

Phylogenetic Inference using RevBayes

Historical biogeography

Michael Landis

1 Introduction

How did species come to live where they're found today? Observing where species live today, we can leverage phylogenetic and geological information to model their distribution as the outcome of biogeographic processes to answer this. These natural processes require some additional considerations, such as how ranges are inherited following speciation events, and how geological events might influence dispersal rates. Using **RevBayes** for biogeographic inference, standard Bayesian techniques are available, such as model selection, Metropolis-coupled MCMC, and stochastic mapping. This tutorial describes how to perform Bayesian inference of historical biogeography using **RevBayes**. Currently, this primarily covers range evolution models, where a species may occupy multiple discrete areas simultaneously. Tutorials to model individual specimens' geographical locations (sometimes called phylogeographic models) are under development.

Outline

2 Dispersal-Extinction-Cladogenesis model

2.1 Range characters

These models will rely on presence-absence data, where a species is observed or not observed across multiple discrete areas. For example, say there are three areas: A, B, and C. Say a species is present in areas A and C, then its range equals AC, which can also be encoded into the length-3 bit vector, 101. Bit vectors may also be transformed into (decimal) integers, e.g., the binary number 101 equals the decimal number 5. Typically, decimal representation is not used in discussion, but it is useful to keep in mind when considering the total number of possible ranges for a species.

2.2 Modeling anagenic range evolution

How might we model the dynamics of species range evolution? In this section, we'll cover the Dispersal-Extinction-Cladogenesis model proposed by [Ree et al. \(2005\)](#). To begin, we'll focus on anagenesis, evolution that occurs between speciation events. Since we have discrete characters we'll use the continuous-time Markov chain, which allows us to compute transition probability of a character changing from i to j in time t through matrix exponentiation

$$\mathbf{P}_{i,j}(t) = [\exp \{ \mathbf{Q}t \}]_{i,j},$$

where \mathbf{Q} is the instantaneous rate matrix defining the rates of change between all pairs of characters, and \mathbf{P} is the transition probability rate matrix. Remember, i and j represent different ranges, each of which is encoded as the set of areas occupied by the species. Exponentiation of the rate matrix is powerful because

it integrates over all possible scenarios of character transitions that could occur during t so long as the chain begins in state i and ends in state j .

We can then encode \mathbf{Q} to reflect the allowable classes of range evolution events with biologically meaningful parameters. We'll take a simple model of range expansion (e.g. $BC \rightarrow ABC$) and range contraction (e.g. $BC \rightarrow C$). (Range expansion may also be referred to as dispersal or area gain and range contraction as extirpation or area loss.) The rates in the transition matrix for three areas might appear as

$$\mathbf{Q} = \begin{array}{c|cccccccc} & \emptyset & A & B & C & AB & AC & BC & ABC \\ \hline \emptyset & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ A & e_A & - & 0 & 0 & d_{AB} & d_{AC} & 0 & 0 \\ B & e_B & 0 & - & 0 & d_{BA} & 0 & d_{BC} & 0 \\ C & e_C & 0 & 0 & - & 0 & d_{CA} & d_{CB} & 0 \\ AB & 0 & e_B & e_A & 0 & - & 0 & 0 & d_{AC} + d_{BC} \\ AC & 0 & e_C & 0 & e_A & 0 & - & 0 & d_{AB} + d_{CB} \\ BC & 0 & 0 & e_C & e_B & 0 & 0 & - & d_{BA} + d_{CA} \\ ABC & 0 & 0 & 0 & 0 & e_C & e_B & e_A & - \end{array},$$

where $e = (e_A, e_B, e_C)$ are the (local) extinction rates per area, and $d = (d_{AB}, d_{AC}, d_{BC}, d_{CB}, d_{CA}, d_{BA})$ are the dispersal rates between areas.

[?] For the three-area DEC rate matrix above, what is the rate of leaving state 101 (which is the coded state for range AB)? That is, what is the absolute value of the diagonal term in the rate matrix for $Q_{AB,AB}$, because the diagonal term is equal to minus the sum of all other entries in the row?

Note the rate of more than one event occurring simultaneously is zero, so a range must expand twice by one area in order to expand by two areas.

[?] What series of events might explain a transition from range ABC to range A? From range AB to range C?

Of course, this model can be specified for more than three areas.

[?] Imagine a DEC rate matrix with four areas, ABCD. What would be the dispersal rate for $Q_{BC,BCD}$? How many states does a DEC rate matrix with four areas have? What is the relationship between the number of areas and the number of states under the DEC model?

Let's consider what happens to the size of \mathbf{Q} when the number of areas, N , becomes large. For three areas, \mathbf{Q} is size 8×8 . For ten areas, \mathbf{Q} is size $2^{10} \times 2^{10} = 1024 \times 1024$, which approaches the largest size matrices that can be exponentiated in a practical amount of time. For twenty areas, \mathbf{Q} is size $2^{20} \times 2^{20} \approx 10^6 \times 10^6$, exponentiation is not viable.

2.3 Modeling cladogenic range evolution

Cladogenesis describes evolutionary change coincident with speciation. In some well-studied cases, daughter species are not expected to inherit their ancestral range identically. For each internal node in the reconstructed tree, one of two cladogenic events can occur: sympatry or allopatry. Say the range of a species is A the moment before speciation occurs at an internal phylogenetic node. Since the species range is size one, both daughter lineages necessarily inherit the ancestral species range (A). In DEC parlance,

this is called a narrow sympatry event.

Now suppose the ancestral range is ABC . Under subset sympatric cladogenesis, one lineage identically inherits the ancestral species range, ABC , while the other lineage inherits only a single area, i.e. only A or B or C . For widespread sympatric cladogenesis, both lineages inherit the ancestral range, ABC .

Under allopatric cladogenesis, the ancestral range is split evenly among daughter lineages, e.g. one lineage may inherit AB and the other inherits C . Both daughter lineages inherit the entire ancestral species range following widespread sympatric cladogenic events. For an excellent overview of described state transitions for cladogenic events, see [Matzke \(2013\)](#).

[?] What are the 6 possible states in the daughter lineages after cladogenesis given the state is AB before cladogenesis?

The probabilities of anagenic change along lineages must account for all combinations of starting states and ending states. For 3 areas, there are 8 states, and thus $8 \times 8 = 64$ probability terms for pairs of states. For cladogenic change, we need transition probabilities for all combinations of states before cladogenesis, after cladogenesis for the left lineage, and after cladogenesis for the right lineage. Like above, for three areas, there are 8 states, and $8 \times 8 \times 8 = 512$ cladogenic probability terms.

[?] For three areas, there are $3 + 12 + 6 = 21$ possible sympatry events and $6 + 6 = 12$ possible allopatry events. How many terms in the cladogenesis matrix are zero?

The DEC model ignores speciation events hidden by extinction or incomplete taxon sampling. The probability of cladogenesis and local extinction events would ideally be linked to a birth-death process, as it is in the GeoSSE model ([Goldberg et al. 2011](#)). Unfortunately, since the numerical method for SSE models scale poorly, and DEC models remain the only option when the geography has more than two or three areas. For more than ten areas, data augmentation may be used to infer ancestral ranges, as described in [Section 4](#).

The rest of this section will describe how to run a simple DEC analysis using RevBayes.

2.3.1 Specifying a simple DEC model

We'll use the primate dataset with 23 taxa. To keep the model simple, we'll discretize their ranges into areas: the New World, Africa, and Eurasia. For simplicity, we'll assume their phylogeny is time-calibrated, errorless, and fixed.

Create some `String` variables for file handling,

```
data_fn = "data/primates_bg_n3.tsv"
tree_fn = "data/primates.tree"
out_fn = "output/bg_1"
```

then read in our character data,

```
data = readTSVCharacterData(data_fn, type="NaturalNumbers")
```

and our tree

```
psi <- readTrees(tree_fn)[1]
```

Next, compute the number of states from the number of areas

```
n_areas = 3
n_states = 2^n_areas
```

Declare index variables for our move vectors for future use

```
mi = 0
```

Now, we'll begin to construct the rate matrix for anagenic events. First create a matrix, 8-by-8 in size, initialized with all zeroes

```
for (i in 1:n_states) {
  for (j in 1:n_states) {
    r[i][j] <- 0.0
  }
}
```

Now we need to populate the non-zero rate matrix elements, which are in terms of dispersal and extinction rates. We'll use one dispersal rate and one extinction rate for this tutorial, and explore more complex models in later sections. For later reference, this will be called the “common rate” model.

First, create a extinction rate parameter and assign it a scale move

```
r_e ~ dnExponential(10.0)
mv[+mi] = mvScale(r_e, weight=5)
```

Before assigning the rates to the rate matrix, we'll create a vector to hold the per-area extinction rates

```
for (i in 1:n_areas) {
  e[i] := r_e
}
```

Now create the dispersal rate and scale move

```
r_d ~ dnExponential(10.0)
mv[+mi] = mvScale(r_d, weight=5)
```

then assign the between-area dispersal rates as determined by r_d

```
for (i in 1:n_areas) {
  for (j in 1:n_areas) {
    if (i != j) {
      d[i][j] := r_d
    }
  }
}
```

Next, we'll populate the non-zero rate matrix elements. Rates are indexed by the natural number value of the range, e.g. the range spanning Eurasia and Africa is coded as 011, which is state 4.

First assign the extinction (range loss) rates

```
r[4][2] := e[2]      # 011 -> 001 : Extirpate in area 2
r[4][3] := e[3]      # 011 -> 010 : Extirpate in area 3
r[6][2] := e[1]      # 101 -> 001 : Extirpate in area 1
r[6][5] := e[3]      # 101 -> 100 : Extirpate in area 3
r[7][3] := e[1]      # 110 -> 010 : Extirpate in area 1
```

```

r[7][5] := e[2]           # 110 -> 100 : Extirpate in area 2
r[8][4] := e[1]           # 111 -> 011 : Extirpate in area 1
r[8][6] := e[2]           # 111 -> 101 : Extirpate in area 2
r[8][7] := e[3]           # 111 -> 110 : Extirpate in area 3

```

then the dispersal (range gain) rates

```

r[2][4] := d[3][2]        # 001 -> 011 : Disperse from area 3 to 2
r[2][6] := d[3][1]        # 001 -> 101 : Disperse from area 3 to 1
r[3][4] := d[2][3]        # 010 -> 011 : Disperse from area 2 to 3
r[3][7] := d[2][1]        # 010 -> 110 : Disperse from area 2 to 1
r[5][6] := d[1][3]        # 100 -> 101 : Disperse from area 1 to 3
r[5][7] := d[1][2]        # 100 -> 110 : Disperse from area 1 to 2
r[4][8] := d[2][1] + d[3][1] # 011 -> 111 : Disperse from area 2 to 1 and from 3 to 1
r[6][8] := d[1][2] + d[3][2] # 101 -> 111 : Disperse from area 1 to 2 and from 3 to 2
r[7][8] := d[1][3] + d[2][3] # 110 -> 111 : Disperse from area 1 to 3 and from 2 to 3

```

Show the value of `r` and compare it to the matrix in Section 2.2.

Of course we did not need to declare `d` and `e` to assign `r`, but we'll see these intermediate variables act as a template expose the structure of `r` for modification.

So far, we only have the desired parameterization of the rate matrix, but we still haven't created a rate matrix function. Converting the vector-of-vectors, `r`, into a simplex allows us to use existing rate matrix functions.

First, we'll convert `r` into a one-dimensional vector, skipping the diagonal elements.

```

k = 1
for (i in 1:n_states) {
  for (j in 1:n_states) {
    if (i != j) {
      er_nat[k] := r[i][j]
      k += 1
    }
  }
}

```

Finally, normalize `er_nat` using a simplex, then pass the resulting exchangeability rates as arguments into the rate matrix function, `q`.

```
er := simplex(er_nat)
bf <- simplex(rep(1,n_states))
q := fnFreeK(er, bf)
```

This yields the desired three-area DEC rate matrix modeling anagenic character change. In contrast, cladogenic event probabilities are given by a transition probability matrix and do not require a rate matrix.

First, we will create a vector of prior weights on cladogenesis events. Here, we assign a flat prior to all cladogenic events

```
widespread_sympatry_wt <- 1.0
subset_sympatry_wt    <- 1.0
allopatry_wt          <- 1.0
clado_prior           <- [ widespread_sympatry_wt, subset_sympatry_wt, allopatry_wt ]
```

then create the distribution over cladogenic event types and add its MCMC move

```
clado_type ~ dnDirichlet(clado_prior)
mv[+mi]    = mvSimplexElementScale(clado_type, alpha=10, weight=5)
```

To give the simplex elements descriptive names when monitored, assign the values to deterministic nodes

```
widespread_sympatry := clado_type[1]
subset_sympatry     := clado_type[2]
allopatry           := clado_type[3]
```

Then create the cladogenic transition probability matrix, which assigns probabilities to cladogenic event classes according to `clado_type_prob`

```
clado_prob := fnCladoProbs(clado_type, n_areas, 2)
```

Add a parameter for a biogeographical clock, which scales the overall rate of range evolution. As a prior, an exponential distribution with rate 10 generates one dispersal or extinction event per 10 million years.

```
clock_bg ~ dnExponential(10)
mv[+mi] = mvScale(clock_bg, weight=5)
```

Finally, all our model components are encapsulated in the `dnPhyloCTMCClado` distribution, which is similar to `dnPhyloCTMC` except specialized to integrate over cladogenic events. Although this dataset has three areas, it is recognized single character with states valued from 1 to 2^3 , hence `nSites=1`.

```
m ~ dnPhyloCTMCClado( tree=psi, Q=q, cladoProbs=clado_prob, branchRates=clock_bg, nSites
  =1, type="NaturalNumbers" )
```

The remaining tasks should be familiar by now, so we can proceed briskly. Attach the observed ranges to the model.

```
m.clamp(data)
```

Compose the model.

```
mdl = model(m)
```

Add the monitors.

```
mn[1] = mnScreen(clock_bg, d[1][2], d[1][3], d[2][1], d[2][3], d[3][1], d[3][2], e[1], e
  [2], e[3], widespread_sympatry, subset_sympatry, allopatry, printgen=1000)
mn[2] = mnFile(clock_bg, d[1][2], d[1][3], d[2][1], d[2][3], d[3][1], d[3][2], e[1], e
  [2], e[3], widespread_sympatry, subset_sympatry, allopatry, file=out_fn+".params.txt
  ")
```

Create the MCMC object, and run the chain after burn-in.

```
ch = mcmc(mv,mn,mdl)
ch.burnin(1000, 10)
ch.run(10000)
```

2.3.2 Per-area rates

Biologically, local extinction events probably do not occur at equal rates across all areas, as done above. Ecological factors, geographical distances, etc. might cause these parameters to be weakly correlated or completely uncorrelated. Dispersal rates, also, might not be the same between pairs of areas, or even symmetric depending on the direction of dispersal. Rather than constraining all events of a type to share a common rate, instead you might give each area it's own extinction parameter


```
for (i in 1:3) {
  e[i] ~ dnExponential(10.0)
  mv[+mi] = mvScale(e[i], weight=5)
}
```

or give each ordered pair of areas a it's own dispersal rate

```
for (i in 1:3) {
  for (j in 1:3) {
    if (i != j) {
      d[i][j] ~ dnExponential(10.0)
      mv[+mi] = mvScale(d[i][j], weight=5)
    }
  }
}
```

Note that you don't need the global dispersal rate **r_d** and extinction rate **r_e** anymore and you can remove the variable from your analysis. **RevBayes** might give you an error message if you have left them in.

This parameterization is identical to matrix introduced in Eq XXX. In Section XXX, we'll extend this idea further to parameterize features of range evolution, and incorporate paleogeological information.

2.4 Exercises

Exercises are independent of each other, except for Exercises 3a and 3b.

- 1) Widespread sympatric speciation is thought to be evolutionarily rare. Set the Dirichlet prior on cladogenic event types to heavily disfavor these events. Using Tracer, describe how changing the cladogenesis prior affects the extinction rate when compared with the “common rate” model.
- 2) Modifying the **RevBayes** script, re-parameterize the rate matrix so ranges may not grow beyond two areas in size.
- 3a) Saving your commands to a file, create a script to produce the “per-area rate” model.
- 3b) Determine if the data support the “common rate” model over the “per-area rate” model. Use the stepping stone method to compute marginal likelihoods, which will let you compute Bayes factors for model selection.

2.5 (Advanced) Joint inference of phylogeny and historical biogeography

This is a slightly more advanced question that you may want to skip if you will always use known, fixed trees in your analyses. Using what you learned in this tutorial and the CTMC tutorial, perform a joint analysis of molecular and biogeographic evolution for primates.

Start by loading the sequence data matrix specified in **data/primates_cytb.nex**.

```
seqData <- readDiscreteCharacterData("data/primates_cytb.nex")
```

We need to get some useful variables from the data so that we will be able to specify the tree prior below. These variables are the number of tips, the number of nodes and the names of the species which we all can query from the continuous character data object.

```
numTips = seqData.n taxa()
names = seqData.names()
numNodes = numTips * 2 - 1
```

Instead of having a fixed tree as in the previous, we should now define a *random* tree. We use a birth death prior with prior distributions on the **diversification** rate and **turnover** rate. The **diversification** rate corresponds to the rate of growth of the tree and the **turnover** rate corresponds to the rate how quickly a species is replaced by a new one. This parametrization has the advantage that we can use meaningful prior distributions from the fossil record.

```
diversification ~ dnLognormal(0,1)
turnover ~ dnGamma(4,4)
```

Then, we compute deterministically the **speciation** and **extinction** rate from the **diversification** rate and **turnover** rate.

```
speciation := diversification + turnover
extinction := turnover
```

Instead, you could also use directly the speciation and extinction rates, e.g., endowed with some diffuse exponential prior.

```
# rescaling moves on speciation and extinction rates
moves[++mi] = mvScale(diversification, lambda=1, tune=true, weight=3.0)
moves[++mi] = mvScale(turnover, lambda=1, tune=true, weight=3.0)
```

The phylogeny that we used are obviously not a complete sample of all the species and you should take the incomplete sampling into account. We will simply use an empirical estimate of the fraction of species which we included in this study. For more information about incomplete taxon sampling see ? and ?.

```
sampling_fraction <- 23 / 270 # 23 out of the ~ 270 primate species
```

Now we are able to specify of tree variable **psi** which is drawn from a constant rate birth-death process. We will condition the age of the tree to be 75 million years old which is approximately the crown age of primates, although this estimate is still debated. We only condition here on the crown age for simplicity because we do not use any other fossil calibration.

```
psi ~ dnBDP(lambda=speciation, mu=extinction, rho=sampling_fraction, rootAge=75, nTaxa=numTips,
            names=names)
```

Note that, here, we do not have included any fossil information: we are merely doing *relative* dating.

The first moves on the tree which we specify are moves that change the node ages. The first move randomly picks a subtree and rescales it, and the second move randomly pick a node and uniformly proposes a new node age between its parent age and oldest child's age.

```
moves[++mi] = mvSubtreeScale(psi, weight=5.0)
moves[++mi] = mvNodeTimeSlideUniform(psi, weight=10.0)
```

We also need moves on the tree topology to estimate the phylogeny. The two moves which you use are the nearest-neighbor interchange (NNI) and the fixed-nodeheight-prune-and-regraft (FNPR) (?).

```
moves[++mi] = mvNNI(psi, weight=5.0)
moves[++mi] = mvFNPR(psi, weight=5.0)
```

In the next step we set up the substitution model. First, we create a substitution model, just like what you probably did in previous tutorial (e.g., RB_CTMC_Tutorial). In a first step, we will use a GTR+Gamma model. We can use a flat Dirichlet prior density on the exchangeability rates **er_mol** and the the base frequencies **pi_mol**.

```
er_mol_prior <- v(1,1,1,1,1,1)
er_mol ~ dnDirichlet(er_mol_prior)
pi_mol_prior <- v(1,1,1,1)
pi_mol ~ dnDirichlet(pi_mol_prior)
```

Now add the simplex scale move one each the exchangeability rates **er_mol** and the stationary frequencies **pi_mol** to the moves vector:

```
moves[++mi] = mvSimplexElementScale(er_mol)
moves[++mi] = mvSimplexElementScale(pi_mol)
```

We can finish setting up this part of the model by creating a deterministic node for the GTR instantaneous-rate matrix **Q_mol**. The **fnGTR()** function takes a set of exchangeability rates and a set of base frequencies to compute the instantaneous-rate matrix used when calculating the likelihood of our model.

```
Q_mol := fnGTR(er_mol,pi_mol)
```

The next part of the substitution process is the rate variation among sites. We will model this using the commonly applied 4 discrete gamma categories which only have a single parameter **alpha**. Let us specify the rate of **alpha** to 0.05 (thus the mean will be 20.0).

```
alpha_prior <- 0.05
```

Then create a stochastic node called **alpha** with an exponential prior:

```
alpha ~ dnExponential(alpha_prior)
```

Initialize the **gamma_rates** deterministic node vector using the **fnDiscretizeGamma()** function with 4 bins:

```
gamma_rates := fnDiscretizeGamma( alpha, alpha, 4 )
```

The random variable that controls the rate variation is the stochastic node **alpha**. We will apply a simple scale move to this parameter.

```
moves[++mi] = mvScale(alpha, weight=2.0)
```

This finishes the substitution process part of the model.

Then next part of the model is the clock model. Here we need a clock model because we work on a time tree. We use our exponentiated uniform distribution to specify a flat prior distribution which is scale invariant.

```
logClockRate ~ dnUniform(-5,5)
clockRate := 10^logClockRate
```

We will apply a sliding window move to the **logClockRate**.

```
moves[++mi] = mvSlide(logClockRate, delta=0.1, tune=true, weight=2.0)
```

Introduce a common prior for the molecular and biogeographical clocks.

```
clock_scale_bg ~ dnGamma(2.0,2.0)
clock_bg       := clockRate * clock_scale_bg
```

Remember that you need to call the **PhyloCTMC** constructor to include the new site-rate parameter:

```
seq ~ dnPhyloCTMC(tree=psi, Q=Q_mol, siteRates=gamma_rates, branchRates=clockRate,  
  type="DNA")
```

Finally we need to attach the molecular sequence data to our model.

```
seq.clamp(seqData)
```

We are essentially done now. We only need to add a new monitor for the tree so that we can monitor and build the maximum a posteriori tree later.

```
monitors[3] = mnFile(filename="output/biogeography_DEC_joint.trees", printgen=100, separator =  
  TAB, psi)
```

3 Epoch models and ancestral state reconstruction

(Under construction.)

4 Large numbers of areas

4.1 Data augmentation

For small rate matrices, transition probabilities of beginning in state i and ending in state j equal the matrix exponential of the underlying rate matrix, scaled by the elapsed time of the process. This integrates over all unobserved transition events during the time interval t . Unfortunately, computing the matrix exponential scales poorly as the state space increases, i.e., $O(n^4)$ for n states.

Alternatively, the probability of beginning in state i and ending in state j can be computed easily when the explicit series of event types and times are known. While we will never know the exact history of events, we can use stochastic mapping in conjunction with Markov chain Monte Carlo (MCMC) to repeatedly sample range evolution histories that are consistent with the ranges observed in the study taxa at the tips of the phylogeny. This technique is called data augmentation. It was first applied in phylogenetics to tertiary structure-dependent evolution of protein-coding nucleotide sequences (Robinson et al. 2003), and later extended to range evolution in Landis et al. (2013).

Using data augmentation, we will approximate $\text{Prob}(\mathbf{X}_{aug}, \theta \mid \mathbf{X}_{obs}, T, M)$, where \mathbf{X}_{obs} is the range data observed at the tips, \mathbf{X}_{aug} is the distribution of ancestral range reconstructions over the phylogeny, T , where \mathbf{X}_{aug} is inferred jointly with the parameters, θ , assuming the range evolution model, M , that describes \mathbf{Q} above. From a Bayesian perspective, \mathbf{X}_{aug} is effectively treated as part of the parameter space in the posterior distribution. Just as MCMC evaluates the posterior distribution for sampled parameter values, the same is done for sampled character histories. Thus, ancestral range reconstructions are a by-product of data augmentation, which are often of primary interest to biogeographers.

You may wonder why matrix exponentiation works fine for molecular substitution models and large multiple sequence alignments. It is the non-independence of character evolution that induces large state spaces, along with cladogenic processes that affect the entire set of characters (e.g. non-identical range inheritance following speciation). Molecular substitution models typically assume each character in an alignment evolves independently, which may be justified by the ability of recombination to degrade linkage disequilibrium over geological time scales. Conveniently, this keeps \mathbf{Q} small even for datasets with many sites.

4.2 Large rate matrices and distance-dependent dispersal

A rate matrix may be represented compactly as the rate function

$$q_{\mathbf{y}, \mathbf{z}}^{(a)} = \begin{cases} \lambda_0 & \text{if } z_a = 0 \\ \lambda_1 & \text{if } z_a = 1 \\ 0 & \mathbf{y} \text{ and } \mathbf{z} \text{ differ in more than one area} \end{cases}.$$

where \mathbf{y} and \mathbf{z} are the “from” and “to” ranges and a is the area that changes, λ_0 is the per-area rate of loss, and λ_1 is the per-area rate of gain. For example, $q_{011,111}^{(1)}$ is the rate of range expansion for $011 \rightarrow 111$ to gain area 1, which equals λ_1 . Note the rate of more than one event occurring simultaneously is zero, so

a range must expand twice by one area in order to expand by two areas. This model is analogous to the Jukes-Cantor model for three independent characters with binary states, except the all-zero “null range” is forbidden.

Extending this idea, we may reasonably expect that a range expansion event into an area depends on which nearby areas are currently inhabited, which imposes non-independence between characters. The transition rate might then appear as

$$q_{\mathbf{y},\mathbf{z}}^{(a)} = \begin{cases} \lambda_0 & \text{if } z_a = 0 \\ \lambda_1 \eta(\mathbf{y}, \mathbf{z}, a, \beta) & \text{if } z_a = 1 \\ 0 & \mathbf{y} = 00\dots 0 \\ 0 & \mathbf{y} \text{ and } \mathbf{z} \text{ differ in more than one area} \end{cases}.$$

For this tutorial, you can take $\eta(\cdot)$ to adjust the rate of range expansion into area a by considering how close it is to the current range, \mathbf{y} relative to the closeness of all other areas unoccupied by the taxon. The β parameter rescales the importance of geographic distance between two areas by a power law. Importantly, $\eta(\cdot) = 1$ when $\beta = 0$, meaning geographic distance between areas is irrelevant. Moreover, when $\beta > 0$, $\eta(\cdot) < 1$ when area a is relatively distant and $\eta(\cdot) > 1$ when area a is relatively close. See [Landis et al. \(2013\)](#) for a full description of the model.

? Write down a rate function where the per-area rates of gain and loss depend on the number of currently occupied areas.

4.3 Specifying a data augmented DEC model

Start to create helper variables for file handling,

```
area_fn = "data/earth25.still.atlas.txt"
data_fn = "data/primates_bg_n25.nex"
tree_fn = "data/primates.tree"
out_fn = "output/bg_3"
```

read in our tree,

```
psi <- readTrees( tree_fn )[1]
```

populate our range observations,

```
data = readDiscreteCharacterData( data_fn )
```

and read in our geographical information (for details, see Section XX).

```
atlas = readAtlas( area_fn)
n_areas = atlas.nAreas()
```

Lastly, create index variables to populate our move and monitor vectors,

```
mi = 0
```

and assign the number of generations to run the MCMC analysis

```
ngen = 500000
```

To later interpret ancestral state monitors phylogenetically, save a copy of **tree** annotated with internal node indexes.

```
write(psi,filename=tree_fn+".index.tre")
```

Proceeding with the model configuration, we'll first create our rate matrix that determines the rate of per-area gain and loss given the current geographical layout of the range. In **RevBayes**, data-augmented CTMC analyses require **RateMap** functions to determine event rates, which differ from the familiar **RateMatrix** functions that include **fnJC** and **fnGTR** as members. In their simplest form, **RateMap** functions generate the rates of change over the full set of characters, where each character evolves according to a provided **RateMatrix** function. Additionally, a **RateMap** accepts a rate modifier function that induces some correlation structure to character change evolution. In this section, we'll be creating a biogeographic **RateMap** function for the dispersal-extinction process given in Section 4.2.

First, create a biogeographical clock to scale the rate of range evolution.

```
clock_bg ~ dnExponential(10)
moves[++mi] = mvScale(clock_bg, lambda=0.5, weight=5.0)
```

Next, instantiate a simplex of gain and loss rates, distributed by a flat Dirichlet prior,

```
glr ~ dnDirichlet([1,1])
moves[++mi] = mvSimplexElementScale(glr, alpha=30.0, weight=5.0)
```

and use deterministic nodes to assign the rates nicknames.


```
r_gain := glr[1]
r_loss := glr[2]
```

Insert the simplex into the rate matrix q_{area} , which gives the average rate of area gain and loss per area.

```
q_area := fnFreeBinary(glr)
```

Next, we will create `dp` to represent the β parameter, which determines the importance of geographical distance to dispersal. Remember that values of β far from zero means distance is important. So, if we assign a prior that pulls β towards zero, then posterior values of β far from zero indicate the range data are informative of the importance of distance to dispersal. We'll use an exponential distribution with mean 1.0 as a prior for `dp`.

```
dp ~ dnExponential(10.0)
moves[++mi] = mvScale(x=dp, lambda=0.5, tune=true, weight=5.0)
```

We will also create a deterministic node to modify the rate of dispersal between areas by evaluating `dp` and `atlas`. This node is determined by the function `fnBiogeoGRM`, where GRM stands for “geographical rate modifier”, and plays the role of the $\eta(\cdot)$ rate-modifier function mentioned earlier. We will tell the `fnBiogeoGRM` function to modify dispersal rates based on distances and whether or not the area exists during an epoch.

```
grm := fnBiogeoGRM(atlas=atlas, distancePower=dp, useDistance=true)
```

Now we need a deterministic node to represent the rate matrix, \mathbf{Q} . To determine the value of this node, we'll use the function `fnBiogeoDE` to assign our model parameters to transition rates as described in the introduction. As input, we'll pass our gain and loss rates, `q_area`, our geographical rate modifier, `grm`, and the biogeographical clock `clock_bg`. In addition, we'll inform the function of the number of areas in our analysis and whether we will allow species to be absent in all areas (i.e. have the null range).

```
q_range := fnBiogeoDE(gainLossRates=q_area, branchRates=clock_bg, geoRateMod=grm,
  numAreas=n_areas, forbidExtinction=true)
```

As with the simple DEC model, we assign a flat Dirichlet prior over cladogenic event type probabilities

```
clado_prob ~ dnDirichlet( [1, 1, 1] )
widespread_sympatry := clado_prob[1]
```

```
subset_sympatry := clado_prob[2]
allopatry       := clado_prob[3]
moves[++mi] = mvSimplexElementScale(clado_prob, alpha=20.0, weight=5.0)
```

Finally, the data evolve according to a phylogenetic CTMC process. Here, we declare a stochastic node using `dnPhyloDACTMC` where DA indicates the distribution requires data augmentation to compute the likelihood rather than Felsenstein's pruning algorithm. To create the distribution, we must pass it our `tree` and `q_range` objects, but additionally inform the distribution that it will be using a biogeographic model, that it will introduce the simple cladogenic range evolution events described in [Ree and Smith \(2008\)](#) (`useCladogenesis=true`), and that it will assign zero probability to a transition away from the null range state.

```
m ~ dnPhyloDACTMC(tree=psi, Q=q_range, cladoProbs=clado_prob, type="Biogeo",
  forbidExtinction=true, useCladogenesis=true)
```

So we may evaluate the graphical model's likelihood, we tell the CTMC to observe the `data` object, which then primes the model with data-augmented character histories.

Now `m` has a defined likelihood value.

```
m.clamp(data)
m.lnProbability()
-156.0288
```

To integrate over the space of possible range histories, we still need to add moves to propose new data augmented histories. The major challenge to sampling character histories is ensuring the character histories are consistent with the observations at the tips of the tree. The proposals in this tutorial use [Nielsen \(2002\)](#)'s rejection sampling algorithm, with some modifications to account for cladogenic events and epoch-based rate maps.

The basic idea is simple. Each time a character history proposal is called, it selects a node at random from the tree. Branch history proposals propose a new character history to a single randomly-chosen branch. Node history proposals propose a new character history for the three branches connecting to a randomly chosen node, in addition to the cladogenic state of the node itself.

For each proposal, each character's history is resampled with probability `lambda` – i.e. `lambda=0.01` resamples 1% of characters, while `lambda=1.` resamples all characters. Once the new character history is proposed, the likelihood of the model is evaluated and the MCMC accepts or rejects the new state according to e.g. the Metropolis-Hastings algorithm.

Cladogenic events require special considerations during sampling, so we indicate `type="Biogeo"`. Currently, only rejection-sampling is available for cladogenic histories, hence `proposal="rejection"`. Finally, we apply high `weight` values to the proposals since the larger the state space is, the more character history proposals needed to effectively integrate over the space of sample paths.

Let's create the character history moves as follows: conservative character history updates for paths and nodes, with `lambda=0.05`

```
moves[++mi] = mvCharacterHistory(ctmc=m, qmap=q_range, tree=psi, lambda=0.05,
                                type="Biogeo", graph="node", proposal="rejection", weight=100.0)
moves[++mi] = mvCharacterHistory(ctmc=m, qmap=q_range, tree=psi, lambda=0.05,
                                type="Biogeo", graph="branch", proposal="rejection", weight=100.0)
```

and the same proposals for more radical character history updates, with `lambda=1.0`

```
moves[++mi] = mvCharacterHistory(ctmc=m, qmap=q_range, tree=psi, lambda=1.0,
                                type="Biogeo", graph="node", proposal="rejection", weight=40.0)
moves[++mi] = mvCharacterHistory(ctmc=m, qmap=q_range, tree=psi, lambda=1.0,
                                type="Biogeo", graph="branch", proposal="rejection", weight=40.0)
```

Next, create the model object,

```
my_model = model(m)
```

and monitors for our simple parameters.

```
monitors[1] = mnScreen(clock_bg, r_gain, r_loss, dp, subset_sympatry, allopatry,
                      widespread_sympatry, printgen=100)
monitors[2] = mnFile(clock_bg, r_gain, r_loss, dp, subset_sympatry, allopatry,
                    widespread_sympatry, filename=out_fn+".params.txt", printgen=10)
```

Like any parameter, we can sample the augmented range histories from the MCMC to approximate the posterior distribution of range histories. This is statistically equivalent to generating ancestral state reconstructions from a posterior distribution via stochastic mapping. We will extract these reconstructions using special monitors designed for the `dnPhyloDACTMC` distribution.

Next, we will create `mnCharHistoryNewick` monitors to record the sampled character history states for each node in the tree. This monitor has two `style` options: `counts` reports the number of gains and losses per branch in a tab-delimited Tracer-readable format; `events` reports richer information of what happens along a branch, anagenically and cladogenically, using an extended Newick format. How to read these file formats will be discussed in more detail in Section 4.4.

```
monitors[3] = mnCharHistoryNewick(filename=out_fn+".events.txt", ctmc=m, tree=psi,
                                printgen=100, style="events")
monitors[4] = mnCharHistoryNewick(filename=out_fn+".counts.txt", ctmc=m, tree=psi,
                                printgen=100, style="counts")
```

As our last monitor, `mnCharHistoryNhx` records character history values throughout the MCMC analysis, then stores some simple posterior summary statistics as a Nexus file. These summary statistics could be computed from the previously mentioned monitor output files, but `mnCharHistoryNhx` provides a simple way to produce Phylowood-compatible files. We will also discuss this file's format in more detail later in the tutorial.

```
monitors[5] = mnCharHistoryNhx(filename=out_fn+".phw.txt", ctmc=m, tree=psi, atlas=atlas
, samplegen=100, maxgen=ngen, burnin=0.5)
```

Finally, create the MCMC object

```
my_mcmc = mcmc(my_model, monitors, moves)
```

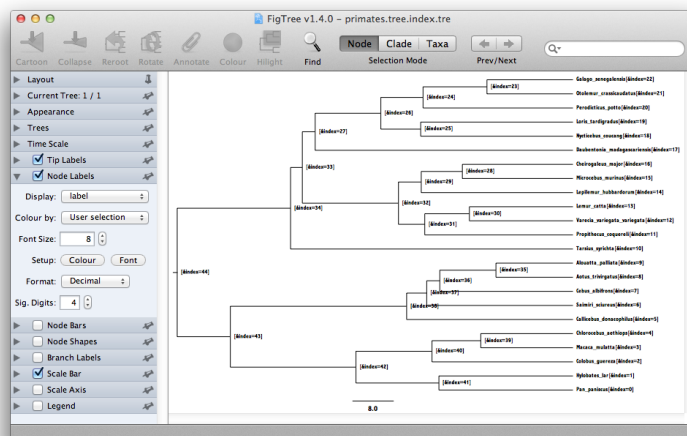
and run

```
my_mcmc.run(generations=ngen)
```

The posterior surface over character histories is highly multimodal, so this analysis takes a long time to mix. If you're following this tutorial in a lab setting, you should proceed using the pre-analysed output files provided in `./completed_output`.

4.4 Analysis output

You will need a node-indexed phylogeny to interpret ancestral state reconstructions. First, open FigTree and load the Newick tree stored in `data/primates.tree.index.tre`. Expand “Node Labels” and change “Display” to the custom node label. All sampled characters have a node index, which corresponds to the branch ancestral to the indexed node.



We'll focus some attention on node 42, the most recent common ancestor of chimps and macaques, designated as the Old World most recent common ancestor (MRCA).

4.4.1 Biogeographic event counts from `mnCharHistoryNewick`

Recording stochastic mappings in a Tracer-compatible format requires some summarization. This monitor generates a tab-delimited file where the number of events of each type for each branch is recorded.

Open `./output/bg_3.counts.txt` in a text editor.

Iter	Posterior		Likelihood	Prior	t_s0		t_s1		t_c0		t_c1		t_c2		t_c3		b0_s0
	b0_s1	b0_c	...														
0	-6518.54	-6519.25	0.706823		977	839	22	0	0	0	38	28	0	...			
10	-583.147	-585.333	2.186240		66	37	3	7	9	3	2	1	2	...			
20	-296.540	-297.340	0.799755		20	25	15	1	3	3	0	1	0	...			
30	-276.284	-277.257	0.972918		17	24	14	0	4	4	0	1	0	...			
40	-266.569	-266.948	0.379624		18	23	15	1	3	3	0	1	3	...			
...																	

For example, `b0_s1` gives the number of areas that are gained for the branch leading to the node indexed 0. Referencing FigTree, we see this corresponds to chimpanzees. `b0_c` gives the cladogenic event type that gives rise to the chimp lineage, where narrow sympatry, widespread sympatry, subset sympatry, and allopatry are recorded as 0, 1, 2, and 3, respectively. The columns `t_s0` and `t_s1` give the sum of events over all branches. `t_c0`, `t_c1`, `t_c2`, and `t_c3` give the total number of narrow sympatric, widespread sympatric, subset sympatric, and allopatric cladogenic events over the entire tree.

4.5 Biogeographic event histories from `mnCharHistoryNewick`

For more detailed data exploration, this analysis also provides annotated Newick strings with the complete character mappings for the tree. (This file format is a bit unwieldy and under revision.)

→ Open `./output/bg_3.events.txt` in a text editor.

Each iteration records the data-augmented character history (stochastic mapping) using metadata labels, which, for an internal node, looks like

```
[&index=42;nd=0000000000000101000000000;pa=0000000000000100000000000;ch0
=0000000000000101000000000;ch1=000000000000010000000000;cs=s;bn=41;ev={{t:0.138033,
a:39.4458,s:1,i:15}}]
```

Having consulted FigTree earlier, we know node 42 is the Old World MRCA. The branch began with the range `pa=0000000000000100000000000` and terminated in the state `nd=0000000000000101000000000`. Since this node is not a tip node, it represents a speciation event, so the daughter ranges are also given, `ch0=0000000000000101000000000` and `ch1=000000000000010000000000`. The cladogenic state for this speciation event was subset sympatric, `cs=s`, rather than sympatric (wide or narrow; `w` or `n`) or allopatric (`a`).

Although we have stochastic mappings under the posterior distribution, the remaining challenge is to summarize them into something useful. The Python script `bg_parse.py` is provided to manipulate this data format. Below are a few examples of interesting features of the posterior.

```
> cd RevBayes_scripts
> python

...

>>> from bg_parse import *
>>> dd=get_events(fn="../output/bg_3.events.txt")
```

```
>>> dd[42].keys()
['ch1', 'iteration', 'bn', 'nd', 'ch0', 'prior', 'posterior', 'pa', 'cs', 'ev', 'likelihood']
>>> dd[42]['cs'][-5:]
['widespread_sympatry', 'widespread_sympatry', 'widespread_sympatry', 'widespread_sympatry', 'widespread_sympatry']
```

```
>>> get_best(dd[42],n=1,p='posterior')
{'prior': [-0.524716], 'iteration': [46700], 'bn': [41], 'nd': [[0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]], 'ch0': [[0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]], 'ch1': [[0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]], 'posterior': [-95.7676],
'pa': [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]],
'cs': ['widespread sympatry'], 'ev': [[]], 'likelihood': [-95.2429]}
```

More data-exploration functions are found in `bg_parse.py`.

4.6 Phylowood animation file (`./output/bg_3.nhx.txt`)

Because this data is very high-dimensional, we'll use an external data exploration tool to look at range evolution. This file summarizes the MCMC output from a RevBayes biogeographical analysis as a Nexus-formatted file, which is used by Phylowood to generate interactive animations to explore biogeographic reconstructions.

- Open <http://mlandis.github.io/phylowood>.
- Drag and drop `./output/bg_3.phw.txt` into the text field.
- Click the Play button to view the animation.

Notice the primate MRCA range is very widespread with six areas. This is

There are three control panels to help you filter data: the media panel, the map panel, and the phylogeny panel. The media buttons correspond to Beginning, Slow/Rewind, Play, Stop, Fast Forward, Ending (from left to right). The animation will play the timeframe corresponding to the slider.

- Drag the slider to the right (the present).
- Pan and zoom around the map.

Marker colors correspond to the phylogenetic lineages in the phylogeny panel. Markers are split into slices and (loosely) sorted phylogenetically, so nearby slices are generally closely related. At divergence events, a marker's radius is proportional to the marginal posterior probability the node was present in the area at that time. Between divergence events, marker's radius is simply an interpolation of the values at the two endpoints. Some information about geological constraints and cladogenic events is lost.

- Mouseover an area to learn which lineage it belongs to and its presence probability.

Since it's difficult to see how specific clades evolve with so many taxa, Phylowood offers two ways to filter taxa from the animation. We call the set of a lineage, all its ancestral lineages towards the root, and all descendant lineages a phylogenetic heritage. The root's heritage is the entire clade. A leaf node's heritage is a path from the tip to the root.

- Mouseover a lineage to temporarily highlight the lineage's heritage. Remove the mouseover to remove the highlight effect.

The highlight effect is temporary and quickly allows you to single out lineages of interest during animation. Phylowood also offers a masking effect that persists until an unmask command is issued.

- Double-click the white root branch to mask the root node’s heritage (all lineages). Single click a lineage to unmask that lineage’s heritage.

Now that the masking effects are in place, you’re free to interact with other map components. In addition, the area of marker sizes is only distributed among unmasked lineages.

Visit <https://github.com/mlandis/phylowood/wiki> to learn more about Phylowood.

4.7 Exercises

- (Under construction.)

References

- Goldberg, E. E., L. T. Lancaster, and R. H. Ree. 2011. Phylogenetic inference of reciprocal effects between geographic range evolution and diversification. *Systematic Biology* 60:451–465.
- Landis, M. J., N. J. Matzke, B. R. Moore, and J. P. Huelsenbeck. 2013. Bayesian analysis of biogeography when the number of areas is large. *Systematic biology* 62:789–804.
- Matzke, N. J. 2013. Probabilistic historical biogeography: new models for founder-event speciation, imperfect detection, and fossils allow improved accuracy and model-testing. *Frontiers of Biogeography* 5.
- Nielsen, R. 2002. Mapping mutations on phylogenies. *Systematic Biology* 51:729–739.
- Ree, R. H., B. R. Moore, C. O. Webb, and M. J. Donoghue. 2005. A likelihood framework for inferring the evolution of geographic range on phylogenetic trees. *Evolution* 59:2299–2311.
- Ree, R. H. and S. A. Smith. 2008. Maximum likelihood inference of geographic range evolution by dispersal, local extinction, and cladogenesis. *Systematic Biology* 57:4–14.
- Robinson, D. M., D. T. Jones, H. Kishino, N. Goldman, and J. L. Thorne. 2003. Protein evolution with dependence among codons due to tertiary structure. *Molecular Biology and Evolution* 20:1692–1704.

Version dated: February 5, 2015

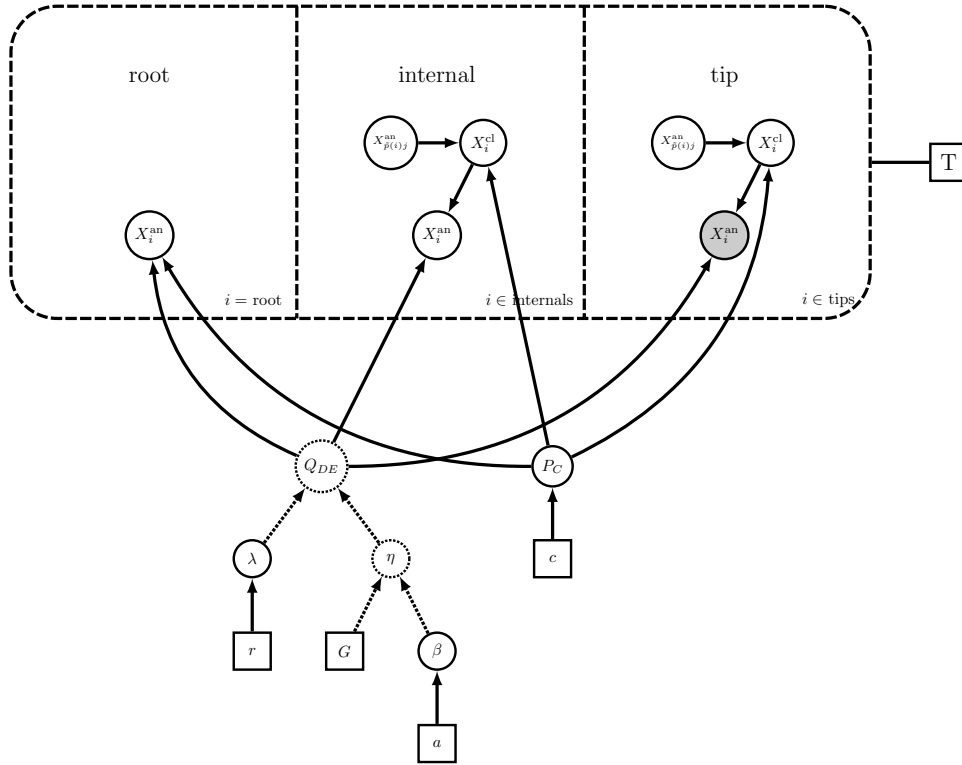


Figure 1: Graphical model of DEC. The tree plate's topology is fixed by T , where each internal node has both an anagenetic and cladogenic random variable (X_i^{an} and X_i^{cl} , resp.) that represents an ancestral species before and after it speciated. Anagenetic change is modeled by a continuous time Markov process, where Q_{DE} is the instantaneous rate matrix of area gain and loss, as parameterized by λ . The geographic distance rate modifier function, η , takes in the geographical distances and strata as G , and the distance power parameter, β . Cladogenic change is modeled by P_C , a Dirichlet-distributed simplex with a flat prior.

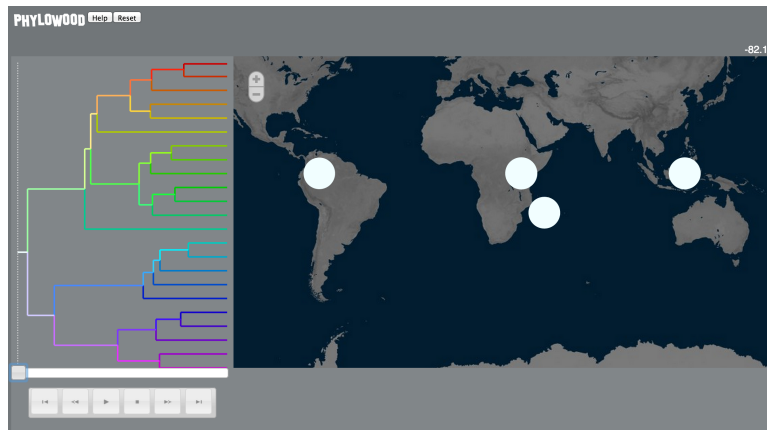


Figure 2: PhyloWood frame showing posterior ancestral range of root node.

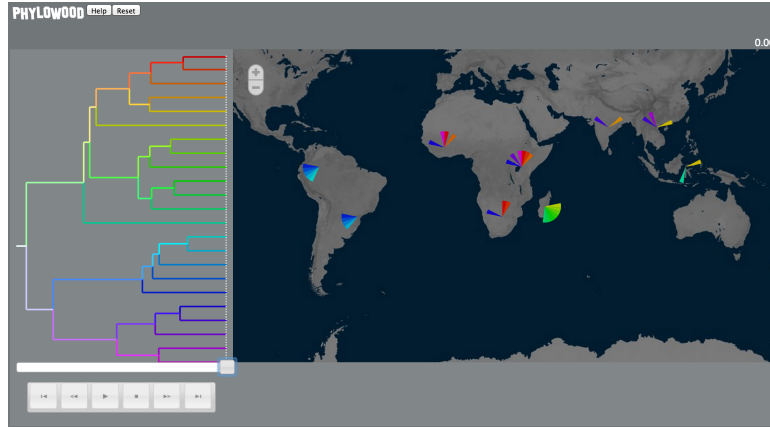


Figure 3: PhyloWood frame showing distribution of extant taxon ranges.

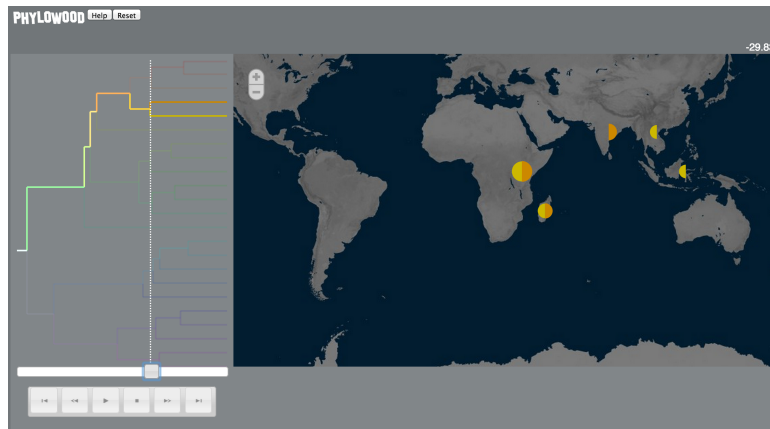


Figure 4: PhyloWood frame highlighting the ancestral range for the MRCA extant lorises.