



HAL
open science

Estimating a time-calibrated phylogeny of fossil and extant taxa using RevBayes

Joëlle Barido-Sottani, Joshua A. Justison, April M. Wright, Rachel C.M. Warnock, Walker Pett, Tracy A. Heath

► To cite this version:

Joëlle Barido-Sottani, Joshua A. Justison, April M. Wright, Rachel C.M. Warnock, Walker Pett, et al.. Estimating a time-calibrated phylogeny of fossil and extant taxa using RevBayes. Céline Scornavacca; Frédéric Delsuc; Nicolas Galtier. Phylogenetics in the Genomic Era, No commercial publisher | Authors open access book, pp.5.2:1–5.2:23, 2020. hal-02536394v2

HAL Id: hal-02536394

<https://hal.science/hal-02536394v2>

Submitted on 26 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.


L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.




Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Chapter 5.2 Estimating a Time-calibrated Phylogeny of Fossil and Extant Taxa using RevBayes


Joëlle Barido-Sottani

Department of Ecology, Evolution, & Organismal Biology
Iowa State University
Ames, IA 50011 USA
joellebs@iastate.edu
 <https://orcid.org/0000-0002-5220-5468>

Joshua A. Justison

Department of Ecology, Evolution, & Organismal Biology
Iowa State University
Ames, IA 50011 USA
justison@iastate.edu
 <https://orcid.org/0000-0002-0233-4413>


April M. Wright

Department of Biological Sciences
Southeastern Louisiana University
Hammond, LA 70402 USA
april.wright@selu.edu
 <https://orcid.org/0000-0003-4692-3225>


Rachel C. M. Warnock

Department of Biosystems Science & Engineering
Eidgenössische Technische Hochschule Zürich
Swiss Institute of Bioinformatics (SIB)
4058 Basel, Switzerland
rachel.warnock@bsse.ethz.ch
 <https://orcid.org/0000-0002-9151-4642>

Walker Pett

Department of Ecology, Evolution, & Organismal Biology
Iowa State University
Ames, IA 50011 USA
willpett@iastate.edu
 <https://orcid.org/0000-0003-3733-0815>

Tracy A. Heath

Department of Ecology, Evolution, & Organismal Biology
Iowa State University
Ames, IA 50011 USA
phylo@iastate.edu
 <https://orcid.org/0000-0002-0087-2541>

Abstract

The fossil record is the primary source of time-stamped information useful for dating phylogenetic trees; and many statistical approaches are available for integrating data from fossil and living species. In this tutorial, we demonstrate how to perform joint inference of divergence times and phylogenetic relationships of fossil and extant taxa from morphological data using the program RevBayes. RevBayes (<http://revbayes.com>) is a flexible and powerful tool for Bayesian



© Joëlle Barido-Sottani, Joshua A. Justison, April Wright, Rachel C.M. Warnock, Walker Pett, and Tracy A. Heath.

Licensed under Creative Commons License CC-BY-NC-ND 4.0.

Phylogenetics in the genomic era.

Editors: Celine Scornavacca, Frédéric Delsuc and Nicolas Galtier; chapter No. 5.2; pp. 5.2:1–5.2:23



A book completely handled by researchers.
No publisher has been paid.

5.2:2 Estimating a time-calibrated phylogeny of fossil and extant taxa using RevBayes

phylogenetic inference. Statistical models in RevBayes are built using probabilistic graphical models and described via an interpreted programming language. As a result, RevBayes offers a wide range of statistical models—ranging from very simple models with few parameters to hierarchical models describing complex biological processes—that are useful in many biological applications. The exercise described here provides instructions on how to construct a phylogenetic model combining the fossilized birth-death process and models describing the generation of morphological data, which is then used to execute an analysis that unites modern and extinct taxa in a dated phylogenetic tree. The content and associated files for this tutorial are kept up-to-date at: http://revbayes.com/tutorials/fbd_simple.

How to cite: Joëlle Barido-Sottani, Joshua A. Justison, April M. Wright, Rachel C. M. Warnock, Walker Pett, and Tracy A. Heath (2020). Estimating a time-calibrated phylogeny of fossil and extant taxa using RevBayes. In Scornavacca, C., Delsuc, F., and Galtier, N., editors, *Phylogenetics in the Genomic Era*, chapter No. 5.2, pp. 5.2:2–5.2:23. No commercial publisher | Authors open access book. The book is freely available at <https://hal.inria.fr/PGE>.

Funding This work was supported by National Science Foundation (USA) grants DEB-1556615, DEB-1556853, and DBI-1759909 (JBS, JAJ, WP, and TAH); and an Institutional Development Award (IDeA) from the National Institute of General Medical Sciences of the National Institutes of Health under grant number P2O GM103424-18 (AMW).

1 Introduction to RevBayes

RevBayes (Höhna et al., 2016) is an open-source software program for Bayesian phylogenetic inference. It offers a flexible framework for hierarchical Bayesian modeling of complex and biologically realistic models of evolution. This flexibility is possible because RevBayes uses probabilistic graphical models (Höhna et al., 2014) and an interpreted programming language—called Rev—to specify and represent statistical models. For an review of the concepts and techniques used in this chapter, see Chapters 1.4 and 5.1 (Lartillot 2020; Pett and Heath 2020).

Links to RevBayes software and documentation

- Website: <http://revbayes.com>
- Download: <http://revbayes.com/download>
- Open source projects on GitHub: <https://github.com/revbayes>
- Tutorials: <http://revbayes.com/tutorials>
- Rev language reference: <http://revbayes.com/documentation>

In the probabilistic graphical modeling framework of RevBayes, model components (parameters and distributions) are interchangeable building blocks for constructing a complete statistical model (Höhna et al., 2016). This modularity enables users to easily modify a model to match their prior assumptions. When applying Bayesian analysis approaches, RevBayes uses a Markov chain Monte-Carlo (MCMC) algorithm to sample the posterior distributions of unknown parameters in a model. While inference using MCMC is the primary analysis approach in RevBayes, there are several other available statistical approaches, including model comparison using Bayes factors, and posterior predictive model checking and analysis of model adequacy (Höhna et al., 2018).

The core RevBayes library (written in C++) implements the various objects and functions that define a model and perform statistical analyses. Currently, the main interface to the RevBayes core is Rev, the interpreted programming language that users access via a RevBayes console or through writing Rev scripts. Members of the RevBayes Development Team are currently working to expand the set of interfaces for working with RevBayes and the Rev language. These include RevScripter¹ a graphical user interface for generating Rev analysis scripts, a Jupyter kernel² for running RevBayes in the Jupyter notebook environment, the RevKnitr³ R package for using Rev interactively in RStudio, and the RevGadgets⁴ R package for summarizing output from RevBayes analyses. Additional information on installing alternative graphical interfaces can be found on the RevBayes website⁵.

The modular framework of RevBayes has facilitated the rapid expansion of available statistical methods for investigating evolutionary hypotheses. The tutorial presented here provides a mere glimpse at what is possible in RevBayes, focusing explicitly on inference of a time-calibrated phylogeny using paleontological and neontological data. However, there are a wide range of approaches for inferring macroevolutionary parameters in a phylogenetic framework. Throughout the tutorial, we refer to alternative or more advanced models and methods available in RevBayes. Thus, we hope that the exercises described here will introduce the reader to the potential for conducting analyses in RevBayes that may elucidate the evolutionary processes underpinning the generation of their biological data.

2 Background: Inferring the Timing and Phylogeny of Fossil and Extant Taxa

This tutorial and associated files (i.e., data and script files) are maintained on the RevBayes website: http://revbayes.com/tutorials/fbd_simple.

The exercise described in Section 3 is a guide to using RevBayes to perform a simple phylogenetic analysis of extant and fossil bear species (family Ursidae), using morphological data as well as the occurrence times of lineages observed in the fossil record. To get an overview of the model, it is useful to think of the model as a generating process for our data. Suppose we would like to simulate our fossil and morphological data; we would consider two components (Figure 1):

- **Time tree model:** This is the diversification process that describes how a phylogeny is generated as well as when fossils are sampled along each lineage on the phylogeny. This component generates the phylogeny, divergence times, and the fossil occurrence data. The tree topology and node ages are parameters of the model that generates our morphological characters.
- **Discrete morphological character change model:** This model describes how discrete morphological character states change over time on the phylogeny. The generation of observed morphological character states is governed by other model components including

¹ RevScripter: <http://revbayes.com/revscripter>

² RevBayes Jupyter kernel: https://github.com/revbayes/revbayes_kernel

³ RevKnitr: <https://github.com/revbayes/RevKnitr>

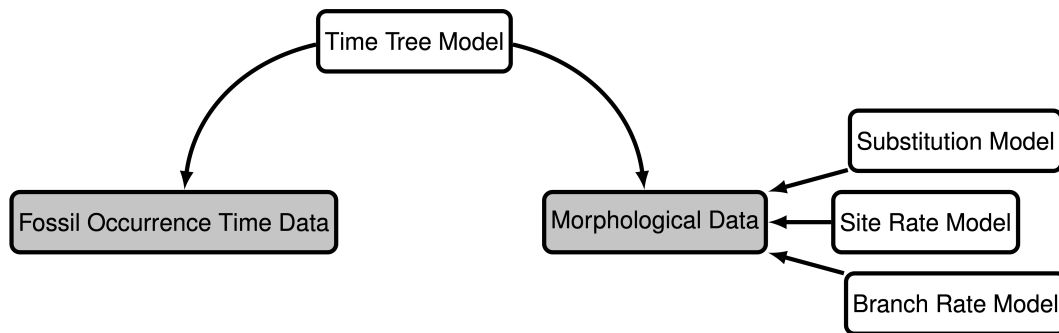
⁴ RevGadgets R package <https://github.com/revbayes/RevGadgets>

⁵ Graphical User Interface installation: <http://revbayes.com/gui-setup>

5.2:4 Estimating a time-calibrated phylogeny of fossil and extant taxa using RevBayes

the substitution process and variation among characters in our matrix and among branches on the tree.

These two components, or modules, form the backbone of the inference model and reflect our prior beliefs on how the tree, fossil data, and morphological trait data are generated. We will provide a brief overview of the specific models used within each component while pointing to other tutorials that implement alternative models.



■ **Figure 1** Modular components of the graphical model used in the analysis described in this tutorial. The gray boxes indicate the observed data: fossil ages and discrete morphological characters. The white boxes represent the models that generated the data. See also Section 5 of Chapter 5.1 [Pett and Heath 2020], and, in particular Figure 5, for other hierarchical models.

2.1 Time tree model: the fossilized birth-death process

The fossilized birth-death (FBD) process provides a joint distribution on the divergence times of living and extinct species, the tree topology, and the sampling of fossils (Stadler, 2010; Heath et al., 2014). The FBD model can be broken into two sub-processes, the birth-death process and the fossilization process.

2.1.1 Birth-death process

The birth-death process is a branching process that provides a distribution for the tree topology and divergence times on the tree. We will consider a constant-rate birth-death process (Kendall, 1948; Thompson, 1975). Specifically, we will assume every lineage has the same constant rate of speciation λ and rate of extinction μ at any moment in time (Nee et al., 1994; Höhna, 2015). Speciation and extinction events occur with rate parameters λ and μ respectively, whereby the waiting time between events is exponentially distributed with parameter $(\lambda + \mu)$. Then, given an event occurred, the probability of the event being a speciation is $(\lambda / (\lambda + \mu))$ while the probability of the event being an extinction is $(\mu / (\lambda + \mu))$.

The birth-death process depends on two other parameters as well, the origin time and the sampling probability. The origin time, denoted ϕ , represents the starting time of the stem lineage, which is the age of the entire process. The sampling probability, denoted ρ , gives the probability that an extant species is sampled.

The assumption that, at any given time, each lineage has the same speciation rate and extinction rate may not be realistic or valid in some systems. Several models are currently implemented in RevBayes that relax the assumption of constant rates such as,

episodic diversification rates⁶ (Höhna, 2015), environment-dependent diversification rates⁷ (Condamine et al., 2018), branch-specific diversification rates⁸ (Höhna et al., 2019), or diversification rates tied to a species trait⁹ (Maddison et al., 2007; Freyman and Höhna, 2018, 2019).

2.1.2 Fossilization process

Given a phylogeny, in this case a phylogeny generated by a birth-death process, the fossilization process provides a distribution for sampling fossilized occurrences of lineages in the tree (Heath et al., 2014). Much like speciation and extinction, fossil sampling is modeled according to a Poisson process with rate parameter ψ . This means that each lineage has the same constant rate of producing a fossil. As a result, along a given lineage, the time between fossilization events is exponentially distributed with rate ψ .

One key assumption of the FBD model is that each fossil represents a distinct fossil specimen. However, if certain taxa persist through time and fossilize particularly well, then the same taxon may be sampled at different stratigraphic ages. These fossil data are commonly represented by only the first and last appearances of a fossil morphospecies. In this case one might want to consider the fossilized birth-death range process¹⁰ (Stadler et al., 2018) in RevBayes to model the stratigraphic ranges of fossil occurrences.

2.1.3 Accounting for fossil age uncertainty

Often, there is uncertainty around the age of each fossil, which is typically represented as an interval of the minimum and maximum possible ages. Moreover, a recent study demonstrated using simulated data that ignoring uncertainty in fossil occurrence dates can lead to biased estimates of divergence times (Barido-Sottani et al., 2019). RevBayes allows fossil occurrence time uncertainty to be modeled by directly treating it as part of the likelihood of the fossil data given the time tree. We model this by assuming the likelihood of a particular fossil occurrence \mathcal{F}_i is zero if the inferred age t_i occurs outside the time interval (a_i, b_i) and some non-zero likelihood when the fossil is placed within the interval. Specifically, we will assume the fossil could occur anywhere within the observed interval with uniform probability, this means that the likelihood is equal to one if the inferred fossil age is consistent with the observed fossil interval:

$$f[\mathcal{F}_i | a_i, b_i, t_i] = \begin{cases} 1 & \text{if } a_i < t_i < b_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The incorporation of uncertainty around the fossil occurrence data is shown graphically as a part of our model in (Figure 2).

2.2 Modeling discrete morphological character change

Given a phylogeny, the discrete morphological character change model will describe how traits change along each lineage, resulting in the observed character states of fossils and

⁶ Episodic diversification rates tutorial: <http://revbayes.com/tutorials/divrate/ebd>

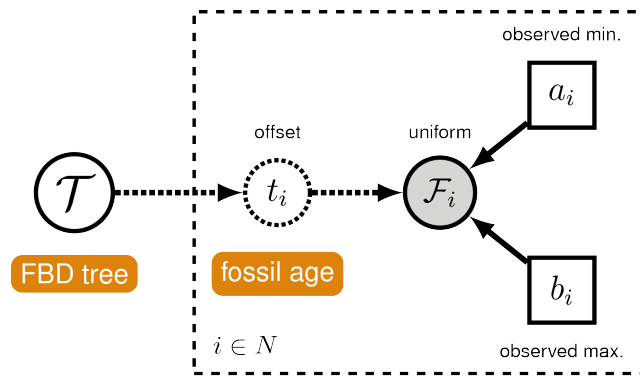
⁷ Environment-dependent diversification rates tutorial: <http://revbayes.com/tutorials/divrate/env>

⁸ Branch-specific diversification tutorial: http://revbayes.com/tutorials/divrate/branch_specific

⁹ State-dependent diversification tutorial: <http://revbayes.com/tutorials/sse/bisse-intro>

¹⁰ Fossilized birth-death range process tutorial: http://revbayes.com/tutorials/fbd_range

5.2:6 Estimating a time-calibrated phylogeny of fossil and extant taxa using RevBayes



■ **Figure 2** A graphical model of the fossil age likelihood model used in this tutorial. The likelihood of fossil observation \mathcal{F}_i is uniform and non-zero when the inferred fossil age t_i falls within the observed time interval (a_i, b_i) .

living species. In our case, the phylogeny and fossil occurrences are generated from the FBD process and we will be modeling the evolution of discrete morphological characters with two states. There are three main components to consider with modeling discrete morphological traits (as shown in Figure 1): the substitution model, the branch rate model, and the site rate model.

2.2.1 Substitution model

The substitution model describes how discrete morphological characters evolve over time. We will be using the Mk model (Lewis, 2001), a generalization of the Jukes-Cantor (Jukes and Cantor, 1969) model described for nucleotide substitutions (see Chapter 1.1 [Pupko and Mayrose 2020]).

The Mk model assumes that all transitions from one state to another occur at the same rate, for all k states. Since the characters used in this tutorial all have two states, we will specifically be using a model where $k = 2$. Thus, a transition from state 0 to state 1 is equally as likely as a transition from state 1 to state 0. For this tutorial, we focus on binary (2-state) characters for simplicity, but it is important to note that RevBayes can also accommodate multistate characters¹¹.

The evolution of discrete morphological characters is thought to occur at a very slow rate. Moreover, once some characters transition to a certain state, they rarely transition back, which means that the assumption of symmetric rates is likely violated by many empirical datasets (Wright et al., 2016; Wright, 2019). We can accommodate asymmetric transition rates¹² for each state using alternative models in RevBayes. Additionally, if some characters change symmetrically while others change asymmetrically, it is possible to partition¹³ the matrix to account for model heterogeneity among characters.

¹¹ Multistate discrete morphology tutorial: http://revbayes.com/tutorials/morph_tree/V2

¹² Asymmetric transition rates tutorial: http://revbayes.com/tutorials/morph_tree

¹³ Partitioned data analysis tutorial: <http://revbayes.com/tutorials/partition>

2.2.2 Branch-rate model

The branch-rate model describes how rates of morphological state transitions vary among branches in the tree. Each lineage in the phylogeny is assigned a value that acts as a scalar for the rate of character evolution. In our case we assume each branch has the same rate of evolution, this is a strict morphological clock (analogous to a strict molecular clock [Zuckerlandl and Pauling, 1962](#)). It is also possible to account for variation in rates among branches. These “relaxed-clock” models are commonly applied to molecular datasets and are currently implemented in RevBayes¹⁴ (see Chapter 4.4 [[Bromham 2020](#)]).

2.2.3 Site-rate model

The rate of character evolution can often vary from site to site, i.e., from one column in the matrix to another (see Chapter 1.1 [[Pupko and Mayrose 2020](#)]). Under the site-rate model, a scalar is applied to each character to account for variation in relative rates. In our case we will assume that each character belongs to one of four rate categories from the discretized gamma distribution ([Yang, 1994](#)), which is parameterized by shape parameter α and number of rate categories n . Normally a gamma distribution requires shape α and rate β parameters, however, we set our site rates to have a mean of one, which results in the constraint $\alpha = \beta$, thus eliminating the second parameter. The parameter n breaks the gamma distribution into n equiprobable bins where the rate value of each bin is equal to its mean or median.

2.3 Putting together the complete phylogenetic model

We have outlined the specific components forming the processes that govern the generation of the time tree and morphological character data; and together these modules make up the complete phylogenetic model. Figure 3 shows the complete probabilistic graphical model that includes all of the parameters we will use in this tutorial (for more on graphical models for statistical phylogenetics see [Höhna et al., 2014](#)).

The parameters represented as stochastic nodes (solid white circles) in Figure 3 are unknown random variables that are estimated in our analysis. For each of these parameters, we assume a prior distribution that describes our uncertainty in that parameter’s value. For example, we apply an exponential distribution with a rate of 10 as a prior on the mutation rate: $\mu \sim \text{Exponential}(10)$. The parameters represented as constant nodes (white boxes) are fixed to “known” or asserted values in the analysis.

2.4 Alternative models and analyses

The model choices and analysis in this tutorial focus on a simple example. Importantly, the modular design of RevBayes allows for many model choices to be swapped with more complex or biologically relevant processes for a given system. Analyses of a wide range of data types are also implemented in RevBayes (e.g., nucleotide sequences¹⁵, historical biogeographic ranges¹⁶). Moreover, it is possible to fully integrate models describing the generation of data from different sources like in the “combined-evidence” approach¹⁷ ([Ronquist et al., 2012](#); [Zhang et al., 2016](#); [Gavryushkina et al., 2017](#)) in a single, hierarchical Bayesian model. Some

¹⁴ Relaxed clock models tutorial: <http://revbayes.com/tutorials/clocks>

¹⁵ Nucleotide substitution models tutorial: <http://revbayes.com/tutorials/ctmc>

¹⁶ Modeling discrete biogeography tutorial: http://revbayes.com/tutorials/biogeo/biogeo_intro

¹⁷ FBD combined evidence tutorial: http://revbayes.com/tutorials/fbd/fbd_specimen

5.2:8 Estimating a time-calibrated phylogeny of fossil and extant taxa using RevBayes

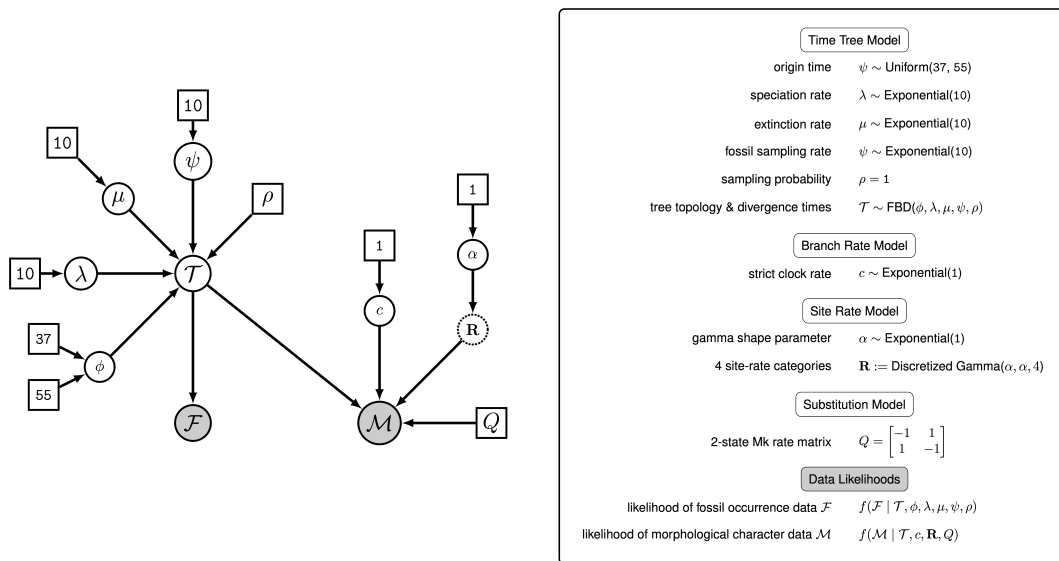


Figure 3 The complete graphical model used in the analysis described in this tutorial. This explicit representation of the model expands on the modular version depicted in Figure 1. The model components are defined in the box on the right. To simplify the model, we do not represent the components accounting for fossil age uncertainty illustrated in Figure 2.

researchers may wish to perform analyses with node calibrations¹⁸, and this approach is also possible in RevBayes. Ultimately, for any statistical analysis of empirical data, it is important to consider the processes governing the generation of those data and how they can be represented in a hierarchical model.

3 Exercise: Phylogenetic Inference under the Fossilized Birth-Death Process

In this exercise, we will create a script in Rev, the interpreted programming language used by RevBayes, that defines the model outlined above and specifies the details of the MCMC simulation. This script can be executed in RevBayes to complete the full analysis. We conclude the exercise by evaluating the performance of the MCMC and summarizing the results.

3.1 Data and files

This tutorial is maintained online at http://revbayes.com/tutorials/fbd_simple. On this page, you will also find links to other RevBayes tutorials that will supplement some of the information provided here. Additionally, this site gives links to the data files and the completed script file.

¹⁸ Molecular dating with node calibrations tutorial: <http://revbayes.com/tutorials/dating/nodedate>

On your own computer or your remote machine, create a directory called **RB_FBD_Tutorial** (or any name you like).

Then, navigate to the folder you just created and make a new directory called **data**.

In the **data** folder, add the following files (you can download these files from the URLs provided):

- **bears_taxa.tsv**: a tab-separated table listing the 18 bear species in our analysis (both fossil and extant) and their occurrence age ranges (minimum and maximum ages). For extant taxa, the minimum age is 0.0 (i.e., the present).
http://revbayes.com/tutorials/fbd_simple/data/bears_taxa.tsv
- **bears_morphology.nex**: a matrix of 62 discrete, binary (coded 0 or 1) morphological characters for our 18 species of fossil and extant bears.
http://revbayes.com/tutorials/fbd_simple/data/bears_morphology.nex

Now you can create a separate file for the Rev script.

In the **RB_FBD_Tutorial** directory created above, create a blank file called **FBD_tutorial.Rev** and open it in a text editor.

It is also possible (though not recommended) to execute this entire tutorial in the RevBayes console.

The file **FBD_tutorial.Rev** will contain all of the instructions required to load the data, assemble the different model components used in the analysis, and configure and run the Markov chain Monte Carlo (MCMC) analysis. Once you finish writing this file, you can compare your script with the **FBD_tutorial.Rev** file on the tutorial webpage.

3.2 Importing data into RevBayes

We will begin our Rev script by loading in the two data files that were downloaded and saved to the **data** directory. In RevBayes, we use functions to read the contents of files and assign them to variables in our workspace. First, we will create a variable called **taxa** that will contain the data read in from **bears_taxa.tsv**.

```
taxa <- readTaxonData("data/bears_taxa.tsv")
```

Next, we will import the morphological character matrix from **bears_morphology.nex** and assign it to the variable **morpho**. In this exercise, we are using a NEXUS-formatted data file, but it is worth noting that several other file-types are acceptable depending on the kind of data (e.g., FASTA for molecular data).

```
morpho <- readDiscreteCharacterData("data/bears_morphology.nex")
```

Here, we use the function **readDiscreteCharacterData** to load a data matrix to the workspace from a formatted file. This function can be used for discrete morphological data as well as molecular sequence data (e.g., nucleotides, amino acids).

3.3 Helper variables

Before we begin specifying the hierarchical model, it is useful to instantiate some “helper variables” that will be used in our model and MCMC specification throughout our script.

First, we will create a new constant node called `n_taxa` that is equal to the number of species in our analysis (18).

```
n_taxa <- taxa.size()
```

Next, we will create a workspace variable called `moves`, which is a vector that will contain all of the MCMC moves used to propose new states for every stochastic node in the model graph. Each time a new stochastic node is created in the model, we can append the corresponding moves to this vector.

```
moves = VectorMoves()
```

One important distinction here is that `moves` is part of the RevBayes workspace and not the hierarchical model. Thus, we use the workspace assignment operator `=` instead of the constant node assignment operator `<-`.

3.4 The fossilized birth-death process

3.4.1 Speciation and extinction rates

Two key parameters of the FBD process are the speciation rate (the rate at which lineages are added to the tree, denoted by λ in Figure 3) and the extinction rate (the rate at which lineages are removed from the tree, μ in Figure 3). We will place exponential priors on both of these values, meaning we assume each parameter is drawn independently from a different exponential distribution, where each distribution has a rate parameter equal to 10. Note that an exponential distribution with a rate of 10 has an expected value (mean) of 1/10.

Create the exponentially distributed stochastic nodes for the `speciation_rate` and `extinction_rate` using the `~` stochastic assignment operator.

```
speciation_rate ~ dnExponential(10)
extinction_rate ~ dnExponential(10)
```

The `~` operator in Rev instantiates a stochastic node in the model (i.e., a solid circle in Figure 3). Every stochastic node must be defined by a distribution. In this case, we use the exponential. In the Rev language, every distribution has the prefix `dn` to make it easier to locate the various distributions in the Rev language documentation (<http://revbayes.com/documentation>). When a stochastic node is created in the model, the distribution function assigns it an initial value by drawing a random value from the prior distribution and assigns the node to the named variable.

For every stochastic node we declare, we must also specify proposal algorithms (called *moves*) to sample the value of the parameter in proportion to its posterior probability (see Chapter 1.4 [Lartillot 2020]). If a move is not specified for a stochastic node, then it will not be estimated, but fixed to its initial value.

The extinction rate and speciation rate are both positive, real numbers (i.e., non-negative floating point variables). For both of these nodes, we will use a scaling move (`mvScale`), which proposes multiplicative changes to a parameter.

```
moves.append(mvScale(speciation_rate, weight=1))
moves.append(mvScale(extinction_rate, weight=1))
```

You will also notice that each move has a specified **weight**. This option indicates the frequency a given move will be performed in each MCMC cycle. In RevBayes, the MCMC is executed by default with a *schedule* of moves at each step of the chain, instead of just one move per step, as is done in MrBayes (Ronquist and Huelsenbeck, 2003) or BEAST (Drummond et al., 2012; Bouckaert et al., 2014). Here, if we were to run our MCMC with our current vector of two moves each with a weight of 1, then our move schedule would perform two moves in each cycle. Within a cycle, an individual move is chosen from the move list in proportion to its weight. Therefore, with both moves assigned **weight=1**, each has an equal probability of being executed and will be performed on average one time per MCMC cycle. For more information on moves and how they are performed in RevBayes, please refer to the tutorials introducing Markov chain Monte Carlo¹⁹ and nucleotide substitution models²⁰.

In addition to the speciation (λ) and extinction (μ) rates, we may also be interested in inferring the net diversification rate ($\lambda - \mu$) and the turnover (μ/λ). Since these parameters can each be expressed as a deterministic transformation of the speciation and extinction rates, we can monitor their values (i.e., track their values and print them to a file) by creating two deterministic nodes using the **:=** deterministic assignment operator.

```
diversification := speciation_rate - extinction_rate
turnover := extinction_rate/speciation_rate
```

Deterministic nodes are represented by circles with dotted borders in a probabilistic graphical model. To maintain the simplicity of the model in Figure 3, the diversification rate and turnover are not shown.

3.4.2 Extant sampling probability

Every extant bear species is represented in this dataset. Therefore, we will fix the probability of sampling an extant lineage (ρ in Figure 3) to 1. The parameter **rho** will be specified as a constant node (new values for **rho** will not be sampled in the MCMC) using the **<-** constant assignment operator.

```
rho <- 1.0
```

Because ρ is a constant node, we do not have to assign a move to this parameter because we assume the value is known and fixed.

3.4.3 Fossil sampling rate

Since our data set includes serially sampled lineages, we must also account for the rate of sampling through time. This is the fossil sampling (or recovery) rate (ψ in Figure 3), which we will instantiate as a stochastic node named **psi**. As with the speciation and extinction rates (see Section 3.4.1), we will use an exponential prior on this parameter and apply a scale move to sample values from the posterior distribution.

```
psi ~ dnExponential(10)
moves.append(mvScale(psi, weight=1))
```

¹⁹ Introduction to MCMC tutorial: <http://revbayes.com/tutorials/mcmc/>

²⁰ Nucleotide substitution models tutorial: <http://revbayes.com/tutorials/ctmc>

3.4.4 Origin time

The FBD process is conditioned on the origin time (ϕ in Figure 3), which requires specification of a node representing the age of the clade. We will set a uniform distribution on the origin age, with the lower bound set at the age of the oldest bear fossil (37 My) and the higher bound of 55 My set to the age of the most recent common ancestor of crown Carnivora estimated by recent studies (dos Reis et al., 2012). For the move, we will use a sliding window move (**mvSlide**), which samples a parameter uniformly within an interval (defined by the half-width “delta”, which is set to 1 by default). Sliding window moves can be problematic for small values, as the window may overlap zero. However, our prior on the origin age excludes values ≤ 37.0 , so this is not an issue.

```
origin_time ~ dnUnif(37.0, 55.0)
moves.append(mvSlide(origin_time, weight=1.0))
```

3.4.5 The FBD tree

Now that we have specified all of the parameters of the FBD process (λ , μ , ϕ , ψ), we will use these parameters to create the stochastic node representing the time-calibrated tree that we will call **fbd_tree**. The **fbd_tree** (\mathcal{T} in Figure 3) is generated by a fossilized birth-death distribution and is conditionally dependent on λ , μ , ϕ , and ψ . The FBD distribution function **dnFBDP** takes the FBD parameters as arguments as well as the **taxa** variable which specifies the number of terminal taxa as well as the taxon labels.

```
fbd_tree ~ dnFBDP(origin=origin_time, lambda=speciation_rate,
                 mu=extinction_rate, psi=psi, rho=rho, taxa=taxa)
```

Next, in order to sample from the posterior distribution of trees, we need to specify moves that propose changes to the topology (**mvFNPR**) and node times (**mvNodeTimeSlideUniform**). We also include a proposal (**mvCollapseExpandFossilBranch**) that will collapse or expand a fossil branch, thus sampling trees where a given fossil is either a sampled ancestor or a sampled tip. In addition, when conditioning on the origin time, we also need to explicitly sample the root age (**mvRootTimeSlideUniform**).

```
moves.append(mvFNPR(fbd_tree, weight=15.0))
moves.append(mvCollapseExpandFossilBranch(fbd_tree, origin_time,
                                         weight=6.0))

moves.append(mvNodeTimeSlideUniform(fbd_tree, weight=40.0))
moves.append(mvRootTimeSlideUniform(fbd_tree, origin_time,
                                    weight=5.0))
```

Note that we specified a higher move **weight** for each of the proposals operating on **fbd_tree** than we did for the previous stochastic nodes. This means that our move schedule will propose fifteen times as many new topologies via the **mvFNPR** move as it will new values of **speciation_rate** using **mvScale**, for example. By increasing the number of times new values are proposed for a parameter, we are increasing the sampling intensity for that parameter. Typically, we do this for parameters that we are particularly interested in or for parameters that tend to induce long mixing times. A node like \mathcal{T} in our graphical model (Figure 3) represents a complex set of variables: the tree topology and all divergence times. Moreover, the likelihoods of our fossil occurrence data and the morphological character data are both conditionally dependent on the time tree. Such complex variables require more extensive sampling than other nodes.

3.4.6 Sampling fossil occurrence times

We need to account for uncertainty in the age estimates of our fossils using the observed minimum and maximum stratigraphic ages that are provided in the file `bears_taxa.tsv`. We can represent the fossil likelihood using any uniform distribution that is non-zero when the likelihood is equal to one (see Section 2.1.3). For example, if t_i is the inferred fossil age and (a_i, b_i) is the observed stratigraphic interval, we know the likelihood is equal to one when $a_i < t_i < b_i$, or equivalently $t_i - b_i < 0 < t_i - a_i$. So we can represent this likelihood using a uniform random variable, uniformly distributed in $(t_i - b_i, t_i - a_i)$ and clamped at zero.

To do this, we will get all the fossils from the tree and use a `for` loop to iterate over them. For each fossil observation, we will create a uniform random variable representing the likelihood, based on the minimum and maximum ages specified in the file `bears_taxa.tsv`.

```
fossils = fbd_tree.getFossils()
for(i in 1:fossils.size())
{
  t[i] := tmrca(fbd_tree, clade(fossils[i]))

  a_i = fossils[i].getMinAge()
  b_i = fossils[i].getMaxAge()

  F[i] ~ dnUniform(t[i] - b_i, t[i] - a_i)
  F[i].clamp( 0 )
}
```

Finally, we will add a move that samples the ages of all the fossils on the tree.

```
moves.append(mvFossilTimeSlideUniform(fbd_tree, origin_time,
                                       weight=5.0))
```

3.4.7 Monitoring parameters of interest

There are additional parameters that may be of particular interest to us that are not directly sampled as part of the graphical model defined thus far. As with the diversification and turnover nodes specified in Section 3.4.1, we can create deterministic nodes to sample the posterior distributions of these parameters. Here we will create a deterministic node called `num_samp_anc` that will compute the number of sampled ancestors in our `fbd_tree`.

```
num_samp_anc := fbd_tree.numSampledAncestors()
```

We are also interested in the age of the most-recent-common ancestor (MRCA) of all living bears. To monitor this age in our MCMC sample, we must use the `clade()` function to identify the node corresponding to the MRCA. Once this clade is defined we can instantiate a deterministic node called `age_extant` that will record the age of the MRCA of all living bears, using the `tmrca()` function.

```
clade_extant = clade("Ailuropoda_melanoleuca", "Tremarctos_ornatus",
                   "Melursus_ursinus", "Ursus_arctos",
                   "Ursus_maritimus", "Helarctos_malayanus",
                   "Ursus_americanus", "Ursus_thibetanus")
age_extant := tmrca(fbd_tree, clade_extant)
```

In the same way we monitored the MRCA of the extant bears, we can also monitor the age of a fossil taxon that we may be interested in recording. We will monitor the marginal

5.2:14 Estimating a time-calibrated phylogeny of fossil and extant taxa using RevBayes

distribution of the age of *Kretzoiarctos beatrix* (Abella et al., 2012), which is sampled between 11.2–11.8 My.

```
age_Kretzoiarctos_beatrix := tmrca(fbd_tree,
                                clade("Kretzoiarctos_beatrix"))
```

3.5 Modeling the evolution of binary morphological characters

The next part of the graphical model, we will define specifies the model of morphological character evolution. This component includes the substitution model, the model of rate variation among characters, and the model of rate variation among branches (Figure 3).

As stated in Section 2.2.1, we will use the Mk model to describe the substitution process. Because the Mk model is a generalization of the Jukes-Cantor model (Jukes and Cantor, 1969), we will initialize our instantaneous rate matrix from a Jukes-Cantor matrix (see Chapter 1.1 [Pupko and Mayrose 2020]). The constant node **Q_morpho** corresponds to the two-state rate matrix Q in Figure 3.

```
Q_morpho := fnJC(2)
```

We will assume that rates vary among characters in our data matrix according to a discretized gamma distribution (described in Section 2.2.3). For this model, we create a vector of rates named **rates_morpho** which is the product of a function **fnDiscretizeGamma()** that divides up a gamma distribution into a set of equal-probability bins (**R** in Figure 3). Here, our only stochastic node is **alpha_morpho** (α in Figure 3), which is the shape parameter of the discretized gamma distribution.

```
alpha_morpho ~ dnExponential(1.0)
rates_morpho := fnDiscretizeGamma(alpha_morpho, alpha_morpho, 4)

moves.append(mvScale(alpha_morpho, weight=5.0))
```

The phylogenetic model also assumes that each branch has a rate of morphological character change. For simplicity, we will assume a strict morphological clock—meaning that every branch has the same rate represented by the stochastic node **clock_morpho** (c in Figure 3), which is drawn from an exponential distribution (see Section 2.2.2).

```
clock_morpho ~ dnExponential(1.0)
moves.append(mvScale(clock_morpho, weight=4.0))
```

3.5.1 The phylogenetic CTMC

If you refer to Figure 3, you will see that we have defined almost all of the components of the complete model except for the observed node representing our morphological character data (\mathcal{M}). The character matrix is a clamped stochastic node that is generated by a phylogenetic continuous-time Markov chain (CTMC) distribution (see Chapter 1.1 [Pupko and Mayrose 2020]). This node is conditionally dependent on the time tree (\mathcal{T} : **fbd_tree**), clock rate (c : **clock_morpho**), site rates (**R**: **rates_morpho**), and the two-state Mk rate matrix (Q : **Q_morpho**). With all of these nodes instantiated in the graphical model, we can now connect the components by defining the node representing our observed morphological data.

There are some unique aspects to specifying a phylogenetic CTMC for morphological data. You will notice that we have an option called **coding**. This option allows us to condition on

biases in the way the morphological data were collected (i.e., ascertainment bias). By setting `coding=variable` we can correct for coding only variable characters (as discussed in [Lewis, 2001](#)).

```
phyMorpho ~ dnPhyloCTMC(tree=fbd_tree, siteRates=rates_morpho,
                        branchRates=clock_morpho, Q=Q_morpho,
                        type="Standard", coding="variable")
phyMorpho.clamp(morpho)
```

Now that we have defined our complete model, we can create a workspace variable that packages the entire model graph. This makes it easy to pass the whole model to functions that will set up our MCMC analysis. This variable is created using the `model()` function, which takes only a single node in the graph. We will use the `fbd_tree` node, but you can try this with an alternative node (e.g., `clock_morpho`, `rho`, etc.). As long as you have established all of the connections among the model parameters, the `model()` function will find every other node by traversing the edges of the graph (Figure 3).

```
mymodel = model(fbd_tree)
```

3.6 Monitoring variables

We have defined the full probabilistic graphical model shown in Figure 3 and now we are ready to specify the details of our MCMC analysis. The first step in setting up the analysis is to create *monitors* that will record the values of each parameter in our model for every sampled cycle of the MCMC. The sampled values are saved to file (or printed to screen) and can be summarized when our MCMC simulation is complete.

Let's create three different monitor objects for this analysis. To manage the monitors in RevBayes, we create another workspace variable called `monitors` that is a vector containing the three monitor variables.

```
monitors = VectorMonitors()
```

We will append our first monitor to the `monitors` vector. This will create a file called `bears.log` in a directory called `output` (if this directory does not already exist, RevBayes will create it). The function `mnModel()` initializes a monitor that saves all of the numerical parameters in the model to a tab-delimited file. This file is useful for summarizing marginal posteriors in statistical plotting tools like Tracer ([Rambaut et al., 2018](#)) or R ([R Core Team, 2020](#)). We will exclude the `F` vector from logging, as it is purely used as an auxiliary variable for estimating fossil ages, and is clamped to 0. Additionally, we also specify how frequently we sample our Markov chain by setting the `printgen` option. We will sample every 10 cycles of our MCMC.

```
monitors.append(mnModel(filename="output/bears.log", printgen=10,
                       exclude=["F"]))
```

You may think that sampling every 10 generations may be too frequent to avoid correlation between samples in our MCMC. However, recall that a single “generation” in RevBayes performs a schedule of moves that is determined by the number of moves in the `moves` vector and the weights assigned to those moves (see Section 3.4.1). Thus, a single generation in this analysis will involve 26 moves, so if we record every 10 generations, there will be 260 moves between each sample.

5.2:16 Estimating a time-calibrated phylogeny of fossil and extant taxa using RevBayes

We want to create a separate file containing samples of the tree and branch lengths since these will not be saved by the monitor defined above. To save the tree parameter, we can use the `mnFile()` function that saves specific parameters to a file. We indicate the parameters by including them in the function's options.

```
monitors.append(mnFile(filename="output/bears.trees", printgen=10,
                      fbd_tree))
```

The final monitor will print updates of our MCMC to the screen. The screen monitor function `mnScreen()` allows us to add parameters in our model that will be displayed along with a few default values (including the current iteration, posterior, likelihood, and prior). We will monitor the age of the MRCA of the living bears, the number of sampled ancestors, and the origin time in the screen output.

```
monitors.append(mnScreen(printgen=10, age_extant, num_samp_anc,
                        origin_time))
```

3.7 Setting up and running the MCMC sampler

Our Rev script specifies the three major parts of our MCMC analysis: a model (`myModel`), a list of MCMC proposals (`moves`), and a way to save the values sampled by our Markov chain (`monitors`). With these three components, we can set up our analysis using the `mcmc()` function. This function creates a workspace variable that we can use to execute the MCMC simulation.

```
myMCMC = mcmc(myModel, monitors, moves)
```

Using our variable `myMCMC`, we can execute the `run()` member method to start our MCMC sampler.

```
myMCMC.run(generations=10000)
```

Finally, since we are going to save this analysis in a script file and run it in RevBayes, it is useful to include a statement that will quit the program when the run is complete.

```
q()
```

Your script is now complete! Note that you can compare your script to the `FBD_tutorial.Rev` file provided on the tutorial webpage.

Save the `FBD_tutorial.Rev` file in the `RB_FBD_Tutorial` directory.

3.8 Execute the analysis script in RevBayes

With your script complete and data files in the proper location, you can execute the `FBD_tutorial.Rev` script in RevBayes.

Run the RevByes executable.

On Unix systems, if the RevBayes is in your path, you simply need to navigate to the `RB_FBD_Tutorial` directory and type `rb`.

If the RevBayes executable is not in your path, you can execute it and then change your working directory within the program using the `setwd()` function which takes the absolute path to your directory as an argument.

```
setwd("<path to>/RB_FBD_Tutorial")
```

Once RevBayes is in the correct working directory (`RB_FBD_Tutorial`), you can then use the `source()` function to feed RevBayes your master script file (`FBD_tutorial.Rev`).

```
source("FBD_tutorial.Rev")
```

This will execute the analysis and you should see the various parameters—specified when you initialized the screen monitor—printed to the screen every 10 generations. When the analysis is complete, RevBayes will quit and you will have a new directory called `output` that will contain all of the files you specified with the monitors.

3.9 Results

Two files are created by the monitors in Section 3.6. These files, located in the `output` directory contain the record of values sampled for the various parameters of the model over the course of the MCMC. In the following sections, we will assess the performance of our MCMC sampler and summarize the marginal posterior distributions of numerical parameters (in the file `bears.log`) and the time-calibrated phylogeny (in the file `bears.trees`).

3.9.1 Evaluating the MCMC sampler

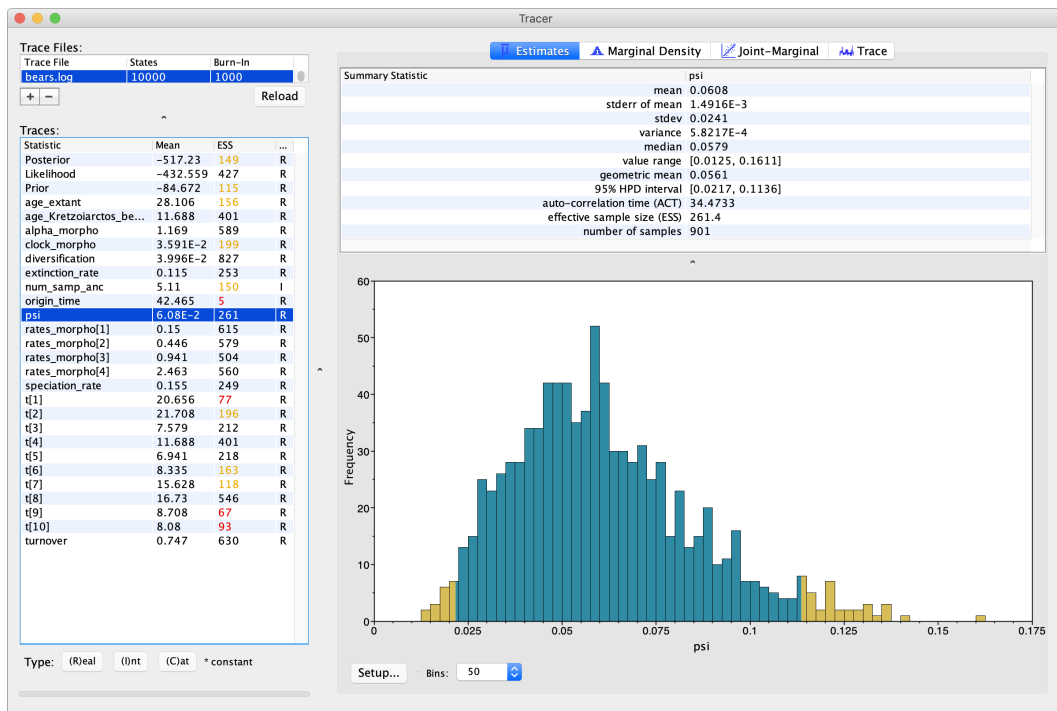
The first step when analyzing the output of an MCMC run is to check whether the chain has converged on the stationary distribution and sampled effectively (i.e., achieved “good mixing”). This can be done by loading the parameter log, in our case the file `bears.log`, in a program such as Tracer²¹ (Rambaut et al., 2018), shown in Figure 4.

On the left side is a panel summarizing all the parameters appearing in the log, with their mean estimate and ESS value (effective sample size). The ESS of a parameter determines whether the chain has adequately sampled the associated variable: values above 200 are considered “good”, whereas values below 200, highlighted by Tracer in yellow or red, indicate poor mixing. Explicitly, the ESS measures the degree of independence between samples and parameters with signatures of autocorrelation between samples are indicative of an inadequate sampler.

Here we can see that the chain has mixed well for some parameters, but not others. In particular, we see low ESS values for the origin time (`origin_time`) and the ages of some fossil tips (`t[1]`, `t[9]` and `t[10]`). This may indicate that the MCMC sampler has not converged on the stationary distribution for these parameters, which are associated with the FBD tree. What this assessment reveals is that we did not perform enough proposals for these parameters. Thus, it will be important to run the MCMC for more generations

²¹Tracer: <http://beast.community/tracer>

5.2:18 Estimating a time-calibrated phylogeny of fossil and extant taxa using RevBayes



■ **Figure 4** Analysis in Tracer of the parameter estimates obtained on the bears dataset.

(specified in Section 3.7) and/or increase the weights of moves applied to these stochastic nodes (e.g., the `mvSlide` applied to `origin_time` in Section 3.4.4). For more details on diagnosing convergence of MCMC samples under the FBD model, please see the tutorial on combined-evidence analysis in RevBayes²².

3.9.2 Summarizing the tree

Once we are certain that our MCMC has effectively sampled the joint posterior distribution of our model parameters, we can summarize the tree topology, branch times, and fossil ages that were saved to `output/bears.trees` using some built-in RevBayes functions.

Run the RevBayes executable, making sure that the working directory is `RB_FBD_Tutorial`.

The file `bears.trees` contains the trees and associated parameters that were sampled every 10 generations by our monitor. In RevBayes, we often refer to a set of samples from our MCMC as a “trace”.

Begin by loading the tree trace into RevBayes from the `bears.trees` file.

```
trace = readTreeTrace("output/bears.trees")
```

By default, a burn-in of 25% is used when reading in the tree trace (250 trees in our case). Note that this is different from Tracer, which uses a burn-in fraction of 10% by

²²FBD combined evidence tutorial: http://revbayes.com/tutorials/fbd/fbd_specimen

default. You can specify a different burn-in fraction, say 50%, by typing the command `trace.setBurnin(500)`.

Now we will use the `mccTree()` function to return a maximum clade credibility (MCC) tree. The MCC tree is the tree with the maximum product of the posterior clade probabilities. When considering trees with sampled ancestors, we refer to the maximum sampled ancestor clade credibility (MSACC) tree (Gavryushkina et al., 2017).

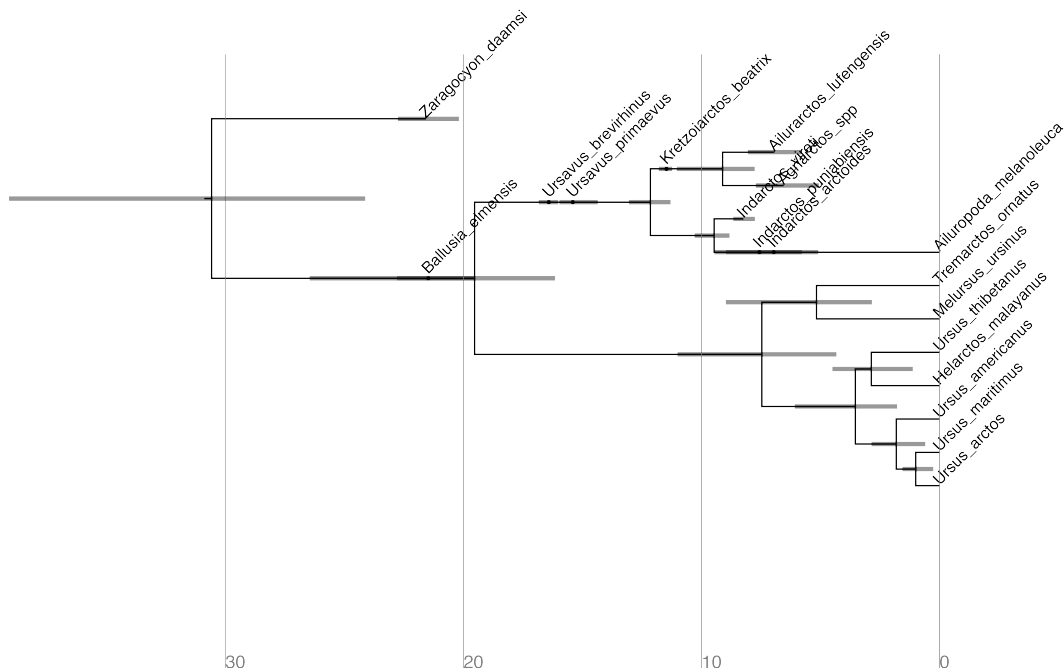
```
mccTree(trace, file="output/bears.mcc.tre")
```

When there are sampled ancestors present, visualizing the tree can be fairly difficult in traditional tree viewers. We will make use of a browser-based tree visualization tool called IcyTree (Vaughan, 2017), which can be accessed at <https://icytree.org>. IcyTree has many unique options for visualizing phylogenetic trees and can produce publication-quality vector image files (i.e., SVG). Additionally, it correctly represents sampled ancestors on the tree as nodes, each with only one descendant (Figure 5).

Navigate to <https://icytree.org> and open the file `output/bears.mcc.tre` in IcyTree.

Try to replicate the tree in Figure 5 (Hint: *Style > Mark Singletons*).

- ★ Why might a node with a sampled ancestor be referred to as a singleton?
- ★ How can you see the names of the fossils that are putative sampled ancestors?
- ★ What is the posterior probability that *Zaragocyon daamsi* is a sampled ancestor?



■ **Figure 5** Maximum sampled ancestor clade credibility (MSACC) tree of bear species used in this tutorial.

3.10 Summary

In this tutorial, we have introduced core information about how morphological and age information are modeled for use with the FBD model in RevBayes. We have also discussed important aspects of executing and summarizing MCMC analysis. This exercise uses a simplified data set and set of models for analysis of fossil and extant data. Most researchers working on living taxa have access to molecular (including genomic) data and may be interested in applying these methods to much larger datasets and more complex problems. Note that the goal of this tutorial is to provide a concise introduction to the framework for analysis of paleontological and neontological data in RevBayes. For more information on how to apply RevBayes datasets combining morphological and molecular characters, please refer to the tutorial describing this approach: http://revbayes.com/tutorials/fbd/fbd_specimen.

4 Bayesian Phylogenetic Inference in RevBayes

This tutorial provided a very focused look at the range of models and methods available in RevBayes. There are currently numerous approaches available and under active development by RevBayes team members. These include (but are not limited to):

- Model selection using Bayes factors
- Model averaging of substitution models
- Approaches for assessing model adequacy using posterior prediction
- Analysis of multi-state discrete morphological characters under asymmetric models
- Various relaxed-clock models
- Models that vary diversification over time
- State-dependent diversification models
- Analysis of chromosome evolution
- Lineage specific diversification rate variation
- Analysis of continuous characters under Brownian motion and Ornstein-Uhlenbeck models
- Ancestral area estimation and phylogenetic analysis of historical biogeography
- Gene-tree/species-tree inference under the multi-species coalescent

The flexibility of the modeling framework implemented in RevBayes provides a rich tool-kit for phylogenetic analysis under complex models. Moreover, the RevBayes core and probabilistic graphical models make it possible for new developers to readily implement their ideas in an existing code base. Members of the RevBayes Development Team are working to expand the documentation for new developers (<http://revbayes.com/developer>) to facilitate the growth of new statistical models and methods available in RevBayes.

Acknowledgements

We wish to thank the editors C. Scornavacca, F. Delsuc, and N. Galtier, for the opportunity to contribute this tutorial to *Phylogenetics in the Genomic Era*. We also thank S. Höhna for providing comments on this manuscript. All RevBayes tutorials benefit from the generous feedback provided by researchers applying RevBayes and workshop participants. The methods described in this tutorial are available because of the efforts of the RevBayes Developer Team, a collaborative network of scientific programmers working on phylogenetic problems.

References

- Abella, J., Alba, D. M., Robles, J. M., Valenciano, A., Rotgers, C., Carmona, R., Montoya, P., and Morales, J. (2012). *Kretzoiarctos* gen. nov., the oldest member of the giant panda clade. *PLoS One*, 17:e48985.
- Barido-Sottani, J., Aguirre-Fernández, G., Hopkins, M. J., Stadler, T., and Warnock, R. (2019). Ignoring stratigraphic age uncertainty leads to erroneous estimates of species divergence times under the fossilized birth–death process. *Proceedings of the Royal Society B: Biological Sciences*, 286(1902):20190685.
- Bouckaert, R., Heled, J., Kühnert, D., Vaughan, T., Wu, C.-H., Xie, D., Suchard, M. A., Rambaut, A., and Drummond, A. J. (2014). BEAST 2: a software platform for Bayesian evolutionary analysis. *PLoS Computational Biology*, 10(4):e1003537.
- Bromham, L. (2020). Substitution rate analysis and molecular evolution. In Scornavacca, C., Delsuc, F., and Galtier, N., editors, *Phylogenetics in the Genomic Era*, chapter 4.4, pages 4.4:1–4.4:21. No commercial publisher | Authors open access book.
- Condamine, F. L., Rolland, J., Höhna, S., Sperling, F. A., and Sanmartín, I. (2018). Testing the role of the Red Queen and Court Jester as drivers of the macroevolution of Apollo butterflies. *Systematic Biology*, 67(6):940–964.
- dos Reis, M., Inoue, J., Hasegawa, M., Asher, R. J., Donoghue, P. C., and Yang, Z. (2012). Phylogenomic datasets provide both precision and accuracy in estimating the timescale of placental mammal phylogeny. *Proceedings of the Royal Society B: Biological Sciences*, 279(1742):3491–3500.
- Drummond, A., Suchard, M., Xie, D., and Rambaut, A. (2012). Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Molecular Biology and Evolution*, 29:1969–1973.
- Freyman, W. A. and Höhna, S. (2018). Cladogenetic and anagenetic models of chromosome number evolution: a Bayesian model averaging approach. *Systematic Biology*, 67(2):1995–215.
- Freyman, W. A. and Höhna, S. (2019). Stochastic character mapping of state-dependent diversification reveals the tempo of evolutionary decline in self-compatible Onagraceae lineages. *Systematic Biology*, 68(3):505519.
- Gavryushkina, A., Heath, T. A., Ksepka, D. T., Stadler, T., Welch, D., and Drummond, A. J. (2017). Bayesian total-evidence dating reveals the recent crown radiation of penguins. *Systematic Biology*, 66:57–73.
- 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. (2015). The time-dependent reconstructed evolutionary process with a key-role for mass-extinction events. *Journal of Theoretical Biology*, 380:321–331.
- Höhna, S., Coghill, L. M., Mount, G. G., Thomson, R. C., and Brown, J. M. (2018). P³: Phylogenetic posterior prediction in RevBayes. *Molecular Biology and Evolution*, 35(4):1028–1034.
- Höhna, S., Freyman, W. A., Nolen, Z., Huelsenbeck, J. P., May, M. R., and Moore, B. R. (2019). A Bayesian approach for estimating branch-specific speciation and extinction rates. *bioRxiv*, <https://doi.org/10.1101/555805>.
- Höhna, S., Heath, T. A., Boussau, B., Landis, M. J., Ronquist, F., and Huelsenbeck, J. P. (2014). Probabilistic graphical model representation in phylogenetics. *Systematic Biology*, 63(5):753–771.

5.2:22 REFERENCES

- Höhna, S., Landis, M. J., Heath, T. A., Boussau, B., Lartillot, N., Moore, B. R., Huelsenbeck, J. P., and Ronquist, F. (2016). RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Systematic Biology*, 65(4):726–736.
- Jukes, T. and Cantor, C. (1969). Evolution of protein molecules. *Mammalian Protein Metabolism*, 3:21–132.
- Kendall, D. G. (1948). On the generalized “birth-and-death” process. *The Annals of Mathematical Statistics*, 19(1):1–15.
- Lartillot, N. (2020). The bayesian approach to molecular phylogeny. In Scornavacca, C., Delsuc, F., and Galtier, N., editors, *Phylogenetics in the Genomic Era*, chapter 1.4, pages 1.4:1–1.4:17. No commercial publisher | Authors open access book.
- Lewis, P. O. (2001). A likelihood approach to estimating phylogeny from discrete morphological character data. *Systematic Biology*, 50(6):913–925.
- Maddison, W., Midford, P., and Otto, S. (2007). Estimating a binary character’s effect on speciation and extinction. *Systematic Biology*, 56(5):701.
- Nee, S., May, R. M., and Harvey, P. H. (1994). The Reconstructed Evolutionary Process. *Philosophical Transactions: Biological Sciences*, 344(1309):305–311.
- Pett, W. and Heath, T. A. (2020). Inferring the timescale of phylogenetic trees from fossil data. In Scornavacca, C., Delsuc, F., and Galtier, N., editors, *Phylogenetics in the Genomic Era*, chapter 5.1, pages 5.1:1–5.1:18. No commercial publisher | Authors open access book.
- Pupko, T. and Mayrose, I. (2020). A gentle introduction to probabilistic evolutionary models. In Scornavacca, C., Delsuc, F., and Galtier, N., editors, *Phylogenetics in the Genomic Era*, chapter 1.1, pages 1.1:1–1.1:21. No commercial publisher | Authors open access book.
- R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Rambaut, A., Drummond, A. J., Xie, D., Baele, G., and Suchard, M. A. (2018). Posterior summarization in Bayesian phylogenetics using Tracer 1.7. *Systematic Biology*, 67(5):901–904.
- Ronquist, F. and Huelsenbeck, J. (2003). MrBayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics*, 19(12):1572–1574.
- Ronquist, F., Klopfstein, S., Vilhelmsen, L., Schulmeister, S., Murray, D. L., and Rasnitsyn, A. P. (2012). A total-evidence approach to dating with fossils, applied to the early radiation of the Hymenoptera. *Systematic Biology*, 61(6):973–999.
- Stadler, T. (2010). Sampling-through-time in birth-death trees. *Journal of Theoretical Biology*, 267(3):396–404.
- Stadler, T., Gavryushkina, A., Warnock, R. C., Drummond, A. J., and Heath, T. A. (2018). The fossilized birth-death model for the analysis of stratigraphic range data under different speciation modes. *Journal of Theoretical Biology*, 447:41–55.
- Thompson, E. A. (1975). *Human Evolutionary Trees*. Cambridge University Press, Cambridge, UK.
- Vaughan, T. G. (2017). IcyTree: rapid browser-based visualization for phylogenetic trees and networks. *Bioinformatics*, 33(15):2392–2394.
- Wright, A. M. (2019). A systematist’s guide to estimating Bayesian phylogenies from morphological data. *Insect systematics and diversity*, 3(3):2.
- Wright, A. M., Lloyd, G. T., and Hillis, D. M. (2016). Modeling character change heterogeneity in phylogenetic analyses of morphology through the use of priors. *Systematic Biology*, 65(4):602–611.

- Yang, Z. (1994). Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *Journal of Molecular Evolution*, 39(3):306–314.
- Zhang, C., Stadler, T., Klopstein, S., Heath, T. A., and Ronquist, F. (2016). Total-evidence dating under the fossilized birth-death process. *Systematic Biology*, 65(2):228–249.
- Zuckerkandl, E. and Pauling, L. (1962). Molecular disease, evolution, and genetic heterogeneity. In Kasha, M. and Pullman, B., editors, *Horizons in Biochemistry*, pages 189–225. Academic Press, New York.