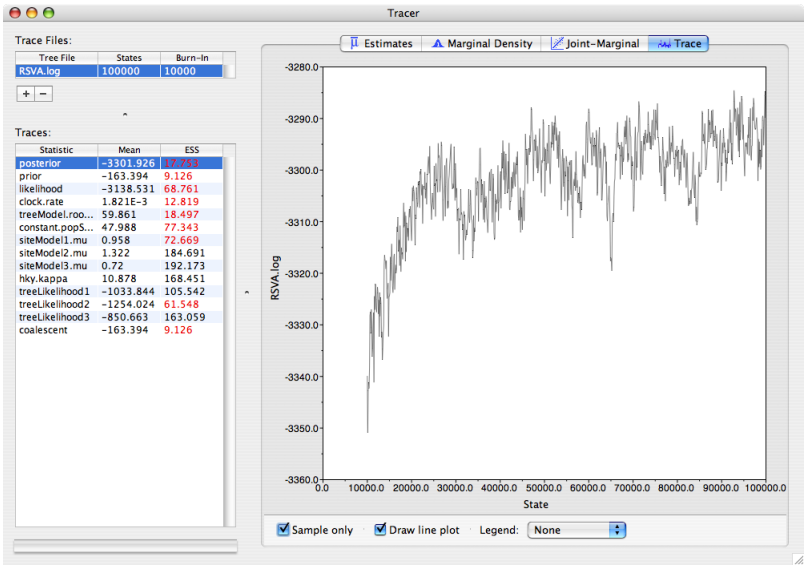


Figure 18.6: The trace of posterior against chain length in `Tracer` for a run of 100,000 steps.

Here you can see how the samples are correlated. There are 1000 samples in the trace (we ran the MCMC for 100,000 steps sampling every 100) but it is clear that adjacent samples often tend to have similar values. The ESS for the age of the root (`treeModel.rootHeight` is about 18 so we are only getting 1 independent sample to every 56 actual samples. It also seems that the default burn-in of 10% of the chain length is inadequate (the posterior values are still increasing over most of the chain). Not excluding enough of the start of the chain as burn-in will bias the results and render estimates of ESS unreliable.

The simple response to this situation is that we need to run the chain for longer. Given the lowest ESS (for the `prior`) is 9, it would suggest that we have to run it at least 12 times longer to get ESSs that are >100. However it would be better to aim higher so lets go for a chain length of 5,000,000. Go back to Section 18.2.7 and create a new `BEAST` XML file with a longer chain length. Now run `BEAST` and load the new log file into `Tracer` (you can leave the old one loaded for comparison). Click on the Trace tab and look at the raw trace plot

(Figure 18.7).

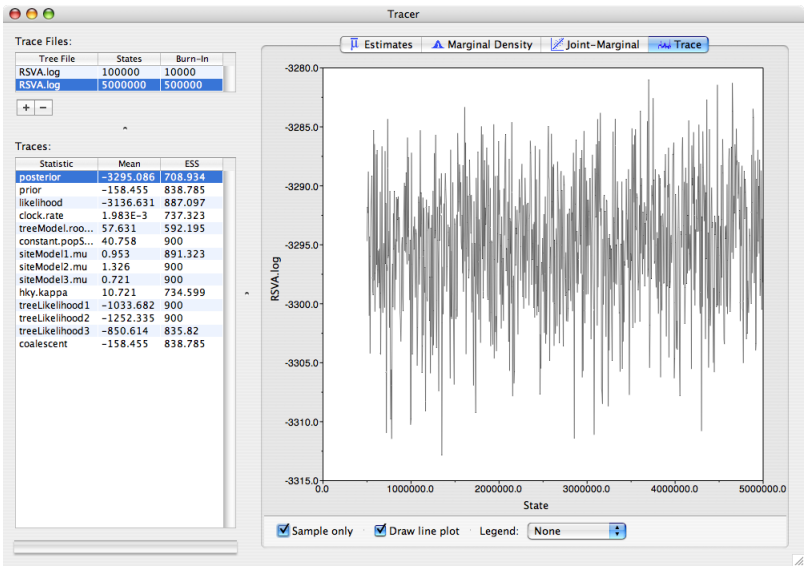


Figure 18.7: The trace of posterior against chain length in `Tracer` for a run of 5,000,000 steps.

Again we have chosen options that produce 1000 samples and with an ESS of about 500 there is still auto-correlation between the samples but 500 effectively independent samples will now provide a good estimate of the posterior distribution. There are no obvious trends in the plot which would suggest that the MCMC has not yet converged, and there are no large-scale fluctuations in the trace which would suggest poor mixing. As we are happy with the behaviour of log-likelihood we can now move on to one of the parameters of interest: substitution rate. Select `clock.rate` in the left-hand table. This is the average substitution rate across all sites in the alignment. Now choose the density plot by selecting the tab labeled Density. This shows a plot of the posterior probability density of this parameter. You should see a plot similar to Figure 18.8.

As you can see the posterior probability density is roughly bell-shaped. There is some sampling noise which would be reduced if we ran the chain for longer but we already have a good estimate of the mean and HPD interval. You can overlay the density plots of multiple traces in order to compare them (it is up to the user to determine whether they are comparable on the the same axis or not). Select the relative substitution rates for all three codon positions in the table to the left (labelled `siteModel1.mu`, `siteModel2.mu` and `siteModel3.mu`. You will now see the posterior probability densities for the relative substitution rate at all three codon positions overlaid (Figure 18.9).

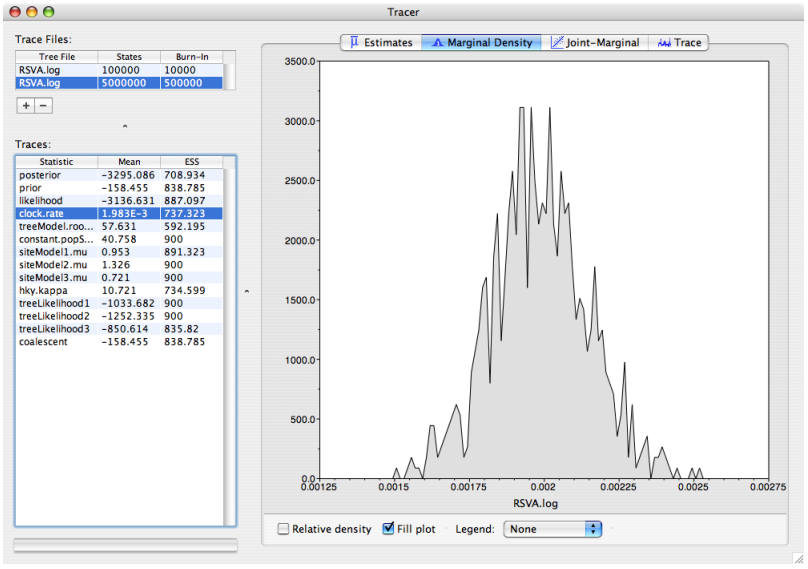


Figure 18.8: The posterior density plot for the subtitution rate.

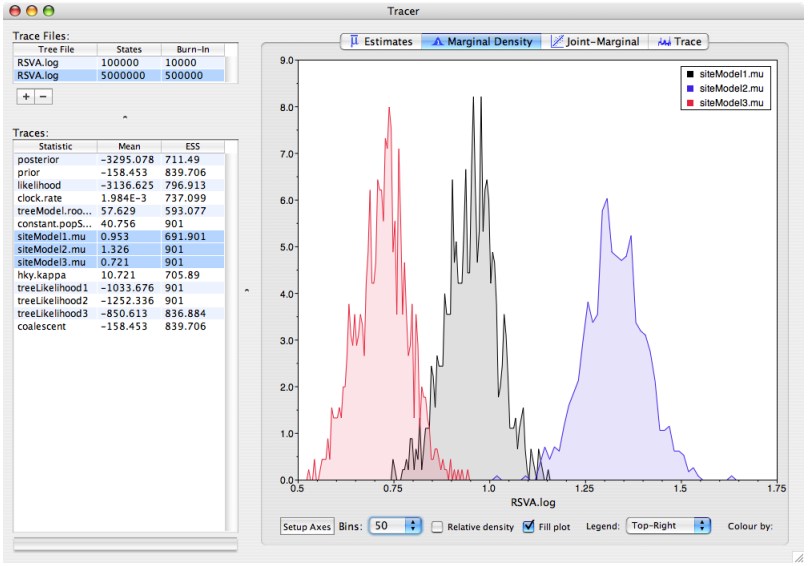


Figure 18.9: The posterior density plots for the relative rate of evolution at each codon position.

### 18.2.10 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 geneologies) at the same time as the other parameters of the model. These are written to a separate file called the 'trees' file. This file is a standard NEXUS format file. As such it can easily be loaded into other software in order to examine the trees it contains. One possibility is to load the trees into a program such as PAUP* and construct a consensus tree in a similar manner to summarizing a set of bootstrap trees. In this case, the support values reported for the resolved nodes in the consensus tree will be the posterior probability of those clades.

In this tutorial, however, we are going to use a tool that is provided as part of the `BEAST` package to summarize the information contained within our sampled trees. The tool is called '`TreeAnnotator`' and once running, you will be presented with a window like the one in Figure 18.10.

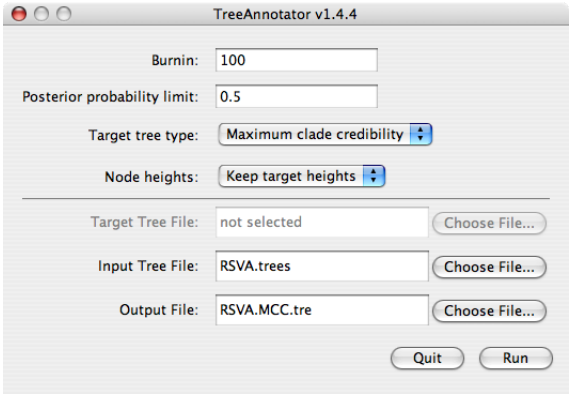


Figure 18.10: The user-interface for the `TreeAnnotator` tool.

`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 trees in the input file that should be excluded from the summarization. This value is given as the number of trees rather than the number of steps in the MCMC chain. Thus for the example above, with a chain of 1,000,000 steps, sampling every 1000 steps, there are 1000 trees in the file. To obtain a 10% burnin, set this value to 100.

- *Posterior probability limit* - This is the minimum posterior probability for a node in order for `TreeAnnotator` to store the annoted information. The default is 0.5 so only nodes with this posterior probability or greater will have information summarized (the equivalent to the nodes in a majority-rule consensus tree). Set this value to 0.0 to summarize all nodes in the target tree.

- *Target tree type* - This has two options "Maximum clade credibility" or "User target tree". For the latter option, a NEXUS tree file can be specified as the Target Tree File, below. For the former option, `TreeAnnotator` will examine every tree in the Input Tree File and select the tree that has the highest 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. The other two options give node heights as an average (Mean or Median) over the sample of trees.

- *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.

Once you have selected all the options, above, press the "Run" button. `TreeAnnotator` will analyse 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.

### 18.2.11 Viewing the annotated tree

Run `FigTree` now and select the *Open...* command from the *File* menu. Select the tree file you created using `TreeAnnotator` in the previous section. The tree will be displayed in the `FigTree` window. On the left hand side of the window are the options and settings which control how the tree is displayed. In this case we want to display the posterior probabilities of each of the clades present in the tree and estimates of the age of each node (see Figure 18.11). In order to do this you need to change some of the settings.

First 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)

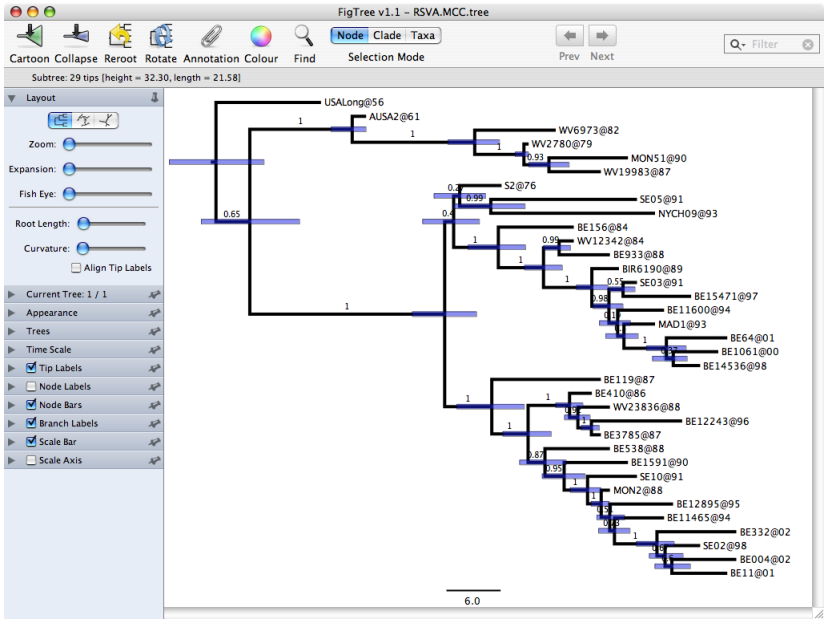


Figure 18.11: The annotated tree displayed in `FigTree`.

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.

Finally, open the *Appearance* panel and alter the *Line Weight* to draw the tree with thicker lines. 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.

### 18.2.12 Conclusion and Resources

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

- The `BEAST` website: http://beast.bio.ed.ac.uk/
- Tutorials: http://beast.bio.ed.ac.uk/Tutorials/
- Frequently asked questions: http://beast.bio.ed.ac.uk/FAQ/

# Bibliography


[1] Huelsenbeck JP, Ronquist F: `MrBayes`: Bayesian inference of phylogenetic trees. *Bioinformatics* 2001, **17**:754-755.

[2] Drummond AJ, Nicholls GK, Rodrigo AG, Solomon W: Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. *Genetics* 2002, **161**(3):1307-1320.

[3] Wilson IJ, Weale ME, Balding DJ: Inferences from DNA data: population histories, evolutionary processes and forensic match probabilities. *J Royal Stat Soc A-Statistics in Society* 2003, **166**:155-188.

[4] Beaumont MA: Detecting population expansion and decline using microsatellites. *Genetics* 1999, **153**(4):2013-2029.

[5] Felsenstein J: Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol* 1981, **17**: 368-376.

[6] Rannala B, Yang ZH: Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. *Genetics* 2003, **164**(4):1645-1656.

[7] Pybus OG, Drummond AJ, Nakano T, Robertson BH, Rambaut A: The epidemiology and iatrogenic transmission of hepatitis C virus in Egypt: a Bayesian coalescent approach. *Mol Biol Evol* 2003, **20**(3):381-387.

[8] Kuhner, MK: `LAMARC` 2.0: Maximum likelihood and Bayesian estimation of population parameters. *Bioinformatics* 2006 **22**(6):768-770.

[9] Redelings BD, Suchard MA: Joint Bayesian Estimation of Alignment and Phylogeny. *Syst Biol* 2005, **54**(3):401-418.

[10] Lunter G, Miklos I, Drummond A, Jensen JL, Hein J: Bayesian coestimation of phylogeny and sequence alignment. *BMC Bioinformatics* 2005, **6**(1):83.

[11] Hastings WK: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 1970, **57**:97-109.

[12] Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E: Equations of state calculations by fast computing machines. *J Chem Phys* 1953, **21**:1087-1091.

[13] Zuckerkandl E, Pauling L: Evolutionary divergence and convergence in proteins. In: *Evolving genes and proteins*. Edited by Bryson V, Vogel HJ. Academic Press: New York; 1965: 97-166.

[14] Aris-Brosou S, Yang Z: Bayesian models of episodic evolution support a late Precambrian explosive diversification of the Metazoa. *Mol Biol Evol* 2003, **20**(12):1947-1954.

[15] Kishino H, Thorne JL, Bruno WJ: Performance of a divergence time estimation method under a probabilistic model of rate evolution. Molecular Biology & Evolution 2001, 18:352-361.

[16] Sanderson MJ: Estimating absolute rates of molecular evolution and divergence times: A penalized likelihood approach. *Mol Biol Evol* 2002, **19**:101-109.

[17] Thorne JL, Kishino H: Divergence time and evolutionary rate estimation with multilocus data. *Syst Biol* 2002, **51**(5):689-702.

[18] Thorne JL, Kishino H, Painter IS: Estimating the rate of evolution of the rate of molecular evolution. *Mol Biol Evol* 1998, **15**:1647-1657.

[19] Yoder AD, Yang ZH: Estimation of primate speciation dates using local molecular clocks. *Mol Biol Evol* 2000, **17**:1081-1090.

[20] Suchard MA, Redelings BD: `Bali-Phy`: simultaneous Bayesian inference of alignment and phylogeny. *Bioinformatics* 2006 **22**(16):2047-2048.

[21] Rambaut A, Drummond AJ: `Tracer` [computer program] Available from http://evolve.zoo.ox.ac.uk/software/ 2003

[22] Shapiro B, Drummond AJ, Rambaut A, Wilson MC, Matheus PE, Sher AV, Pybus OG, Gilbert MT, Barnes I, Binladen J et al: Rise and fall of the Beringian steppe bison. *Science* 2004, **306**(5701):1561-1565.

[23] Rodriguez F, Oliver JL, Marin A, Medina JR: The general stochastic model of nucleotide substitution. *J Theor Biol* 1990, **142**(4):485-501.

[24] Hasegawa M, Kishino H, Yano T: Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol* 1985, **22**(2):160-174.

[25] Goldman N, Yang Z: A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Mol Biol Evol* 1994, **11**(5):725-736.

[26] Yang Z: Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *J Mol Evol* 1994, **39**(3):306-314.

[27] Drummond AJ, Pybus OG, Rambaut A, Forsberg R, Rodrigo AG: Measurably evolving populations. *Trends Ecol Evol* 2003, **18**(9):481-488.

[28] Griffiths RC, Tavare S: Sampling theory for neutral alleles in a varying environment. *Philos Trans R Soc Lond B Biol Sci* 1994, **344**(1310):403-410.

[29] Kingman JFC: The coalescent. *Stochastic Processes and Their Applications* 1982, **13**:235-248.

[30] Drummond AJ, Rambaut A, Shapiro B, Pybus OG: Bayesian coalescent inference of past population dynamics from molecular sequences. *Mol Biol Evol* 2005, **22**(5):1185-1192.

[31] Aldous DJ: Stochastic models and descriptive statistics for phylogenetic trees, from Yule to today. *Statistical Science* 2001, **16**(1):23-34.

[32] Drummond AJ, Ho SYW, Phillips MJ, Rambaut A: Relaxed phylogenetics and dating with confidence. *PLoS Biology* 2006, **4**(5)

[33] Thorner JL, Kishino H, Felsenstein J: An evolutionary model for maximum likelihood alignment of DNA sequences. *J Mol Evol* 1991, **33**(2): 114-124.

[34] Lemey P, Pybus OG, Rambaut A, Drummond AJ, Robertson DL, Roques P, Worobey M, Vandamme AM: The molecular population genetics of HIV-1 group O. *Genetics* 2004, **167**(3):1059-1068.

[35] Zlateva KT, Lemey P, Vandamme AM, Van Ranst M: Molecular evolution and circulation patterns of human respiratory syncytial virus subgroup a: positively selected sites in the attachment g glycoprotein. *J Virol* 2004, **78**(9): 4675-4683.

[36] Lanciotti RS, Gubler DJ, Trent DW: Molecular evolution and phylogeny of dengue-4 viruses. *Journal of General Virology* 1997, **78**:2279-2284.

[37] Rambaut A: Estimating the rate of molecular evolution: incorporating non-contemporaneous sequences into maximum likelihood phylogenies. *Bioinformatics* 2000, **16**(4):395-399.