Phylogenetic Inference using RevBayes

Total-evidence Dating under the FBD Model

Tracy A. Heath, April Wright, and Walker Pett

1 Introduction

Ronquist et al. (2012)

Exercise is in Section 3.

1.1 Models

1.1.1 Sequence Evolution

Point to other tutorials (e.g., GTR stuff)

1.1.2 Morphological Character Change

Mk models and ascertainment bias

1.1.3 Lineage-Specific Substitution Rates

Clocks (Zuckerkandl and Pauling 1962) and relaxing them

1.1.4 Lineage Diversification and Sampling

Birth-death processes and FBD

2 Prerequisites

What do you need to know before doing this?

2.1 Requirements

We assume that you have read and hopefully completed the following tutorials:

- RB_Getting_Started
- RB_Basics_Tutorial

Note that the RB_Basics_Tutorial introduces the basic syntax of `Rev` but does not cover any phylogenetic models. You may skip the RB_Basics_Tutorial if you have some familiarity with `R`. We tried to keep this tutorial very basic and introduce all the language concepts on the way. You may only need the RB_Basics_Tutorial for a more in-depth discussion of concepts in `Rev`.

3 Exercise: Estimating the Phylogeny and Divergence Times of Fossil and Extant Bears

Information about the exercise, citations for the data, questions.

3.1 Data files

We provide the data files which we will use in this tutorial. You may want to use your own data instead. In the **data** folder, you will find the following files

- **bears_taxa.tsv**: a tab-separated table listing every bear species (both fossil and extant) and their occurrence dates. For extant taxa, the occurrence date is `0.0` (*i.e.,* the present) and for fossil species, the occurrence date is equal to the mean of the age range (the ranges are defined in a separate file).
- **bears_cytb.nex**: an alignment in NEXUS format of 1,000 bp of cytochrome-b sequences for 10 bear species. This alignment includes 8 living bears and 2 extinct sub-fossil bears.
- **bears_morphology.nex**: a matrix of 62 discrete, binary (coded `0` or `1`) morphological characters for 18 species of fossil and extant bears.
- **bears_fossil_intervals.tsv**: a tab-separated table containing the age ranges (minimum and maximum in millions of years) for 14 fossil bears.

3.2 Getting Started

On your own computer, create a directory called `RB_TotalEvidenceDating_FBD_Tutorial` (or any name you like).

In this directory download and unzip the archive containing the data files: `data.zip`.

Additionally, create a new directory (in **RB_TotalEvidenceDating_FBD_Tutorial**) called `scripts`

When you execute `RevBayes` in this exercise, you will do so within the main directory you created (**RB_TotalEvidenceDating_FBD_Tutorial**).

3.3 Creating `Rev` Files

For complex models and analyses, it is best to create `Rev` script files that will contain all of the model parameters, moves, and functions. In this exercise, you will work primarily in your text editor and create a set of modular files that will be easily managed and interchanged. You will write the following files from scratch in a text editor* and save them in the **scripts** directory:

- **mcmc_TEFBD.Rev**: the master `Rev` file that loads the data, the separate model files, and specifies the monitors and MCMC sampler.
- **model_FBDP_TEFBD.Rev**: specifies the model parameters and moves required for the fossilized birth-death prior on the tree topology, divergence times, fossil occurrence times, and diversification dynamics.
- **model_UCExp_TEFBD.Rev**: specifies the components of the uncorrelated exponential model of lineage-specific substitution rate variation.

- **model_GTRG_TEFBD.Rev**: specifies the parameters and moves for the general-time reversible model of sequence evolution with gamma-distributed rates across sites (GTR+Γ).
- **model_Morph_TEFBD.Rev**: specifies the model describing discrete morphological character change (binary characters) under a strict morphological clock.

*Note: if you do not already have a good text editor that you like, we recommend one that has features for syntax coloring, easy navigation between different files, line numbers, etc. A good option is Sublime Text, which is available for Mac OSX, Windows, and Linux.

All of the files that you will create are also provided in the RevBayes tutorial repository. Please refer to these files to verify or troubleshoot your own scripts.

3.4 Start the Master Rev File and Import Data

Open your text editor and create the master Rev file called mcmc_TEFBD.Rev in the **scripts** directory.

Enter the Rev code provided in this section in the new model file.

The file you will begin in this section will be the one you load into RevBayes when you've completed all of the components of the analysis. In this section you will begin the file and write the Rev commands for loading in the taxon list and managing the data matrices. Then, starting in Section 3.5, you will move on to writing modular files for each of the model components. Once the model files are complete, you will return to editing **mcmc_TEFBD.Rev** and complete the Rev script with the instructions given in Section 3.9.

3.4.1 Load Taxon List

Begin the Rev script by loading in the list of taxon names from the **bears_taxa.tsv** file using the **readTaxonData()** function.

```
taxa <- readTaxonData("data/bears_taxa.tsv", delimiter=TAB)
```

This function reads the tab-delimited file and creates a variable called **taxa** that is a list of all of the taxon names relevant to this analysis. This list includes all of the fossil and extant bears.

3.4.2 Load Data Matrices

RevBayes uses the function **readDiscreteCharacterData()** to load a data matrix to the workspace from a formatted file. This function can be used for both molecular sequences and discrete morphological characters.

Load the cytochrome-b sequences from file and assign the data matrix to a variable called **cytb**.

```
cytb <- readDiscreteCharacterData("data/bears_cytb.nex")
```

Next, import the morphological character matrix and assign it to the variable **morpho**.

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

3.4.3 Add Missing Taxa

In the descriptions of the files in Section 3.1, we mentioned that the two data matrices have different numbers of taxa. Thus, we must add any taxa that are not found in the molecular (**cytb**) partition (*i.e.,* are only found in the fossil data) to that data matrix as missing data, and do the same with the morphological data partition (**morpho**). In order for all the taxa to appear on the same tree, they all need to be part of the same dataset, as opposed to present in separate datasets. This ensures that there is a unified taxon set that contains all of our tips.

```
cytb.addMissingTaxa( taxa )
morpho.addMissingTaxa( taxa )
```

3.4.4 Create Helper Variables

Before we begin writing the `Rev` scripts for each of the model components, we need to instantiate a couple "helper variables" that will be used by downstream parts of our model specification files. These variables will be used in more than one of the module files so it's best to initialize them in the master file.

Create a new constant node called **n_taxa** that is equal to the number of species in our analysis (22).

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

Next, create a *workspace variable* called **mvi**. This variable is an iterator that will build a vector containing all of the MCMC moves used to propose new states for every stochastic node in the model graph. After each time a new move is added to the vector, **mvi** will be incremented by a value of **1**.

```
mvi = 1
```

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

Save your current working of **model_FBDP_TEFBD.Rev** in the **scripts** directory.

We will now move on to the next `Rev` file and will complete **model_FBDP_TEFBD.Rev** in Section 3.9.

3.5 The Fossilized Birth-Death Process

Open your text editor and create the fossilized birth-death model file called **model_FBDP_TEFBD.Rev** in the **scripts** directory.

Enter the Rev code provided in this section in the new model file.

Two key parameters of the FBD process are the birth rate (the rate at which lineages are added to the tree) and the death rate (the rate at which lineages are removed from the tree). We'll place exponential priors on both of these values. An exponential prior with a $\lambda = 10$ places a higher probability on values closer to zero than one for these parameters.

```
birth_rate ~ dnExponential(10)
death_rate ~ dnExponential(10)
```

Now that the priors have been specified, we give RevBayes some information on how to sample values for our parameters. We'll use a scaling move, which changes the value sampled multiplicatively with the tuning parameter. We will use three different tuning parameters, which govern the size of the move. Including multiple tuning parameters improves mixing.

```
moves[mvi++] = mvScale(birth_rate, lambda=0.01, weight=3.0)
moves[mvi++] = mvScale(birth_rate, lambda=0.1, weight=3.0)
moves[mvi++] = mvScale(birth_rate, lambda=1.0, weight=3.0)
moves[mvi++] = mvScale(death_rate, delta=0.01, weight=3.0)
moves[mvi++] = mvScale(death_rate, delta=0.1, weight=3.0)
moves[mvi++] = mvScale(death_rate, delta=1, weight=3.0)
```

In order to print the states of model variables output files (also called *monitoring*), we need to create deterministic nodes for the diversification and turnover. Deterministic nodes are value transformations between existing stochastic nodes. So we will define diversification and turnover as deterministic nodes.

```
diversification := birth_rate - death_rate
turnover := death_rate/birth_rate
```

All extant bears are represented in this dataset. Therefore, we can fix the sampling probability of extant lineages to 1.

```
rho <- 1.0
```

The rate of sampling fossils (ψ), on the other hand is not known. We will use an exponential prior on this parameter as well, and use a slide move to sample values from our distribution.

```
psi ~ dnExponential(10)
moves[mvi++] = mvScale(birth_rate, lambda=0.01, weight=3.0)
moves[mvi++] = mvScale(birth_rate, lambda=0.1, weight=3.0)
moves[mvi++] = mvScale(birth_rate, lambda=1.0, weight=3.0)
```

Under the FBD model, the process is conditioned on the age of the origin, or the start of the process. We will specify a uniform distribution on the age of the origin. If you looked in the bears taxa file, you might notice that the age of the oldest fossil is slightly younger than the upper bound of the uniform distribution on the origin age. For this parameter, we will use a sliding window move. A sliding window move samples within an interval (defined by **delta**). Sliding window moves can be tricky for small values, as the window may overlap zero. However, for parameters such as the origin, there is little risk of this being an issue.

```
origin_time ~ dnUnif(37.0, 55.0)
moves[mvi++] = mvSlide(origin_time, delta=0.01, weight=10.0)
moves[mvi++] = mvSlide(origin_time, delta=0.1, weight=10.0)
moves[mvi++] = mvSlide(origin_time, delta=1, weight=10.0)
```

All the parameters of the FBD process are now defined. The next step is to combine these parameters to define the tree prior as the FBD.

```
tree_prior = dnFBDP(origin=origin_time, lambda=birth_rate, mu=death_rate, psi=psi, rho
    =rho, taxa=taxa)
```

Next, we will define the **fbd_tree** variable as a random variable. It will be used to generate trees under the FBD process that conform to our clade constraints.

```
fbd_tree ~ dnConstrainedTopology(tree_prior, constraints)
```

Finally, we can also create deterministic nodes for other quantities we might be interested in monitoring. Below, we will define a monitor that prints the number of fossils that are inferred to be 'sampled ancestors' - lineages that are present in the phylogeny, and have descendants present on the tree. We will also define a deterministic node for the age of the crown group of bears, using the previously-defined extant bear constraint (Section **??**).

```
sa := fbd_tree.numSampledAncestors();
crown := tmrca(fbd_tree, clade_extant)
```

3.6 The Uncorrelated Exponential Relaxed-Clock Model

Open your text editor and create the lineage-specific branch-rate model file called `model_UCExp_TEFBD.Rev` in the **scripts** directory.

Enter the `Rev` code provided in this section in the new model file.

3.7 The General-Time Reversible Gamma-Rates Model of Sequence Evolution

Open your text editor and create the molecular substitution model file called `model_GTRG_TEFBD.Rev` in the **scripts** directory.

Enter the `Rev` code provided in this section in the new model file.

3.8 Modeling the Evolution of Binary Morphological Characters

Open your text editor and create the morphological character model file called `model_Morph_TEFBD.Rev` in the **scripts** directory.

Enter the `Rev` code provided in this section in the new model file.

Morphology has traditionally been assumed to evolve according to a generalized Jukes-Cantor matrix in which all characters have the same transition rates, and all characters have an equal probability of making forwards or backwards transitions (i.e., the same probability of going from 0 to 1 as 1 to 0). This model is called the Mk model. We will use a hyperprior on state frequencies to relax these two assumptions. Because we are working with binary data, we can use a discrete Beta distribution to describe the variation in stationary state frequencies across characters. The Beta distribution has two parameters, α and β which describe its shape. For simplicity, we will assume that $\alpha = \beta$. The below code draws a value for β from an exponential distribution and places a move on it.

```
beta_hp ~ dnExponential( 1.0 )

moves[mvi++] = mvScale(beta_hp, lambda=1, weight=1.0 )
moves[mvi++] = mvScale(beta_hp, lambda=0.1, weight=1.0 )
moves[mvi++] = mvScale(beta_hp, lambda=0.01, weight=1.0 )
```

Next, we'll create a vector containing four different site stationary state frequencies. This is similar to allowing gamma-distributed rate variation across sites. We will then use these stationary frequencies to generate a set of new Q-matrices which do not enforce the assumptions of the same transition rates at every site and equal forward and backwards transition rates.

```
beta_cats := fnDiscretizeBeta( beta_hp, beta_hp, 4)
for(i in 1:beta_cats.size())
{
    Q_morpho[i] := fnFreeBinary(v(1-beta_cats[i], beta_cats[i]))
}
```

As in the molecular data partition, we will allow gamma-dstributed rate heterogeneity among sites.

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

moves[mvi++] = mvScale(alpha_morpho, lambda=0.01, weight=1.0)
moves[mvi++] = mvScale(alpha_morpho, lambda=0.1, weight=1.0)
moves[mvi++] = mvScale(alpha_morpho, lambda=1, weight=1.0)
```

Each data partition has to have a clock rate. For simplicity, we will assume a strict clock rate drawn from an exponential distribution.

```
clock_morpho ~ dnExponential(1.0)

moves[mvi++] = mvScale(clock_morpho, lambda=0.01, weight=4.0)
moves[mvi++] = mvScale(clock_morpho, lambda=0.1, weight=4.0)
moves[mvi++] = mvScale(clock_morpho, lambda=1, weight=4.0)
```

As in our molecular data partition, we now combine our data and our model. There are some unique aspects to doing this for morphology.

You will notice that we have a variable called 'coding'. This variable allows us to condition on biases in the way the morphological data were collected. Morphology is often collected to maximize the amount of variation present in the dataset. This has traditionally been accomplished by collecting characters that exhibit parsimony informativity (i.e., those that can be used under the parsimony optimality criterion to discriminate among tree hypotheses) in the group of interest, or those that exhibit any variation in the group of interest. This means few datasets contain invariant characters. The lack of invariant characters can bias estimates of branch lengths towards unrealistically long lengths. Therefore, we specify a correction to account for the fact that invariant sites have no been observed. In the case of this dataset, parsimony non-informative variable characters, such as autapomorphies, have been collected. We will, therefore, use the 'variable' correction to account for this.

We use the flag 'siteMatrices=true' to indicate that we are providing multiple Q matrices generated as a function of our state frequency variation model.

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

3.9 Complete MCMC File

Return to the master `Rev` file you created in Section 3.4 called `mcmc_TEFBD.Rev` in the **`scripts`** directory.

Enter the `Rev` code provided in this section in this file.

3.9.1 Source Model Scripts

This step will load in the model scripts we have written in the text editor.

```
source("scripts/model_FBDP_TEFBD.Rev")

source("scripts/model_UCExp_TEFBD.Rev")

source("scripts/model_GTRG_TEFBD.Rev")

source("scripts/model_Morph_TEFBD.Rev")
```

3.9.2 Create Model Object

```
mymodel = model(sf)
```

3.9.3 Specify Monitors and Output Filenames

```
mni = 1
monitors[mni++] = mnModel(filename="output/bears.log", printgen=10)
monitors[mni++] = mnFile(filename="output/bears.trees", printgen=10, fbd_tree)
monitors[mni++] = mnScreen(printgen=10, age_extant, num_samp_anc, origin_time)
```

3.9.4 Set up the MCMC

```
mymcmc = mcmc(mymodel, monitors, moves)

mymcmc.run(generations=10000)
```

Save and close all files.

3.10 Run it

```
./rb
```

Execute the MCMC analysis:

```
source("scripts/mcmc_TEFBD.Rev")
```

```
   Processing file "scripts/mcmc_TEFBD.Rev"
   Successfully read one character matrix from file 'data/bears_cytb.nex'
   Successfully read one character matrix from file 'data/bears_morphology.nex'
   Processing file "scripts/model_FBDP_TEFBD.Rev"
   Processing of file "scripts/model_FBDP_TEFBD.Rev" completed
   Processing file "scripts/model_UCExp_TEFBD.Rev"
   Processing of file "scripts/model_UCExp_TEFBD.Rev" completed
   Processing file "scripts/model_GTRG_TEFBD.Rev"
   Processing of file "scripts/model_GTRG_TEFBD.Rev" completed
   Processing file "scripts/model_Morph_TEFBD.Rev"
   Processing of file "scripts/model_Morph_TEFBD.Rev" completed

   Running MCMC simulation
   This simulation runs 1 independent replicate.
   The simulator uses 163 different moves in a random move schedule with 267 moves per iteration

Iter      |     Posterior  |    Likelihood  |        Prior  |    age_extant  |  num_samp_anc  |   origin_time  |    elapsed  |       ETA  |
-------------------------------------------------------------------------------------------------------------------------------------------------
0         |      -8174.01  |       -8053.8  |      -120.209  |       34.8641  |            0   |       44.4332  |  00:00:00   |  --:--:--  |
10        |      -4654.95  |       -4611.2  |      -43.7495  |       4.32618  |            7   |       45.4494  |  00:00:01   |  --:--:--  |
20        |      -4294.05  |      -4266.91  |      -27.1443  |       4.58804  |            7   |       46.5636  |  00:00:01   |  00:08:19  |
30        |      -4267.35  |      -4233.41  |       -33.94  |        6.8467  |            6   |       45.9177  |  00:00:02   |  00:11:04  |
40        |      -4226.63  |      -4188.32  |      -38.3037  |       6.40484  |            8   |       44.3696  |  00:00:02   |  00:08:18  |
...
```

3.11 Summarize Your Results

3.11.1 Evaluate MCMC

3.11.2 Summarize Tree

Start up RevBayes at the command line. You should do this from within the **RB_TotalEvidenceDating_FBD_Tutorial** directory.

```
./rb
```

Read in the MCMC sample of trees from file.

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

for(i in 1:trace.size())
{
    trees[i] = fnPruneTree(trace.getTree(i), pruneTaxa=v(taxa[17],taxa[20]))
}

trace_pruned = treeTrace(trees)
mccTree(trace_pruned, "output/bears.mcc.tre" )
```

See Fig. 1

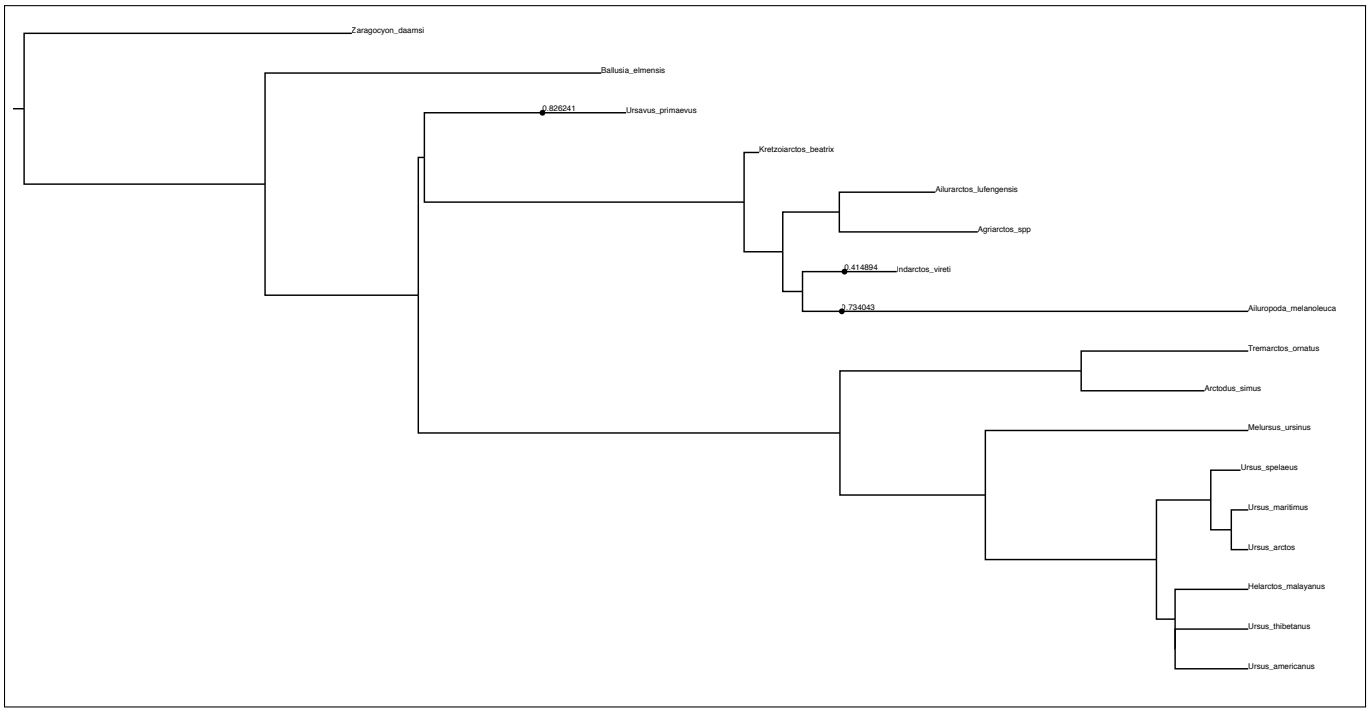

Figure 1: This is a place-holder figure.

References

Ronquist, F., S. Klopfstein, L. Vilhelmsen, S. Schulmeister, D. L. Murray, and A. P. Rasnitsyn. 2012. A total-evidence approach to dating with fossils, applied to the early radiation of the Hymenoptera. Systematic Biology 61:973–999.

Zuckerkandl, E. and L. Pauling. 1962. Molecular disease, evolution, and genetic heterogeneity. Pages 189–225 *in* Horizons in Biochemistry (M. Kasha and B. Pullman, eds.) Academic Press, New York.

Version dated: December 20, 2016