Phylogeny Simulation and Diversification Rate Analysis with TESS

Sebastian Höhna, Michael R. May and Brian R. Moore

October 27, 2015

This tutorial describes statistical approaches for inferring rates of lineage diversification (speciation – extinction) from empirical (*i.e.,* estimated) phylogenetic trees—and for generating simulated phylogenetic trees—under various stochastic-branching process models using the R package TESS. TESS provides a flexible framework for specifying diversification models—where diversification rates are constant, vary continuously, or change episodically through time (including explicit models of mass-extinction events)—and implements numerical methods to estimate parameters of these models from estimated phylogenies. A major feature of TESS is the ability to include various methods of incomplete taxon sampling. Additionally, we provide robust Bayesian methods for assessing the *relative* fit of these models of lineage diversification to a given study tree—*e.g.,* where stepping-stone simulation is used to estimate the marginal likelihoods of competing models, which can then be compared using Bayes factors. We also provide Bayesian methods for evaluating the *absolute* fit of these branching-process models to a given study tree—*i.e.,* where posterior-predictive simulation is used to assess the ability of a candidate model to generate the observed phylogenetic data. Finally, we show how TESS can be used to efficiently simulate phylogenies, and how these simulations can provide invaluable null hypotheses.

Contents

1 Getting Started **4**
- 1.1 Types of research questions involving diversification rates 4
- 1.2 Scope of research questions addressed by TESS 10
- 1.3 Empirical data 11

2 Models **14**
- 2.1 The birth-death branching process 14
- 2.2 The space of birth-death branching-process models 16
- 2.3 Simulating data 19
- 2.4 Estimating parameters using Markov chain Monte Carlo (MCMC) . 23
  - 2.4.1 Birth-death processes with constant rates 23
  - 2.4.2 Birth-death processes with continuously varying rates 26
  - 2.4.3 Birth-death processes with episodically varying rates 29
  - 2.4.4 Birth-death processes with explicit mass-extinction events . 33

3 Accommodating Incomplete Taxon Sampling **38**
- 3.1 Patterns of incomplete sampling 38
- 3.2 Uniform taxon sampling 41
- 3.3 Diversified taxon sampling 44

4 Model Evaluation **48**
- 4.1 Comparing models with Bayes factors 49
  - 4.1.1 Stepping-stone simulation 50
  - 4.1.2 Estimating marginal likelihoods of birth-death models 52
- 4.2 Assessing model adequacy with posterior predictive simulation . . . 54
  - 4.2.1 Posterior-predictive simulation 54
  - 4.2.2 Assessing the adequacy of branching-process models 55
- 4.3 Model selection with reversible-jump MCMC and CoMET 60
  - 4.3.1 Specifying hyperpiors *a priori* 60
  - 4.3.2 Empirical hyperpiors 67
  - 4.3.3 Without diversification-rate shifts 70
  - 4.3.4 Without mass-extinction events 72

5 MCMC Diagnosis, auto-tuning and auto-stopping **74**
- 5.1 MCMC diagnosis for a constant-rate birth-death model 74
- 5.2 MCMC diagnosis for the CoMET model 77
  - 5.2.1 Single-chain diagnostics 77
  - 5.2.2 Multiple-chain diagnostics 81
- 5.3 Auto-tuning MCMC proposals 82

5.4 Auto-stopping MCMC simulations 89

1 Getting Started

We assume that the reader has some experience using R and has installed the TESS package (including all dependent packages, such as ape and coda). We also assume some familiarity with Bayesian inference and models of lineage diversification. Nevertheless, we intend this guide to be relatively self-contained: we provide brief explanations of the methods and models in the corresponding tutorials, and direct the reader to the relevant primary literature for more detailed descriptions of the corresponding topics.
We originally developed TESS as a tool for efficiently simulating phylogenies in order to test and validate new inference methods and models (Höhna, 2013). However, TESS has since evolved to include several methods for estimating diversification rates from empirical phylogenies (*e.g.,* Höhna, 2014; May et al., 2015). This is a natural extension, as both simulation and inference methods are based on the same equations and underlying theory.

1.1 Types of research questions involving diversification rates

Many evolutionary phenomena entail differential rates of diversification (speciation – extinction); *e.g.*, adaptive radiation, diversity-dependent diversification, key innovations, and mass extinction. The innumerable specific study questions regarding lineage diversification may be classified within five fundamental categories of inference problems. Admittedly, this classification scheme is somewhat arbitrary, but it is nevertheless useful, as it allows users to navigate the ever-increasing number of available phylogenetic methods. Below, we describe each of the fundamental questions regarding diversification rates and provide a few examples of software packages that are available to address them.

(1) Rate estimation *What is the (constant) rate of diversification in my study group?* Methods have been developed to estimate parameters of the stochastic-branching process (*i.e.*, rates of speciation and extinction, or composite parameters such as net-diversification and relative-extinction rates) under the assumption that rates have remained constant across lineages and through time; *i.e.*, under a constant-rate birth-death stochastic-branching process model. Statistical phylogenetic methods developed *specifically* to estimate diversification-rate parameters include:

- `DivBayes` is a program to estimate the net-diversification rate (speciation–extinction rate) and the relative-extinction rate (speciation ÷ extinction

rate) given the estimated stem age of a group and the number of extant species that belong to it (Ryberg et al., 2011).

- `SubT` is a program for estimating the net-diversification rate and the relative-extinction rate using the method described by Bokma (2008), which allows inclusion of unsampled species. This is a nice feature, as `SubT` models unsampled species explicitly—rather than assuming some artificial sampling scheme—by virtue of estimating the divergence times of the missing taxa.

Both `DivBayes` and `SubT` are implemented in a Bayesian statistical framework, and therefore estimate posterior probability distributions for the rate parameters of interest, which provides a natural means of accommodating uncertainty in these parameter estimates. Of course, rate parameters may also be estimated by many other methods that relax the assumption that rates of diversification are constant across lineages or constant through time (*i.e.*, the constant-rate birth-death branching model is a special case of these more general rate-variable branching process models, which we describe below).
Similarly, diversification-rate parameters are also included as nuisance parameters of other phylogenetic models—*i.e.*, where these diversification-rate parameters are not of direct interest. For example, many methods for estimating species divergence times—such as `BEAST` (Drummond et al., 2012), `MrBayes` (Ronquist et al., 2012), and `RevBayes` (Höhna et al., 2015)—implement 'relaxed-clock models' that include a constant-rate birth-death branching process as a prior model on the distribution of tree topologies and node ages (*c.f.*, **?**). Although the parameters of these 'tree priors' are not typically of direct interest, they are nevertheless estimated as part of the joint posterior probability distribution of the relaxed-clock model, and so can be estimated simply by querying the corresponding marginal posterior probability densities. In fact, this may provide more robust estimates of the diversification-rate parameters, as they accommodate uncertainty in the other phylogenetic-model parameters (including the tree topology, divergence-time estimates, and the other relaxed-clock model parameters).

(2) Detecting diversification-rate variation across branches *Is there evidence that diversification rates have varied significantly across the branches of my study group?* Methods have been developed to detect departures from rate constancy across lineages; these tests are analogous to methods that test for departures from a molecular clock—i.e., to assess whether *substitution* rates vary significantly across lineages. Like molecular-clock tests, these diversification-rate methods only indicate whether *diversification* rates vary significantly across lineages, but they do not identify the location(s) of any rate shifts. These methods are important for

assessing whether a given tree violates the assumptions of other inference methods. For example, statistical phylogenetic methods that detect diversification-rate variation through time (see below) typically assume that rates are constant across branches at every instant in time (even though they may vary through time). Several methods are available to detect diversification-rate variation across lineage.

- `SymmeTREE` (Chan and Moore, 2005) implements a number of so-called 'whole-tree' indices that summarize the shape of a tree as a number (Chan and Moore, 2002), and uses these indices as test statistics to identify significant diversification-rate variation across lineages using Monte Carlo simulation (Moore et al., 2004).

- `apTreeshape` (Bortolussi et al., 2006) is an R package that implements two of the seven 'whole-tree' indices implemented in `SymmeTREE`, and implements the Monte Carlo simulation method of Moore et al. (2004) to provide tests for significant diversification-rate variation across lineages.

- `TreeStat` (developed by Andrew Rambaut) also implements several 'whole-tree' indices to measure tree shape, but does not provide tests for significant diversification-rate variation across lineages.

(3) Detecting diversification-shifts along branches There are two distinct questions that fall under this general inference category, depending upon whether we are testing an *a priori* hypothesis about the predicted location(s) of diversification-rate shifts in our study tree, or if we are instead agnostically surveying our study tree for possible location(s) of significant diversification-rate shifts across lineages. The first type of question asks: *Was there a significant diversification-rate shift along a specified branch in my study group?* Several statistical phylogenetic methods have been developed to detect significant diversification-rate shifts along pre-specified branches of the tree.

- `r8s` (Sanderson, 2003) implements a maximum-likelihood approach (Magallón and Sanderson, 2001) to identify pre-specified lineages that have diversified at anomalous (either significantly elevated or decreased) diversification rates.

- `BayesRate` (Silvestro et al., 2011) can identify significant diversification-rate shift across branches by estimating the marginal likelihood (using robust thermodynamic integration methods) of a study tree that has been partitioned into one or more rate categories, and then selecting the partition scheme (diversification-rate model) that provides the best fit to the data

using Bayes factors. `BayesRate` implements constant and exponentially decaying birth-death models, and can accommodate phylogenetic uncertainty by averaging inferences over a sample of trees.

The second type of question asks: *Have there been significant diversification-rate shifts along branches in my study group, and if so, how many shifts and along which branches?* Several statistical phylogenetic methods have been developed to detect significant diversification-rate shifts along pre-specified branches of the tree.

- `SymmeTREE` (Chan and Moore, 2005) implements a maximum-likelihood approach, in which various 'shift statistics' compute the probability of a diversification-rate shift along each internal node, and uses Monte Carlo simulation to assess their significance (Moore et al., 2004).

- `MEDUSA` (Alfaro et al., 2009) implements a maximum-likelihood approach in which birth-death models of increasing complexity (with 0, 1, 2 . . . diversification-rate shifts) are first fit to the tree, and then AIC is used to select the preferred diversification-rate model.

- `BAMM` (Rabosky, 2014) implements a compound Poisson process model in a Bayesian framework to provide estimates of the number and location of diversification-rate shifts across the branches of a tree, and also infers the diversification-rate parameters (speciation, extinction, and diversity-dependence) on each branch of the tree.

(4) Detecting diversification-rate correlates *Are diversification rates correlated with some variable in my study group?* Several methods have been developed to identify overall correlations between diversification rates and organismal features (binary and multi-state discrete morphological traits, continuous morphological traits, geographic range, etc.), many of which are implemented in the excellent `DiversiTree` package (FitzJohn, 2012). These methods include the following:

- `BiSSE` (Binary State Speciation and Extinction; Maddison et al., 2007) models the evolution of a binary trait—with parameters q_{01} and q_{10} that specify the instantaneous rates of change between the two states, 0 and 1—where the rate of lineage diversification depends on the current state. When a lineage is in state 0, the stochastic-branching process has rate parameters $\phi_0 = \{\lambda_0, \mu_0\}$, and when it is in state 1, the process has rate parameters $\phi_1 = \{\lambda_1, \mu_1\}$. The `BiSSE` model is implemented in both maximum likelihood and Bayesian frameworks, although in practice, selection among candidate state-specific models is typically performed in a maximum-likelihood framework using likelihood-ratio tests.

- `MuSSE` (Multiple State Speciation and Extinction; FitzJohn et al., 2009) extends the `BiSSE` model to identify correlations for multi-state discrete states.

- `GeoSSE` (Geographic State Speciation and Extinction; Goldberg et al., 2011) extends the `BiSSE` model to identify correlations between diversification rates among a set of discrete geographic areas.

- `BiSSENESS` (`BiSSE`-Node Enhanced State Shift; Magnuson-Ford and Otto, 2012) extends the `BiSSE` model to identify correlations between diversification rates and a discrete binary traits, while also assessing whether the trait evolves gradually or episodically.

- `QuaSSE` (Quantitative State Speciation and Extinction; FitzJohn, 2010) is the continuous-trait analogue of the `BiSSE` model.

(5) Detecting diversification-rate shifts through time There are several distinct and common types of questions that fall under this general inference category. First, we might ask whether there is evidence of an episodic, tree-wide increase in diversification rates (associated with a sudden increase in speciation rate and/or decrease in extinction rate), as might occur during an episode of adaptive radiation. A second question asks whether there is evidence of a continuous/gradual decrease in diversification rates through time (associated with decreasing speciation rates and/or increasing extinction rates), as might occur because of diversity-dependent diversification (*i.e.*, where competitive ecological interactions among the species of a growing tree decrease the opportunities for speciation and/or increase the probability of extinction). A final question in this category asks whether our study tree was impacted by a mass-extinction event (where a large fraction of the standing species diversity is suddenly lost). This is the category of methods to which `TESS` belongs, which is shared with several other methods for detecting tree-wide variation in diversification rates.

- `DDD` (Diversity-Dependent Diversification; Etienne et al., 2012) implements an explicit model of diversity-dependent diversification in a maximum-likelihood framework. This flexible model allows diversity-dependent diversification where: (1) speciation rate is a function of time; (2) extinction rate is a function of time; (3) speciation and extinction rates are both functions of time; (4) speciation rate is a function of species diversity; (5) extinction rate is a function of species diversity, and; (6) speciation and extinction rates are both functions of species diversity. Additionally, `DDD` can identify the effect of events (tree-wide 'key innovations') that alter the carrying capacity of the tree (Etienne and Haegeman, 2012), and can accommodate incomplete species sampling under the assumption of uniform species sampling.

Finally, `DDD` can be used to simulate trees under various diversity-dependent stochastic-branching processes. This method is primarily used to address two research questions:

  - *Does the study tree exhibit diversity-dependent diversification?*
  - *Have key innovations changed the carrying capacity in the study tree?*

- `RPANDA` (Phylogenetic ANalyses of DiversificAtion; Morlon et al., prep) implements the time-dependent birth-death process described in Morlon et al. (2010) and Morlon et al. (2011). `RPANDA` estimates maximum likelihood parameter values for any time-dependent speciation and extinction rate function, providing specification of a virtually infinite number of diversification rate through time functions. Missing species are modeled by uniform taxon sampling. This method is primarily used to address two research questions:

  - *Are diversification rates constant through time?*
  - *How have diversification rates changed through time?*

- `TreePar` is the inference counterpart to the simulation package, `TreeSim` (described below), and enables maximum-likelihood estimation under a wide range of diversification models, including constant-rate, episodic, or continuously varying birth-death branching process models. Under the episodic model, tree-wide diversification rates may change instantaneously at a shift event, but are constant between those shift events (Stadler, 2011a). A special case of the episodic model is an explicit mass-extinction model, in which a large fraction of the standing species diversification is lost when an event occurs. These (piecewise) constant-rate models are complemented by continuously varying rate models, including diversity-dependent diversification rates (where the net-diversification rate is a function of the species diversity; Etienne et al., 2012; Leventhal et al., 2014), and age-dependent models (where the extinction rate is a function of species age; Lambert et al., 2014). Incomplete species sampling is incorporated by uniform species sampling or using information based on the diversity of more inclusive taxonomic groups. An intriguing feature of `TreePar` (and `TreeSim` and `expoTree`, see below) is the ability to include serially samples species (which are common for viral datasets) to enable inference of diversification dynamics from non-ultrametric trees (Stadler, 2010). This method is primarily used to address three research questions:

  - *Are diversification rates constant through time?*
  - *How have diversification rates changed through time?*

– *Is there evidence that my study tree experienced mass extinction?*

- `TreeSim` enables flexible simulation of reconstructed and complete phylogenetic trees under constant or episodic birth-death bracing process models. Trees can be simulated for a specified time interval (duration) or to a species species diversity (tree size) (Stadler, 2011b). Additionally, tips can be sampled sequentially through time (*e.g.,* sampling fossil taxa).

- `expoTree` (Leventhal et al., 2014) estimates maximum likelihood parameter values for diversity-dependent diversification branching process, and accommodates missing species under the assumption of uniform taxon sampling. Like `TreePar`, `expoTree` accommodates sequential species sampling, which makes it well suited for the study of epidemiological inference problems. This method is primarily used to address the following research question:

  – *Does the study tree exhibit diversity-dependent diversification?*

1.2 Scope of research questions addressed by TESS

There are three fundamental questions that can be addressed using TESS:

1. *What are the rates of the process that gave rise to my study tree?*
2. *Have diversification rates changed through time in my study tree?*
3. *Is there evidence that my study tree experienced mass extinction?*

Questions regarding diversification rates can be addressed using TESS simply by estimating the parameters of the branching-process model—*i.e.,* rates of speciation (λ), extinction (μ), net-diversification ($\lambda - \mu$), and relative-extinction ($\mu \div \lambda$). We estimate these parameters in a Bayesian statistical framework, which provides a natural means to accommodate our uncertainty in estimates of the parameters—*i.e.,* rather than inferring rate parameters as point estimates, TESS provides estimates as marginal posterior probability densities. We describe the branching-process models implemented in TESS—and the methods for estimating parameters of these models—in Section 2 of this guide.

Questions regarding temporal variation in diversification rates can be addressed using TESS by comparing the relative fit of the study tree to candidate branching-process models—*i.e.,* by performing Bayes factor comparisons to assess the relative support for models in which diversification rates are either constant or change through time. Note that the models we have implemented in TESS assume that diversification rates are homogeneous across lineages. Accordingly, even though

diversification rates may change—gradually or episodically—through time, diversification rates are nevertheless identical across all lineages at any instant in time. We describe how to use TESS to compare the fit of candidate diversification models to a given dataset in Section 4 of this guide. Special attention is given to different methods of incomplete taxon sampling in Section 3.

Questions regarding mass-extinction events can be inferred using TESS by performing specific hypothesis tests (see Section 2.4.4) or analyses under the CPP on Mass-Extinction Times (`CoMET`) model (May et al., 2015). These analyses can identify whether your study tree has been impacted by mass extinction, and if so, can identify the number and timing of these events. Additionally, the `CoMET` model can be used to explore events other than mass extinction—such as the number of tree-wide diversification-rate shifts, the timing of those events, and the rate parameters (*e.g.,* speciation and extinction rates) associated with those events. We describe how to use TESS and `CoMET` to explore mass-extinction events in Section 4.3 of this guide.

1.3 Empirical data

Rates of lineage diversification are typically estimated from phylogenies that, in turn, have been inferred from molecular sequence data. For example, consider the conifer phylogeny that is included with the TESS distribution:

```
library(TESS)      # load the package
data(conifers)     # load the conifers dataset
```

More information on this phylogeny can be found in Leslie et al. (2012). You will, of course, want to use your own tree for your diversification-rate analyses. You can do this using the `read.nexus` function and `read.tree` provided in the `ape` package:

```
myTree <- read.nexus("data/myTree.nex")
```

You can extract the node ages from the tree using the `ape` function `branching.times`. We often use the node ages for estimating parameters of birth-death processes, so we'll extract them and store them in a variable for later use.

```
times <- as.numeric( branching.times(conifers) )
```

You then can view the phylogeny (Figure 1).

```
plot(conifers,show.tip.label=FALSE,no.margin=TRUE)
```

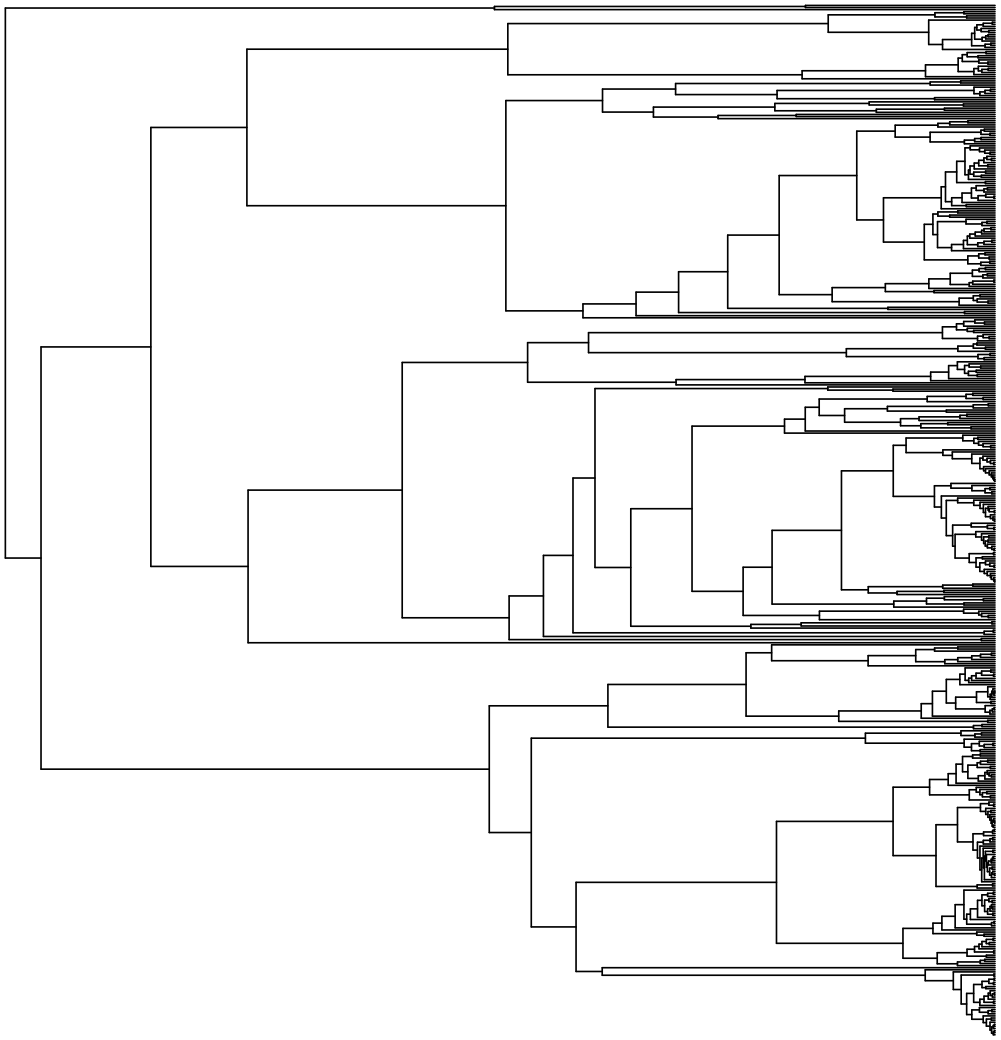


Figure 1: Conifer phylogeny from Leslie et al. (2012) without taxon labels.

Notice that this is an *ultrametric* tree; that is, it is rooted and all of the tips are sampled at the same time horizon (*i.e.,* the present). The models implemented in TESS are only valid for ultrametric trees. Other trees—*e.g.,* where tips are sampled sequentially through time (Stadler, 2010; Heath et al., 2014)—are currently not supported.

Additionally, you can look at the lineage-through-time (LTT) plot (Figure 2).

```
ltt.plot(conifers,log="y")
```

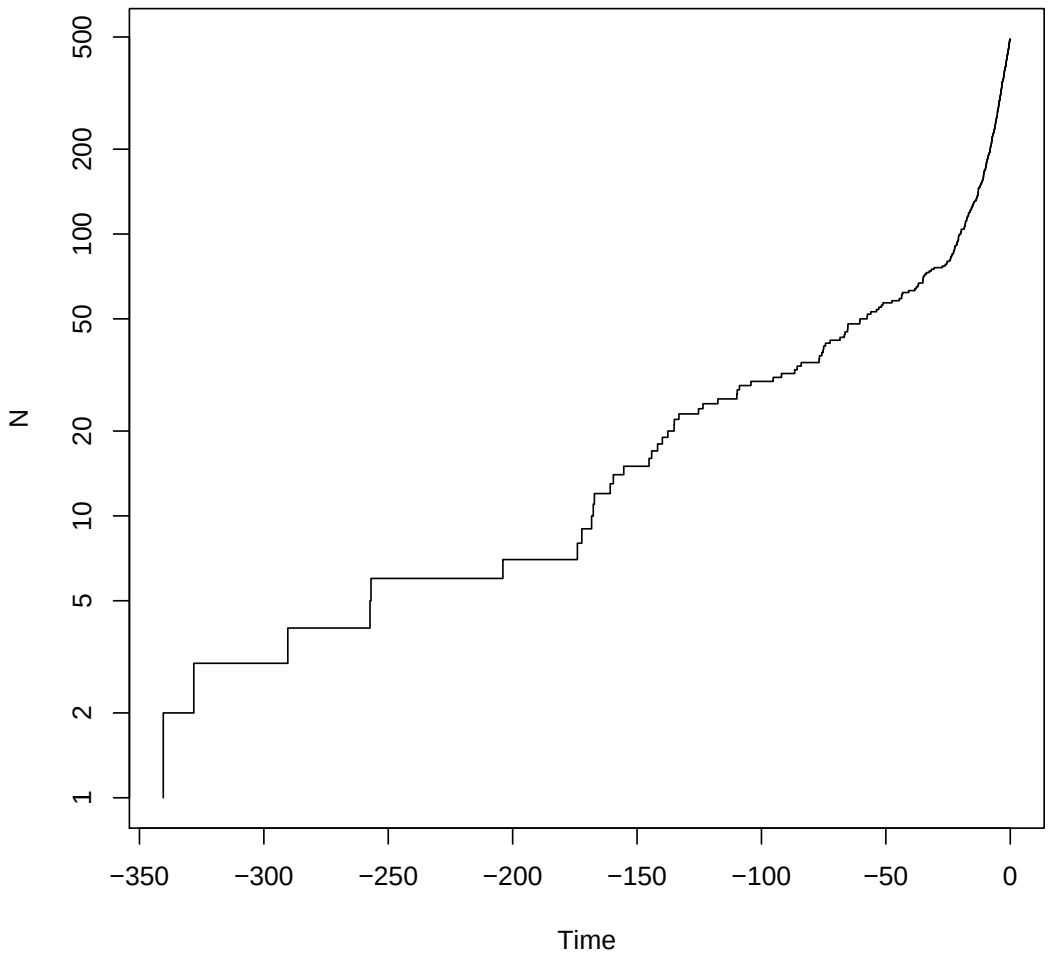


Figure 2: Lineage-through-time plot of the conifer phylogeny.

The LTT plot allows us to visualize the phylogenetic information that is used for estimating diversification rates. For example, it appears that the slope of the LTT plot changes slightly at $\approx 175, 70$, and 20 million years ago.

2 Models

We begin this section with a general introduction to the stochastic birth-death branching process that underlies inference of diversification rates in TESS. This primer will provide some details on the relevant theory of stochastic-branching process models. We appreciate that some readers may want to skip this somewhat technical primer; however, we believe that a better understanding of the relevant theory provides a foundation for performing better inferences. We then disscuss a variety of specific birth-death models, but emphasize that these examples represent only a tiny fraction of the possible diversification-rate models that can be specified in TESS.

2.1 The birth-death branching process

Our approach is based on the *reconstructed evolutionary process* described by Nee et al. (1994); a birth-death process in which only sampled, extant lineages are observed. Let $N(t)$ denote the number of species at time t. Assume the process starts at time t_1 (the 'crown' age of the most recent common ancestor of the study group, t_{MRCA}) when there are two species. Thus, the process is initiated with two species, $N(t_1) = 2$. We condition the process on sampling at least one descendant from each of these initial two lineages; otherwise t_1 would not correspond to the t_{MRCA} of our study group. Each lineage evolves independently of all other lineages, giving rise to exactly one new lineage with rate $b(t)$ and losing one existing lineage with rate $d(t)$ (Figure 3 and Figure 4). Note that although each lineage evolves independently, all lineages share both a common (tree-wide) speciation rate $b(t)$ and a common extinction rate $d(t)$ (Nee et al., 1994; Höhna, 2015). Additionally, at certain times, $t_{\mathbb{M}}$, a mass-extinction event occurs and each species existing at that time has the same probability, ρ, of survival. Finally, all extinct lineages are pruned and only the reconstructed tree remains (Figure 3).

To condition the probability of observing the branching times on the survival of both lineages that descend from the root, we divide by $P(N(T) > 0|N(0) = 1)^2$. Then, the probability density of the branching times, $\mathbb{T}$, becomes

$$P(\mathbb{T}) = \frac{\overbrace{P(N(T) = 1 \mid N(0) = 1)^2}^{\text{both initial lineages have one descendant}}}{\underbrace{P(N(T) > 0 \mid N(0) = 1)^2}_{\text{both initial lineages survive}}} \times \prod_{i=2}^{n-1} \overbrace{i \times b(t_i)}^{\text{speciation rate}} \times \overbrace{P(N(T) = 1 \mid N(t_i) = 1)}^{\text{lineage has one descendant}},$$

and the probability density of the reconstructed tree (topology and branching

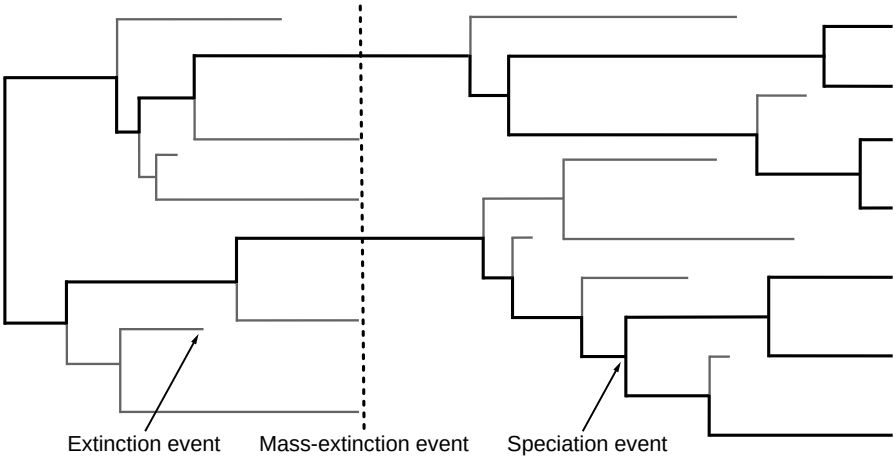


Figure 3: A realization of the birth-death process with mass extinction. Lineages that have no extant or sampled descendant are shown in gray and surviving lineages are shown in a thicker black line.

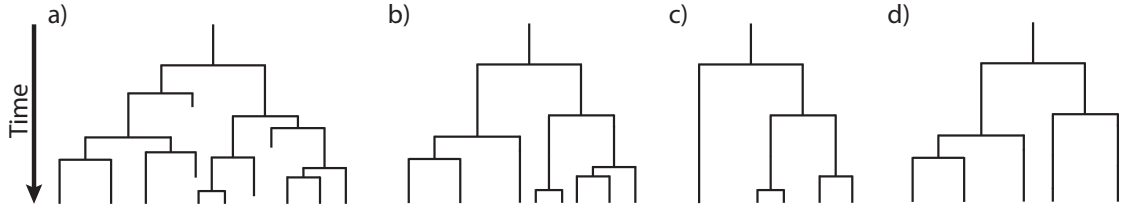


Figure 4: Examples of trees produced under a birth-death process. The process is initiated at the first speciation event (the 'crown-age' of the MRCA) when there are two initial lineages. At each speciation event the ancestral lineage is replaced by two descendant lineages. At an extinction event one lineage simply terminates. (A) A complete tree including extinct lineages. (B) The reconstructed tree of tree from A with extinct lineages pruned away. (C) A *uniform* subsample of the tree from B, where each species was sampled with equal probability, ρ. (D) A *diversified* subsample of the tree from B, where the species were selected so as to maximize diversity.

times) is then

$$P(\Psi) = \frac{2^{n-1}}{n!(n-1)!} \times \left(\frac{P(N(T) = 1 \mid N(0) = 1)}{P(N(T) > 0 \mid N(0) = 1)} \right)^2 \times \prod_{i=2}^{n-1} i \times b(t_i) \times P(N(T) = 1 \mid N(t_i) = 1) \quad (1)$$

We can expand Equation (1) by substituting $P(N(T) > 0 \mid N(t) = 1)^2 \exp(r(t, T))$ for $P(N(T) = 1 \mid N(t) = 1)$, where $r(u, v) = \int_u^v d(t) - b(t)dt$; the above equation

becomes

$$
\begin{aligned}
P(\Psi) &= \frac{2^{n-1}}{n!(n-1)!} \times \left(\frac{P(N(T) > 0 \mid N(0) = 1)^2 \exp(r(0,T))}{P(N(T) > 0 \mid N(0) = 1)} \right)^2 \\
&\quad \times \prod_{i=2}^{n-1} i \times b(t_i) \times P(N(T) > 0 \mid N(t_i) = 1)^2 \exp(r(t_i, T)) \\
&= \frac{2^{n-1}}{n!} \times \Big(P(N(T) > 0 \mid N(0) = 1) \exp(r(0,T)) \Big)^2 \\
&\quad \times \prod_{i=2}^{n-1} b(t_i) \times P(N(T) > 0 \mid N(t_i) = 1)^2 \exp(r(t_i, T)). \qquad (2)
\end{aligned}
$$

For a detailed description of this substitution, see Höhna (2015). Additional information regarding the underlying birth-death process can be found in (Thompson, 1975, Equation 3.4.6) and Nee et al. (1994) for constant rates and Lambert (2010); Lambert and Stadler (2013); Höhna (2013, 2014, 2015) for arbitrary rate functions.

To compute the equation above we need to know the rate function, $r(t,s) = \int_t^s d(x) - b(x)dx$, and the probability of survival, $P(N(T) > 0|N(t) = 1)$. Yule (1925) and later Kendall (1948) derived the probability that a process survives ($N(T) > 0$) and the probability of obtaining exactly n species at time T ($N(T) = n$) when the process started at time t with one species. Kendall's results were summarized in Equation (3) and Equation (24) in Nee et al. (1994)

$$
\begin{aligned}
P(N(T)>0|N(t)=1) &= \left(1 + \int_t^T \Big(\mu(s) \exp(r(t,s)) \Big) ds \right)^{-1} \qquad (3) \\
P(N(T)=n|N(t)=1) &= (1 - P(N(T)>0|N(t)=1) \exp(r(t,T)))^{n-1} \\
&\quad \times P(N(T)>0|N(t)=1)^2 \exp(r(t,T)) \qquad (4)
\end{aligned}
$$

An overview for different diversification models is given in Höhna (2015).

2.2 The space of birth-death branching-process models

Our preceding discussion of the birth-death process makes it clear that we can define countless birth-death models that specify different speciation- and extinction-rate functions over time. We could assume, for example, that the extinction rate is constant over time, $d(t) = \mu$, or that the speciation rate decreases exponentially, $b(t) = \lambda * \exp(-\alpha * t)$. Furthermore, the constant-rate birth-death process can be parameterized in various ways, for example, by adopting parameters for the rate of speciation, $b(t) = \lambda$, and extinction, $d(t) = \mu$. Alternatively, we could

describe the birth-death process using parameters for the net-diversification rate, $\delta = \lambda - \mu$, and relative-extinction rate, $\epsilon = \mu/\lambda$, such that $b(t) = \delta/(1-\epsilon)$ and $d(t) = \epsilon * (\delta/(1-\epsilon))$. Finally, we could describe the birth-death process using parameters for the net-diversification rate, $\delta = \lambda - \mu$, and turnover rate, $\tau = \mu$, such that $b(t) = \delta + \tau$ and $d(t) = \tau$. Depending on the inference scenario, each of these parameterizations may offer advantages in terms of interpretation.

Below, we list several birth-death process models (*e.g.,* used in Höhna, 2014) to provide a sense of the types of models that can be specified and how they are parametrized in TESS (Table 1).

Table 1: Six different birth-death models with the corresponding parameters.

Model	$b(t)$	$d(t)$
Model 1	λ_0	0
Model 2	$\lambda_1 * \exp(-\alpha * t)$	0
Model 3	λ_0	μ
Model 4	$\lambda_0 + \lambda_1 * \exp(-\alpha * t)$	0
Model 5	$\lambda_1 * \exp(-\alpha * t)$	μ
Model 6	$\lambda_0 + \lambda_1 * \exp(-\alpha * t)$	μ

- Model 1: A constant-rate pure-birth (Yule) process (Yule, 1925). Under this process, the number of species increases monotonically and exponentially.
- Model 2: A decreasing-rate pure-birth process where the speciation rate declines toward zero. This process is equivalent to the decreasing-rate pure-birth process used in Rabosky and Lovette (2008). Under this process, the number of species increases monotonically.
- Model 3: A constant-rate birth-death process, as used in Thompson (1975). Under this process, the expected number of species increases exponentially.
- Model 4: A pure-birth process with a decaying rate of speciation but a constant, non-zero speciation rate the longer the process continues ($\lambda(t) = \lambda_0 + \lambda_1 * \exp(-\alpha * t)$). Thus, the process does not stop producing new species after the initial burst, as in Model 2. As in the other two pure-birth processes, the number of species increases monotonically.
- Model 5: A birth-death process with an initial expansion phase (where the speciation rate exceeds the extinction rate) that subsequently converges to a critical-branching process, *i.e.,* where the speciation and extinction rates

are equal, $\lambda(t) = \mu + \lambda_1 * \exp(-\alpha * t)$ and $\mu(t) = \mu$. Although one might assume that the expected number of species will remain constant for a critical-branching process, this does not hold if the process is conditioned on survival.

- Model 6: A birth-death process where the extinction rate remains constant, but speciation rate has an initially constant phase followed by a decreasing phase. This model corresponds to an early phase of radiation, followed by a phase of steady increase, $\lambda(t) = \lambda_0 + \lambda_1 * \exp(-\alpha * t)$ and $\mu(t) = \mu$.

The parametrizations of these models are listed in Table 1, and the expected number of species, $E[N(T)]$, at time T under each model is depicted in Figure 5. We derive $E[N(T)]$ analytically by using the fact that $N(T)$ is geometrically distributed (see Equation 5 in Höhna, 2013). Note that the process is conditioned on survival to the present, such that $E[N(T)]$ increases even if $\lambda(t) = \mu(t)$.

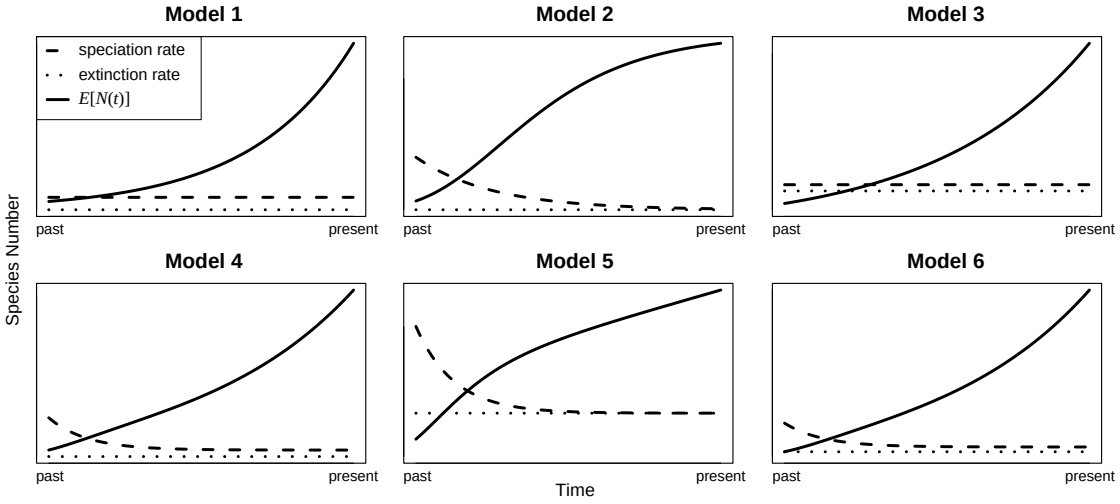


Figure 5: Six possible birth-death models. Each plot shows the speciation and extinction rates over time, and also the expected number of species ($E[N(t)]$). Model 1: A constant-rate pure-birth process. Model 2: A decreasing-rate pure-birth process with speciation rate declining to zero. Model 3: A constant-rate birth-death process. Model 4: A pure-birth process, where the speciation rate passes through a constant phase to a decreasing phase. Model 5: A birth-death process with an initial expansion phase (speciation rate > extinction rate) that later converges to a critical-branching process. Model 6: A birth-death process with a constant extinction rate, where the speciation rate is initially constant and later decreases.

2.3 Simulating data

Simulating phylogenies is critical for validating methods/models of lineage diversification, and is also invaluable for developing our intuition about the behavior of these models. Simulations are also crucial for assessing the adequacy (absolute fit) of a model for a given dataset, which we will describe later. In the previous section we described the expected form of lineage-accumulation curves under different branching-process models. We will now briefly explain how to simulate phylogenies using TESS.

We will explore some common diversification models, including the constant-rate pure-birth process, the constant-rate birth-death process, and the exponentially decaying pure-birth process. Specifically, we will use TESS to simulate 50 trees under these models and look at the corresponding LTT plots. You can experiment with the parameter settings to better understand their impact, *e.g.,* the influence of the extinction rate.

We will first simulate trees under a constant-rate pure-birth process, where we specify a speciation rate of 1.0 and the duration of the process as 3.0 time units.

```
speciation <- 1.0
extinction <- 0.0
tmrca <- 3.0
```

Here, we are explicitly conditioning the simulation on the time of the process. Because it is a stochastic process, this will result in simulated trees of different sizes (number of species), which may be relevant to our question. We might, for example, wish to know whether the observed species diversity in our study tree is improbable under the current model and parameterization. In the next subsection we will show how to simulate trees conditioned on the number of extant species.

We simulate 50 trees under the specified model as follows:

```
trees <- tess.sim.age(n = 50,
                      age = tmrca,
                      lambda = speciation,
                      mu = extinction,
                      MRCA = TRUE)
```

Note that we are initializing the simulation with two species; *i.e.,* from the 'crown age' of the most recent common ancestor (MRCA). Accordingly, the resulting trees will not have 'stem' branches subtending their root nodes; instead, these trees begin at the root node that corresponds to the first speciation event in each tree (*c.f.*, Figure 4). This scenario corresponds well with empirical trees, where (by definition) at least one species from both of these two initial lineages will survive

to the present (otherwise we would not recognize this node as the root of our study tree).

Next, we will generate the lineage-through-time plots for all 50 simulated trees.

```
mltt.plot(trees,
          log = "y",
          dcol = FALSE,
          legend = FALSE,
          backward = FALSE)
```

For a fully specified model, TESS can calculate the expected number of lineages through time. We will overlay a curve describing the the expected number of lineages on the LTT plot.

```
expected <- function(t)
  tess.nTaxa.expected(begin = 0,
                      t = t,
                      end = tmrca,
                      lambda = speciation,
                      mu = extinction,
                      MRCA = TRUE,
                      reconstructed = TRUE)

curve(expected,add=FALSE,col="red",lty=2,lwd=5)
legend("topleft",col="red",lty=2,"Expected Diversity")
```

The results of this simulation are shown in Figure 6A. Here, you can see that the shape of the LTT curve is clearly linear (in log-scale) under a constant-rate pure-birth process. All other curves will be rendered in log-scale for convenience. Notice also that we used the argument `reconstructed = TRUE` which means that we compute the expected number of species (diversity) of a reconstructed phylogeny. This must be a monotonically increasing function. You could plot the expected diversity at any given time and compare it to the diversity of reconstructed phylogeny.

We will now repeat the above simulation under a constant-rate birth-death process. First, we set the parameters of the model.

```
speciation <- 5.0
extinction <- 4.0
tmrca <- 3.0
```

Then simulate 50 trees under these parameters.

```
trees <- tess.sim.age(n = 50,
                      age = tmrca,
                      lambda = speciation,
                      mu = extinction,
                      MRCA = TRUE)
```

Next, we plot the lineage-through-time curves for the simulated trees:

```
mltt.plot(trees,
          log = "y",
          dcol = FALSE,
          legend = FALSE,
          backward = FALSE)
```

Finally, we overlay the expected number of lineages on our LTT plot. In this example you may notice that the expected number of lineages under the birth-death process diverges from the expected number of lineages in the reconstructed tree. This is simply because the expected number of lineages in the reconstructed tree only considers lineages that have at least one descendant sampled at the present time, whereas the expected number of lineages gives the expected diversity at the time without that constraint.

```
expected <- function(t)
  tess.nTaxa.expected(begin = 0,
                      t = t,
                      end = tmrca,
                      lambda = speciation,
                      mu = extinction,
                      MRCA = TRUE,
                      reconstructed = TRUE)

curve(expected,add=TRUE,col="red",lty=2,lwd=5)
legend("topleft",col="red",lty=2,"Expected Diversity")
```

The results of this simulation are shown in Figure 6B. Notice that the slope of the LTT plot increases sharply near the present: this is commonly referred the 'pull-of-the-present' effect. This effect becomes more pronounced as the relative-extinction rate (*i.e.*, extinction ÷ speciation) increases.

Finally, we will consider a pure-birth process with exponentially decreasing speciation rate. In TESS you can either specify a simple numeric value for the speciation and extinction rates or you can specify a function that takes the time t as a parameter. Here, we will use the second option.

```
speciation <- function(t) 0.5 + 2 * exp(-1.0*t)
extinction <- 0.0
tmrca <- 3.0
```

We again simulate 50 trees conditioned on the survival of the two initial lineages.

```
trees <- tess.sim.age(n = 50,
                      age = tmrca,
                      lambda = speciation,
                      mu = extinction,
                      MRCA = TRUE)
```

We generate the LTT plots for the simulated trees.

```
mltt.plot(trees,
          log = "y",
          dcol = FALSE,
          legend = FALSE,
          backward = FALSE)
```

And then add the expected number of lineages in the reconstructed phylogeny.

```
expected <- function(t)
  tess.nTaxa.expected(begin = 0,
                      t = t,
                      end = tmrca,
                      lambda = speciation,
                      mu = extinction,
                      MRCA = TRUE,
                      reconstructed = TRUE)

curve(expected,add=TRUE,col="red",lty=2,lwd=5)
legend("topleft",col="red",lty=2,"Expected Diversity")
```

The results of the three simulations are shown in Figure 6. We will return to simulating reconstructed trees in Section 4.2 when we discuss model adequacy testing.

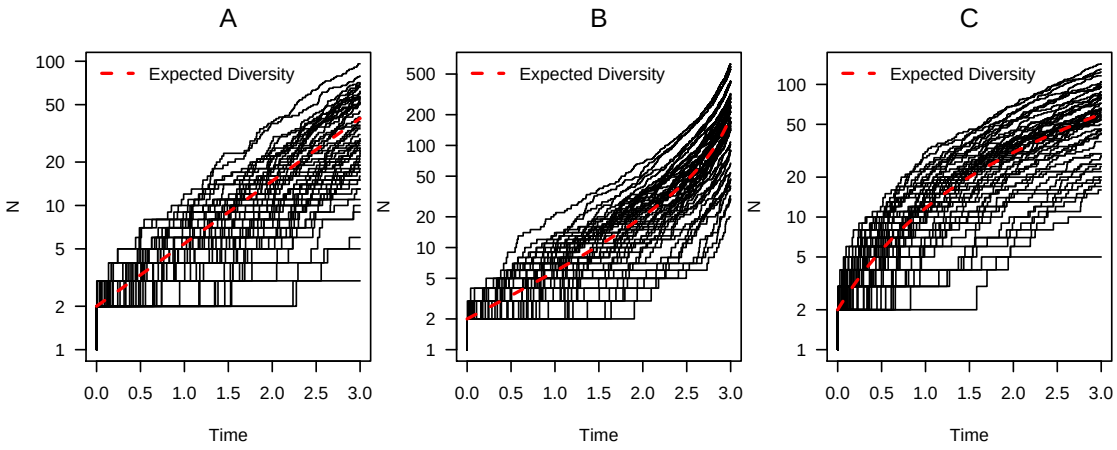


Figure 6: Lineage-through-time curves for pure-birth trees (panel A), birth-death trees (panel B), and pure-birth trees with exponentially decreasing speciation rate (panel C).

2.4 Estimating parameters using Markov chain Monte Carlo (MCMC)

In the previous section we introduced some stochastic-branching process models, and demonstrated how to simulate trees under those models. Here, we turn to the issue of estimating parameters of branching-process models from empirical data. We estimate parameters within a Bayesian statistical framework, which adopts the perspective that parameters are random variables. Accordingly, it is necessary to specify a probability distribution for each parameter that describes the nature of that random variation. These *prior* probability distributions describe our beliefs about the parameter values before evaluating the data at hand. Prior probabilities are updated by the information in the data (via the likelihood function) to provide the corresponding *posterior* probability distributions. These posterior probability distributions reflect our belief about the parameter values after incorporating the new information in our data. We estimate the joint posterior probability density of the model parameters from the data using numerical methods—Markov chain Monte Carlo (MCMC) algorithms.

2.4.1 Birth-death processes with constant rates

We first consider the constant-rate birth-death process. Although we do not explicitly consider the constant-rate pure-birth process, it can easily be specifed by simply setting the extinction rate of the constant-rate birth-death process to zero.

First, we specify prior distributions for our parameters. The constant-rate birth-death process has two parameters; the speciation rate and extinction rate.

There are many possible prior distributions that we might adopt for these two parameters, *e.g.,* the exponential, gamma, lognormal distributions. Here, we will use an exponential distribution, which has a single parameter (the rate parameter) that describes the shape of the distribution. We will specify a value of 0.1 for the rate parameter (such that the mean of the exponential is 1/rate = 10.0). We will use identical priors for both the speciation- and extinction-rate parameters.

In TESS the prior distribution must be functions that can be computed for all values that can be realized by the corresponding parameter (*e.g.,* priors for rates must only include positive-real values). Furthermore, the prior distributions need to return log-transformed probabilities (this is a standard convention adopted to avoid underflow in computer memory).

```
prior_delta <- function(x) { dexp(x,rate=10.0,log=TRUE) }
prior_tau <- function(x) { dexp(x,rate=10.0,log=TRUE) }
priorsConstBD <- c("diversification"=prior_delta,
                   "turnover"=prior_tau)
```

If you provide names for the prior distributions, as we did here, then these names will be used to label that parameters in the MCMC output. Currently, only the names of the priors are used.

Next, we set up the likelihood of the constant-rate birth-death process as an R function. Here, the actual likelihood computation is performed by the function `tess.likelihood`. It is necessary to wrap the TESS likelihood into another R function because you need to specify how the speciation and extinction rates are assembled and which assumptions/conditions are applied. This approach enables maximal flexibility for using TESS.

```
likelihoodConstBD <- function(params) {

  speciation <- params[1] + params[2]
  extinction <- params[2]

  lnl <- tess.likelihood(times,
                         lambda = speciation,
                         mu = extinction,
                         samplingProbability = 1.0,
                         log = TRUE)

  return (lnl)

}
```

It is also possible to specify prior distributions on other parameterizations of the constant-rate birth-death model, *e.g.,* using parameters for the net-diversification rate (speciation−extinction) and the relative-extinction rate (extinction/speciation). This alternative parmaterization of the model would, of course, require modification of the likelihood function.

Next, we use the function `tess.mcmc` to run an MCMC simulation. The function takes in several arguments to describe the MCMC algorithm. Specifically, you must specify the `likelihoodFunction`, `priors`, and initial values for the `parameters`. Additionally, you can specify whether the MCMC proposal mechanisms should operate on the log-transformed parameters, which is advisable for rate parameters but not for location parameters.

We will also specify the value for the `delta` parameter, which defines the (initial) width of the sliding-window proposal mechanism. This `delta` tuning parameter determines the scale (severity) of the proposal mechanism: larger values will specify more severe changes to the current parameter value when that parameter is being updated during the MCMC. We will discuss these issues in more detail in Section 5 of this guide. The remaining parameters specify the number of iterations of the MCMC simulation, the number of iterations for the pre-burnin phase, the thinning schedule, and whether the scale of the poposal mechanisms are to be automatically tuned.

```
set.seed(12345)    # remove this line to obtain a random seed
samplesConstBD <- tess.mcmc(likelihoodFunction = likelihoodConstBD,
                            priors = priorsConstBD,
                            parameters = runif(2,0,1),
                            logTransforms = c(TRUE,TRUE),
                            delta = c(1,1),
                            iterations = 10000,
                            burnin = 1000,
                            thinning = 10,
                            adaptive = TRUE,
                            verbose = TRUE)
```

Note that we have specified a starting seed for the random-number generator. We have done this only to ensure that your results will be identical to those in this guide. However, you *should not* specify the starting seed for your analyses, but instead use a random starting seed that is automatically generated from the system clock (*i.e.,* just delete or comment out the line that sets the seed). This is important, as you will want to perform multiple independent MCMC simulations to assess convergence. The basic idea is to compare parameter estimates from multiple independent analyses: if the chains have converged to the target (joint

posterior probability) distribution, then the parameter estimates from the replicate chains should be identical up to some stochastic uncertainty. However, this important diagnostic would be rendered meaningless if the replicate analyses were performed under the same starting seed; in this case, the results are guaranteed to be identical.

Note that we ran a short MCMC simulation above for covenience. In practice, MCMC simulations are commonly run for 10^5 to 10^8 iterations. We will use the R package coda (which is automatically loaded with TESS) to summarize the samples from our MCMC simulation. TESS saves samples in the coda format, which allows us to easily summarize our samples:

```
summary(samplesConstBD)

##
## Iterations = 1:1001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                     Mean      SD  Naive SE Time-series SE
## diversification 0.006218 0.00230 7.269e-05      7.269e-05
## turnover        0.149091 0.01237 3.908e-04      3.908e-04
##
## 2. Quantiles for each variable:
##
##                     2.5%      25%      50%      75%   97.5%
## diversification 0.001992 0.004586 0.006133 0.007725 0.01094
## turnover        0.126253 0.140378 0.148587 0.157682 0.17364
```

We can also visualize the trace plots and marginal posterior probability densities for these samples (Figure 7).

```
plot(samplesConstBD)
```

2.4.2 Birth-death processes with continuously varying rates

Here we consider a birth-death process with an exponentially decreasing speciation rate. Specifically, we define the speciation rate as $\lambda(t) = \delta + \lambda \exp(-\alpha * t)$ and

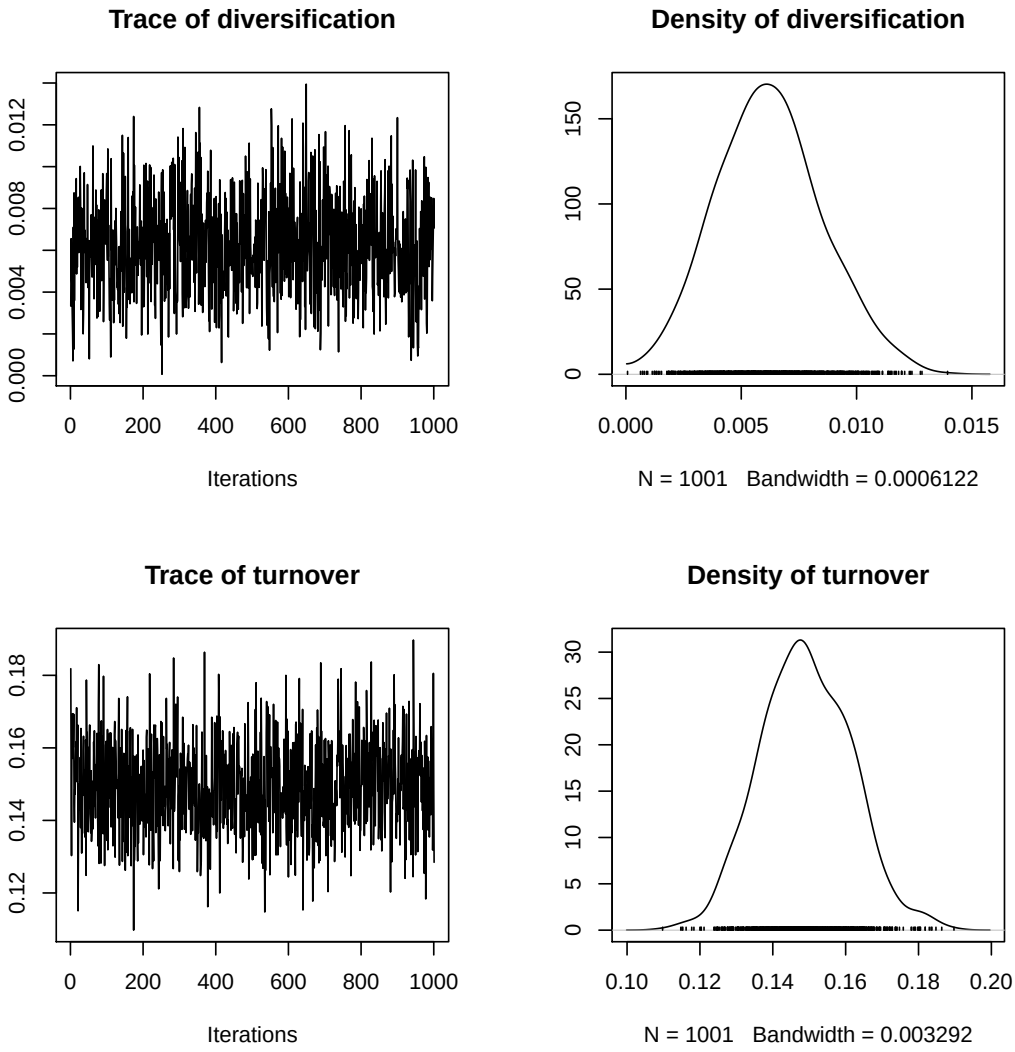


Figure 7: Trace plots (left) and marginal posterior probability densities (right) for the diversification rate (top) and turnover rate (bottom) from the MCMC simulation under the constant-rate birth-death process.

extinction rate as $\mu(t) = \delta$ (Höhna, 2014). It is not possible to analytically compute the probability density (or likelihood) under this process. Instead, we approximate these quantities using numerical integration techniques. These numerical methods are implemented in TESS and will be performed automatically if you provide functions instead of numerical arguments for the speciation and/or extinction rate. The numerical integration is very convenient but, of course, imposes a higher computational cost that will make these analyses run more slowly.

The decreasing speciation rate birth-death model has three parameters: δ, λ_1, and α. We will use an exponential prior probability distribution with a rate of 0.1 (*i.e.*, with a mean of 10.0) for all three parameters. As before, the prior distributions must be functions that return the log-transformed probability for a given value of the parameter.

```
prior_delta <- function(x) { dexp(x,rate=0.1,log=TRUE) }
prior_lambda <- function(x) { dexp(x,rate=10.0,log=TRUE) }
prior_alpha <- function(x) { dexp(x,rate=0.1,log=TRUE) }
priorsDecrBD <- c("turnover"=prior_delta,
                  "initial speciation"=prior_lambda,
                  "speciation decay"=prior_alpha)
```

We now specify the speciation and extinction rates as functions and pass them into the likelihood, which again must be provided as a function.

```
likelihoodDecrBD <- function(params) {

  speciation <- function(t) params[1] + params[2] * exp(-params[3]*t)
  extinction <- function(t) params[1]

  lnl <- tess.likelihood(times,
                         lambda = speciation,
                         mu = extinction,
                         samplingProbability = 1.0,
                         log = TRUE)

  return (lnl)

}
```

Next, we start the analysis by calling the MCMC function in TESS. (The details of this MCMC simulation are similar to those described in the constant-rate birth-death example, above.)

```
set.seed(12345)
samplesDecrBD <- tess.mcmc(likelihoodFunction = likelihoodDecrBD,
                           priors = priorsDecrBD,
                           parameters = runif(3,0,1),
                           logTransforms = c(TRUE,TRUE,TRUE),
```

```
                              delta = c(1,1,1),
                              iterations = 10000,
                              burnin = 1000,
                              thinning = 10,
                              adaptive = TRUE,
                              verbose = TRUE)
```

We then summarize the parameter estimates from our MCMC samples:

```
summary(samplesDecrBD)

##
## Iterations = 1:1001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                       Mean      SD  Naive SE Time-series SE
## turnover            0.1616 0.01192 0.0003767      0.0003767
## initial speciation 0.0986 0.09446 0.0029856      0.0029856
## speciation decay   9.8532 9.89787 0.3128419      0.3128419
##
## 2. Quantiles for each variable:
##
##                         2.5%     25%     50%     75%   97.5%
## turnover            0.139719 0.15374 0.16101  0.1695  0.1857
## initial speciation 0.003187 0.02881 0.07291  0.1396  0.3490
## speciation decay   0.142774 2.79831 6.96848 13.4007 36.5878
```

We can also visualize the trace plots and marginal posterior probability densities for these samples:

```
plot(samplesDecrBD)
```

2.4.3 Birth-death processes with episodically varying rates

The next model we consider is a birth-death process with piecewise-constant rates. Under this model, rates of speciation and extinction change at some (discrete)

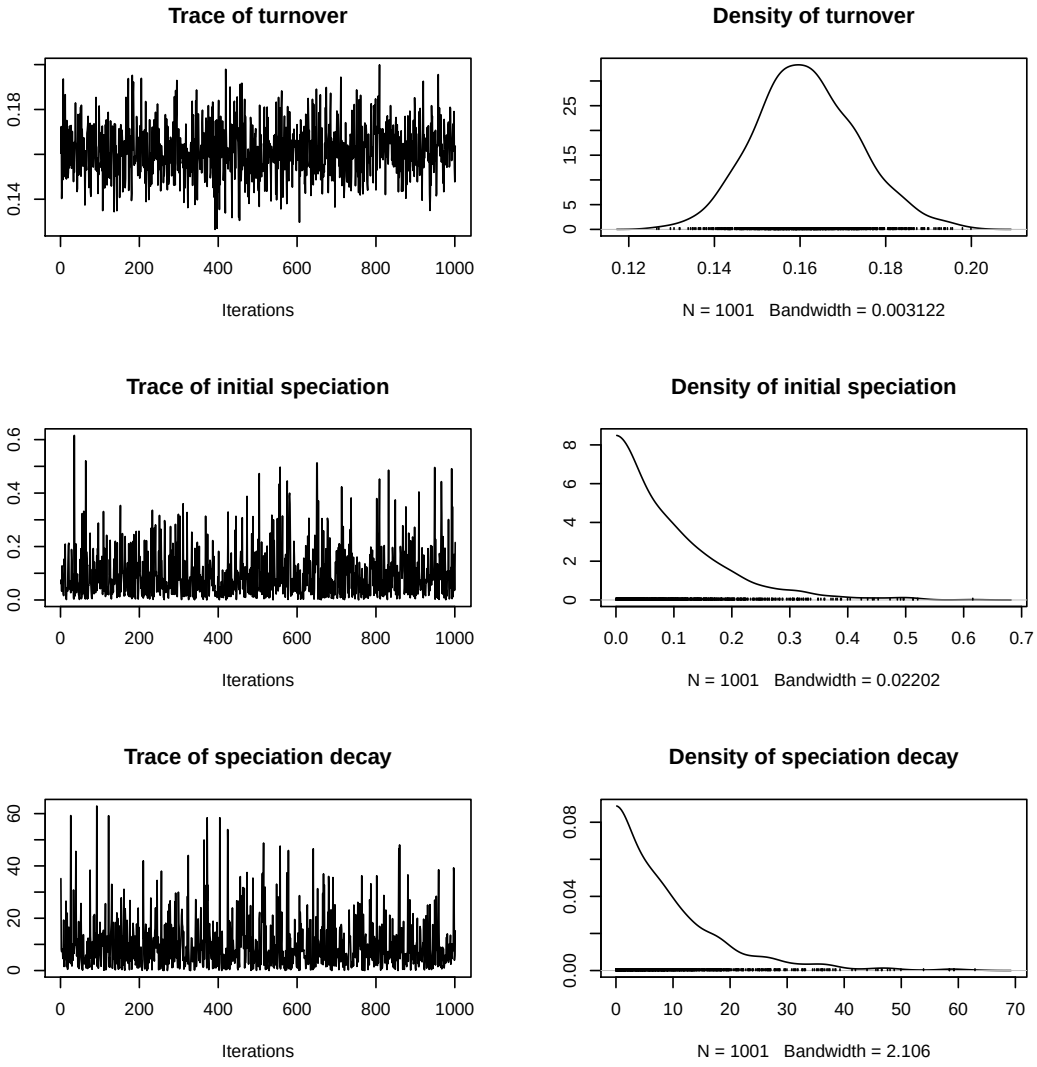


Figure 8: Trace plots and estimated posterior distribution of the parameter under the decreasing speciation rate birth-death model.

number of events; between these rate-shift events, however, the diversification-rate parameters remain constant (Stadler, 2011a; Höhna, 2015).

The number of parameters included in the episodic model varies depending on the number of rate-shift events. In general, there are $k_{\mathbb{B}} + 1$ speciation-rate parameters and $k_{\mathbb{D}} + 1$ extinction-rate parameters, where $k_{\mathbb{B}}$ is the number of speciation-rate shifts and $k_{\mathbb{D}}$ is the number of extinction-rate shifts.

In this example, we will assume there is a single speciation-rate shift and a single extinction-rate shift, both occurring at the mid-point of the duration spanned by the conifer tree. First, we specify the time of the rate-shift event.

```
rateChangeTime <- max( times ) / 2
```

Next, we specify priors for the parameters. There are a total of four parameters (the speciation and extinction rates before and after the rate-shift event). Accordingly, we specify four identical exponential priors for these parameters, all with a rate of 10.0 (and a mean of 0.1).

```
prior_delta_before <- function(x) { dexp(x,rate=10.0,log=TRUE) }
prior_tau_before <- function(x) { dexp(x,rate=10.0,log=TRUE) }
prior_delta_after <- function(x) { dexp(x,rate=10.0,log=TRUE) }
prior_tau_after <- function(x) { dexp(x,rate=10.0,log=TRUE) }
priorsEpisodicBD <- c("diversification before"=prior_delta_before,
                      "turnover before"=prior_tau_before,
                      "diversification after"=prior_delta_after,
                      "turnover after"=prior_tau_after)
```

Next, we specify a likelihood function using the rate-shift model implemented in TESS, `tess.likelihood.rateshift`.

```
likelihoodEpisodicBD <- function(params) {

  speciation <- c(params[1]+params[2],params[3]+params[4])
  extinction <- c(params[2],params[4])

  lnl <- tess.likelihood.rateshift(times,
                                   lambda = speciation,
                                   mu = extinction,
                                   rateChangeTimesLambda = rateChangeTime,
                                   rateChangeTimesMu = rateChangeTime,
                                   samplingProbability = 1.0,
```

```
                                    log = TRUE)

  return (lnl)

}
```

Now we can start the analysis by calling the MCMC function in TESS. (The details of this MCMC simulation are similar to those described in the constant-rate birth-death example, above.)

```
set.seed(12345)
samplesEpisodicBD <- tess.mcmc(likelihoodFunction = likelihoodEpisodicBD,
                               priors = priorsEpisodicBD,
                               parameters = runif(4,0,1),
                               logTransforms = c(TRUE,TRUE,TRUE,TRUE),
                               delta = c(1,1,1,1),
                               iterations = 10000,
                               burnin = 1000,
                               thinning = 10,
                               adaptive = TRUE,
                               verbose = TRUE)
```

We then summarize the parameter estimates from our MCMC samples:

```
summary(samplesEpisodicBD)

##
## Iterations = 1:1001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                            Mean       SD  Naive SE Time-series SE
## diversification before 0.011450 0.006883 2.175e-04      2.443e-04
## turnover before        0.119879 0.081695 2.582e-03      2.729e-03
## diversification after  0.006041 0.002725 8.613e-05      9.921e-05
## turnover after         0.148364 0.012362 3.907e-04      4.495e-04
```

```
##
## 2. Quantiles for each variable:
##
##                              2.5%      25%      50%      75%   97.5%
## diversification before 0.0007131 0.006239 0.010795 0.015865 0.02693
## turnover before        0.0152805 0.061084 0.103623 0.158882 0.32575
## diversification after  0.0012304 0.004041 0.005865 0.007869 0.01168
## turnover after         0.1243451 0.140137 0.148350 0.156782 0.17259
```

Finally, we can visualize the trace plots and marginal posterior probability densities for these samples:

```
plot(samplesEpisodicBD)
```

2.4.4 Birth-death processes with explicit mass-extinction events

The final model we consider is one where speciation and extinction rates are constant, but where there is a single mass-extinction event at some unknown time. We'll assume that 10% of the species survive the mass-extinction event.

```
survivalProbability <- 0.1
```

There are three parameters in the model: the speciation rate, the extinction rate, and the mass-extinction time. We must specify priors for each of these parameters. For simplicity, we'll assume *a priori* that the mass-extinction event could happen at any time in the most recent half of the tree with equal probability.

```
prior_delta <- function(x) { dexp(x,rate=10.0,log=TRUE) }
prior_tau <- function(x) { dexp(x,rate=10.0,log=TRUE) }
prior_time <- function(x) { dunif(x,min=max(times)/2,max=max(times),log=TRUE)}
priorsMassExtinctionBD <- c("diversification"=prior_delta,
                            "turnover"=prior_tau,
                            "mass-extinction time"=prior_time)
```

Next, we specify a likelihood function. We can use either the standard likelihood function `tess.likelihood` or the likelihood function of the rate-shift model `tess.likelihood.rateshift`.

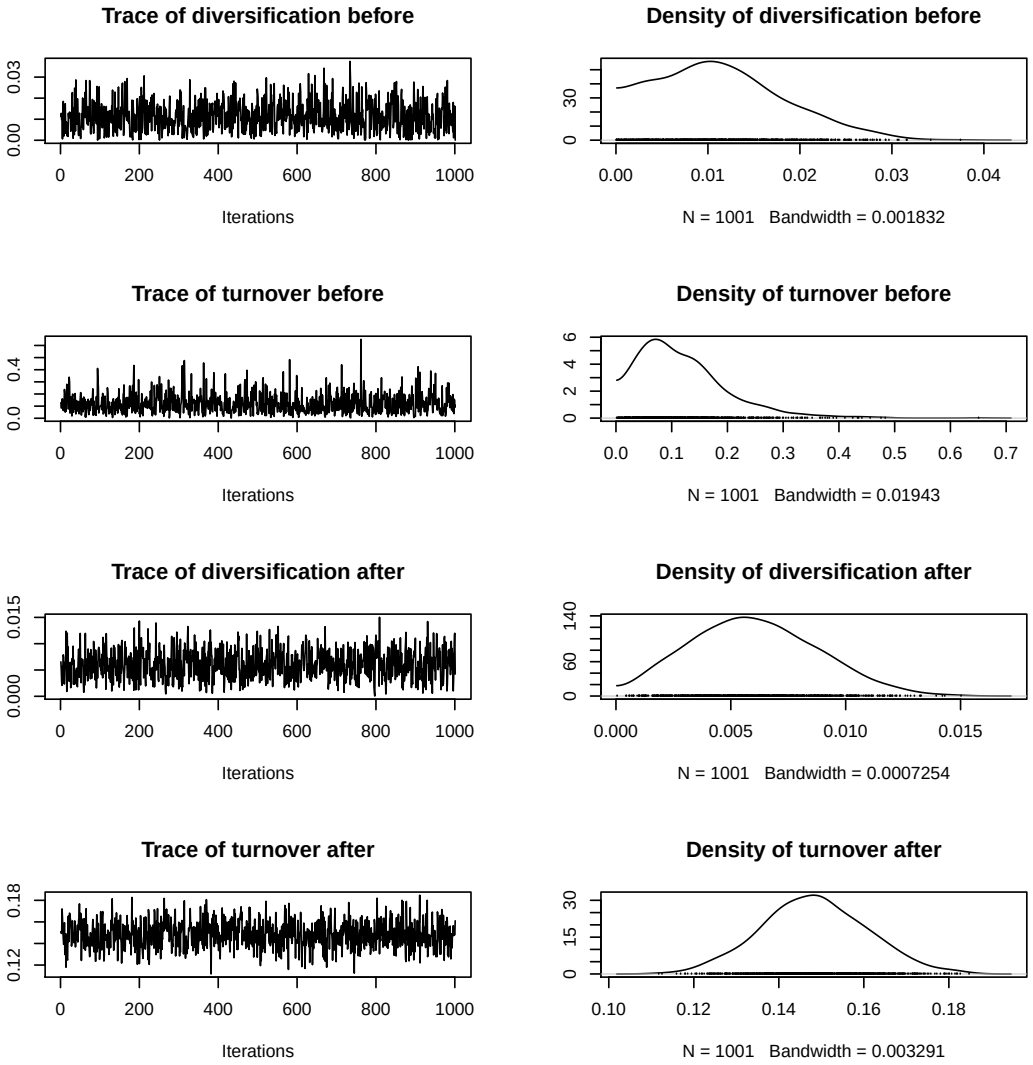


Figure 9: Trace plots and estimated posterior distributions of the parameters under a birth-death-shift model.

```
likelihoodMassExtinctionBD <- function(params) {

  speciation <- params[1]+params[2]
  extinction <- params[2]
  time <- params[3]

  lnl <- tess.likelihood(times,
                         lambda = speciation,
                         mu = extinction,
                         massExtinctionTimes = time,
                         massExtinctionSurvivalProbabilities =
                                     survivalProbability,
                         samplingProbability = 1.0,
                         log = TRUE)

  return (lnl)

}
```

Now we can start the analysis by calling the MCMC function in TESS. (The details of this MCMC simulation are similar to those described in the constant-rate birth-death example, above.)

```
set.seed(12345)
samplesMassExtinctionBD <- tess.mcmc(likelihoodFunction =
                                       likelihoodMassExtinctionBD,
                                       priors = priorsMassExtinctionBD,
                                       parameters = c(runif(2,0,1),max(times)*3/4)
                                       logTransforms = c(TRUE,TRUE,FALSE),
                                       delta = c(1,1,1),
                                       iterations = 10000,
                                       burnin = 1000,
                                       thinning = 10,
                                       adaptive = TRUE,
                                       verbose = TRUE)
```

We then summarize the parameter estimates from our MCMC samples:

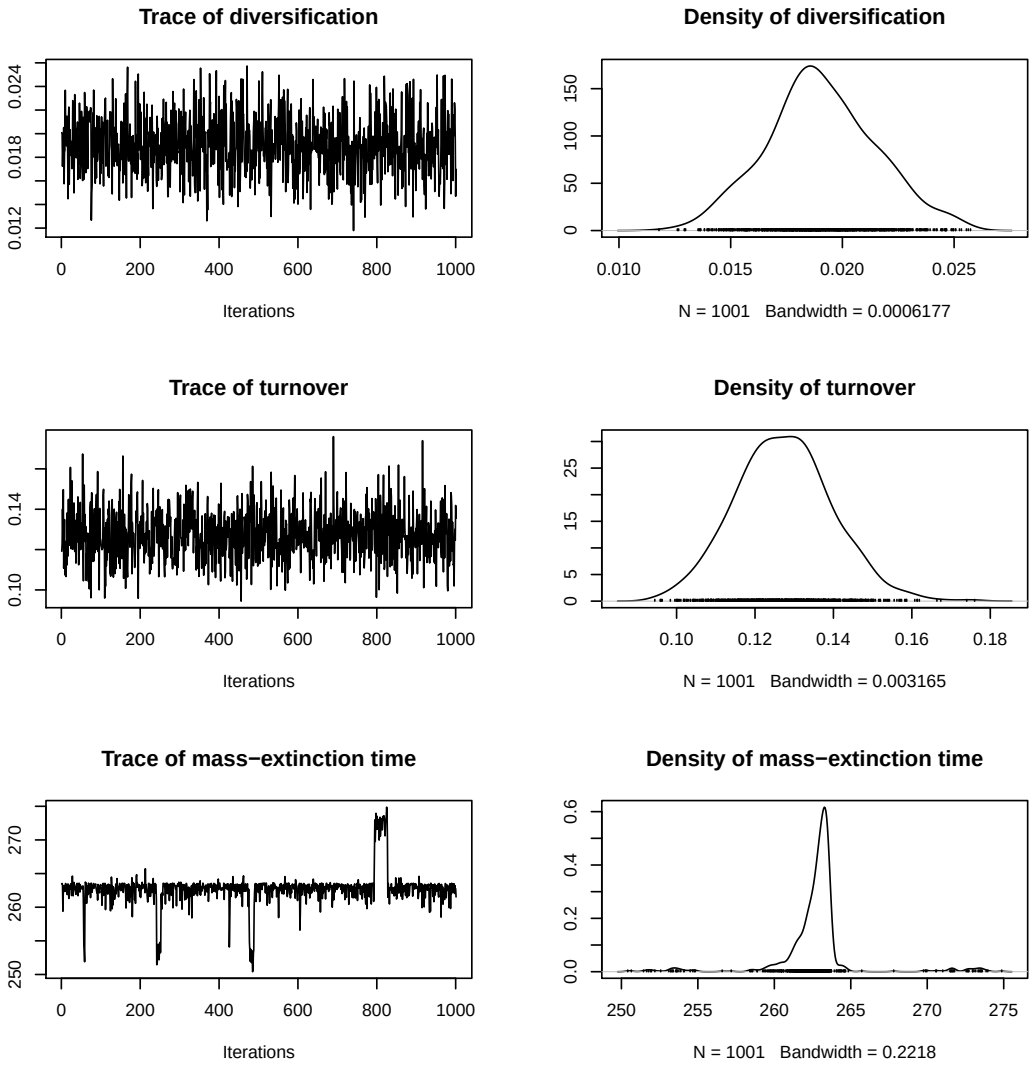
```
summary(samplesMassExtinctionBD)

##
```

```
## Iterations = 1:1001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                           Mean       SD  Naive SE Time-series SE
## diversification        0.01906 0.002448 7.737e-05      8.239e-05
## turnover               0.12733 0.012369 3.909e-04      3.895e-04
## mass-extinction time 262.74704 2.569482 8.121e-02      4.031e-01
##
## 2. Quantiles for each variable:
##
##                           2.5%      25%       50%       75%     97.5%
## diversification        0.01446   0.0175   0.01891   0.02061   0.02424
## turnover               0.10407   0.1190   0.12698   0.13496   0.15198
## mass-extinction time 254.77080 262.2503 262.96772 263.36666 271.51602
```

Finally, we visualize the trace plots and marginal posterior probability densities for these samples:

```
plot(samplesMassExtinctionBD)
```


Trace of diversification
0.012
0.018
0.024
0
200
400
600
800
1000
Iterations
Density of diversification
0
50
100
150
0.010
0.015
0.020
0.025
N = 1001 Bandwidth = 0.0006177
Trace of turnover
0.10
0.14
0
200
400
600
800
1000
Iterations
Density of turnover
0
5
15
25
0.10
0.12
0.14
0.16
0.18
N = 1001 Bandwidth = 0.003165
Trace of mass−extinction time
250
260
270
0
200
400
600
800
1000
Iterations
Density of mass−extinction time
0.0
0.2
0.4
0.6
250
255
260
265
270
275
N = 1001 Bandwidth = 0.2218

3 Accommodating Incomplete Taxon Sampling

Most phylogenies do not contain all species of the group under study. Instead, only an *incomplete sample*—or subsample—of all described species are included. Assuming complete taxon sampling for trees that are actually incomplete is known to bias estimates of diversification rates (Cusimano and Renner, 2010). Additionally, the sampling strategy (*e.g.,* whether species are sampled uniformly at random or to maximize diversity) also influences the parameter estimates (Höhna et al., 2011). Fortunately, methods for modeling incomplete taxon sampling exist to correct for the introduced bias (Cusimano and Renner, 2010; Höhna et al., 2011; Cusimano et al., 2012; Stadler and Bokma, 2013; Höhna, 2014).

Here we consider two approach of incomplete taxon sampling: *uniform* sampling and *diversified* sampling. A sketch of the two sampling methods was provided in Figure 4. As we will demonstrate below, the sampling strategy has a substantial influence on the distribution of branching times in the tree, which results in different patterns in the lineage-through-time curves for the different sampling schemes. Additionally, we note that the patterns induced by incomplete sampling can mimic patterns of decreasing rates of lineage diversification (Pybus and Harvey, 2000; Cusimano and Renner, 2010; Höhna et al., 2011), and thus it is critical to incorporate incomplete sampling in any study of lineage diversification rates.

We will begin by simulating trees under different kinds of sampling schemes, and then demonstrate how we can incorporate these sampling schemes into branching-process models in TESS.

3.1 Patterns of incomplete sampling

To demonstrate the impact of incomplete taxon sampling, we will simulate trees under each sampling strategy and plot the resulting LTT curves. We simulate $n = 50$ trees, each conditioned on the specified age. First, we simulate trees with complete taxon sampling to compare against the other sampling schemes. Next, we simulate trees under uniform taxon sampling with a sampling probability of $\rho = 0.25$ (which means that each species at the present has the same probability of $\rho = 0.25$ being included in the phylogeny; if a species is not sampled then its lineage is removed from the reconstructed tree). Finally, we simulate trees under diversified taxon sampling with a sampling probability of $\rho = 0.25$ (*i.e.,* only the oldest 25% of divergence events are included in the reconstructed phylogeny, and all later divergence events are excluded).

```
# Birth-death
birthDeathSpeciationSampling <- 2.0
birthDeathExtinctionSampling <- 1.0
```

```
birthDeathTreesComplete <- tess.sim.age(n = 50,
                                age = 3.0,
                                lambda = birthDeathSpeciationSampling,
                                mu = birthDeathExtinctionSampling,
                                MRCA = TRUE)

birthDeathTreesUniform <- tess.sim.age(n = 50,
                                age = 4.0,
                                lambda = birthDeathSpeciationSampling,
                                mu = birthDeathExtinctionSampling,
                                samplingProbability = 0.25,
                                samplingStrategy = "uniform",
                                MRCA = TRUE)

birthDeathTreesDiversified <- tess.sim.age(n = 50,
                                age = 4.0,
                                lambda = birthDeathSpeciationSampling,
                                mu = birthDeathExtinctionSampling,
                                samplingProbability = 0.25,
                                samplingStrategy = "diversified",
                                MRCA = TRUE)



par(mfrow=c(1,3),mar=c(5,4,3,0.1),las=1)

# Plot the trees
mltt.plot(birthDeathTreesComplete,log = "y",dcol = FALSE,
          legend = FALSE,backward = FALSE)
mtext("A", line = 1)

mltt.plot(birthDeathTreesUniform,log = "y",dcol = FALSE,
          legend = FALSE,backward = FALSE)
mtext("B", line = 1)

mltt.plot(birthDeathTreesDiversified,log = "y",dcol = FALSE,
          legend = FALSE,backward = FALSE)
mtext("C", line = 1)
```

For the remainder of this section, we focus on the biases stemming from incomplete

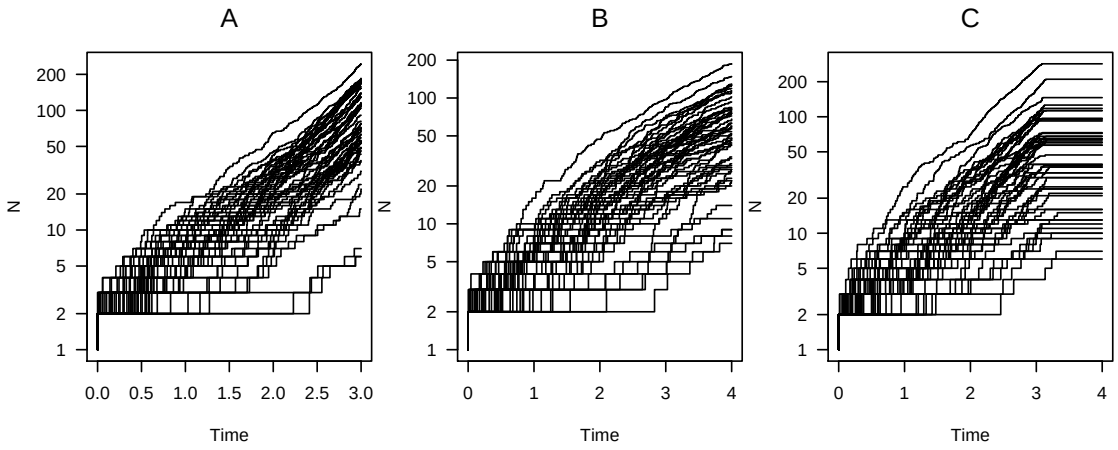


Figure 10: Lineage-through-time plots for completely sampled trees (panel A), incomplete trees with uniform sampling (panel B), and incomplete trees with diversified sampling (panel C).

taxon sampling. We will simulate a single incompletely sampled tree under a diversified sampling strategy with sampling fraction $\rho = 0.25$, and in the following sections we will estimate parameters under various birth-death processes from this tree. We have simulated the tree under diversified sampling with known speciation and extinction rates, which allows us to compare estimates of parameter values using various approaches to the true parameter values in order to better understand the influence of the sampling strategy on parameter estimates.

```
# simulate a tree under diversified taxon sampling
tree.diversified <- tess.sim.age(n = 1,
                                 age = 4.0,
                                 lambda = 2.0,
                                 mu = 1.0,
                                 samplingProbability = 0.25,
                                 samplingStrategy = "diversified",
                                 MRCA = TRUE)[[1]]

# extract the branching times from the tree
times.diversified <- as.numeric( branching.times(tree.diversified) )
```

First, we will take a look at the simulated tree: it looks similar to many empirical phylogenies.

```
plot(tree.diversified,show.tip.label=FALSE,no.margin=TRUE)
```

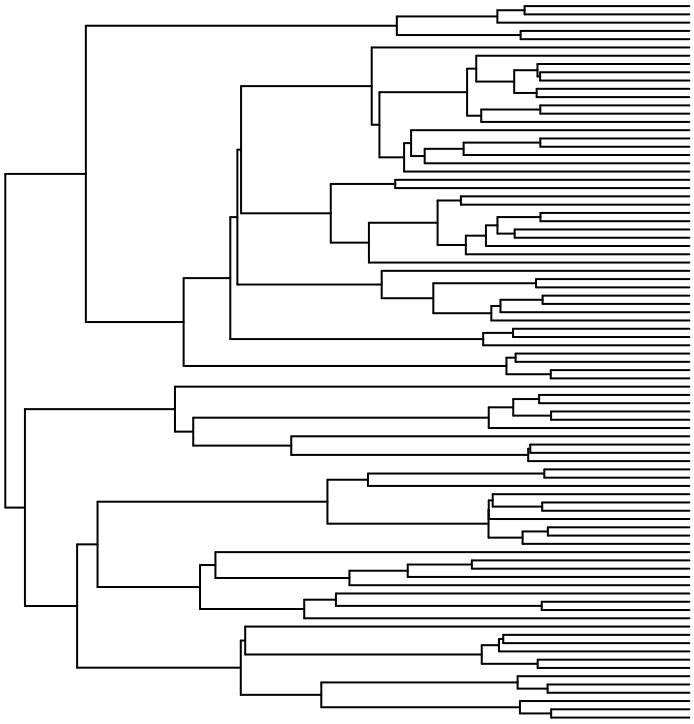


Figure 11: The simulated tree under *diversified* sampling with sampling fraction $\rho = 0.25$.

In the following sections we will infer diversification rates on this simulated tree.

3.2 Uniform taxon sampling

Uniform taxon sampling, which is sometimes also called *random* taxon sampling, assumes that every species at present has the same probability ρ of being included in the sample (Nee et al., 1994; Yang and Rannala, 1997; Stadler, 2009; Höhna et al., 2011; Höhna, 2014). That is, regardless of age or phylogenetic relationship, this method assumes that a researcher flips a coin for each species to decide whether it will be included in the analysis. This sampling scheme may not be realistic, as many factors typically influence the probability that a researcher will include a species in their study. However, the uniform taxon sampling scheme was initially adopted because it is mathematically convenient. Moroever, the approximation to uniform sampling scheme improves when the sampling fraction is large, *i.e.,* if more than 80% of the species are included.

In principle, we could treat the sampling probablity ρ as a random variable and estimate it from the data. However, estimating all three parameters of the sampled constant-rate birth-death process model—the speciation rate, the extinction rate and the sampling probability—is not possible because the parameters are nonidentifiable (Stadler, 2009). Therefore, we use the empirical sampling fraction; simply the number of included species divided by the total number of known species (630 for conifers).

```
# There are 630 known conifer species
samplingFraction <- (conifers$Nnode + 1) / 630
```

We can then use this empirical sampling fraction as our sampling probability. All the likelihood functions in TESS, as described in the previous sections, have an argument called `samplingProbability`. You can incorporate incomplete sampling in any of these analyses by setting this argument to the empirical sampling probability.

How well does the *uniform* taxon sampling scheme perform on our simulated phylogeny? We will assess the performance of this method by estimating the joint posterior density of the diversification-rate parameters by performing an MCMC simulation. As usual, we start by specifying the prior distributions. Here, we use the constant-rate birth-death process with the net-diversification rate (speciation − extinction) and turnover rate (extinction). We choose exponential prior distributions with a mean of 1.0, which corresponds to the true value. Notice that this is essentially an ideal setting, as the true parameter values are clearly unknown for empirical analyses. Therefore, this represents a best case scenario for the impact of taxon sampling on parameter estimates.

```
prior_delta <- function(x) { dexp(x,rate=1.0,log=TRUE) }
prior_tau <- function(x) { dexp(x,rate=1.0,log=TRUE) }
priorsSampling <- c("diversifiation"=prior_delta,
                    "turnover"=prior_tau)
```

We then define the likelihood function. This is similar to the likelihood function used previously for the constant-rate birth-death process. However, in this case we will specify the sampling probability ($\rho = 0.25$) and the sampling strategy.

```
likelihoodUniform <- function(params) {

  speciation <- params[1] + params[2]
  extinction <- params[2]
```

```
  lnl <- tess.likelihood(times.diversified,
                         lambda = speciation,
                         mu = extinction,
                         samplingProbability = 0.25,
                         samplingStrategy = "uniform",
                         log = TRUE)

  return (lnl)

}
```

Now we are ready to estimate the diversification-rate parameters. The settings for the MCMC simulation are the same as those used previously (*c.f.*, Section 2.4).

```
set.seed(12345)
samplesUniform <- tess.mcmc(likelihoodFunction = likelihoodUniform,
                            priors = priorsSampling,
                            parameters = runif(2,0,1),
                            logTransforms = c(TRUE,TRUE),
                            delta = c(1,1),
                            iterations = 10000,
                            burnin = 1000,
                            thinning = 10,
                            adaptive = TRUE,
                            verbose = TRUE)
```

Recall that the true parameter values are $\lambda = 2.0$ and $\mu = 1.0$. This correspods to a diversification rate of 1.0 and a turnover rate of 1.0. The estimated parameters under *uniform* taxon sampling are

```
summary(samplesUniform)

##
## Iterations = 1:1001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
```

```
##
##                   Mean     SD Naive SE Time-series SE
## diversifiation 0.8048 0.1641 0.005186       0.005614
## turnover       0.2726 0.2686 0.008489       0.008988
##
## 2. Quantiles for each variable:
##
##                     2.5%     25%    50%    75% 97.5%
## diversifiation 0.470512 0.70908 0.8002 0.9150 1.121
## turnover       0.007851 0.08001 0.1977 0.3699 1.009
```

```
plot(samplesUniform)
```

Our estimate of the diversification rate is actually quite good. This is because most information about the diversification rate comes from the age of the phylogeny and the number of sampled species, which does not depend on the sampling scheme or the divergence times. However, the estimated turnover rate is quite biased. As a result, the speciation- and extinction-rate estimates are both biased. This is caused by the underestimation of the extinction rate (Höhna et al., 2011). The bias is quite severe: the true values are not even contained in the 95% credible interval (see Figure 12).

3.3 Diversified taxon sampling

Now we will consider diversified sampling in more detail. We assume that our study group contains m species from which we have n sampled species. Under *diversified* sampling, this means that the most recent $m - n$ speciation events have been discarded. This is the strictly mathematical interpretation of *diversified* sampling (Höhna et al., 2011; Höhna, 2014).

This sampling strategy is intended to mimic empirical datasets where species were selected to include "representatives" from some number of distinct lineages (*e.g.,* all families or all genera). This sampling implicitly maximizes species diversity and comes close to the mathematical description of *diversified* sampling. However, diversified taxon sampling is not a perfect mathematical description of this sort of sampling. For example, not all "major lineages" are of the same age and size, and we may therefore sometimes include species that are recently diverged.

As we did for *uniform* taxon sampling, we want to test how well the method performs in estimating parameters given the simulated phylogeny. We will use the same constant-rate birth-death process with the only difference being the sampling strategy. We therefore use the same prior distributions as in the *uniform* sampling

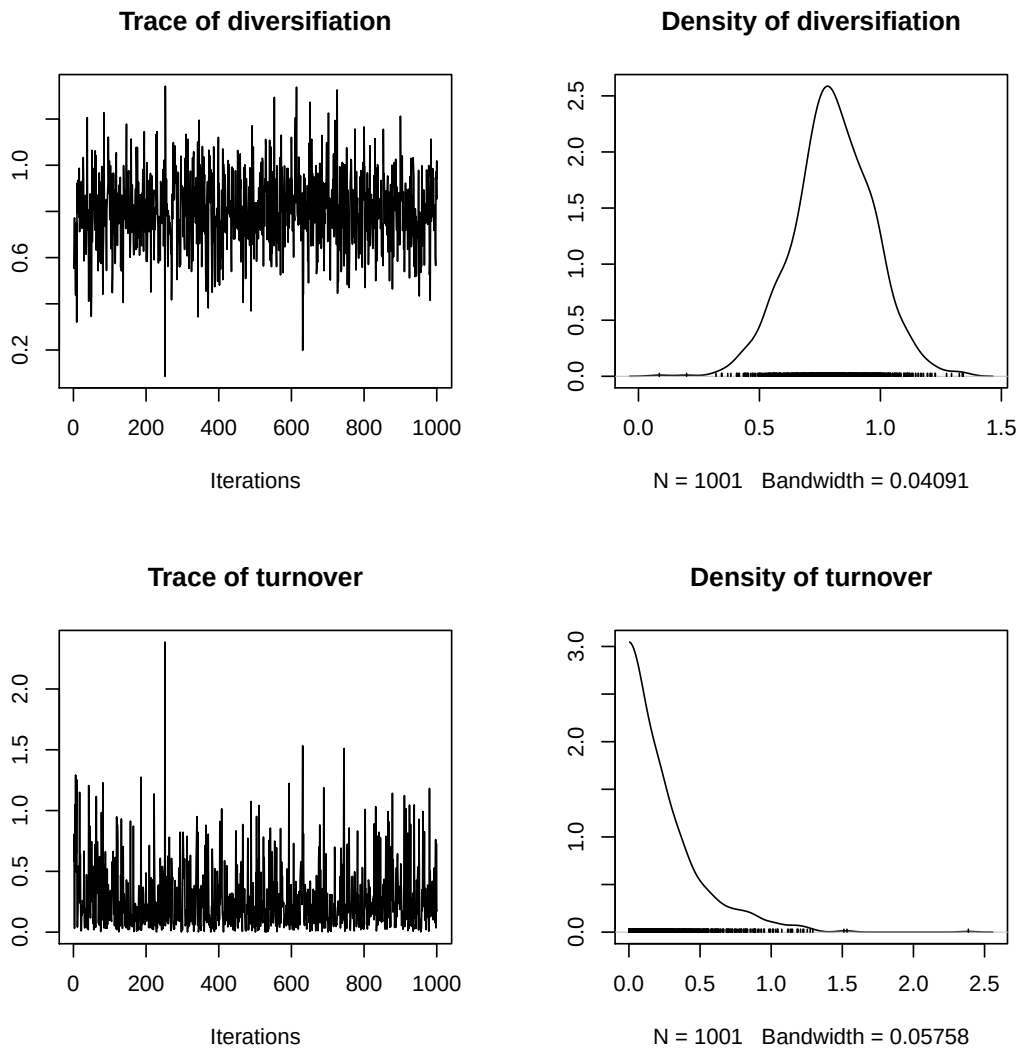


Figure 12: Trace plots (left) and marginal posterior probability densities (right) for the diversification rate (speciation - extinction) and turnover rate (extinction) under *uniform* sampling from the MCMC simulation.

analysis. The likelihood function is adapted by changing the `samplingStrategy` to "diversified".

```
likelihoodDiversified <- function(params) {

  speciation <- params[1] + params[2]
  extinction <- params[2]
```

```
  lnl <- tess.likelihood(times.diversified,
                         lambda = speciation,
                         mu = extinction,
                         samplingProbability = 0.25,
                         samplingStrategy = "diversified",
                         log = TRUE)

  return (lnl)

}
```

Then, we perform a (short) MCMC simulation to sample from the posterior distribution of the parameters.

```
samplesDiversified <- tess.mcmc(likelihoodFunction = likelihoodDiversified,
                                priors = priorsSampling,
                                parameters = runif(2,0,1),
                                logTransforms = c(TRUE,TRUE),
                                delta = c(1,1),
                                iterations = 10000,
                                burnin = 1000,
                                thinning = 10,
                                adaptive = TRUE,
                                verbose = TRUE)
```

Finally, our estimates of the diversification rate and turnover rate under *diversified* taxon sampling are

```
summary(samplesDiversified)

##
## Iterations = 1:1001
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1001
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                  Mean     SD Naive SE Time-series SE
## diversifiation 0.465 0.2903 0.009174        0.01462
```

```
## turnover       1.106 0.5884 0.018599        0.02829
##
## 2. Quantiles for each variable:
##
##                    2.5%    25%    50%    75% 97.5%
## diversifiation 0.03026 0.2277 0.4298 0.6754 1.052
## turnover       0.07549 0.6671 1.1259 1.4815 2.319

plot(samplesDiversified)
```

Here we see that the true values fall within the 95% credible interval. The mean estimate of the *diversification* rate might be slightly worse, but the transformed speciation and extinction rate estimates are significantly better. Thus, we can conclude that only if we know the true sampling strategy, and sampling fraction, are we able to make unbiased estimates the speciation and extinction rates.

You may wish to repeat the above experiment for the case when you simulate the tree under *uniform* taxon sampling, and/or when you assume an incorrect sampling fraction (*i.e.,* that is either too large or too small). For more information about incomplete taxon sampling, we refer the reader to Höhna et al. (2011) and Höhna (2014).

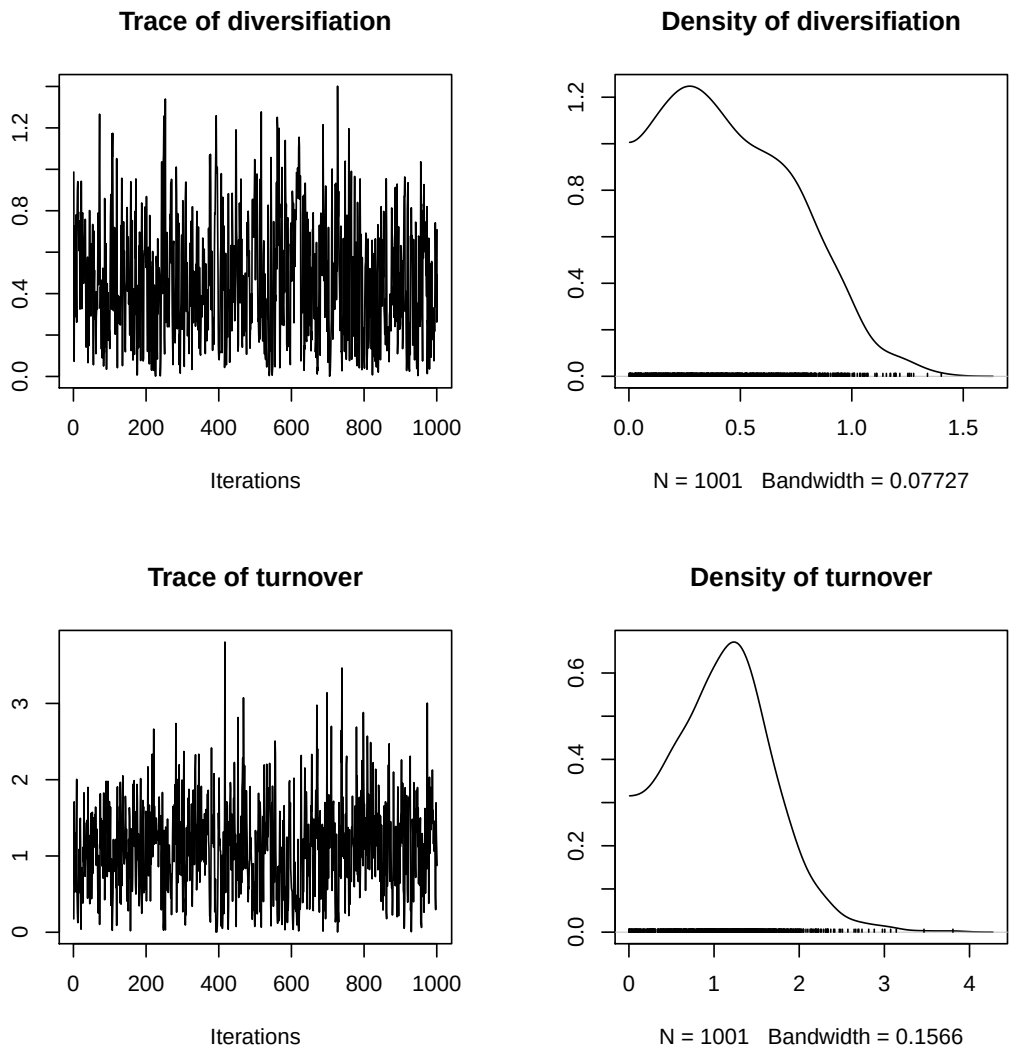


Figure 13: Trace plots (left) and marginal posterior probability densities (right) for the diversification rate (speciation - extinction) and turnover rate (extinction) under *diversified* sampling from the MCMC simulation.

4 Model Evaluation

Model-based inference is, by definition, based on the model. The model describes the process that gave rise to our observed data—in the present case, it describes the stochastic-branching process that gave rise to our study tree. Accordingly, if the model provides a 'poor fit' to the data—*i.e.,* provides a poor description of the process that gave rise to the observed data—then all bets are off.

The model must balance two competing criteria: (1) it must include the relevant parameters to describe important aspects of the diversification process, but; (2) it must exclude any superfluous parameters that only capture stochastic fluctuations in the data. Failure to satisfy criterion (1) will result in biased estimates of parameters; *e.g.,* speciation and extinction rates. Failure to satisfy criterion (2) will inflate the variance in the error of the parameter estimates (overconfidence in estimates).

Accordingly, we must be mindful both regarding our choice of model and also with respect to assessing our ability to perform reliable inference under the chosen model. Model evaluation entails three closely related issues. *Model selection* entails assessing the relative fit of our dataset to the pool of candidate models. In a Bayesian statistical framework, we compare the relative fit of candidate models based on their marginal likelihood (which measures the average fit of the candidate models to the data). *Model adequacy* entails assessing the absolute fit of the dataset to a given model. *Model uncertainty* is related to the common scenario when multiple candidate models provide a similar fit to the data and model averaging provides a method to (a) estimate the probability of a model being true given the set of available models, and (b) infers the parameters of interest by averaging over all specified models. Below, we demonstrate how to address each of these model-evaluation issues using TESS.

4.1 Comparing models with Bayes factors

For most groups of species, several (possibly many) branching-process models of varying complexity are plausible *a priori*. We therefore need a way to objectively identify the model that balances estimation bias and inflated error variance associated with under- and over-parameterized models, respectively. Increasingly, model selection is based on *Bayes factors* (*e.g.*, Kass and Raftery, 1995; Suchard et al., 2001; Lartillot and Philippe, 2006; Xie et al., 2011; Fan et al., 2011; Baele et al., 2012, 2013). This procedure requires that we first calculate the marginal likelihood of each candidate model, and we then compare the ratio of the marginal likelihoods for the set of candidate models.

Note that interpreting Bayes factors (BF) involves a measure of subjectivity. That is, it is up to you to decide what BF values appropriately reflect the level of significance in the competing models. Despite the absence of an absolutely objective model-selection threshold, we can refer to the scale (outlined by Jeffreys, 1961) that provides a "rule-of-thumb" for interpreting these measures (Table 2).

Given two candidate models, M_0 and M_1, the Bayes-factor comparison to assess

Table 2: The scale for interpreting Bayes factors by Harold Jeffreys (1961).

Strength of evidence	$BF(M_0, M_1)$	$\ln(BF(M_0, M_1))$	$\log_{10}(BF(M_0, M_1))$
Negative (supports M_1)	< 1	< 0	< 0
Barely worth mentioning	1 to 3.2	0 to 1.16	0 to 0.5
Substantial	3.2 to 10	1.16 to 2.3	0.5 to 1
Strong	10 to 100	2.3 to 4.6	1 to 2
Decisive	> 100	> 4.6	> 2

For a detailed description of Bayes factors see Kass and Raftery (1995)

the relative fit of each model to the data, $BF(M_0, M_1)$, is:

$$BF(M_0, M_1) = \frac{\mathbb{P}(M_0 \mid \mathbf{X})}{\mathbb{P}(M_1 \mid \mathbf{X})} = \frac{\mathbb{P}(M_0)}{\mathbb{P}(M_1)} \frac{\mathbb{P}(\mathbf{X} \mid M_0)}{\mathbb{P}(\mathbf{X} \mid M_1)},$$

where $\mathbb{P}(\mathbf{X} \mid M_i)$ is the *marginal likelihood* of the data (this may be familiar to you as the denominator of Bayes Theorem, which is variously referred to as the *model evidence* or *integrated likelihood*). Formally, the marginal likelihood is the probability of the observed data ($\mathbf{X}$) under a given model (M_i) that is averaged over all possible values of the parameters of the model (θ_i) with respect to the prior density on θ_i

$$\mathbb{P}(\mathbf{X} \mid M_i) = \int \mathbb{P}(\mathbf{X} \mid \theta_i)\mathbb{P}(\theta_i)d\theta. \tag{5}$$

This makes it clear that more complex (parameter-rich) models are penalized by virtue of the associated prior: each additional parameter entails integration of the likelihood over the corresponding prior density.

Exact solutions for calculating marginal likelihoods are not avaiable for most models, which requires that we resort to numerical integration methods to approximate these values. Below, we first provide a brief description of a robust method for estimating marginal likelihoods—stepping-stone simulation (Xie et al., 2011; Fan et al., 2011)—and then demonstrate how to use the implementation of the stepping-stone algorithm in TESS to estimate the marginal likelihoods for two birth-death branching-process models.

4.1.1 Stepping-stone simulation

Recent developments provide robust methods for estimating marginal likelihoods, including stepping-stone (Xie et al., 2011; Fan et al., 2011) and path-sampling estimators (Lartillot and Philippe, 2006; Baele et al., 2012). These algorithms are similar to the familiar MCMC algorithms, which are intended to sample from

(and estimate) the joint posterior probability of the model parameters. Stepping-stone algorithms are like a series of MCMC simulations that iteratively sample from a specified number of discrete steps between the posterior and the prior probability distributions. The basic idea is to estimate the probability of the data for all points between the posterior and the prior—effectively summing the probability of the data over the prior probability of the parameters to estimate the marginal likelihood. Technically, the steps correspond to a series of *power-posteriors*: a series of numbers between 1 and 0 that are iteratively applied to the posterior. When the posterior probability is raised to the power of 1 (typically the first stepping stone), samples are drawn from the (untransformed) posterior. By contrast, when the posterior probability is raised to the power of 0 (typically the last stepping stone), samples are drawn from the prior (Figure 14).

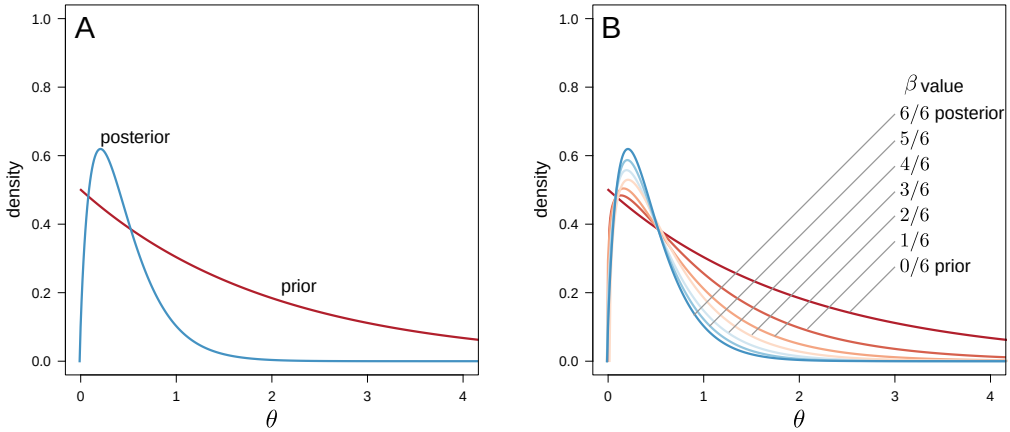


Figure 14: Estimating marginal likelihoods using stepping-stone simulation. Estimating the marginal likelihood involves integrating the likelihood of the data over the entire prior probability density for the model parameters. MCMC algorithms target the posterior probability density, which is typically concentrated in a small region of the prior probability density (A). Accordingly, standard MCMC simulation cannot provide unbiased estimates of the marginal likelihood because it will typically fail to explore most of the prior density. (B) Stepping-stone algorithms estimate the marginal likelihood by means of a series of MCMC-like simulations, where the likelihood is iterativey raised to a series of powers, effectivey forcing the simulation to more fully explore the prior density of the model parameters. Six uniformly spaced stones span the posterior, where the power posterior is $\beta = 6/6 = 1$, to the prior, where the power posterior is $\beta = 0/6 = 0$.

To perform a stepping-stone simulation, we need to specify (1) the number of stepping stones (power posteriors) that we will use to traverse the path between the posterior and the prior (*e.g.,* we specify 50 or 100 stones), (2) the spacing of the stones between the posterior and prior (*e.g.,* we may specify that the stones are distributed according to a beta distribution), (3) the number (and thinning)

of samples to be drawn from each stepping stone, and

This method computes a vector of powers from a beta distribution, then executes an MCMC run for each power step while raising the likelihood to that power. As implementated in TESS, the vector of powers starts with 1, initially sampling the likelihood close to the posterior, and incrementally sampling closer and closer to the prior as the simulation progresses across the stepping stones.

4.1.2 Estimating marginal likelihoods of birth-death models

To estimate the marginal likelihoods of the branching-process models, we will again make use of the corresponding likelihood functions that we defined in the previous section. However, rather than using the `tess.mcmc` function to sample from (and so estimate) the posterior distribution of model parameters, we will use the `tess.steppingStoneSampling` function to estimate the marginal likelihood of the data under the various models. The commands to execute this function for each of the three branching-process models are as follows:

```
set.seed(12345)
marginalLikelihoodConstBD <- tess.steppingStoneSampling(
                likelihoodFunction = likelihoodConstBD,
                priors = priorsConstBD,
                parameters = runif(2,0,1),
                logTransforms = c(TRUE,TRUE),
                iterations = 1000,
                burnin = 100,
                K = 50)

marginalLikelihoodDecrBD <- tess.steppingStoneSampling(
                likelihoodFunction = likelihoodDecrBD,
                priors = priorsDecrBD,
                parameters = runif(3,0,1),
                logTransforms = c(TRUE,TRUE,TRUE),
                iterations = 1000,
                burnin = 100,
                K = 50)

marginalLikelihoodEpisodicBD <- tess.steppingStoneSampling(
                likelihoodFunction = likelihoodEpisodicBD,
                priors = priorsEpisodicBD,
                parameters = runif(4,0,1),
                logTransforms = c(TRUE,TRUE,TRUE,TRUE),
```

```
                  iterations = 1000,
                  burnin = 100,
                  K = 50)

marginalLikelihoodMassExtinctionBD <- tess.steppingStoneSampling(
                  likelihoodFunction = likelihoodMassExtinctionBD,
                  priors = priorsMassExtinctionBD,
                  parameters = c(runif(2,0,1),max(times)*3/4),
                  logTransforms = c(TRUE,TRUE,FALSE),
                  iterations = 1000,
                  burnin = 100,
                  K = 50)
```

We can now use the estimated marginal likelihoods to perform Bayes factor comparisons of these three candidate branching-process models.

```
# First, construct a vector of the marginal likelhoods named by the
# model to which they refer.
candidateModels <- c("ConstBD"=marginalLikelihoodConstBD,
                     "DecrBD"=marginalLikelihoodDecrBD,
                     "EpisodicBD"=marginalLikelihoodEpisodicBD,
                     "MassExtinctionBD"=marginalLikelihoodMassExtinctionBD)

# Make all possible combinations of the models.
marginalLikelihoodGrid <- expand.grid(M0=names(candidateModels),
                                      M1=names(candidateModels))

# Add a column that is the 2 ln BF for each pair of models.
marginalLikelihoodGrid$BF <- 2 * (candidateModels[marginalLikelihoodGrid$M0] -
                                  candidateModels[marginalLikelihoodGrid$M1])

# Sort the comparisons by their 2 ln BF in descending order.
marginalLikelihoodGrid <- marginalLikelihoodGrid[order(marginalLikelihoodGrid$BF,
                                                       decreasing=TRUE),]

marginalLikelihoodGrid
```

```
##                   M0                M1         BF
## 13          ConstBD MassExtinctionBD  29.362497
## 15       EpisodicBD MassExtinctionBD  24.678734
## 14           DecrBD MassExtinctionBD  21.509682
```

```
## 5           ConstBD           DecrBD   7.852815
## 9           ConstBD       EpisodicBD   4.683763
## 7         EpisodicBD           DecrBD   3.169052
## 1           ConstBD          ConstBD   0.000000
## 6            DecrBD           DecrBD   0.000000
## 11        EpisodicBD       EpisodicBD   0.000000
## 16 MassExtinctionBD MassExtinctionBD   0.000000
## 10           DecrBD       EpisodicBD  -3.169052
## 3         EpisodicBD          ConstBD  -4.683763
## 2            DecrBD          ConstBD  -7.852815
## 8   MassExtinctionBD           DecrBD -21.509682
## 12 MassExtinctionBD       EpisodicBD -24.678734
## 4   MassExtinctionBD          ConstBD -29.362497
```

If we compare these computed Bayes factor values to the thresholds in Table 2, we see that there is decisive support for the constant-rate model (*e.g.,* this model is decisively preferred over either of the variable-rate models; BF $\gg$ 4.6). Furthermore, we see that the decreasing-rate model is decisively preferred over the episodic model.

4.2 Assessing model adequacy with posterior predictive simulation

Bayes factors, dicussed in the previous section, allow us to assess the relative fit of two or more competing models to a given dataset. However, even the very best of the competing models may nevertheless be inadequate in an absolute sense. Fortunately, we can assess the absolute fit of a candidate model to a given dataset using *posterior-predictive simulation*. The basic premise of this approach is as folows: if the model under consideration provides an adequate description of the process that gave rise to our observed dataset, then we should be able to use that model generate new datasets that are in some sense 'similar' to our dataset.

4.2.1 Posterior-predictive simulation

Posterior-predictive simulation involves six main steps:

1. We first calculate a summary statistic for our observed dataset. This is intended to capture—in a single number—a relevant feature of our dataset.

For models of lineage diversification, for example, we might use number of species in the tree or the γ-statistic (Pybus and Harvey, 2000) as our summary statistic.

2. We then estimate parameters of the candidate model from our oberved dataset. This simply involves performing an MCMC simulation to estimate the posterior probability distribution of the candidate model parameters.

3. Next, we specify parameters of the candidate model by drawing values from the inferred joint posterior probability distributions. For example, we would parameterize the diversification model under consideration by drawing rate parameters from the joint posterior densities that we inferred from the study tree.

4. We then use this parameterized model to simulate a tree, and calculate the summary statistic for the resulting tree.

5. We repeat steps $3-4$ many times to generate a distribution of the summary statistic. This is the distribution that is predicted by simulating datasets under the candidate model that has been parameterized using posterior estimates on the observed dataset.

6. Finally, we compare the summary statistic calculated for the observed dataset to the posterior-predictive distribution. If the candidate model provides an adequate description of the process that gave rise to the original dataset, then the statistic for the observed dataset will fall near the center of the simulated distribution. Otherwise, the statistic from the observed data will fall near the tails of the null distribution, indicating that the model cannot be used to predict future data that look like the observed dataset.

We can formalize the relative position of the statistic for the observed data to the posterior-predictive distribution by calculating the *posterior-predictive p-value*. To do so, we simply sum the number of simulated summary statistics that are greater than or equal to the observed value, and divide this by the number of simulated values.

4.2.2 Assessing the adequacy of branching-process models

To assess the adequacy of branching-process models in TESS, we will use the number of species and the γ-statistic (Pybus and Harvey, 2000) as our example test statistics. We will demonstrate how to perform posterior-predictive simulation to assess the absolute fit of the constant-rate birth-death process model to the

conifer dataset. Note that we have already estimated the joint posterior probability distribution of this model using MCMC (described in Section 2.4.1, above).

We will condition our simulated trees on the age of conifer phylogeny.

```
tmrca <- max( times )
```

We first define the function that will perform the simulation. This is analogous to the specification of the likelihood function, which means that it is possible to perform posterior-predictive simulation under any birth-death model.

```
# The simulation function
simConstBD <- function(params) {

  # Same model as above.
  speciation <- params[1] + params[2]
  extinction <- params[2]

  # We need trees with at least three tips for the
  # gamma-statistic.
  repeat {
    tree <- tess.sim.age(n = 1,
                         age = tmrca,
                         lambda = speciation,
                         mu = extinction,
                         samplingProbability = 1.0,
                         MRCA = TRUE)[[1]]
    if (tree$Nnode > 1) break
  }
  return (tree)
}
```

Note that the simulation function needs to return a single tree. The next step is to simulate trees by sampling parameter values (for the speciation and extinction rate) from the corresponding posterior probability distributions that we inferred from the conifer tree (these are referred to as the 'posterior-predictive samples').

```
# simulate trees from the posterior-predictive distribution
treesConstBD <- tess.PosteriorPrediction(simConstBD,samplesConstBD)
```

We will specify the number of species as the summary statistic:

```
# compute the number of species in each simulate tree
numTaxaConstBD <- c()
for (i in 1:length(treesConstBD)){
  numTaxaConstBD[i] <- treesConstBD[[i]]$Nnode + 1
}
```

We then compute the posterior-predictive quantiles for the number of species and plot the posterior-predictive distribution and quantiles. Finally, we compare the observed number of species to the posterior-predictive distribution.

```
# Compute the 95% posterior-predictive interval of the
# number of taxa.
numTaxaPPDI <- quantile(numTaxaConstBD,prob=c(0.025,0.975))

# Plot the posterior-predictive distribution with the
# quantiles. Then, compare it to the observed number
# of species, x.
plot(density(numTaxaConstBD),main="Number of taxa",xlab="",
    ylab="Posterior Predictive Density",lwd=2)

abline(v=numTaxaPPDI,lty=2,col="gray",lwd=2)

points(conifers$Nnode+1,0,pch="x")
```

We can plot the posterior-predictive distribution of the lineage-accumulation curves (*i.e.*, the LTT plots for the simulated trees), and compare this predictive distribution to the LTT plot for the observed tree.

```
ltt.plot(treesConstBD[[1]],backward=FALSE,col="gray",log="y",
         ylim=c(1,max(numTaxaConstBD)),main="LTT-plot")

for (i in 2:min(100,length(treesConstBD))) ltt.lines(treesConstBD[[i]],
                            backward=FALSE, col="gray")

ltt.lines(conifers,backward=FALSE,lwd=3)
```

Additional to using species number as the summary statistic, we can also use the gamma statistic. To do so, we simply compute the value of the gamma statistic for the observed tree, and then compare it to the posterior-predictive distribution of the gamma statistic (*i.e.*, the distribution of γ-statistics computed from the simulated trees).

```
# Compute the observed gamma statistic.
observedGamma <- gammaStat(conifers)

# Perform the posterior predictive test, and compute
# the 95% posterior predictive interval.
ppt <- tess.PosteriorPredictiveTest(treesConstBD,conifers,
                                    gammaStat)
gammaPPDI <- quantile(ppt[[1]],prob=c(0.025,0.975))

# Compare the observed statistic to the posterior
# predictive density.
plot(density(ppt[[1]]),main="Gamma Statistic",xlab="",
                       ylab="Posterior Predictive Density",lwd=2)
abline(v=gammaPPDI,lty=2,col="gray",lwd=2)
points(observedGamma,0,pch="x")
```

Encouragingly, the observed values for both summary statistics—the number of species and the gamma statistic—fall near the center of their respective posterior-predictive distributions (see Figure 15 A and C, respectively). This means that the model under consideration—the constant-rate birth-death model—can be used to simulate trees that look like our conifer study tree, indicating that it provides a good absolute fit to our dataset. Conversely, if we had found that the observed values for the summary statistics fell outside the 95% credible intervals of the posterior-predictive distributions, we would conclude that the contant-rate birth-death model cannot be used to predict trees that look like our conifer stuy tree.

We could further quantify the relative position of the observed summary statistic within the posterior-predictive distribution by calculatig the posterior-predictive p-value as follows:

```
mean(ppt[[1]] >= observedGamma)

## [1] 0.2396804
```

These posterior-predictive p-values can be used to assess the absolute fit of one or more models to a given dataset.

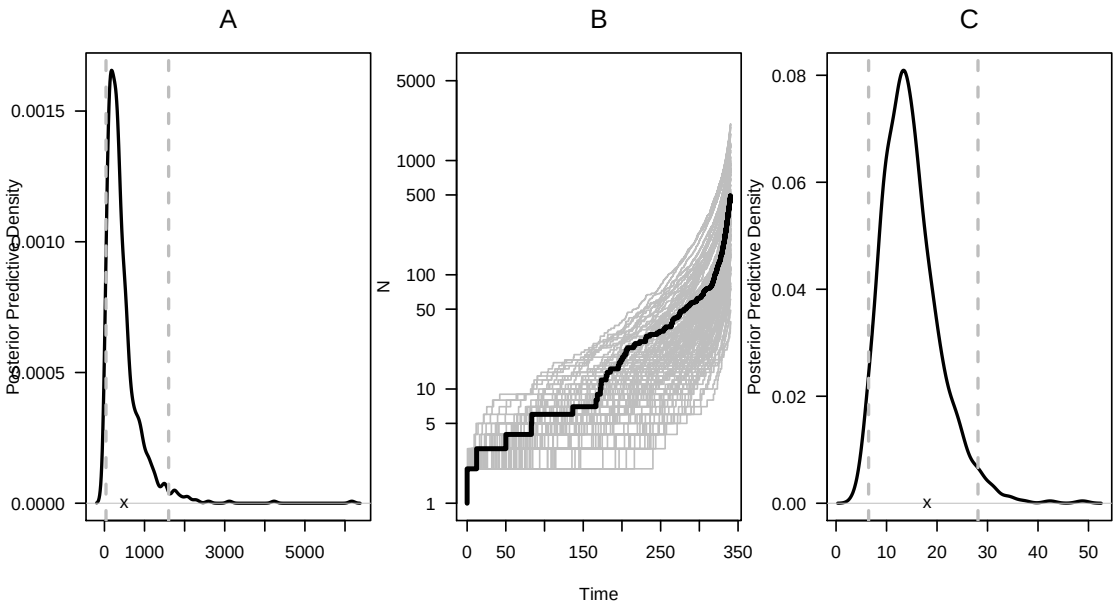


Figure 15: Assessing the absolute fit of the conifer tree to the constant-rate birth-death model using posterior-predictive simulation. (A) The posterior-predictive distribution for the number of species; the dashed gray lines indicate the 95% credible interval, and the 'x' indicates the location of the observed species number. (B) LTT plots for the simulated trees (gray) and for the conifer study tree (black). (C) The posterior-predictive distribution for the gamma statistic; the dashed gray lines indicate the 95% credible interval, and the 'x' indicates the location of the value of the gamma statistic calculated for the conifer tree.

4.3 Model selection with reversible-jump MCMC and CoMET

In the preceding sections, we assessed the relative and absolute fit of birth-death process models using Bayes factors and posterior-predictive simulation. These methods are computationally intensive and so are only practical when the set of candidate branching-process models is quite small; however, the number of total birth-death models is vast. Consider, for example, the episodic birth-death process models: there is an infinite number of nested models that differ in the number of events, and for a specific number of events, there is an infite number of times at which those events could occur. Clearly, the vast space of possible branching-process models precludes their exhaustive pairwise comparison using Bayes factors. This issue may be addressed by means of reversible-jump MCMC (rjMCMC) approaches that treat the model as a random variable (*i.e.,* the number of mass-extinction events). Additionally, the rjMCMC approach in `CoMET` integrates over the (model) uncertainty of the nuisance parameters: the number of diversification rate shifts is integrated over if you are interested only in the number of mass-extinction events.

The `CoMET` method performs rjMCMC simulation over all possible episodically varying birth-death processes with explicitly modeled mass-extinction events. Briefly, this method treats the number of specation-rate shifts, extinction-rate shifts, and mass-extinction events—as well as the parameters associated with these events—as random variables, and estimates their joint posterior distribution. To perform a full `CoMET` analysis, we must therefore specify values for the following quantities: (1) the expected number of speciation-rate shifts, extinction-rate shifts, and mass-extinction events ($\lambda_{\mathbb{B}}, \lambda_{\mathbb{D}}, \lambda_{\mathbb{M}}$, respectively); (2) the hyperpriors describing the speciation and extinction rates ($\mu_{\mathbb{B}}, \sigma_{\mathbb{B}}$ and $\mu_{\mathbb{D}}, \sigma_{\mathbb{D}}$, respectively); and (3) the hyperpriors describing the mass-extinction survival probability (α, β). The settings and accompanying arguments are summarized in Table 3. Note that the method currently assumes that $\lambda_{\mathbb{B}} = \lambda_{\mathbb{D}}$, although the actual number, timing and magnitude of speciation- and extinction-rate shifts are independent of each other. For the full details of this method, and a more complete description of the model parameters, see May et al. (2015).

4.3.1 Specifying hyperpiors *a priori*

Before we can run a `CoMET` analysis, we must specify prior distributions for each of the parameters in the model. We will begin with the most complex model in this section, and then show how to specify special cases of the full model in later sections.

We will start by specifying the prior distributions for the expected number of speciation- and extinction-rate shifts, $\lambda_{\mathbb{B}} = \lambda_{\mathbb{D}}$, and mass-extinction events

Table 3: Settings for the `CoMET` model. The left column lists the priors and hyperpriors used by the `CoMET` model. The middle column lists the associated arguments for use with the `tess.analysis` command. The right column lists the interpretation of the prior or hyperprior.

Prior	Argument	Interpretation
$\lambda_{\mathbb{B}}$	`numExpectedRateChanges`	Expected number of speciation-rate shifts
$\lambda_{\mathbb{D}}$	`numExpectedRateChanges`	Expected number of extinction-rate shifts
$\lambda_{\mathbb{M}}$	`numExpectedMassExtinctions`	Expected number of mass-extinction events
$\mu_{\mathbb{B}}$	`speciationRatePriorMean`	Mean speciation rate
$\sigma_{\mathbb{B}}$	`speciationRatePriorStDev`	Standard deviation of the speciation rate
$\mu_{\mathbb{D}}$	`extinctionRatePriorMean`	Mean extinction rate
$\sigma_{\mathbb{D}}$	`extinctionRatePriorStDev`	Standard deviation of the extinction rate
α	`pMassExtinctionPriorShape1`	Shape parameter of the expected survival probability
β	`pMassExtinctionPriorShape2`	Shape parameter of the expected survival probability

$\lambda_{\mathbb{M}}$. As it turns out, the prior expectation of the number of events does not impact our conclusions because we will use Bayes factors which cancel out the prior assumptions. Thus, the specific prior choices only tweak the performance of the method but should not result in qualitatively different conclusions. Accordingly, we specify $\lambda_{\mathbb{M}}$ to reflect the fossil record as follows:

```
numExpectedMassExtinctions <- 2
```

Similarly, we specify the prior expectation on the number of rate-shift events:

```
numExpectedRateChanges <- 2
```

Next, we consider the prior densities for the diversification-rate parameters themselves. Speciation and extinction rates must be greater than 0, so we will use lognormal prior densities to reflect this fact. We must therefore specify the mean and standard deviation hyperparameters of each lognormal distribution. We start by specifying the hyperparameters in real space, *i.e.,* the mean and standard deviation of the actual speciation and extinction rates.

```
# Specify the mean and standard deviation of the lognormal
# prior on the speciation rate in real space
speciationPriorMu <- 0.2
speciationPriorSigma <- 0.5

# Specify the mean and standard deviation of the lognormal
# prior on the extinction rate in real space
extinctionPriorMu <- 0.15
extinctionPriorSigma <- 0.5
```

We then transform the hyperparameters to reflect the mean and standard of the log-transformed speciation and extinction rates. These are the μ and σ parameters of the lognormal priors on speciation and extinction rates.

```
# Transform the priors on the speciation rate into log space.
speciationRatePriorMean <- log((speciationPriorMu^2)
                           /sqrt(speciationPriorSigma^2+
                           speciationPriorMu^2))

speciationRatePriorStDev <- sqrt( log(1+speciationPriorSigma^2
                            /(speciationPriorMu^2)))

# Transform the priors on the extinction rate into log space.
extinctionRatePriorMean <- log((extinctionPriorMu^2)
                           /sqrt(extinctionPriorSigma^2+
                           extinctionPriorMu^2))

extinctionRatePriorStDev <- sqrt( log(1+extinctionPriorSigma^2
                            /(extinctionPriorMu^2)))
```

Finally, we need to specify the prior density on the survival probability of a mass-extinction event. This value reflects the probability that a lineage survives a particular mass-extinction event, and therefore must be between 0 (each lineage will always go extinct) and 1 (each lineage will always survive). A convenient prior density for this parameter is the beta distribution, which has two shape parameters, α and β. We use the definition of what constitutes to a mass extinction to inform our prior density on this parameter.

We begin parameterizing this distribution by specifying the expected survival probability. The fossil record suggests that between 95% of species diversity was lost during the a major mass-extinction events and 70% during a minor mass-

extinction event. Therefore, the expected survival probability should be quite low. In this example, we assume *a priori* a survival probability of 5%.

```
expectedSurvivalProbability <- 0.05
```

Using the expected survival probability, we compute the α and β parameters of the beta distribution. We set the value of β to be large, which focuses the prior density more tightly around the expected survival probability. Then, we compute α based on the expected survival probability and the specified β value.

```
pMassExtinctionPriorShape2 <- 100
pMassExtinctionPriorShape1 <- - pMassExtinctionPriorShape2 *
                                expectedSurvivalProbability /
                                (expectedSurvivalProbability - 1)
```

We can inspect this beta distribution to confirm that it accurately reflects our prior belief regarding the survival probability (Figure 16).

```
# Plot the density function of our beta distribution.
curve(dbeta(x,shape1=pMassExtinctionPriorShape1,
            shape2=pMassExtinctionPriorShape2),n=1001,
            xlab='survival probability',ylab='density',las=1)

# Plot the 95% prior interval on the survival probability.
abline(v = qbeta(c(0.025,0.975),shape1=pMassExtinctionPriorShape1,
                 shape2=pMassExtinctionPriorShape2),lty=2)
```

This beta distribution seems to reflect our prior belief that the survival probability is expected to be $\sim 5\%$, but can range from about 1% to about 10%.

Having specified prior distributions for all of the parameters of the `CoMET` model, we can now perform an analysis.

```
set.seed(12345)
tess.analysis(conifers,
              empiricalHyperPriors = FALSE,
              initialSpeciationRate = speciationPriorMu,
              speciationRatePriorMean = speciationRatePriorMean,
              speciationRatePriorStDev = speciationRatePriorStDev,
              initialExtinctionRate = extinctionPriorMu,
              extinctionRatePriorMean = extinctionRatePriorMean,
              extinctionRatePriorStDev = extinctionRatePriorStDev,
```

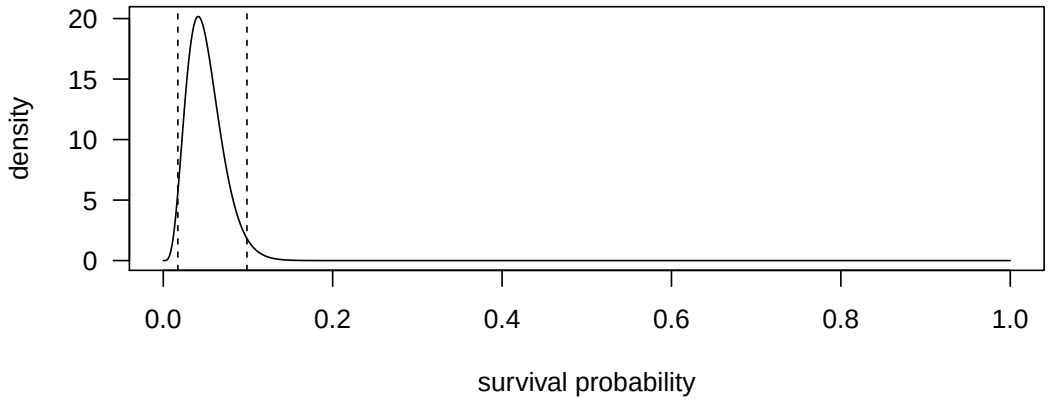


Figure 16: Our prior density on the survival probability of a mass-extinction event.

```
                samplingProbability = samplingFraction,
                numExpectedRateChanges = numExpectedRateChanges,
                numExpectedMassExtinctions = numExpectedMassExtinctions,
                pMassExtinctionPriorShape1 = pMassExtinctionPriorShape1,
                pMassExtinctionPriorShape2 = pMassExtinctionPriorShape2,
                MAX_ITERATIONS = 10000,
                dir = "tess_analysis")
```

Now, we process the output using the function `tess.process.output`

```
output <- tess.process.output("tess_analysis",
                  numExpectedRateChanges = numExpectedRateChanges,
                  numExpectedMassExtinctions = numExpectedMassExtinctions)
```

and visualize the results using `tess.plot.output` (Figure 17).

```
layout.mat <- matrix(1:6,nrow=3,ncol=2,byrow=TRUE)
layout(layout.mat)
tess.plot.output(output,
                 fig.types = c("speciation rates",
                               "speciation shift times",
                               "extinction rates",
                               "extinction shift times",
```

```
                                "mass extinction Bayes factors",
                                "mass extinction times"),
                las=2)
```

There appears to be nearly decisive support for a mass-extinction event about 27 million years ago ($2 \ln \mathrm{BF} \approx 10$), as well as strong support for a mass-extinction event about 173 million year ago ($2 \ln \mathrm{BF} \approx 6$). Additionally, there is support for an extinction-rate shift near the present; however, we caution against interpreting these rate shifts based on a single analysis and recommend assessing the sensitivity of this conclusion to different priors on the expected number of diversification-rate shifts.

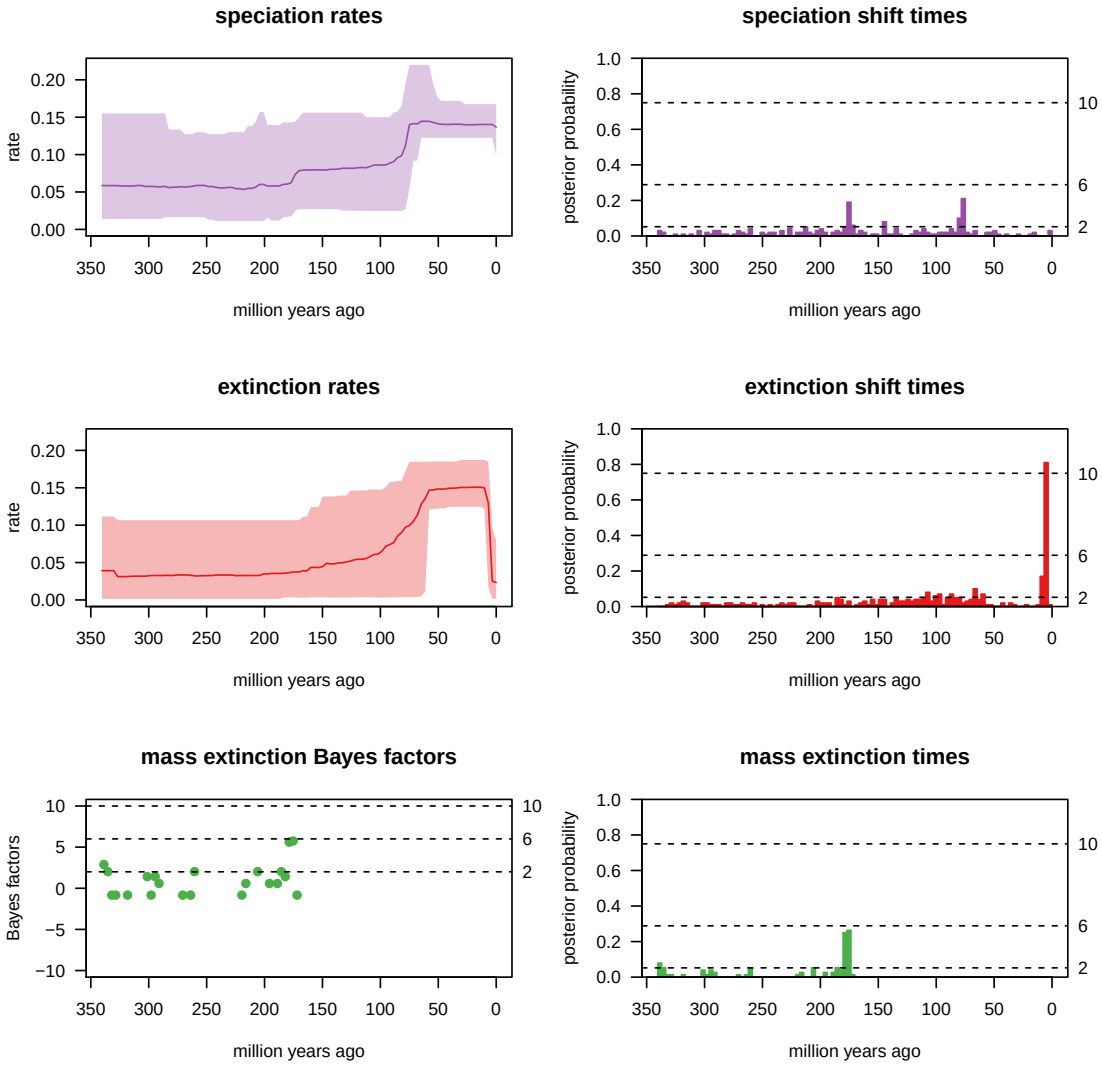


Figure 17: Visualizing the results of a `CoMET` analysis when diversification hyperpriors are specified *a priori*.

4.3.2 Empirical hyperpriors

As mentioned above, it can be difficult to specify the prior distributions for the speciation- and extinction-rate parameters. TESS implements an automatic empirical hyperprior procedure which performs an initial Bayesian MCMC analysis under a constant-rate birth-death process model to determine reasonable values for the hyperparameters of the diversification priors. To perform a hyperprior analysis, we simply have to set `empiricalHyperPriors = TRUE`. Additionally, we may omit the parameters of the lognormal distributions, since they will automatically be estimated from the data.

```
set.seed(12345)
tess.analysis(conifers,
              empiricalHyperPriors = TRUE,
              samplingProbability = samplingFraction,
              numExpectedRateChanges = numExpectedRateChanges,
              numExpectedMassExtinctions = numExpectedMassExtinctions,
              pMassExtinctionPriorShape1 = pMassExtinctionPriorShape1,
              pMassExtinctionPriorShape2 = pMassExtinctionPriorShape2,
              MAX_ITERATIONS = 10000,
              dir = "comet_hyperpriors")
```

As before, we process

```
output <- tess.process.output("comet_hyperpriors",
                numExpectedRateChanges = numExpectedRateChanges,
                numExpectedMassExtinctions = numExpectedMassExtinctions)
```

and visualize the output using `tess.process.output` and `tess.plot.output` (Figure 18).

```
layout.mat <- matrix(1:6,nrow=3,ncol=2,byrow=TRUE)
layout(layout.mat)
tess.plot.output(output,
                 fig.types = c("speciation rates",
                               "speciation shift times",
                               "extinction rates",
                               "extinction shift times",
                               "mass extinction Bayes factors",
                               "mass extinction times"),
                 las=2)
```

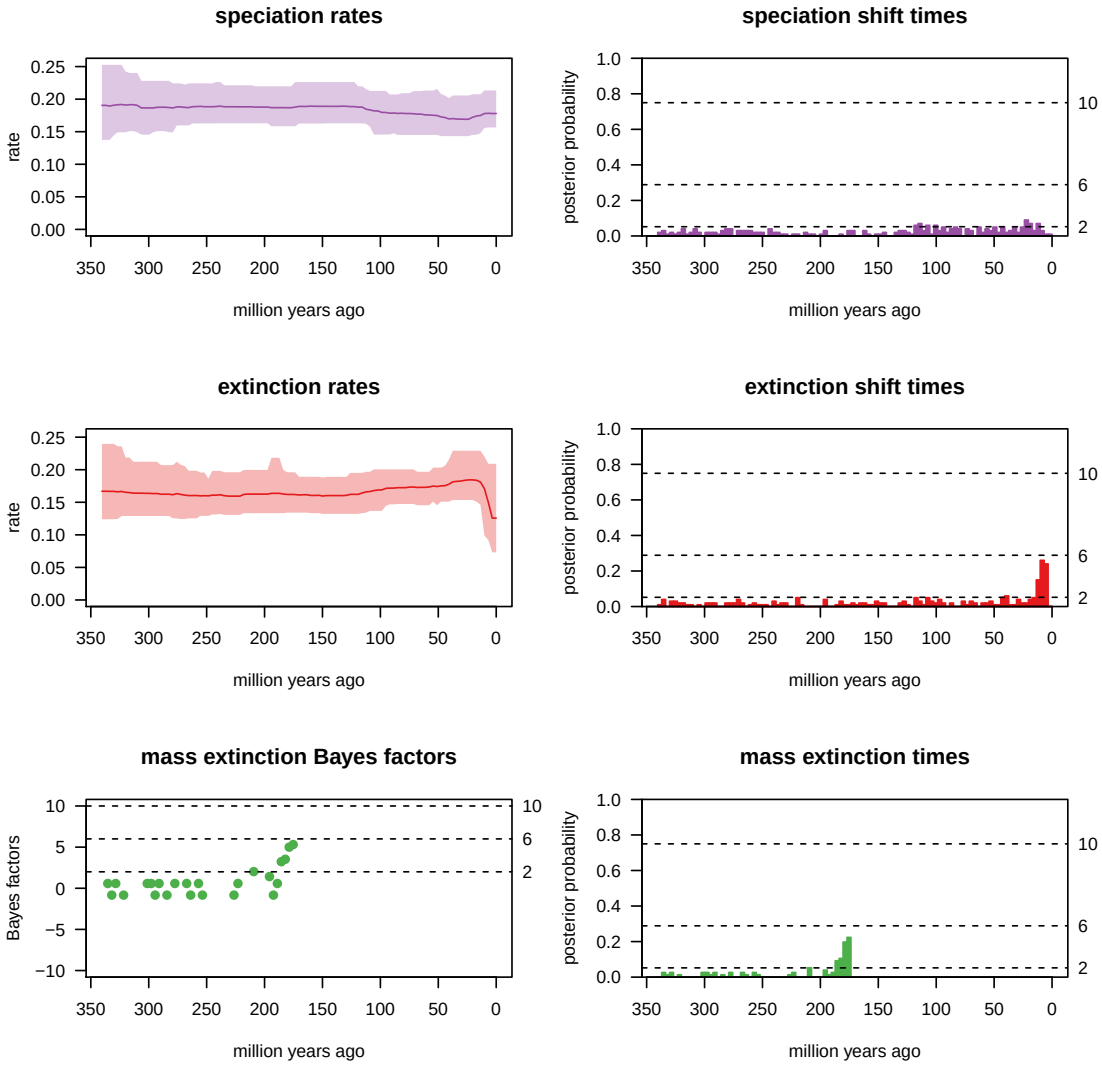


Figure 18: Visualizing the results of a `CoMET` analysis with empirically estimated diversification hyperpriors.

Interestingly, these results are quite similar to those from our *a priori* analysis (Figure 17).

4.3.3 Without diversification-rate shifts

Based on the results of the Bayes factor comparisons (Section 4.1.2) and posterior-predictive tests (Section 4.2.2)—which found support for a constant-rate birth-death process—we may also be interested in performing a `CoMET` analysis without diversification-rate shifts (*i.e.,* where speciation and extinction rates are constant through time). To do so, we use the argument `estimateNumberRateChanges = FALSE`.

```
set.seed(12345)
tess.analysis(conifers,
              empiricalHyperPriors = TRUE,
              samplingProbability = samplingFraction,
              estimateNumberRateChanges = FALSE,
              numExpectedMassExtinctions = numExpectedMassExtinctions,
              pMassExtinctionPriorShape1 = pMassExtinctionPriorShape1,
              pMassExtinctionPriorShape2 = pMassExtinctionPriorShape2,
              MAX_ITERATIONS = 10000,
              dir = "comet_no_rateshifts")

output <- tess.process.output("comet_no_rateshifts",
                numExpectedRateChanges = numExpectedRateChanges,
                numExpectedMassExtinctions = numExpectedMassExtinctions)
```

We visualize the output as before. However, since there are no diversification-rate shifts, we only plot estimates related to mass-extinction events (Figure 19).

```
layout.mat <- matrix(1:2,nrow=2,ncol=1)
layout(layout.mat)
tess.plot.output(output,
                 fig.types = c("mass extinction Bayes factors",
                               "mass extinction times"),
                 las=2)
```

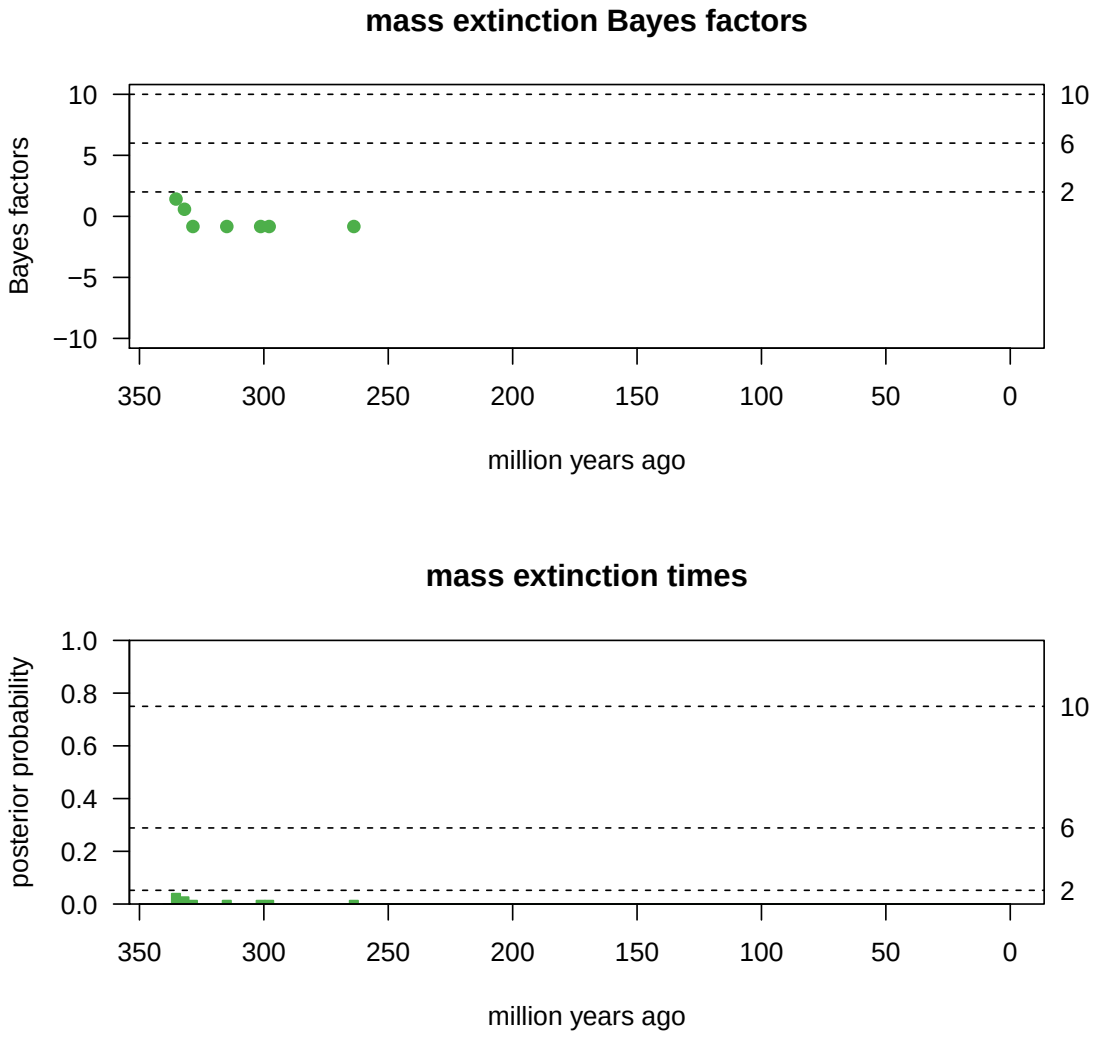


Figure 19: Visualizing the results of a `CoMET` analysis with empirically estimated diversification hyperpriors *and without* diversification rate-shifts.

4.3.4 Without mass-extinction events

Finally, we can also perform a `CoMET` analysis where mass-extinction events are disallowed. This scenario is particularly interesting if you are interested in the diversification rates through time and are certain that there were no mass-extinction events. Nevertheless, the full `CoMET` analyses might be preferred because it includes the case without mass extinction and the data should support this (if true).

We do this with the argument `estimateNumberMassExtinctions = FALSE`. Because mass-extinction events are precluded, we can omit the corresponding parameters for survival probability, `pMassExtinctionPriorShape1` and `pMassExtinctionPriorShape`

```
set.seed(12345)
tess.analysis(conifers,
              empiricalHyperPriors = TRUE,
              samplingProbability = samplingFraction,
              estimateNumberMassExtinctions = FALSE,
              MAX_ITERATIONS = 10000,
              dir = "comet_no_mass_extinctions")

output <- tess.process.output("comet_no_mass_extinctions",
                numExpectedRateChanges = numExpectedRateChanges,
                numExpectedMassExtinctions = numExpectedMassExtinctions)
```

We visualize the output as before, this time omitting estimates of the number and timing of mass-extinction events (Figure 20).

```
layout.mat <- matrix(1:4,nrow=2,ncol=2,byrow=TRUE)
layout(layout.mat)
tess.plot.output(output,
                 fig.types = c("speciation rates",
                               "speciation shift times",
                               "extinction rates",
                               "extinction shift times"),
                 las=2)
```

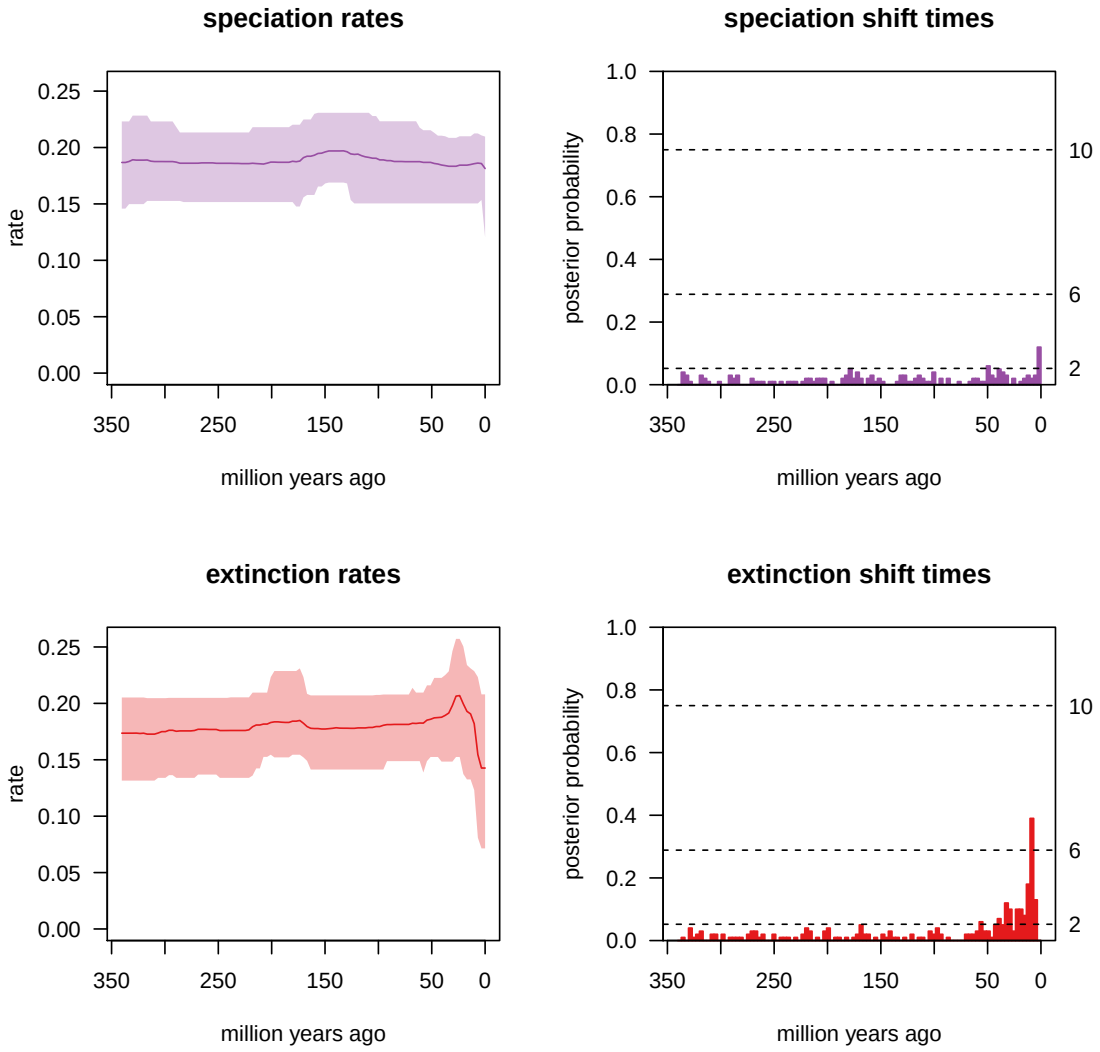


Figure 20: Visualizing the results of a `CoMET` analysis with empirically estimated diversification hyperpriors *and without* mass-extinction events.

5 MCMC Diagnosis, auto-tuning and auto-stopping

In any analysis we should be mindful about our ability to obtain reliable estimates, especially when complex models and stochastic algorithms are used. Bayesian inference is focussed on the joint posterior probability density of the model parameters, which must be approximated using numerical methods (MCMC simulation). It may be comforting to know that, in theory, an appropriately constructed and adequately run MCMC simulation is guaranteed to provide an arbitrarily precise description of the joint posterior probability density. In practice, however, even a given MCMC algorithm that provides reliable estimates in most cases will nevertheless fail in some cases and is not guaranteed to work for any given dataset. This raises an obvious question: "When do we know that an MCMC simulation provides reliable estimates for a given empirical analyses". Convergence can never be established with certainty, only non-convergence may be detected. We will illustrate how to assess convergence of MCMC simulations for two examples: 1) for analyses under a constant-rate birth-death process and 2) for analyses under the CoMET model.

5.1 MCMC diagnosis for a constant-rate birth-death model

In order to properly assess convergence, we need to perform at least two independent MCMC simulations. Otherwise, we run the risk in erroneously concluding that an MCMC has converged when it may have been stuck in a subregion of the joint posterior probability density.

We will use the same likelihood functions an MCMC methods described in Section 2.4. We will first run two short MCMC simulations. For the sake of this demonstration we will use a pre-burnin value of 0, a thinning of 1, and a chain length of 200 cycles to highlight that the runs have not converged.

```
prior_delta <- function(x) { dexp(x,rate=10.0,log=TRUE) }
prior_tau <- function(x) { dexp(x,rate=10.0,log=TRUE) }
my_priors <- c("diversification"=prior_delta,
                    "turnover"=prior_tau)
my_likelihood <- function(params) {

  speciation <- params[1] + params[2]
  extinction <- params[2]

  lnl <- tess.likelihood(times,
                         lambda = speciation,
                         mu = extinction,
```

```
                          samplingProbability = 1.0,
                          log = TRUE)

  return (lnl)

}

samples_run_1 <- tess.mcmc(likelihoodFunction = my_likelihood,
                           priors = my_priors,
                           parameters = runif(2,0,10),
                           logTransforms = c(TRUE,TRUE),
                           delta = c(1,1),
                           iterations = 200,
                           burnin = 0,
                           thinning = 1,
                           adaptive = TRUE,
                           verbose = TRUE)

samples_run_2 <- tess.mcmc(likelihoodFunction = my_likelihood,
                           priors = my_priors,
                           parameters = runif(2,0,10),
                           logTransforms = c(TRUE,TRUE),
                           delta = c(1,1),
                           iterations = 200,
                           burnin = 0,
                           thinning = 1,
                           adaptive = TRUE,
                           verbose = TRUE)
```

We assess convergence using three diagnostics: the effective sample size, the Geweke statistic, and the Gelman-Rubin statistic.

First, we compute the effective sample size (ESS). Samples drawn from an MCMC simulation are correlated. Accordingly, each sample is not independent, and so provides less information. We can compute the number of effectively independent samples by computing the ESS. This is important because we want to draw statistical conclusions from the samples, such as the sample mean. Higher ESS values should provide more precise inferences from the posterior sample. As a rule of thumb, the ESS should be larger than 200.

```
effectiveSize(samples_run_1)

## diversification        turnover
##       13.352224        7.931362

effectiveSize(samples_run_2)

## diversification        turnover
##        48.88930        17.18969
```

Next, we compute the Geweke diagnostic, which assesses convergence by computing the probability that the samples collected during an early window of the MCMC simulation are drawn from the same distribution as samples collected from a later window.

```
geweke.diag(samples_run_1)

##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## diversification        turnover
##           1.080           2.385

geweke.diag(samples_run_2)

##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
## diversification        turnover
##           1.017           1.002
```

Hence, we only need to test if the computed values is smaller than

```
qnorm(0.05/2)

## [1] -1.959964
```

and larger than

```
qnorm(1-0.05/2)

## [1] 1.959964
```

If this is the case, then we reject convergence. Note that we used a significance threshold of $\alpha = 0.05$, but you might use some other threshold, such as $\alpha = 0.01$.

The final MCMC diagnostic—the Gelman-Rubin test—compares samples from two independent simulations. This test compares the variance of sampled parameter values within each simulation to that between two simulations. This test effectively assesses whether we can reject the null hypothesis that samples from the two independent MCMC simulations are drawn from the same distribution.

```
gelman.diag(x=list(run1 = samples_run_1,run2 = samples_run_2),
            confidence = 0.95, transform = FALSE,
            autoburnin = FALSE, multivariate=TRUE)$mpsrf

## [1] 1.067948
```

If the two independent MCMC simulations have converged to the stationary distribution, the ratio of the within-sample variance to the between-sample variance (R) should be close to 1.0.

In general it is unlikely that those very short MCMC runs have converged. You may observe this by the very low ESS values. Depending on the starting values and your random seed, they actually may or may not have converged; and thus we can not say anything definitely. As a rule-of-thumb, you should run your MCMC simulations a bit longer, especially for more complex models than the simple constant-rate birth-death process model.

5.2 MCMC diagnosis for the CoMET model

5.2.1 Single-chain diagnostics

Proper MCMC diagnosis for a CoMET analysis requires examining the convergence and effective sample size of a large number of parameters, including the number of diversification-rate shifts, the number of mass-extinction events, and the interval-specific diversification-rate parameters. We will explore the single chain diagnostics on the hyperprior example. Thus, we read in again the output to be sure that we have it in memory.

```
output <- tess.process.output("comet_hyperpriors",
                numExpectedRateChanges = numExpectedRateChanges,
                numExpectedMassExtinctions = numExpectedMassExtinctions)
```

First, we will assess the MCMC performance for the numerical parameters of the CoMET model (*i.e.,* the number of diversification-rate shifts and mass-extinction events).

```
# Compute the effective sample size and Geweke diagnostic for
# the number of speciation-rate shifts.
effectiveSize(output$numSpeciationCategories)

##     var1
## 44.34624

geweke.diag(output$numSpeciationCategories)

##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1
## 0.4795

# Compute the effective sample size and Geweke diagnostic for
# the number of extinction-rate shifts.
effectiveSize(output$numExtinctionCategories)

##    var1
## 34.1116

geweke.diag(output$numExtinctionCategories)

##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##    var1
## -0.7937

# Compute the effective sample size and Geweke diagnostic for
# the number of mass-extinctionevents.
effectiveSize(output$numMassExtinctions)

##    var1
## 54.3984
```

```
geweke.diag(output$numMassExtinctions)

##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1
## 0.2699
```

Next, we will use the function `tess.plot.singlechain.diagnostics` to examine the effective sample size and Geweke diagnostic for the interval-specific speciation- and extinction-rate parameters.

```
layout.mat <- matrix(1:6,nrow=3,ncol=2,byrow=TRUE)
layout(layout.mat)
tess.plot.singlechain.diagnostics(output,
                    parameters = c("speciation rates",
                                   "extinction rates",
                                   "mass extinction times"),
                    las=2)
```

We can see from this single run that the ESS values for the interval-specific rate-parameter estimates are quite low. The Geweke statistics does not pick up any non-convergence, but this may be entirely due to the large uncertainty in the estimated mean (a low ESS means a large standard error of the mean). We therefore need to extend the length of this MCMC simulation to obtain adequate ESS values (we recommend a minimum ESS of 500) and achieve satisfactory Geweke statistics.

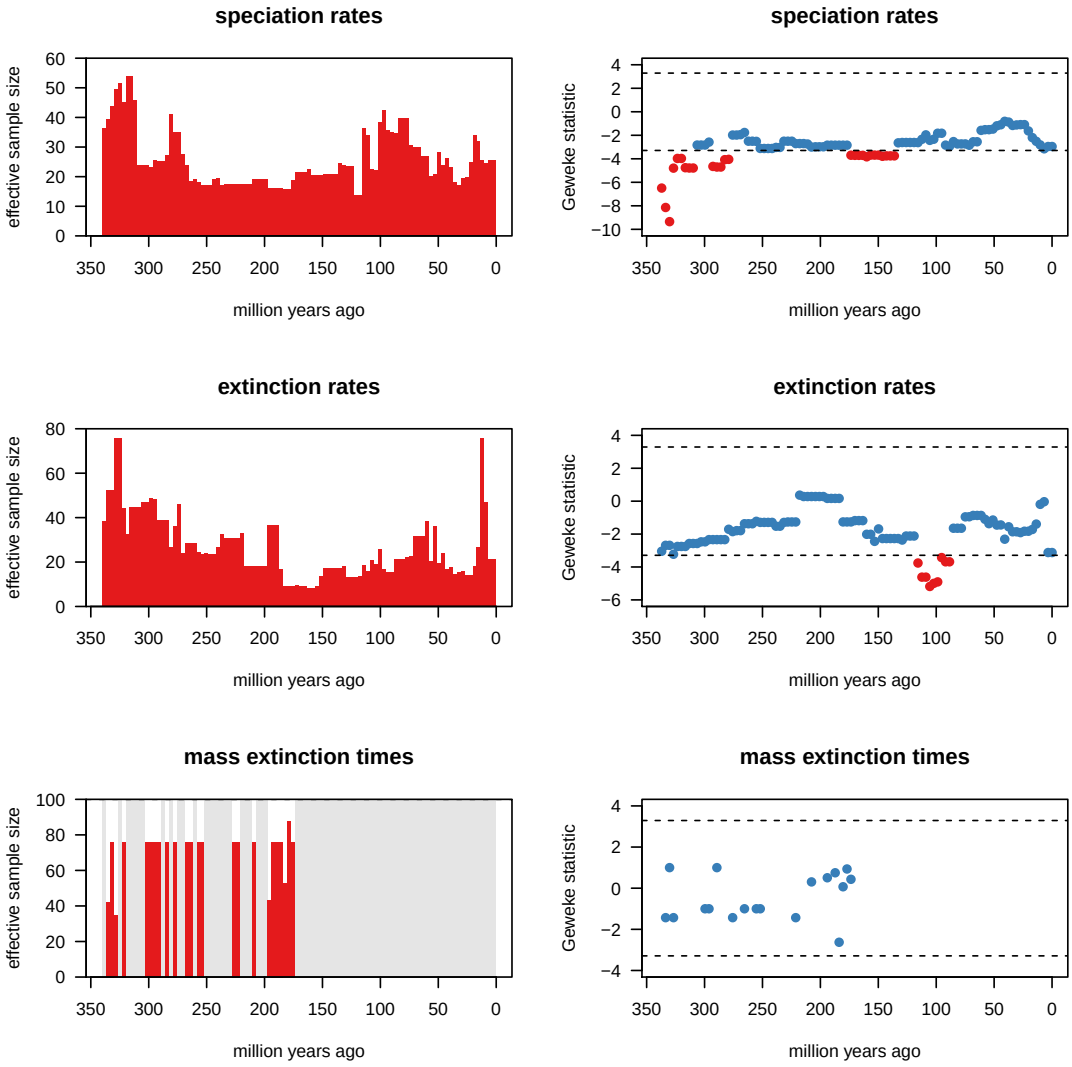


Figure 21: Visualizing the single-chain MCMC diagnostics for a `CoMET` analysis with empirically estimated diversification hyperpriors. Blue bars/dots represent passed tests and red bars/dots mean failed tests (failed convergence).

5.2.2 Multiple-chain diagnostics

To perform multiple-chain diagnostics, we must first obtain samples from multiple MCMC simulations. Accordingly, we will begin by repeating the above CoMET analysis with empirically estimated hyperpriors four times, each time with the same settings.

```
set.seed(12345)
posterior_directories <- paste("comet_posterior_",
                               1:4,sep="")

for(dir in posterior_directories) {
  tess.analysis(conifers,
                empiricalHyperPriors = TRUE,
                samplingProbability = samplingFraction,
                numExpectedRateChanges = numExpectedRateChanges,
                numExpectedMassExtinctions = numExpectedMassExtinctions,
                pMassExtinctionPriorShape1 = pMassExtinctionPriorShape1,
                pMassExtinctionPriorShape2 = pMassExtinctionPriorShape2,
                MAX_ITERATIONS = 10000,
                dir = dir)
}
```

We use the function `tess.plot.multichain.diagnostics` to compute the Gelman-Rubin convergence diagnostic for the interval-specific parameter estimates. First, we have to process each of the CoMET outputs individually.

```
output_1 <- tess.process.output("comet_posterior_1",
                numExpectedRateChanges = numExpectedRateChanges,
                numExpectedMassExtinctions = numExpectedMassExtinctions)

output_2 <- tess.process.output("comet_posterior_2",
                numExpectedRateChanges = numExpectedRateChanges,
                numExpectedMassExtinctions = numExpectedMassExtinctions)

output_3 <- tess.process.output("comet_posterior_3",
                numExpectedRateChanges = numExpectedRateChanges,
                numExpectedMassExtinctions = numExpectedMassExtinctions)

output_4 <- tess.process.output("comet_posterior_4",
                numExpectedRateChanges = numExpectedRateChanges,
```

```
                    numExpectedMassExtinctions = numExpectedMassExtinctions)
```

Next, we make a list of the MCMC outputs and use the Gelman-Rubin diagnostic on the parameters. For brevity, we analyze only a few of the parameters; you should check each parameter to make sure they have all converged to the same posterior distribution!

```
output_list <- list(output_1,output_2,output_3,output_4)

layout.mat <- matrix(1:3,nrow=3,ncol=1,byrow=TRUE)
layout(layout.mat)
tess.plot.multichain.diagnostics(output_list,
                                 parameters = c("speciation rates",
                                                "extinction rates",
                                                "mass extinction times"),
                                 las=2)
```

Generally speaking, those very short independent MCMC runs are unlikely to have converged to their posterior distributions. Depending on the starting values and your random seed, they may or may not have converged; and thus we can not say anything definitely. We recommend running the chains until the PSRF values have converged for all of the intervals.

5.3 Auto-tuning MCMC proposals

Convergence of an MCMC simulation and the ESS depend strongly on the efficiency of the MCMC proposals. In our MCMC algorithm, we use a simple sliding window proposal which is sufficient for these continuous parameters. The sliding window proposal draws a new parameter value x' from a normal distribution with mean x (the current parameter value) and standard deviation δ ($x' \sim \text{norm}(x, \delta)$). You can imagine that with large values of δ we are more likely to propose more different values from the current value than if δ is small. It is very difficult to know what good values for δ are. To demonstrate, we will first run an MCMC with a very small value of δ. We will again use the constant rate birth-death process as an example.

```
prior_delta <- function(x) { dexp(x,rate=10.0,log=TRUE) }
prior_tau <- function(x) { dexp(x,rate=10.0,log=TRUE) }
my_priors <- c("diversification"=prior_delta,
                    "turnover"=prior_tau)
```

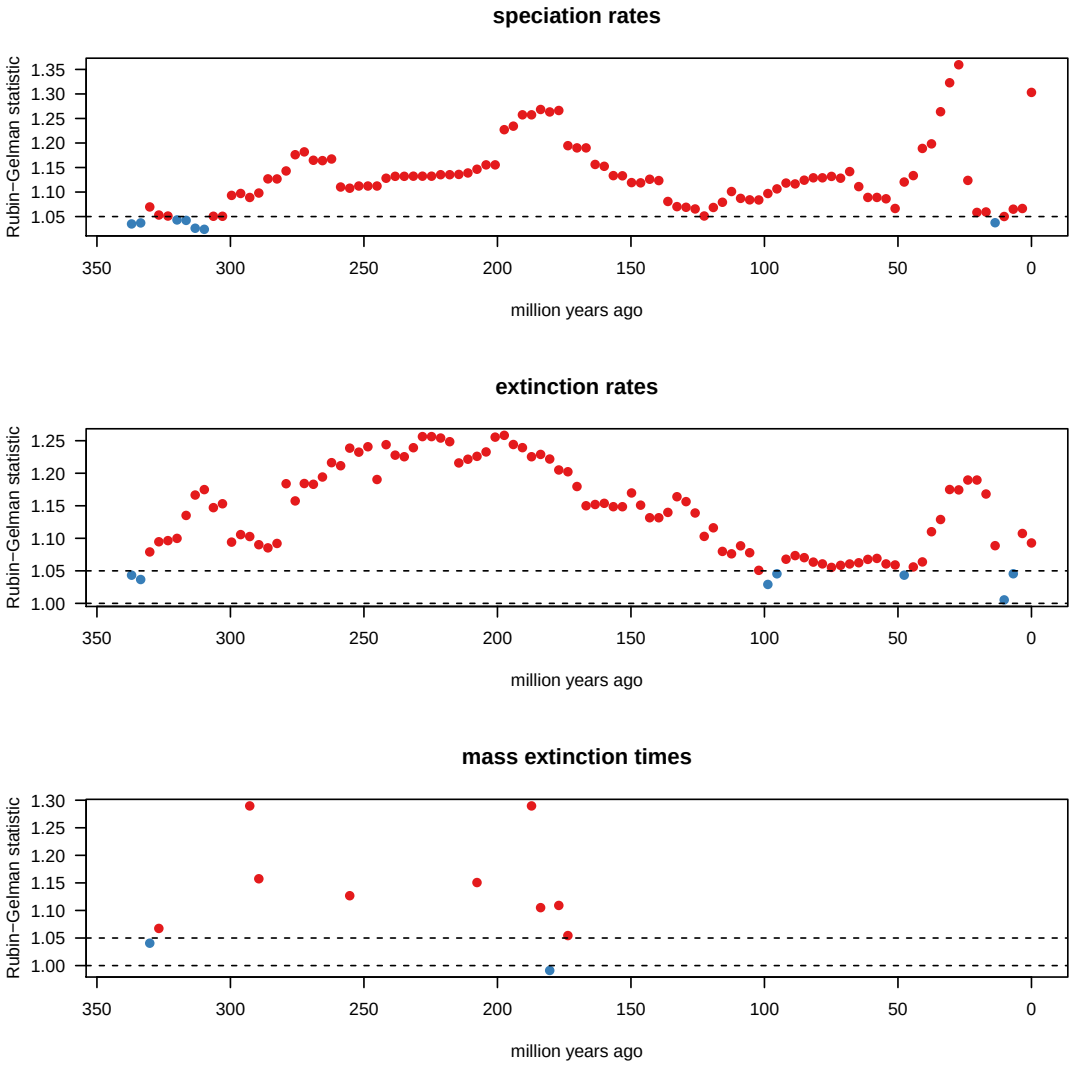


Figure 22: Visualizing the multiple-chain MCMC diagnostics for a `CoMET` analysis with empirically estimated diversification hyperpriors. Blue dots represent passed tests and red dots mean failed tests (failed convergence).

```
my_likelihood <- function(params) {

  speciation <- params[1] + params[2]
  extinction <- params[2]

  lnl <- tess.likelihood(times,
                         lambda = speciation,
                         mu = extinction,
                         samplingProbability = 1.0,
                         log = TRUE)

  return (lnl)

}
```

Specifically, we choose $\delta_1 = 0.1$ for the proposal on the net-diversification rate and $\delta_2 = 0.02$ for the turn-over rate.

```
samples_run_small <- tess.mcmc(likelihoodFunction = my_likelihood,
                               priors = my_priors,
                               parameters = runif(2,0,10),
                               logTransforms = c(TRUE,TRUE),
                               delta = c(0.1,0.02),
                               iterations = 1000,
                               burnin = 200,
                               thinning = 1,
                               adaptive = FALSE,
                               verbose = TRUE)
```

The particular behavior from small δ values is that proposals are frequently accepted but the parameter changes very small, leading to slow convergence and high between-sample correlations (auto-correlation). You can see this in the trace plots of the MCMC.

```
plot(samples_run_small)
```

As a second demonstration we will use intermediate values (which are more approriate for this dataset). Specifically, we choose $\delta_1 = 1.0$ for the proposal on the net-diversification rate and $\delta_2 = 0.2$ for the turn-over rate.

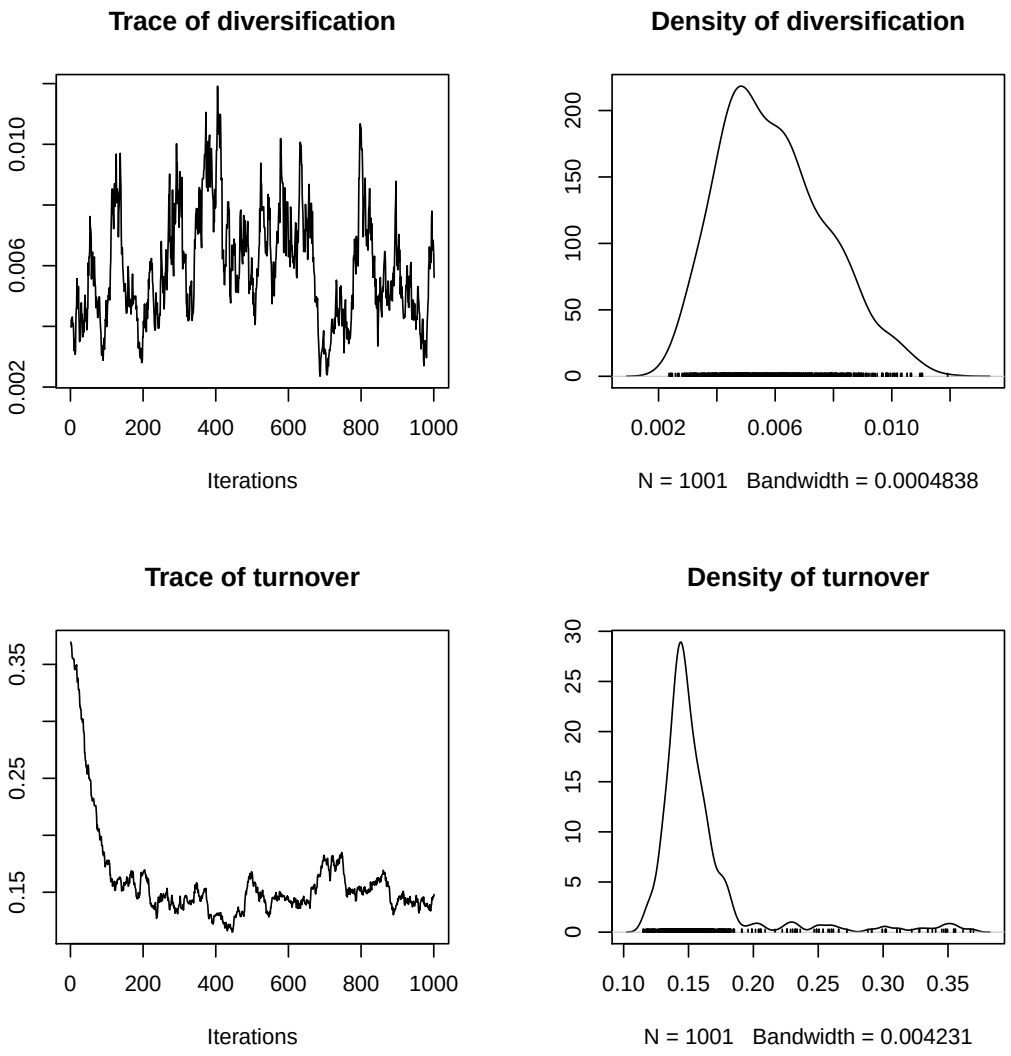


Figure 23: Trace plots (left) and marginal posterior probability densities (right) for the diversification rate (top) and turnover rate (bottom) from the MCMC simulation under the constant-rate birth-death process.

```
samples_run_good <- tess.mcmc(likelihoodFunction = my_likelihood,
                              priors = my_priors,
                              parameters = runif(2,0,10),
                              logTransforms = c(TRUE,TRUE),
                              delta = c(1,0.2),
                              iterations = 1000,
```

```
                              burnin = 200,
                              thinning = 1,
                              adaptive = FALSE,
                              verbose = TRUE)
```

Good MCMC simulations accept between 20% to 50% of the proposals on continuous parameters. If the chain is run long enough the trace plot will look like a *fuzzy caterpillar*.

```
plot(samples_run_good)
```

As a third demonstration we will very large values for δ. Specifically, we choose $\delta_1 = 1.0$ for the proposal on the net-diversification rate and $\delta_2 = 2.0$ for the turn-over rate.

```
samples_run_large <- tess.mcmc(likelihoodFunction = my_likelihood,
                              priors = my_priors,
                              parameters = runif(2,0,10),
                              logTransforms = c(TRUE,TRUE),
                              delta = c(10,2),
                              iterations = 1000,
                              burnin = 200,
                              thinning = 1,
                              adaptive = FALSE,
                              verbose = TRUE)
```

Because the MCMC proposes values that are very different from the current values, very few proposals are accepted. That is, the MCMC moves very slowly from one parameter value to another, which leads to long periods where the MCMC appears to be stuck.

```
plot(samples_run_large)
```

We showed that both large and small values of δ lead to poor mixing in the MCMC simulation. Unfortunately, there is no way to know which values of δ are good for the specific study tree ahead of time. Luckily, we can use methods like auto-tuning (**?**) to overcome this problem. Auto-tuning the MCMC means that we adjust the tuning parameter (δ) during the burnin phase. In TESS, you need to specify `ADAPTIVE = TRUE`. Then, you must specify the number of generations to perform burnin with the `burnin` argument; additionally, you must choose the frequency at which auto-tuning is performed with the `CONVERGENCE_FREQUENCY`

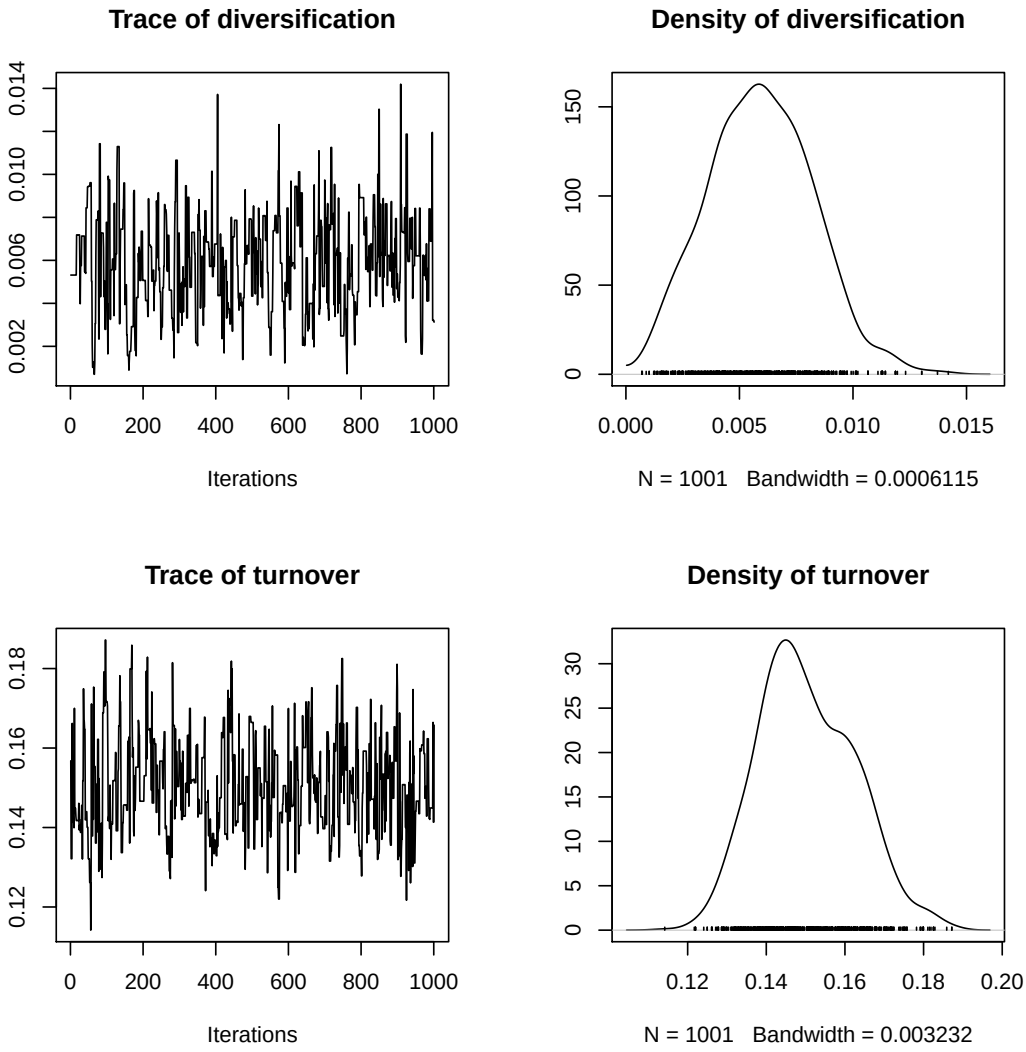


Figure 24: Trace plots (left) and marginal posterior probability densities (right) for the diversification rate (top) and turnover rate (bottom) from the MCMC simulation under the constant-rate birth-death process.

argument. During the burnin phase, every `CONVERGENCE_FREQUENCY` generations, δ values are updated so that proposals are accepted approximately 44% of the time: proposals that are accepted too frequently have their δ values increased, wherease proposals that are accepted too infrequently have their δ values decreased. Once the burnin phase is over, the chain begins logging samples from the MCMC and stops adjusting the δ values.

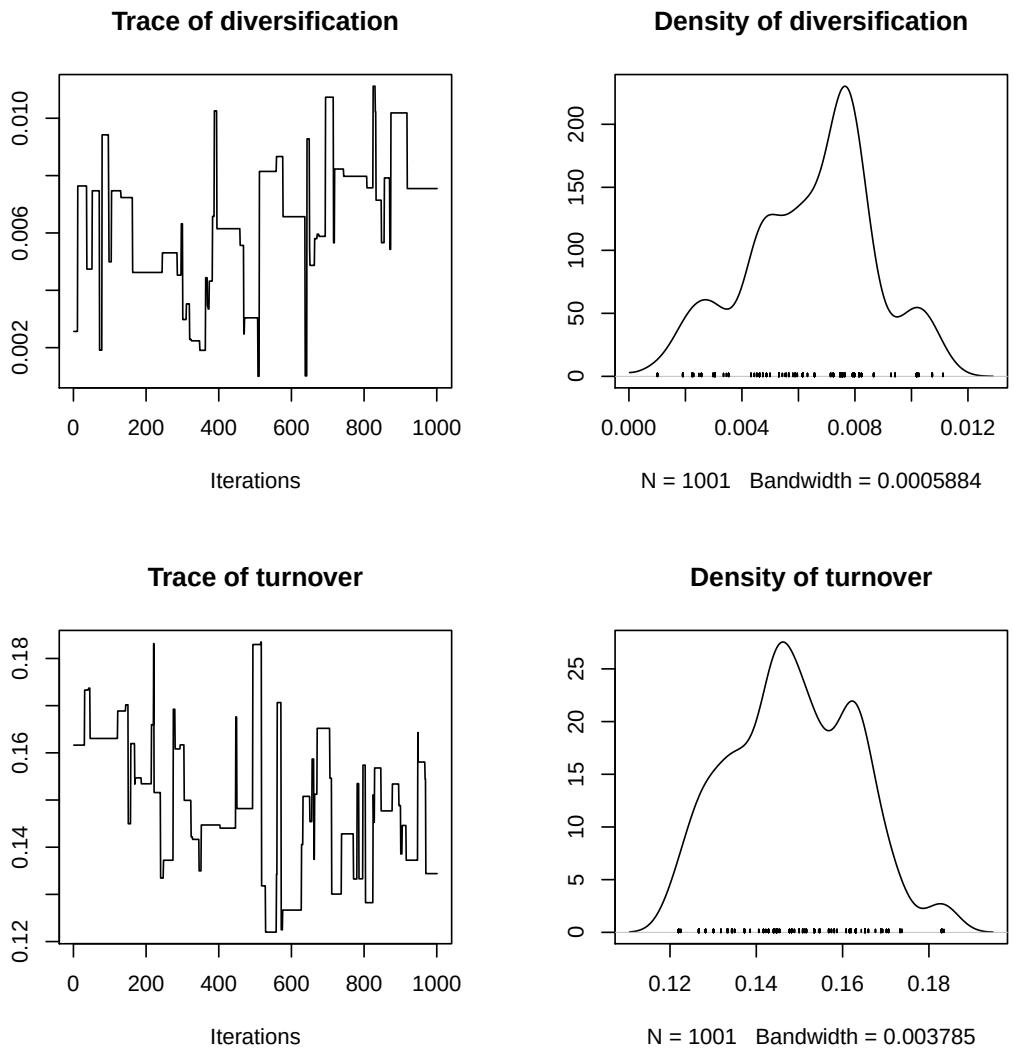


Figure 25: Trace plots (left) and marginal posterior probability densities (right) for the diversification rate (top) and turnover rate (bottom) from the MCMC simulation under the constant-rate birth-death process.

```
samples_run_auto_tuned <- tess.mcmc(likelihoodFunction = my_likelihood,
                                    priors = my_priors,
                                    parameters = runif(2,0,10),
                                    logTransforms = c(TRUE,TRUE),
                                    delta = c(1,1),
                                    iterations = 1000,
```

```
                                burnin = 200,
                                thinning = 1,
                                adaptive = TRUE,
                                verbose = TRUE)
```

At the end of the MCMC simulation, TESS provides some information about the values of δ. You can see that they were tuned roughly to the good values which we chose "arbitrarily" above. You can also see the acceptance rate which is somewhere between 20% and 50%. The trace plot also looks acceptable.

```
plot(samples_run_auto_tuned)
```

We recommend that you always use auto-tuning, and we include it as the default behavior of MCMC algorithms in TESS (with auto-tuning every 1000 burnin generations).

5.4 Auto-stopping MCMC simulations

CoMET analyses are more complicated than the analyses which we showed in the previous section. Auto-tuning helps to improve efficiency, but since a CoMET analyses includes reversible-jump MCMC it intrinsically needs more iterations for convergence. Nowing how long to run the MCMC to acheive sufficient convergence is quite difficult. We have implemented automatic stopping functionality to ensure that MCMCs are run sufficiently long to acheive good convergence. Auto-stopping checks every k iterations to determine if the MCMC simulation has failed to converge (as we described in the MCMC diagnosis section above). For example, we can specify that the minimum number of effective samples should be 500 (`MIN_ESS = 500`) for each parameter; the MCMC will stop once this threshold has been reached. We also provide other hard stopping rules, for example, the maximum number of iterations and the maximum running time; these hard stopping rules are enforced even if the MCMC has not converged. This is particularly useful when the MCMC does not seem to work and would, potentially, never converge.

In our small example we set the maximum number of iterations and the maximum time to large values so that they will not be reached.

```
set.seed(12345)

tess.analysis(conifers,
              empiricalHyperPriors = TRUE,
              samplingProbability = samplingFraction,
```

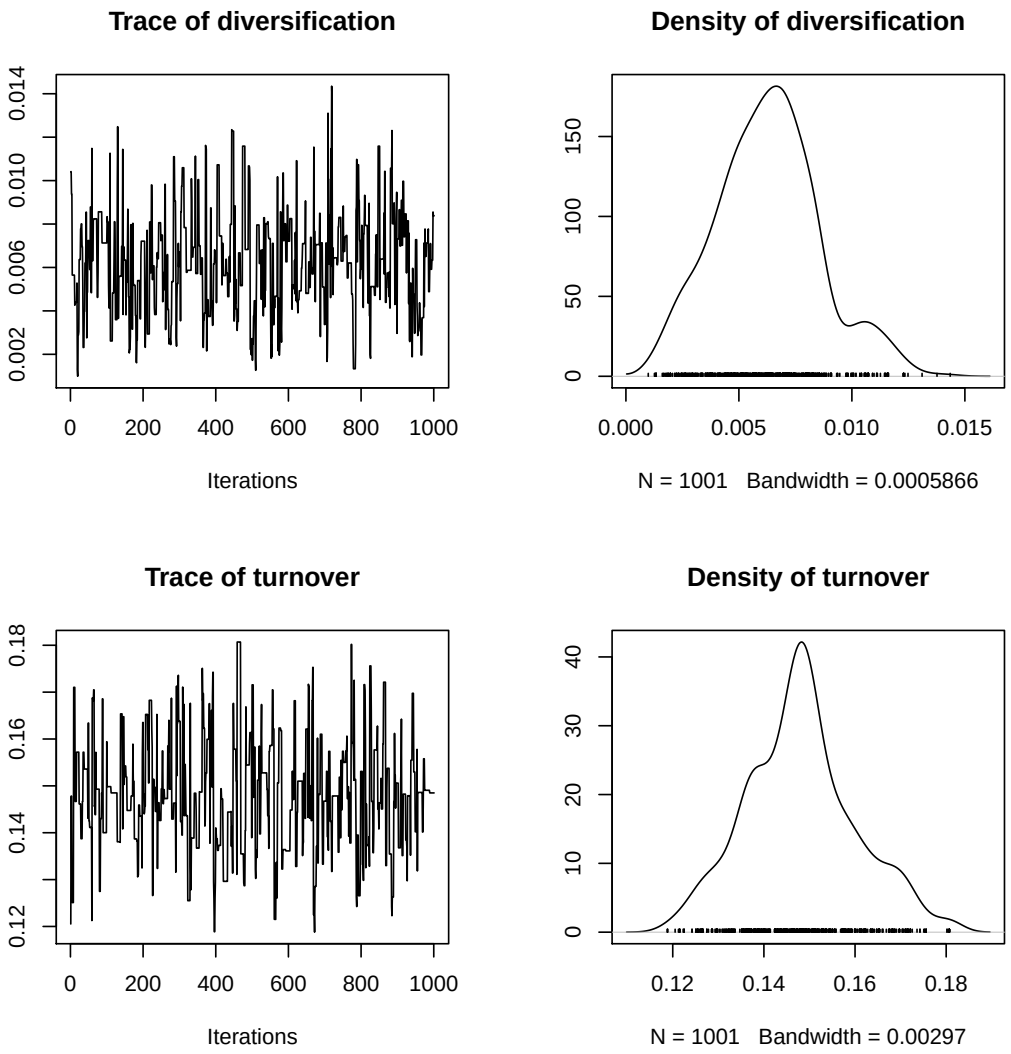


Figure 26: Trace plots (left) and marginal posterior probability densities (right) for the diversification rate (top) and turnover rate (bottom) from the MCMC simulation under the constant-rate birth-death process.

```
            numExpectedRateChanges = numExpectedRateChanges,
            numExpectedMassExtinctions = numExpectedMassExtinctions,
            pMassExtinctionPriorShape1 = pMassExtinctionPriorShape1,
            pMassExtinctionPriorShape2 = pMassExtinctionPriorShape2,
            MAX_ITERATIONS = 100000000,
            MAX_TIME = 24*60*60,
```

```
              MIN_ESS = 500,
              dir = "comet_auto_stop")


output_auto_stop <- tess.process.output("comet_auto_stop",
          numExpectedRateChanges = numExpectedRateChanges,
          numExpectedMassExtinctions = numExpectedMassExtinctions)
```

When we now process the output we see that the chain has converged nicely compared with the short run we showed before.

```
# Compute the effective sample size and Geweke diagnostic for
# the number of speciation-rate shifts.
effectiveSize(output_auto_stop$numSpeciationCategories)

##     var1
## 613.7274

geweke.diag(output_auto_stop$numSpeciationCategories)

##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##    var1
## -0.3335

# Compute the effective sample size and Geweke diagnostic for
# the number of extinction-rate shifts.
effectiveSize(output_auto_stop$numExtinctionCategories)

##     var1
## 622.5807

geweke.diag(output_auto_stop$numExtinctionCategories)

##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##   var1
## 0.8911
```

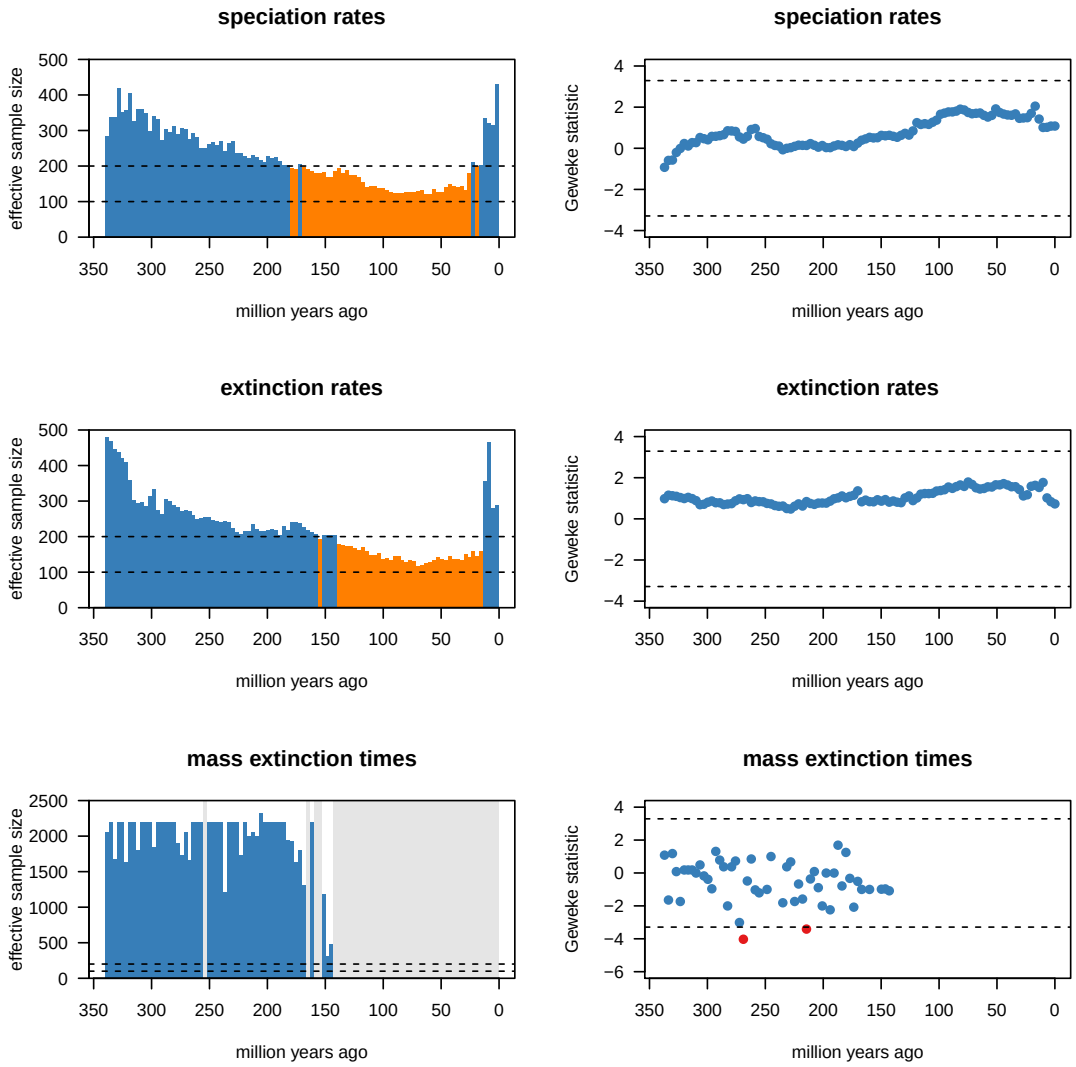
```
# Compute the effective sample size and Geweke diagnostic for
# the number of mass-extinction events.
effectiveSize(output_auto_stop$numMassExtinctions)

##     var1
## 1063.746

geweke.diag(output_auto_stop$numMassExtinctions)

##
## Fraction in 1st window = 0.1
## Fraction in 2nd window = 0.5
##
##    var1
## -0.9541

layout.mat <- matrix(1:6,nrow=3,ncol=2,byrow=TRUE)
layout(layout.mat)
tess.plot.singlechain.diagnostics(output_auto_stop,
                    parameters = c("speciation rates",
                                   "extinction rates",
                                   "mass extinction times"),
                    las=2)
```



Automatic stopping is the default behavior for a `CoMET` analysis; by default, the MCMC will stop once it has acheived 500 effective samples from the posterior.

References

Alfaro, M., Santini, F., Brock, C., Alamillo, H., Dornburg, A., Rabosky, D., Carnevale, G., and Harmon, L. (2009). Nine exceptional radiations plus high turnover explain species diversity in jawed vertebrates. *Proceedings of the National Academy of Sciences*, 106(32):13410–13414.

Baele, G., Lemey, P., Bedford, T., Rambaut, A., Suchard, M., and Alekseyenko, A. (2012). Improving the accuracy of demographic and molecular clock model comparison while accommodating phylogenetic uncertainty. *Molecular Biology and Evolution*, 29(9):2157–2167.

Baele, G., Li, W., Drummond, A., Suchard, M., and Lemey, P. (2013). Accurate Model Selection of Relaxed Molecular Clocks in Bayesian Phylogenetics. *Molecular Biology and Evolution*, 30(2):239–243.

Bokma, F. (2008). Bayesian estimation of speciation and extinction probabilities from (in) complete phylogenies. *Evolution*, 62(9):2441–2445.

Bortolussi, N., Durand, E., Blum, M., and Francois, O. (2006). aptreeshape: statistical analysis of phylogenetic tree shape. *Bioinformatics*, 22(3):363–364.

Chan, K. M. A. and Moore, B. R. (2002). Whole-tree methods for detecting differential diversification rates. *Systematic Biology*, 51(6):85–865.

Chan, K. M. A. and Moore, B. R. (2005). SymmeTREE: whole-tree analysis of differential diversification rates. *Bioinformatics*, 21:1709–1710.

Cusimano, N. and Renner, S. (2010). Slowdowns in diversification rates from real phylogenies may not be real. *Systematic Biology*, 59(4):458.

Cusimano, N., Stadler, T., and Renner, S. S. (2012). A new method for handling missing species in diversification analysis applicable to randomly or nonrandomly sampled phylogenies. *Systematic Biology*, 61(5):785–792.

Drummond, A. J., Suchard, M. A., Xie, D., and Rambaut, A. (2012). Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Molecular Biology And Evolution*, 29:1969–1973.

Etienne, R. and Haegeman, B. (2012). A Conceptual and Statistical Framework for Adaptive Radiations with a Key Role for Diversity Dependence. *The American Naturalist*, 180(4):75–89.

Etienne, R., Haegeman, B., Stadler, T., Aze, T., Pearson, P., Purvis, A., and Phillimore, A. (2012). Diversity-dependence brings molecular phylogenies closer to agreement with the fossil record. *Proceedings of the Royal Society B: Biological Sciences*, 279(1732):1300–1309.

Fan, Y., Wu, R., Chen, M.-H., Kuo, L., and Lewis, P. O. (2011). Choosing among partition models in bayesian phylogenetics. *Molecular Biology and Evolution*, 28(1):523–532.

FitzJohn, R., Maddison, W., and Otto, S. (2009). Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. *Systematic Biology*, 58(6):595–611.

FitzJohn, R. G. (2010). Quantitative traits and diversification. *Systematic Biology*, 59(6):619–633.

FitzJohn, R. G. (2012). Diversitree: comparative phylogenetic analyses of diversification in R. *Methods in Ecology and Evolution*, 3(6):1084–1092.

Goldberg, E. E., Lancaster, L. T., and Ree, R. H. (2011). Phylogenetic inference of reciprocal effects between geographic range evolution and diversification. *Systematic Biology*, 60(4):451–465.

Heath, T. A., Huelsenbeck, J. P., and Stadler, T. (2014). The fossilized birth-death process for coherent calibration of divergence-time estimates. *Proceedings of the National Academy of Sciences*, 111(29):E2957–E2966.

Höhna, S. (2013). Fast simulation of reconstructed phylogenies under global time-dependent birth-death processes. *Bioinformatics*, 29(11):1367–1374.

Höhna, S. (2014). Likelihood Inference of Non-Constant Diversification Rates with Incomplete Taxon Sampling. *PLoS One*, 9(1):e84184.

Höhna, S. (2015). The time-dependent reconstructed evolutionary process with a key-role for mass-extinction events. *Journal of Theoretical Biology*, 380:321–331.

Höhna, S., Landis, M. J., Heath, T. A., Boussau, B., Lartillot, N., Moore, B. R., Huelsenbeck, J. P., and Ronquist, F. (2015). RevBayes: A flexible framework for Bayesian inference of phylogeny. *Systematic Biology*, 64(5):in press.

Höhna, S., Stadler, T., Ronquist, F., and Britton, T. (2011). Inferring speciation and extinction rates under different species sampling schemes. *Molecular Biology and Evolution*, 28(9):2577–2589.

Jeffreys, H. (1961). *The theory of probability*. Oxford University Press.

Kass, R. and Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association*, 90:773–795.

Kendall, D. G. (1948). On the generalized "birth-and-death" process. *The Annals of Mathematical Statistics*, 19(1):1–15.

Lambert, A. (2010). The contour of splitting trees is a lévy process. *The Annals of Probability*, 38(1):348–395.

Lambert, A., Alexander, H. K., and Stadler, T. (2014). Phylogenetic analysis accounting for age-dependent death and sampling with applications to epidemics. *Journal of Theoretical Biology*, 352:60–70.

Lambert, A. and Stadler, T. (2013). Birth–death models and coalescent point processes: the shape and probability of reconstructed phylogenies. *Theoretical Population Biology*, 90:113–128.

Lartillot, N. and Philippe, H. (2006). Computing Bayes factors using thermodynamic integration. *Systematic Biology*, 55(2):195.

Leslie, A. B., Beaulieu, J. M., Rai, H. S., Crane, P. R., Donoghue, M. J., and Mathews, S. (2012). Hemisphere-scale differences in conifer evolutionary dynamics. *Proceedings of the National Academy of Sciences*, 109(40):16217–16221.

Leventhal, G. E., Günthard, H. F., Bonhoeffer, S., and Stadler, T. (2014). Using an epidemiological model for phylogenetic inference reveals density dependence in hiv transmission. *Molecular Biology and Evolution*, 31(1):6–17.

Maddison, W., Midford, P., and Otto, S. (2007). Estimating a binary character's effect on speciation and extinction. *Systematic Biology*, 56(5):701.

Magallón, S. and Sanderson, M. J. (2001). Absolute diversification rates in angiosperm clades. *Evolution*, 55(9):1762–1780.

Magnuson-Ford, K. and Otto, S. P. (2012). Linking the investigations of character evolution and species diversification. *The American Naturalist*, 180(2):225–245.

May, M. R., Höhna, S., and Moore, B. R. (2015). A Bayesian Approach for Detecting Mass-Extinction Events When Rates of Lineage Diversification Vary. *submitted*.

Moore, B. R., Chan, K. M. A., and Donoghue, M. J. (2004). Detecting diversification rate variation in supertrees. In Bininda-Emonds, O. R. P., editor, *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*, pages 487–533. Kluwer Academic Press.

Morlon, H., Condamine, F., Lewitus, E., and Manceau, M. (in prep). RPANDA: an r package for macroevolutionary analyses on phylogenetic trees. *Methods in Ecology and Evolution*. R package version 1.0.

Morlon, H., Parsons, T., and Plotkin, J. (2011). Reconciling molecular phylogenies with the fossil record. *Proceedings of the National Academy of Sciences*, 108(39):16327–16332.

Morlon, H., Potts, M., and Plotkin, J. (2010). Inferring the dynamics of diversification: a coalescent approach. *PLoS Biology*, 8(9):e1000493.

Nee, S., May, R. M., and Harvey, P. H. (1994). The Reconstructed Evolutionary Process. *Philosophical Transactions: Biological Sciences*, 344(1309):305–311.

Pybus, O. and Harvey, P. (2000). Testing macro–evolutionary models using incomplete molecular phylogenies. *Proceedings of the Royal Society B: Biological Sciences*, 267(1459):2267–2272.

Rabosky, D. L. (2014). Automatic detection of key innovations, rate shifts, and diversity-dependence on phylogenetic trees. *PLoS One*, 9(2):e89543.

Rabosky, D. L. and Lovette, I. (2008). Explosive evolutionary radiations: decreasing speciation or increasing extinction through time? *Evolution*, 62(8):1866–1875.

Ronquist, F., Teslenko, M., van der Mark, P., Ayres, D. L., Darling, A., Höhna, S., Larget, B., Liu, L., Suchard, M. A., and Huelsenbeck, J. P. (2012). MrBayes 3.2: Efficient Bayesian phylogenetic inference and model choice across a large model space. *Systematic Biology*, 61(3):539–542.

Ryberg, M., Nilsson, R. H., and Matheny, P. B. (2011). DivBayes and SubT: exploring species diversification using Bayesian statistics. *Bioinformatics*, 27(17):2439–2440.

Sanderson, M. J. (2003). r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. *Bioinformatics*, 19(2):301–302.

Silvestro, D., Schnitzler, J., and Zizka, G. (2011). A Bayesian framework to estimate diversification rates and their variation through time and space. *BMC Evolutionary Biology*, 11(1):311.

Stadler, T. (2009). On incomplete sampling under birth-death models and connections to the sampling-based coalescent. *Journal of Theoretical Biology*, 261(1):58–66.

Stadler, T. (2010). Sampling-through-time in birth-death trees. *Journal of Theoretical Biology*, 267(3):396–404.

Stadler, T. (2011a). Mammalian phylogeny reveals recent diversification rate shifts. *Proceedings of the National Academy of Sciences*, 108(15):6187–6192.

Stadler, T. (2011b). Simulating trees with a fixed number of extant species. *Systematic Biology*, 60(5):676–684.

Stadler, T. and Bokma, F. (2013). Estimating Speciation and Extinction Rates for Phylogenies of Higher Taxa. *Systematic Biology*, 62(2):220–230.

Suchard, M. A., Weiss, R. E., and Sinsheimer, J. S. (2001). Bayesian selection of continuous-time Markov chain evolutionary models. *Molecular Biology and Evolution*, 18(6):1001–1013.

Thompson, E. (1975). *Human evolutionary trees*. Cambridge University Press Cambridge.

Xie, W., Lewis, P., Fan, Y., Kuo, L., and Chen, M. (2011). Improving marginal likelihood estimation for Bayesian phylogenetic model selection. *Systematic Biology*, 60(2):150–160.

Yang, Z. and Rannala, B. (1997). Bayesian phylogenetic inference using DNA sequences: a Markov Chain Monte Carlo Method. *Molecular Biology and Evolution*, 14(7):717–724.

Yule, G. (1925). A mathematical theory of evolution, based on the conclusions of dr. jc willis, frs. *Philosophical Transactions of the Royal Society of London. Series B, Containing Papers of a Biological Character*, 213:21–87.