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

I. Range evolution
- a) Dispersal-Extinction-Cladogenesis
- b) Model

II. Advanced features
- a) Epoch model
- b) GIS layers
- c) Ancestral range reconstructs

III. Data augmented biogeography
- a) BayArea
- b) Model
- c) Phylowood animation

2 Dispersal-Extinction-Cladogenesis model

2.1 Model and method

First, we define the range for taxon i as the bit vector X_i, where $X_{i,j} = 1$ if the taxon is present in area j and $X_{i,j} = 0$ if the taxon is absent. Each taxon range is a bit vector of length N areas. For example, if taxon B is present only in areas 2 and 3 out of $N = 3$ areas, its range is represented as $X_B = (0, 1, 1)$, which is translated to the bit string $X_B = 011$ for short. If we apply the labels A, B, and C to our three areas, this bit vector can also be represented as a set: $X_B = 011$ is the same as $X_B = \{B, C\}$, or $X_B = BC$ for short. The data matrix, $\mathbf{X}$, is analogous to a multiple sequence alignment where each element in the data matrix reports a discrete value for a homologous character shared by all taxa at column j.

2.2 Modeling anagenic range evolution

Next, we need a model of anagenic range evolution. 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

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

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. This technique of matrix exponentiation 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. Remember, i and j represent different ranges, each of which is encoded as a set of occupied areas.

We can then encode range evolution events into the allowed character transitions of $\mathbf{Q}$ and parameterize the events so that we may infer their relative importance to generating our observed ranges. 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.

Q: For the three-area DEC rate matrix above, what is the rate of leaving state 101? That is, what is the absolute value of the diagonal term in the rate matrix for $Q_{AB,AB}$?

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.

Q: What 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.

Q: 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 biogeographic inference under large numbers of areas, see the Tutorial XXX (**?**).

2.3 Modeling cladogenic range evolution

In addition to dispersal and extinction, the DEC models cladogenic range evolution events. For each internal node in the reconstructed tree, one of two cladogenic events can occur: sympatry or allopatry. Say the range of a species approaching an internal node, i.e. that is about to speciate, is A. Since the species range is size one, this always results in sympatry, where both daughter lineages inherit the ancestral species range, so both lineages begin in state A. The notation $A \rightarrow A \mid A$ describes this event: the state is A before cladogenesis, the left daughter inherits range A after cladogenesis, as does the right daughter.

Now suppose the ancestral range is ABC. Under 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.

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 a general description of state transitions for cladogenic events, see **?**.

Q: 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.

Q: 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 (**?**). 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, we need data augmentation to integrate over range evolution, which is used in Section XXX.

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 a dataset for 23 primates, which are the same taxa as used in Section XXX. To keep the model simple, we'll ranges are over three areas: the New World, Africa, and Eurasia. For simplicity, we'll assume their phylogeny is time-calibrated, errorless, and fixed.

First, set your working directory.

```
setwd("/Users/arwallace/projects/RB_Biogeography_tutorial/")
```

First we'll create some `String` variables for file handling

```
fp = "./"
data_fn = fp + "data/primates_bg_n3.tsv"
tree_fn = fp + "data/primates.tree"
```

then read in our character data

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

and our tree

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

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

```
n_areas = data.nchar()
n_states = 2^data.nchar()
```

Declare index variables for our move and monitor vectors for future use

```
mvi = 1
mni = 1
```

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.
    }
}
```

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.

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

```
r_e ~ dnExp(10.)
mv[mvi++] = 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 ~ dnExp(10.)
mv[mvi++] = 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 Equation (XX).

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 probabilites 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[mvi++]             = 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

```
b <- simplex(1,1)
clado_prob := fnCladoProbs(clado_type_prob, b, 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 ~ dnExp(10)
mv[mvi++] = 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=tree, 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[mni++] = 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)
mn[mni++] = 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=fp + "output/out.
    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] ~ dnExp(10.)
    mv[mvi++] = mvScale(e, 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] ~ dnExp(10.)
            mv[mvi++] = mvScale(d[i][j], weight=5)
        }
    }
}
```

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.

2.3.3 Exercises

- 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.
- (1 of 2) Saving your commands to a file, create a script to produce the "per-area" rate model.
- (2 of 2) 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 (see Section XXX).
- (Advanced) Using what you learned in this tutorial and the CTMC tutorial (Section XXX), perform a joint analysis of molecular and biogeographic evolution for primates. Introduce a common prior for the molecular and biogeographical clocks, e.g.

```
clock_mol      ~ dnExp(10.)
clock_scale_bg ~ dnGamma(2.,2.)
clock_bg       := clock_mol * clock_scale_bg
```

3 Epoch models and ancestral state reconstruction

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, ($O(n^4)$ 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 an adaptation of the data-augmented phylogenetic method first described by **?**, and was first applied to tertiary structure-dependent evolution of protein-coding nucleotide sequences.

This is the strategy we will use to infer the posterior distribution approximated by
$\text{Prob}(\mathbf{X}_{aug}, \theta \mid \mathbf{X}_{obs}, T, M)$, 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. Ancestral range reconstructions are often of primary interest in phylogenetic biogeographic analyses, which are generated with support values as a by-product of the MCMC analysis.

You may wonder why matrix exponentiation works fine for molecular substitution models and large multiple sequence alignments. Molecular substitution models typically assume each site in the multiple sequence alignment evolves independently, which may be justified because recombination degrades linkage disequilibrium over geological timescales. Conveniently, this keeps $\mathbf{Q}$ small even for datasets with many sites.

4.2 Large rate matrices

This matrix can be represented compactly as the rate function

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

where $\mathbf{y}$ and $\mathbf{z}$ are the "from" and "to" ranges and a is the area that changes. For example, $q^{(1)}_{011,111}$ is the rate of range expansion for $011 \rightarrow 111$ to gain area 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.

4.3 Distance-dependent dispersal function

Lastly, 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

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

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.

4.4 Analysis

First, we'll assign all our input files to `String` variables.

```
in_fp  <- "./data/"
data_fn <- "psychotria_range.nex"
area_fn <- "hawaii_dynamic.atlas.txt"
out_fp <- "./output/"
out_str <- "bg_2rate"
```

Then we'll create our range data, tree, and atlas objects

```
data <- readDiscreteCharacterData(in_fp + data_fn)
tree <- readTrees(in_fp + data_fn)[1]
atlas <- readAtlas(in_fp + area_fn)
```

Finally, we'll be using a few helper variables to create or move and monitor vectors

```
mvi = 1
mni = 1
```

4.5 Creating the model

Here, we will compose our rate matrix, $\mathbf{Q}$, parameterized by the transition rates, λ, and the distance dependent dispersal power parameter, β.

First, for λ, we will create a vector of two rates, where `glr[1]` corresponds to the rate of area gain (dispersal) and `glr[2]` corresponds to the rate of area loss (local extinction). Each rate will be drawn from an exponential distribution with rate 10.0 (mean 0.1). Because our tree is in units of millions of years, this means our prior expectation is that any given species undergoes one dispersal or extinction event per area per ten million years.

Let's declare the distributions for these priors along with their MCMC moves

```
for (i in 1:2) {
       glr[i]      ~ dnExponential(10.0)
       moves[mvi++] = mvScale(x=glr[i], lambda=0.5, tune=true, weight=5.0)
}
```

These are then inserted 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`, which determines the importance of geographical distance to dispersal. Remember that values of β far from zero means distance is important. So, if we 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 rate 10.0 (mean 0.1) as a prior for `dp`.

```
dp ~ dnExponential(10.0)
moves[mvi++] = 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, useAvailable=true, useDistance=true)
```

Now we need a deterministic node to represent the rate matrix, **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, `glr`, and our geographical rate modifier, `grm`. 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, geoRateMod=grm, numAreas=4, forbidExtinction
   =true)
```

To extract information for the frequencies of different cladogenic event types, we will create a Dirichlet-distributed stochastic node. The simplex is over three events, subset sympatry (index 0), allopatry (index

1), and widespread sympatry (index 2), but not over narrow sympatry whose range size is one. The prior parameter `[1,1,0.1]` is known as a flat prior, meaning all event types are expected to occur at equal frequency. If there is information in the data of a dominant cladogenic mode of range evolution, the posterior simplex values in `csf` will reflect this.

```
csf ~ dnDirichlet([1.,1.,0.5])
narrow_sympatry   := csf[1]
allopatry         := csf[2]
widespread_sympatry := csf[3]
moves[mvi++] = mvSimplexElementScale(csf, alpha=10.0, tune=true, weight=4.0)
```

For the model's final node, we create the stochastic node for the continuous-time Markov chain (CTMC). This node's distribution is `dnPhyloDACTMC` where `DA` indicates the CTMC uses data-augmentation to compute the likelihood rather than Felsenstein's pruning algorithm. To create the distribution, we must pass it our `tree` and `Q` 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 **?** (`useCladogenesis=true`), and that it will assign zero probability to a transition away from the null range state.

```
M ~ dnPhyloDACTMC(tree=tree, Q=q_range, C=csf, type="biogeo", forbidExtinction=true,
    useCladogenesis=true)
```

In addition to proposing new model parameter values, we must also propose new data-augmented states and events to properly integrate over the space of possible range histories. The major challenge to sampling character histories is ensuring the character histories are consistent with the observations at the tip of the tree. The proposals in this tutorial use **?**'s rejection sampling algorithm, with some modifications to account for cladogenic events and epoch-based rate matrices.

The basic idea is simple. Each time a character history proposal is called, it selects a node at random from the tree. Path history proposals (`mvPathCHRS()`) propose a new character history for the lineage leading to that node. Node history proposals (`mvNodeCHRS()`) propose a new character history for the node and for the three lineages incident to that node. The character history proposal also samples some number of areas to update, ranging from one to all of the areas. 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.

Because these `Move` objects update the character histories stored in the data-augmented CTMC node, e.g. `M`, they require access to a `TimeTree` object to know which lineages are sisters and whether the lineages span various epochs, and a `RateMap_Biogeography` object to propose new character histories. The `lambda` argument gives what proportion of areas' character histories to update. Here, if `lambda=0.2`, then the proposal will redraw character histories for each area with probability 0.2 (in addition to one random area with probability 1). Below, we use two moves of each type with `lambda=0.2` and `lambda=1.0` for partial and full character history updates, respectively. Indicating `type="biogeo"` informs the `Move` object to be aware of special character history constraints, such as cladogenic events and forbidden null ranges. The `weight` parameter should be assigned a value proportional to the number of nodes in the analysis to ensure proper mixing.

Let's create the character history moves as follows: conservative character history updates for paths and nodes, with `lambda=0.2`

```
moves[mvi++] = mvCharacterHistory(ctmc=M, qmap=q_range, tree=tree, lambda=0.2, type="
    Biogeo", graph="node", proposal="rejection", weight=40.0)
moves[mvi++] = mvCharacterHistory(ctmc=M, qmap=q_range, tree=tree, lambda=0.2, type="
    Biogeo", graph="branch", proposal="rejection", weight=40.0)
```

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

```
moves[mvi++] = mvCharacterHistory(ctmc=M, qmap=q_range, tree=tree, lambda=1.0, type="
    Biogeo", graph="node", proposal="rejection", weight=10.0)
moves[mvi++] = mvCharacterHistory(ctmc=M, qmap=q_range, tree=tree, lambda=1.0, type="
    Biogeo", graph="branch", proposal="rejection", weight=10.0)
```

So we may evaluate the graphical model's likelihood, we tell the CTMC to observe the `data` object, which will prime the model with data-augmented character histories. Now `M` has a defined likelihood value.

```
M.clamp(data)
M.lnProbability()
   -56.0288
```

Finally, we encapsulate our graphical model into a `Model` object, which can learn the model's structure and dependencies from any model parameter.

```
my_model = model(M)
```

First, we'll create monitors for our simple parameters

```
monitors[mni++] = mnScreen(glr, dp, narrow_sympatry, allopatry, widespread_sympatry,
    printgen=10)
monitors[mni++] = mnFile(glr, dp, narrow_sympatry, allopatry, widespread_sympatry,
    filename=out_fp+params_fn, 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 `Mntr_CharacterHistoryNewickFile` objects 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 **??**.

```
monitors[mni++] = mnCharHistoryNewick(filename=fp+out_str+".events.txt", ctmc=M, tree=
    tree, printgen=100, style="events")
monitors[mni++] = mnCharHistoryNewick(filename=fp+out_str+".counts.txt", ctmc=M, tree=
    tree, printgen=100, style="counts")
```

As our last monitor, the `Mntr_CharacterHistoryNhxFile` 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[mni++] = mnCharHistoryNhx(filename=fp+out_str+".nhx.txt", ctmc=M, tree=tree,
    atlas=atlas, samplegen=100, maxgen=25000, burnin=0.2)
```

WE CAN REPLACE THIS USING WRITETREE->FIGTREE Before we run the MCMC, we'd like to get the node index of the ancestor When analysing the output, we'll take some special interest in the branch for the most recent common ancestor of *P. kaduana* and *P. hatheway*. We will identify this lineage by its index, 23, which is meaningful only for a fixed tree topology,

```
> names = tree.names()
> names[15]
   P_kaduana_PuuKukuiAS
> names[16]
   P_hathewayi_1
> mrcaIndex(tree=tree,clade=clade(names[15],names[16]))
   23
```

4.6 Running an MCMC analysis

Now all that's left is to configure and run our MCMC analysis. For this, we create an `Mcmc` object, which we give our `Move` vector, our `Monitor` vector, and our `Model object`

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

MCMC typically requires some period of burn-in before it reaches stationarity, i.e. from a random starting point, it takes some time for the chain to produce valid samples from the posterior distribution. By running

`burnin()`, we tell the `Mcmc` object to propose and reject new states but *not* to record anything to file. After burn-in is complete, we call `run()`, where we begin recording valid posterior samples under our model.

```
my_mcmc.run(generations=25000)
```

Everything we've done is contained in the file `biogeography_DEC_2rate.Rev`. You can modify this file as you like then re-run the analysis by typing

```
source("./scripts/biogeography_DEC_2rate.Rev")
```

4.7 Exercises

4.8 Output

4.8.1 Sampled parameters from `ScreenMonitor`

→ Open Tracer, select the fields for the posterior probability and area gain rate, `glr[2]`, then click the Joint-Marginal tab.

Here, we see a strong negative correlation between the posterior probability and the area gain rate, which is expected. Next, click the Estimates tab then select the three `csf` parameters.

4.9 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_2rate.counts.txt` in a text editor.

```
0          -51.3307   -56.0288 4.69806      9       9       18      0       0       0
          1       1       0       ...
10         -54.4257   -58.1568 3.73110      9       10      17      0       0       1
          1       1       0       ...
20         -58.0696   -62.0923 4.02274      11      9       15      2       1       0
          2       1       1       ...
30         -46.5049   -51.1197 4.61480      8       8       18      0       0       0
          1       1       0       ...
40         -42.8697   -46.4870 3.61735      7       7       18      0       0       0
          1       1       0       ...
50         -43.5319   -47.4659 3.93394      7       7       18      0       0       0
          1       1       0       ...
...
```

For example, `b2_s1` gives the number of areas that are gained for the branch leading to the node indexed 2. `b2_c` gives the cladogenic event type that gives rise to the node indexed 2, where narrow sympatry, subset sympatry, allopatry, and widespread sympatry 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`, and `t_c2` give the total number of narrow sympatric, subset sympatric, allopatric, and widespread sympatric cladogenic events over the entire tree.

Because the expected number of gain events should be proportional to the area gain rate, we expect to see the same negative correlation between posterior probability and number of events as we did with the posterior and rate in the `parameters.txt` file.

Open Tracer, select the fields for the posterior probability and the number of gained areas over the tree, `t1`, then click the Joint-Marginal tab.

One interesting facet of this output is there are never fewer than six events. In fact, since we assume a stratified geography and that only one event may occur per instant, it is impossible to describe the data we see at the tips with fewer than six gain events. That is, six gain events is part of the maximum parsimony solution.

4.10 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.

→ Open `./output/bg_2rate.events.txt` in a text editor.

```
Iteration      Posterior      Likelihood     Prior  Tree
0      -51.3307       -56.0288       4.69806 (((((((P_hawaiiensis_WaikamoiL1[&index
    =18;nd=0010;pa=0010;ev={}]:0.96 ...
10     -54.4257       -58.1568       3.7311 (((((((P_hawaiiensis_WaikamoiL1[&index
    =18;nd=0010;pa=0010;ev={}]:0.96 ...
20     -58.0696       -62.0923       4.02274 (((((((P_hawaiiensis_WaikamoiL1[&index
    =18;nd=0010;pa=0010;ev={}]:0.96 ...
30     -46.5049       -51.1197       4.6148 (((((((P_hawaiiensis_WaikamoiL1[&index
    =18;nd=0010;pa=0010;ev={}]:0.96 ...
40     -42.8697       -46.4870       3.61735 (((((((P_hawaiiensis_WaikamoiL1[&index
    =18;nd=0010;pa=0010;ev={}]:0.96 ...
50     -43.5319       -47.4659       3.93394 (((((((P_hawaiiensis_WaikamoiL1[&index
    =18;nd=0010;pa=0010;ev={}]:0.96 ...

...
```

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

```
[&index=23;nd=0110;pa=0010;ch0=0010;ch1=0110;cs=s;bn=16;ev={{t:0.2513,a:1.1195,s:1,i:1}}
```

`index=23` indicates this branch leads to the node indexed 23. The branch began in the ancestral state `pa=0100` and terminated in the state `nd=0110`. Since this node is not a tip node, it represents a speciation event, so the daughter ranges are also given, `ch0=0010` and `ch1=0110`. The cladogenic state for this speciation event was subset sympatric, `cs=s`, rather than sympatric (wide or narrow; `w` or `n`) or allopatric (`a`). Anagenic dispersal and extinction events occurring along the lineage leading to node 19 are recorded in `events`, where each event has a time (relative to the absolute branch length), absolute age, state (into), and character index (`t`, `a`, `s`, `i`, resp.). For this posterior sample of the character history for the branch leading to node 22, the species range expanded into Oahu at age 1.1195.

To manipulate this data format, we'll use Python scripts. Below are a few examples of interesting posterior features.

→ Open a Python console and read in the events.

```
> cd scripts
> python27

...

>>> from bg_parse import *
>>> dd=get_events(fn="../output/bg_2rate.events")
```

By default, `get_events()` extracts a dictionary where each node index maps to a branch's character history as reported in `./input/bg_2rate.events.txt`. Each branch is a dictionary whose keys are various parts of the MCMC state and whose values the MCMC samples.

```
>>> dd[23].keys()
['ch1', 'iteration', 'bn', 'nd', 'ch0', 'prior', 'posterior', 'cs', 'ev', 'likelihood']
>>> dd[23]['posterior'][0:5]
[-48.6952, -60.1832, -53.2286, -57.5778, -53.4633]
```

To get the $n = 1$ highest-valued sample for a branch by its posterior value

```
>>> get_best(dd[23],n=1,p='posterior')
{'prior': [4.48225], 'iteration': [14890], 'bn': [22], 'nd': [[0, 1, 1, 0]], 'ch0': [[0,
    1, 1, 0]], 'ch1': [[0, 0, 1, 0]], 'posterior': [-34.7139], 'pa': [[0, 1, 0, 0]], '
   cs': ['subset_sympatry'], 'ev': [[{'age': 1.5637, 'state': 1, 'idx': 2, 'time':
   0.8611}]], 'likelihood': [-39.1962]}
```

To get the probability that area i and area j are both part of the species range as the branch for node 23 terminates, just before the speciation event

```
>>> get_area_pair(dd[23])
[[0.0816, 0.0188, 0.0628, 0.0000],
 [0.0188, 0.7081, 0.4390, 0.0000],
 [0.0628, 0.4390, 0.7141, 0.0000],
 [0.0000, 0.0000, 0.0000, 0.0000]]
```

showing area 3 was occupied nearly with probability 0.71 and both areas 2 and 3 were occupied with probability 0.44. Note, Hawaii was submerged until approximately 0.5 million years ago, and thus the probability of being in that area is 0.0.

If the range is size one during a speciation event, the cladogenic event state is always narrow sympatric, `'narrow_sympatry'`. But given the opportunity for non-sympatric events, i.e. that the range is larger than size one, we can get the probability of cladogenic state using For the probability for cladogenic event state given the range was larger than size one

```
>>> get_clado_state(dd[23])
{'allopatry': 0.0224, 'subset_sympatry': 0.1463, 'widespread_sympatry': 0.3183, '
    narrow_sympatry': 0.5130}
>>> get_clado_state(dd[23],minSize=2)
{'allopatry': 0.0460, 'subset_sympatry': 0.3005, 'widespread_sympatry': 0.6535, '
    narrow_sympatry': 0.0000}
>>> get_clado_state(get_best(dd[23],n=100),minSize=2)
{'allopatry': 0.1290, 'subset_sympatry': 0.6774, 'widespread_sympatry': 0.1936, '
    narrow_sympatry': 0.0000}
```

Depending on your question, different aspects of the posterior cladogenic state will interest you. Narrow sympatry is the favored ancestral state, but wide sympatry is favored for ranges of size $n > 1$. However, when we look at the 100 most probable samples, subset sympatry becomes most favored.

More script functions are found in `./scripts/bg_parse.py`.

4.11 New Hampshire extended format file (`./output/bg_2rate.nhx`)

Because this data is very high-dimensional, we'll use an external data exploration tool to look at range evolution.

This file summarizes the input and output from a BayArea analysis using NEXUS format containing a New Hampshire eXtended (NHX) tree string. NHX allows you to annotate nodes in a Newick string with meta-information, which BayArea uses to report the probabilities in the `my_run.area_probs.txt` file. The `geo` block gives the geographical latitudes and longitudes for the areas in the order they are reported as probabilities. Like the `my_run.area_probs.txt` file, this file is not written until the analysis is complete. This annotation is used for the two visualization programs covered in the next section, Phylowood and BayArea-Fig. The anatomy of the Phylowood and BayArea-Fig settings blocks will also be explained there.

5 Visualization

Here we'll explore two options for visualizing ancestral range reconstructions. I'll walk you through some of the basic functionality, but feel free to play around as you like.

5.1 Phylowood

Phylowood generates interactive animations to explore biogeographic reconstructions.

→ Open `http://mlandis.github.io/phylowood`.

→ Drag and drop `./output/bg_2rate.nhx.txt` into the text field.

→ Click the Play button to view the animation.

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.

Version dated: February 3, 2015

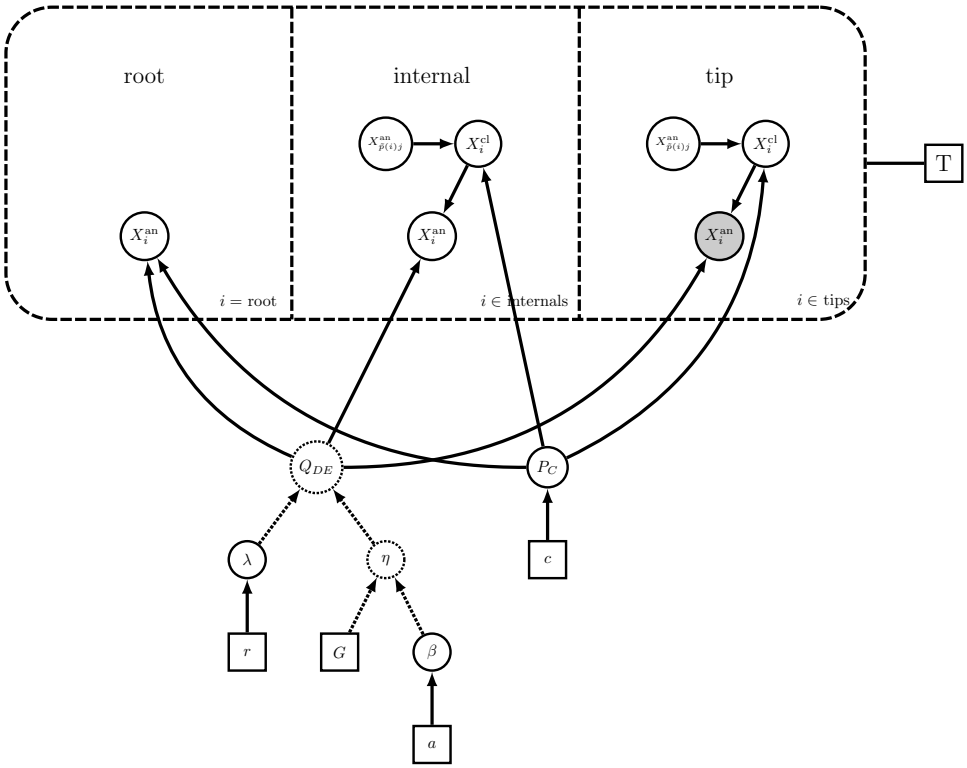

Figure 1: Graphical model of DEC. The tree plate's topology is fixed by T, where each internal node has both an anagenic and cladogenic random variable (X_i^{an} and X_i^{cl}, resp.) that represents an ancestral species before and after it speciated. Anagenic change is modeled by a continuous time Markov 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.

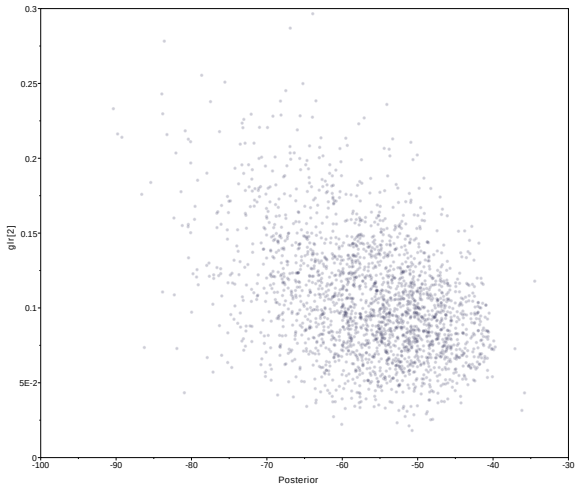

Figure 2: Joint-marginal distribution of posterior and area gain rate, λ_1.

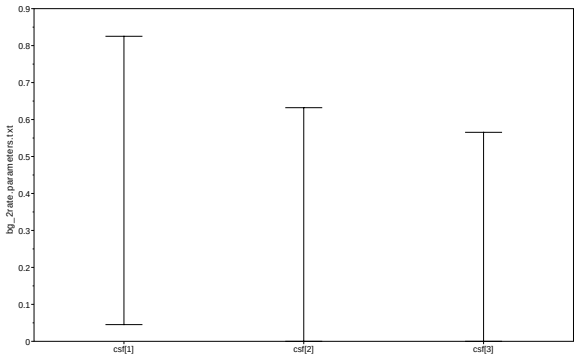

Figure 3: Mean values for the cladogenic state frequency simplex, where `csf[1]`, `csf[2]`, and `csf[3]` correspond to subset sympatry, allopatry, and wide sympatry whose mean posterior values are 0.45, 0.30, and 0.25, respectively.

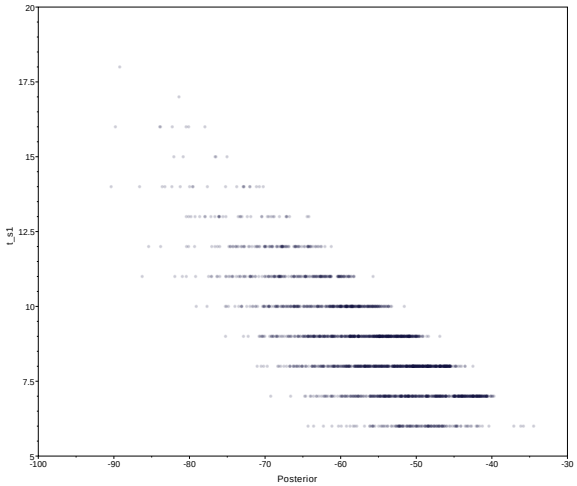

Figure 4: Joint-marginal distribution of posterior and number dispersal events summed over the tree.

Figure 5: Phylowood frame showing posterior ancestral range of root node.

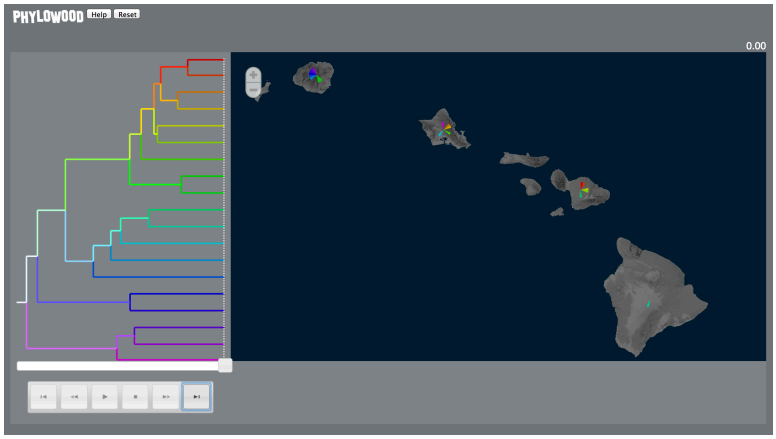

Figure 6: Phylowood frame showing distribution of extant taxon ranges.

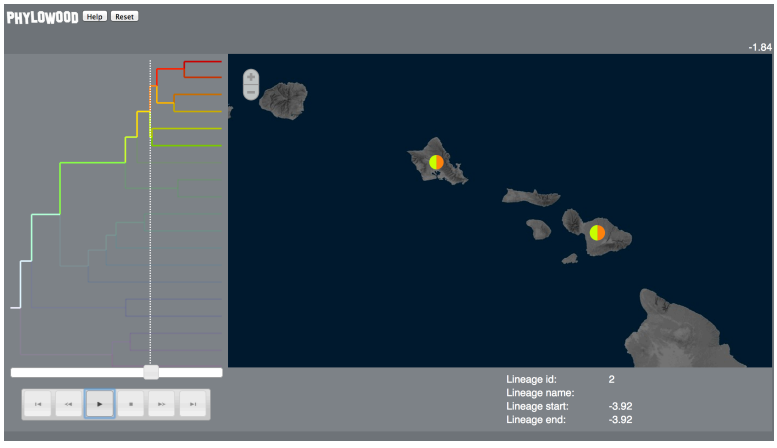

Figure 7: Phylowood frame highlighting the posterior range for the most recent common ancestor of *P. mauiensis* and *P. hawaiiensis*.