Phylogenetic Inference using `RevBayes`

Historical biogeography

Michael Landis

Introduction

How did species come to live where they're found today? To answer this, we can leverage phylogenetic and geographical information to model species distributions as the outcome of biogeographic processes. How to best model these processes requires special consideration, such as how ranges are inherited following speciation events, how geological events might influence dispersal rates, and what factors affect rates of dispersal and extirpation. The major challenge of modeling range evolution is how to translate these natural processes into stochastic processes that remain tractable for inference. This tutorial provides a brief background in some of these models, then describes how to perform Bayesian inference of historical biogeography using `RevBayes`.

1 Overview: Dispersal-Extinction-Cladogenesis model

1.1 Range characters

Discrete biogeographical models typically rely on presence-absence data, that is, where a species is observed or not observed across multiple discrete areas. For example, say there are three areas: A, B, and C. If 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.

Range	Bits	Size	State
$\emptyset$	000	0	0
A	100	1	1
B	010	1	2
C	001	1	3
AB	110	2	4
AC	101	2	5
BC	011	2	6
ABC	111	3	7

Table 1: Example of discrete range representations for an analysis with areas A, B, and C.

Decimal representation is rarely used in discussion, but it is useful to keep in mind when considering the total number of possible ranges for a species.

1.2 Modeling anagenetic range evolution

How might we model the dynamics of species range evolution? In this section, we'll cover the Dispersal-Extinction-Cladogenesis model first described by Ree et al. (2005). To begin, we'll focus on anagenesis: evolution that occurs between speciation events within lineages. 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. 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, (local) extinction, or area loss.) The rates in the transition matrix for three areas might appear as

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

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. Notice that the sum of rates leaving state $\emptyset$ is zero, meaning any species that loses all areas in its range remains permanently extinct.

Assume you have three areas

```
n_areas <- 3
```

First, create a matrix of dispersal rates between area pairs.

```
for (i in 1:n_areas) {
    for (j in 1:n_areas) {
        dr[i][j] <- abs(1)
    }
}
```

Next, let's create the extirpation rates

```
for (i in 1:n_areas) {
    for (j in 1:n_areas) {
        er[i][j] <- abs(0)
    }
    # overwrite the diagonal terms
    er[i][i] <- abs(1)
}
```

When the only non-zero extirpation rates are on the diagonal of the matrix, extirpation rates are independent of what other areas the taxon is occupies. More complex models that penalize widespread ranges spanning disconnected areas are explored in later sections.

Now, create the DEC rate matrix

```
Q_DEC := fnDECRateMatrix(dispersalRates=dr, extirpationRates=er)
Q_DEC
[ [ 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000 ] ,
    1.0000, -3.0000, 0.0000, 0.0000, 1.0000, 1.0000, 0.0000, 0.0000 ] ,
    1.0000, 0.0000, -3.0000, 0.0000, 1.0000, 0.0000, 1.0000, 0.0000 ] ,
    1.0000, 0.0000, 0.0000, -3.0000, 0.0000, 1.0000, 1.0000, 0.0000 ] ,
    0.0000, 1.0000, 1.0000, 0.0000, -4.0000, 0.0000, 0.0000, 2.0000 ] ,
    0.0000, 1.0000, 0.0000, 1.0000, 0.0000, -4.0000, 0.0000, 2.0000 ] ,
    0.0000, 0.0000, 1.0000, 1.0000, 0.0000, 0.0000, -4.0000, 2.0000 ] ,
    0.0000, 0.0000, 0.0000, 0.0000, 1.0000, 1.0000, 1.0000, -3.0000 ] ]
```

Show the anagenetic transition probabilities for a branch of length `0.2`

```
tp_DEC <- Q_DEC.getTransitionProbabilities(rate=0.2)
tp_DEC
[ [ 1.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000, 0.000],
  [ 0.000, 0.673, 0.013, 0.013, 0.123, 0.123, 0.005, 0.050],
  [ 0.000, 0.013, 0.673, 0.013, 0.123, 0.005, 0.123, 0.050],
  [ 0.000, 0.013, 0.013, 0.673, 0.005, 0.123, 0.123, 0.050],
  [ 0.000, 0.107, 0.107, 0.004, 0.502, 0.031, 0.031, 0.218],
  [ 0.000, 0.107, 0.004, 0.107, 0.031, 0.502, 0.031, 0.218],
  [ 0.000, 0.004, 0.107, 0.107, 0.031, 0.031, 0.502, 0.218],
  [ 0.000, 0.021, 0.021, 0.021, 0.107, 0.107, 0.107, 0.616]]
```

Substructures from the rate matrix are reflected in the transition probability matrix. Notice that ranges that are separated by multiple dispersal and extirpation events are the most improbable – e.g. going from A to BC takes three events and has probability 0.005.

By default, the process conditions on the process never entering the null range when computing the transition probabilities. This is crucial so the model can both simulate and infer using identical (correct) probabilities. The `condition=Include` setting computes the raw probabilities of (**?**). Setting `condition=Exclude` is the DEC* model of (**?**) that removes the null range from the state space.

1.3 Modeling cladogenetic range evolution

Cladogenesis describes evolutionary change accompanying speciation. Daughter species are not expected to inherit their ancestral range identically in general. For each internal node in the reconstructed tree, one of two cladogenetic 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 *widspread 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.

For an excellent overview of described state transitions for cladogenetic events, see Matzke (2012).

Make the cladogenetic probability event matrix

```
clado_event_types = [ "s", "a" ]
clado_event_probs <- simplex( 1, 1 )
P_DEC := fnDECCladoProbs(eventProbs=clado_event_probs,
                         eventTypes=clado_event_types,
                         numCharacters=n_areas)
```

`clado_event_types` defines what cladogenetic event types are used. `"a"` and `"s"` indicate allopatry and subset sympatry, as described in (**?**). Other cladogenetic events include jump dispersal (`"j"`) and full sympatry (`"f"`). The cladogenetic event probability matrix will assume that `eventProbs` and `eventTypes` share the same order.

Print the cladogenetic transition probabilities

```
P_DEC
   [
     ( 1 -> 1, 1 ) = 1.0000,
     ( 2 -> 2, 2 ) = 1.0000,
     ( 3 -> 3, 3 ) = 1.0000,
     ...
     ( 7 -> 7, 1 ) = 0.0833,
     ( 7 -> 7, 2 ) = 0.0833,
     ( 7 -> 7, 3 ) = 0.0833
   ]
```

The cladogenetic probability matrix becomes very sparse for large numbers of areas, so only non-zero values are shown.

1.4 Things to consider

The probabilities of anagenetic 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 cladogenetic 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 $8 \times 8 \times 8 = 512$ cladogenetic probability terms.

Of course, this model can be specified for more than three areas. Let's consider what happens to the size of **Q** when the number of areas, N, becomes large. For three areas, **Q** is size 8×8. For ten areas, **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, **Q** is size $2^{20} \times 2^{20} \approx 10^6 \times 10^6$ and exponentiation is not viable.

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 **??**.

1.5 Some questions

? For the three-area DEC rate matrix above, what is the rate of leaving state AC in terms of dispersal and extinction parameters?

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

? 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?

? Given the state is AB before cladogenesis, and allowing subset sympatry, widespread sympatry, and allopatry, what are the 7 possible states in the daughter lineages after cladogenesis?

? For three areas, there are three narrow, four widespread, 18 subset sympatric events and 12 allopatric cladogenesis events. What proportion of terms in the cladogenesis matrix are zero?

2 Simple DEC analysis

The tutorials will reconstruct the ancestral ranges of the silversword alliance (Tribe *Madiinae*), a diverse clade of about 50 species. Although silverswords are endemic to Hawaii, they are nested within a larger clade alongside tarweeds, which are native to the California coastline. The size and age of the clade, combined with our knowledge of Hawaiian island formation, makes it an ideal system to explore concepts in historical biogeography and phylogeny. [Much of this work draws on insights coming from Baldwin and colleagues.]

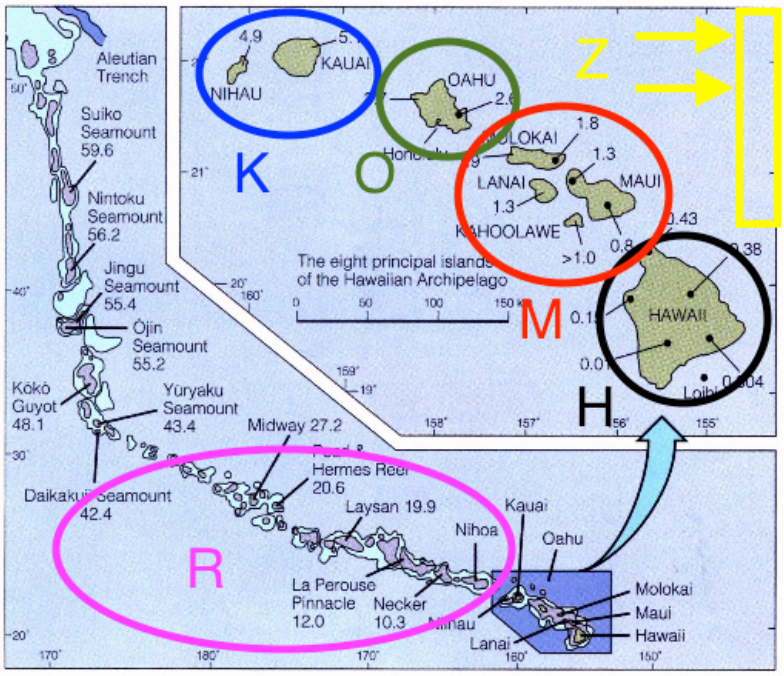

Figure 1: A beautiful figure of the discrete areas for tutorial. Six areas are shown: Kauai and Niihau (K); Oahu (O); Maui-Nui, Lanai, and Molokai (M); Hawaii (H); the remaining Hawaiian islands (R); and the North American mainland (Z).

For this analysis in this section, we will use just four areas

Range	Areas	Size	State
∅	0000	0	0
K	1000	1	1
O	0100	1	2
M	0010	1	3
H	0001	1	4
KO	1100	2	5
KM	1010	2	6
OM	0110	2	7
KH	1001	2	8
OH	0101	2	9
MH	0011	2	10
KOM	1110	3	11
KOH	1101	3	12
KMH	1011	3	13
OMH	0111	3	14
KOMH	1111	4	15

Table 2: Area coding used for four areas: K is Kauai and Nihoa; O is Oahu; M is Maui Nui, Lanai, and Molokai; H is Hawaii island.

First, we'll create some variables to manage files

```
range_fn = "data/n4/silversword.n4.range.nex"
phy_fn = "data/n4/silversword.tre"
out_fn = "output/simple"
```

then read in our character data as binary presence-absence characters

```
dat_range_01 = readDiscreteCharacterData(range_fn)
```

then encode the species ranges into natural numbers

```
dat_range_n   = formatDiscreteCharacterData(dat_geo_01, "DEC")
```

Record the number of areas

```
n_areas = dat_geo_01.nchar()
```

You can view the taxon data to see how characters are coded

```
dat_range_01[1]
 Argyroxiphium_grayanum_East_Maui:
    0010
dat_range_n[1]
 Argyroxiphium_grayanum_East_Maui:
    3
```

For now, we'll assume we know the dated species phylogeny without error.

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

For this analysis, we'll assume that all pairs of areas share the same dispersal rate and all areas share the same extirpation rate.

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

```
r_d ~ dnExponential(1)
mv[++mi] = mvScale(r_d)
```

then create the dispersal rate matrix

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

Next, assign the prior distribution to the extirpation rate and assign it a move

```
r_e ~ dnExponential(1)
mv[++mi] = mvScale(r_e)
```

then create a matrix of extirpation rates

```
for (i in 1:n_areas) {
    for (j in 1:n_areas) {
            er[i][j] <- abs(0)
    }
```

```
        er[i][i] := r_e
}
```

This diagonal matrix results in per-area extirpation rates that are mutually independent. All non-diagonal extirpation rates equal zero. (More on penalized ranges and off-diagonal rates later.)

```
Q_DEC := fnDECRateMatrix(dispersalRates=dr, extirpationRates=er)
```

Note, `fnDECRateMatrix` does not rescale its elements in any way, so transition rates share the same time scale as the underlying tree. This is in contrast to the standard molecular substitution processes (e.g. `fnGTR`) whose rates are rescaled such that the process is expected to produce one event per site per unit time. For our parameterization, we assign the "biogeographic clock" the rate of one such that $\mathbf{Q} = \mu\mathbf{Q} = 1\mathbf{Q}$.

```
rate_bg <- 1
```

In contrast, cladogenetic 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 cladogenetic events

```
clado_event_types <- [ "s", "a" ]
clado_type_probs <- simplex(1, 1)
P_DEC := fnDECCladoProbs(eventProbs=clado_type_probs,
                         eventTypes=clado_event_types,
                         numCharacters=n_areas)
```

Create the cladogenetic transition probability matrix, which assigns probabilities to cladogenetic event classes.

Finally, all our model components are encapsulated in the `dnPhyloCTMCClado` distribution, which is similar to `dnPhyloCTMC` except specialized to integrate over cladogenetic events. Although this dataset has four areas, it is recognized single character with states valued from 1 to 2^4, hence `nSites=1`.

```
m ~ dnPhyloCTMCClado( tree=psi, Q=Q_DEC, cladoProbs=P_DEC, branchRates=rate_bg, nSites
   =1, type="NaturalNumbers" )
```

The remaining tasks should be familiar from previous tutorials, so we can proceed briskly. Attach the observed ranges to the model.

```
m.clamp(dat_geo)
```

Compose the model.

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

Add the monitors. (The `mnJointConditionalAncestralState` monitor will be described later.)

```
mn[1] = mnScreen(r_d, r_d, printgen=1000)
mn[2] = mnModel(file=out_fn+".params.txt", printgen=100)
mn[3] = mnJointConditionalAncestralState(tree=psi, ctmc=m, filename=out_fn+".states.txt
    ", type="NaturalNumbers", printgen=100, withTips=true, withStartStates=true)
```

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

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

2.0.1 Visualizing ancestral state reconstructions

The `mnJointConditionalAncestralState` monitor above created a states file. Each row in the states file lists the joint sample of ancestral states conditioned on the tip values for the whole tree. Each column corresponds to the phylogenetic node index for that particular MCMC sample. The index is used to later correspond the state samples with the tree samples when the topology is a random variable. (More on this in the ancestral state tutorial.)

The script located at `scripts/make_anc_states.Rev` contains code to construct an ancestral state tree. To use it for other analyses, just modify the `out_str` variable below.

Open a new RevBayes session. Set up the files we'll work with.

```
out_str     = "output/simple"
out_state_fn = out_str + ".states.log"
out_phy_fn  = out_str + ".tre"
out_mcc_fn  = out_str + ".mcc.tre"
```

Get the ancestral state trace

```
state_trace = readAncestralStateTrace(file=out_state_fn)
```

Get the ancestral state tree trace. It is important to use `readAncestralTreeTrace` and not `readTreeTrace` to properly annotate the tree with ancestral states.

```
tree_trace = readAncestralStateTreeTrace(file=out_phy_fn, treetype="clock")
```

Read the maximum clade credibility tree and write it to file

```
mccTree(tree_trace, file=out_mcc_fn)
mcc_tree = readTrees(out_mcc_fn)[1]
```

Build the ancestral state tree

```
anc_tree = ancestralStateTree(tree=mcc_tree, ancestral_state_tree_trace_vector=
    state_trace, tree_trace=tree_trace, include_start_states=true, file=out_str+".ase",
    summary_statistic="MAP", site=0)
```

We can review the output in `ancestralStateTree` in FigTree.

Nodes are annotated with the first three most probable ancestral states along with their posterior probabilities. When the tree is a random variable, as it is in later exercises, additional information about phylogenetic uncertainty is reported.

Finally, we can also generate a figure with ancestral states in R using RevGadgets that is suitable for publication.

```
# RevGadgets requires development tools for installation
install.packages("devtools")
library(devtools)

# Install RevGadgets
install_github("revbayes/RevGadgets")

# Note about ggtree dependency:
# RevGadgets requires ggtree version 1.5.14 or greater. This can be installed directly
    from GitHub:
install_github("GuangchuangYu/ggtree")
```

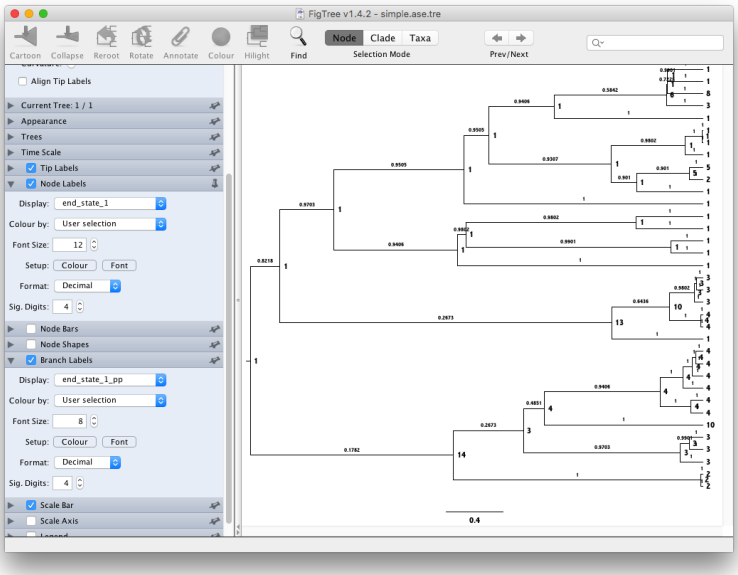

Figure 2: Tree with ancestral state estimates. The most probable end state of each branch (before cladogenesis) is shown at each node. Branches are labeled with the posterior probability for the ancestral state on the tipwards end of the branch.

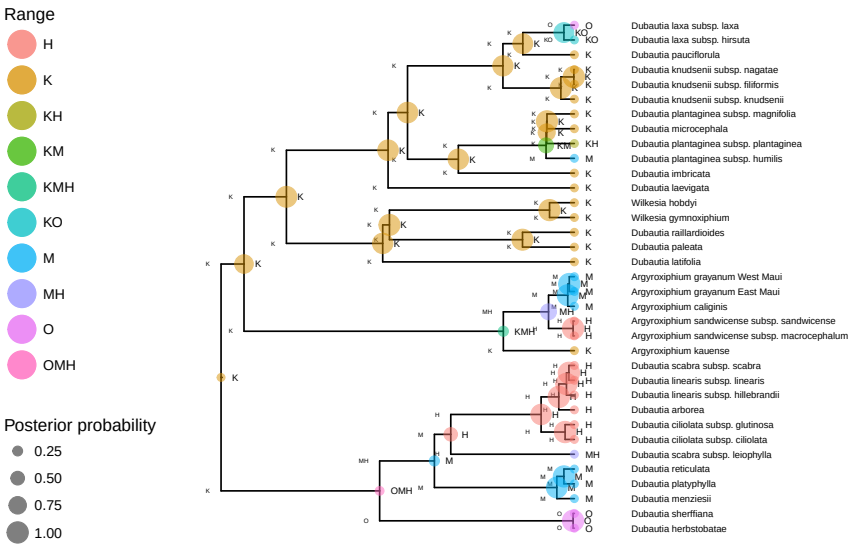

Figure 3: Tree with ancestral state estimates for the "simple" analysis. Nodes are annotated with ancestral states before and after cladogenetic events. Most probable states are shown. Colors of markers indicate the range state. Sizes of markers indicate the posterior probability of that state.

Once this is installed you can generate a figure by executing `source("plot_anc_state.n4.R")` from within an R session.

Modifying the source file allows you to use the script with different datasets.

3 An improved DEC analysis

In this section, we'll focus on techniques that allow permit more realistic biogeographic analyses. Topics include applying range size constraints, area connectivity, stratified/epoch models, function-valued dispersal rates, and incorporating uncertainty in paleogeographic event time estimates.

Start by creating our filename variables

```
range_fn = "data/n4/silversword.n4.range.nex"
phy_fn   = "data/n4/silversword.tre"
out_fn   = "output/complex"
geo_fn   = "data/n4/hawaii.n4"
times_fn = geo_fn + ".times.txt"
dist_fn = geo_fn + ".distances.txt"
```

Create some helper analysis variables

```
mvi = 1
mni = 1
n_gen = 1e3
```

Read in the presence-absence range characters and record the number of areas in the dataset

```
dat_range_01 = readDiscreteCharacterData(range_fn)
n_areas <- dat_range_01.nchar()
```

Often, biogeographers wish to limit to the maximum allowable range size. This prohibits widespread species ranges and to reduce the total number of range states in the analysis, thus benefitting computational efficiency. Suppose we disallowed ranges from including more than two areas. The total number of ranges equals $\sum_{k=0}^{m} \binom{n}{k}$ where n is the total number of areas, m is the maximum number of permissible areas, and $\binom{n}{k}$ is the number of ways to sample k unordered areas from a pool of n areas.

```
max_areas <- 2
n_states <- 0
for (k in 0:max_areas) n_states += choose(n_areas, k)
```

Then format the dataset for the reduced state space

```
dat_range_n = formatDiscreteCharacterData(dat_range_01, "DEC", n_states)
```

Our state space now includes only 11 states (∅, K, O, M, H, KO, KM, OM, KH, OH, MH).

Next, we'll set up the paleogeographic model. Read in the list of minimum and maximum ages of island formation

```
time_bounds <- readDataDelimitedFile(file=times_fn, delimiter=" ")
n_epochs <- time_bounds.size()
```

Read in a vector of matrices that describe the connectivity between areas over time. Note, there is one connectivity matrix per epoch, ordered from oldest to youngest.

```
for (i in 1:n_epochs) {
    connectivity[i] <- readDataDelimitedFile(file=geo_fn+"."+i+".txt", delimiter=" ")
}
```

Read in the matrix of distances between all pairs of areas (km). For simplicity, we will assume that distances remain constant over time, even though they certainly vary.

```
distances <- readDataDelimitedFile(file=dist_fn, delimiter=" ")
```

Dispersal rates might make use of some extrinsic information, such as geographical distances between areas (**??**). We model this as $d_{ij} = ae^{-bg_{ij}}$ where g_{ij} is the geographical distance between areas i and j and a and b are parameters that scale distance on linear and exponential scales, respectively. Note that all dispersal rates equal a when $b = 0$.

```
a ~ dnGamma(2,2)
moves[mvi++] = mvScale(a)
b ~ dnUniform(-5,5)
b.setValue(0)
moves[mvi++] = mvSlide(b, delta=0.1)
```

Now we can assign rates that are functions of distance between all pairs of areas

```
for (i in 1:n_epochs) {
    for (j in 1:n_areas) {
        for (k in 1:n_areas) {
            dr[i][j][k] <- abs(0)
            if (connectivity[i][j][k] > 0) {
                dr[i][j][k] := a * exp( -b * distances[j][k] )
            }
```

```
        }
    }
}
```

It is unlikely that widespread ranges persist across disjunct areas for long periods of time. Extirpation is more likely to occur in fragmented ranges than well-connected ranges, where peripheral populations are continuously reinforced from the center.

```
e ~ dnExp(1)
moves[mvi++] = mvScale(e)

for (i in 1:n_epochs) {
    for (j in 1:n_areas) {
        for (k in 1:n_areas) {
            er[i][j][k] <- abs(0)
            if (j == k) {
                er[i][j][k] := e
            }
        }
    }
}
```

Treat epoch times as random variables. The present is always the present.

```
for (i in 1:n_epochs) {
    time_max[i] <- time_bounds[i][1]
    time_min[i] <- time_bounds[i][2]
    if (i != n_epochs) {
        epoch_times[i] ~ dnUniform(time_min[i], time_max[i])
        moves[mvi++] = mvSlide(epoch_times[i], delta=0.5)
    } else {
        epoch_times[i] <- 0.0
    }
}
```

Build a rate matrix for each time interval

```
for (i in 1:n_epochs) {
    Q_DEC[i] := fnDECRateMatrix(dispersalRates=connectivity[i],
                          extirpationRates=er,
                          maxRangeSize=max_areas)
}
```

Create the epoch rate generator object

```
Q_DEC_epoch := fnDECCladoProbs(Q=Q_DEC, times=epoch_times, rates=rep(1,n_epochs))
```

Here, we treat the probability of different types of cladogenetic events as a random variable to be estimate.

```
clado_event_types <- [ "s", "a" ]
clado_type_probs ~ dnDirichlet( [1,1] )
moves[++mi] = mvSimplexElementScale(clado_type_probs, alpha=10)
P_clado := fnDECCladoProbs(eventProbs=clado_type_probs,
                           eventTypes=clado_event_types,
                           numCharacters=n_areas,
                           maxRangeSize=max_areas)
```

For this dataset, we assume cladogenetic probabilities are constant with respect to geological time.

```
rate_bg <- 1
```

```
rf_DEC <- rep(0, n_states)
rf_DEC[2] <- 1
rf_DEC <- simplex(rf_DEC)
```

Create the phylogenetic model

```
ctmc_bg ~ dnCTMCClado(tree=phy,
                      Q=Q_DEC_epoch,
                      cladoProbs=P_DEC,
                      branchRates=rate_bg,
                      rootFrequencies=rf_DEC,
                      type="NaturalNumbers",
                      nSites=1)
```

Attach the dataset

```
ctmc_bg.clamp(dat_range_n)
```

And the rest we've done before...

```
# make the model
mdl = model(m)

# make the monitors
mn[1] = mnScreen(dispersal_rate, distance_scale, extirpation_rate, printgen=1000)
mn[2] = mnModel(file=out_fn+".params.txt", printgen=100)
mn[3] = mnJointConditionalAncestralState(tree=psi, ctmc=m, filename=out_fn+".states.txt
    ", type="NaturalNumbers", printgen=100, withTips=true, withStartStates=true)

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

The ancestral state estimates look much more realistic, given what we know about when the islands were formed and the speciation times.

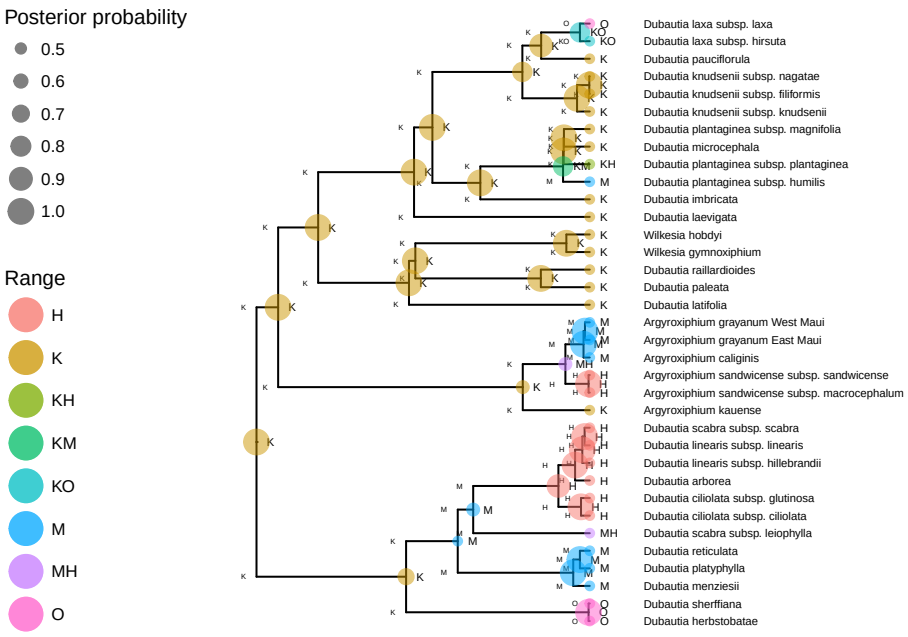

Figure 4: Tree with ancestral state estimates. Nodes are annotated with ancestral states before and after cladogenetic events. Most probable states are shown. Colors of markers indicate the range state. Sizes of markers indicate the posterior probability of that state.

[Look at posterior estimates for distance.]

4 Biogeographic dating using DEC

This analysis will jointly estimate phylogeny and biogeography. One benefit is that the biogeographic analysis will intrinsically accommodate phylogenetic uncertainty. Another is that paleogeographic evidence may provide information about the geological timing of speciation events in the phylogeny. Finally, biogeographic data may lend support to certain phylogenetic relationships that have poor resolution otherwise.

Hawaiian silverswords are nested in a larger group of plants, the tarweeds. Fossil pollen evidence indicates that tarweeds diversified during a period of aridification from 15–5 Ma in the western regions of North America (cite Baldwin papers). Although the oldest Hawaiian island that silverswords inhabit is Kauai, it is possible that silverswords first colonized older islands in the Emperor Island chain that predate the formation of Kauai at about 5.1 Ma.

This makes traditional node-based biogeographic calibrations challenging, because it would require a strong assumption about when and how many times the oldest silversword lineages colonized Kauai. Did silverswords colonize Kauai once directly from the California coast? Or did the colonize the younger islands multiple times from older islands in the chain? And did the event occur immediately after Kauai surfaced or much later? Because we cannot observe the timing and nature of this event directly, this process-based biogeographic dating approach does so through probabilistic inference.

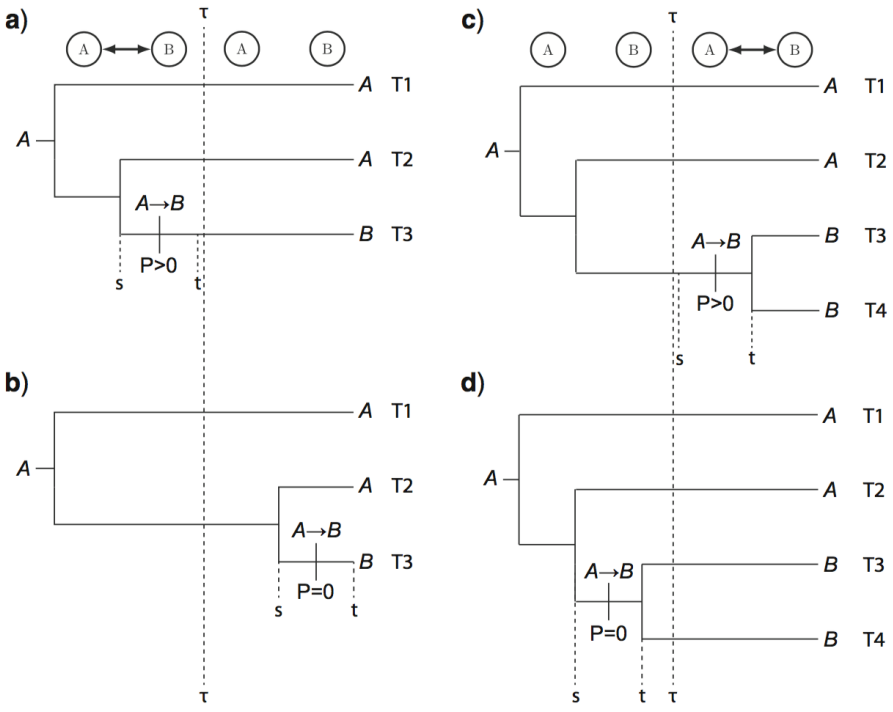

Figure 5: Cartoon of biogeographic transition probabilities as functions of geological time, and how that relates to speciation times. (a) Areas split, dispersal before split, positive probability; (b) Areas split, dispersal after split, zero probability; (c) Areas merge, dispersal after merge, positive probability; (d) Areas merge, dispersal before merge, zero probabilty. [Improve description later.]

We will move from our simpler 4-area model to a 6-area model, where we now include the remaining

Hawaiian islands (R) and the North American mainland (Z). Additionally, we will add three tarweed taxa to our dataset, increasing the total number of taxa to 38. We'll use internal transcribed spacer (ITS) to estimate the phylogeny, which is a 600 bp non-coding locus that is historically important for plant systematics. Because the locus is relatively short, it will also leave us with a fair amount of phylogenetic uncertainty in branch length and topology estimates. However, because we're estimating phylogeny and biogeography, it will be correctly incorporated into our ancestral range estimates.

Much of this tutorial will be similar to the previous section, except we are adding a birth-death process and a molecular substitution process to the model graph.

Create the necessary input/output variables.

```
range_fn = "data/n6/silversword.n6.range.nex"
mol_fn  = "data/n6/silversword.mol.nex"
phy_fn  = "data/n6/silversword.tre"
out_fn  = "output/epoch_phy"
geo_fn  = "data/n6/hawaii.n6"
times_fn = geo_fn + ".times.txt"
dist_fn = geo_fn + ".distances.txt"
```

Add the analysis helper variables

```
mvi = 1
mni = 1
n_gen = 1e5   # more parameters, longer run!
```

Impose limits to the maximum range size, both to prohibit widespread species ranges and to improve the computational efficiency of the method.

First, get the number of areas

```
dat_range_01 = readDiscreteCharacterData(range_fn)
n_areas <- dat_range_01.nchar()
```

Suppose we wanted to forbid ranges from being three or more areas in size. The total number of ranges is $\sum_{k=0}^{m} \binom{n}{k}$ where n is the total number of areas, m is the maximum number of permissible areas, and $\binom{n}{k}$ is the number of ways to sample k unordered areas from a pool of n areas.

```
max_areas <- 2
n_states <- 0
for (k in 0:max_areas) n_states += choose(n_areas, k)
```

Then format the dataset for the reduced state space

```
dat_range_n = formatDiscreteCharacterData(dat_range_01, "DEC", n_states)
```

Read in the list of minimum and maximum ages of island formation

```
time_bounds <- readDataDelimitedFile(file=times_fn, delimiter=" ")
n_epochs <- time_bounds.size()
```

Read in a vector of matrices that describe the connectivity between areas over time. Note, there is one connectivity matrix per epoch, ordered from oldest to youngest.

```
for (i in 1:n_epochs) {
    connectivity[i] <- readDataDelimitedFile(file=geo_fn+"."+i+".txt", delimiter=" ")
}
```

Read in the matrix of distances between all pairs of areas (km). For simplicity, we will assume that distances remain constant over time, even though they certainly vary.

```
distances <- readDataDelimitedFile(file=dist_fn, delimiter=" ")
```

4.0.1 The tree model

In this exercise we will also be estimating the phylogeny (topology and branch lengths), meaning our tree will be a stochastic node with a prior distribution. For this, we'll use a constant rate birth-death process.

Assign root age with a maximum age of 15Ma to reflect the fossil pollen record for Californian tarweeds [cite].

```
root_age ~ dnUniform(0, 15)
moves[mvi++] = mvScale(root_age, weight=2)
```

Assign the proportion of sampled taxa (we have a non-uniform sampling scheme, but this should suffice).

```
rho <- 35/50
```

Assign the birth and death priors

```
birth ~ dnExp(1)
moves[mvi++] = mvScale(birth)
death ~ dnExp(1)
moves[mvi++] = mvScale(death)
```

Instantiate a tree variable generated by a birth-death process

```
phy ~ dnBDP(lambda=birth, mu=death, rho=rho, rootAge=root_age, taxa=taxa)
```

Add topology and branch length moves

```
moves[mvi++] = mvNNI(phy, weight=n_branches/2)
moves[mvi++] = mvFNPR(phy, weight=n_branches/8)
moves[mvi++] = mvNodeTimeSlideUniform(phy, weight=n_branches/2)
```

Provide a starting tree (improves mixing, not essential)

```
phy.setValue(phy_init)
root_age.setValue(phy_init.rootAge())
```

4.0.2 The molecular model

In addition, to inform our branch lengths (in relative time units) and our topology, we will specify a simple HKY+Γ4+UCLN model of molecular substitution.

First specify a base rate for the molecular clock. This prior is uniform over orders of magnitude, between 10^-6 and 10^3

```
log10_rate_mol ~ dnUniform(-6, 3)
log10_rate_mol.setValue(-1)
moves[mvi++] = mvSlide(log10_rate_mol, weight=5, delta=0.2)
rate_mol := 10^log10_rate_mol
```

Assign log-normal relaxed clock rate multipliers to each branch in the tree. These priors have a mean of 1 so each branch prefers a strict clock model in the absence of data.

```
branch_sd <- 1.0
branch_mean <- 0.0 - 0.5 * branch_sd^2
```

```
for (i in 1:n_branches) {
    branch_rate_multiplier[i] ~ dnLognormal(mean=branch_mean, sd=branch_sd)
    moves[mvi++] = mvScale(branch_rate_multiplier[i])
    branch_rates[i] := rate_mol * branch_rate_multiplier[i]
}
```

Now we'll create an HKY rate matrix. First the transition-transversion rate ratio (with prior with mean=1)

```
kappa ~ dnGamma(2,2)
moves[mvi++] = mvScale(kappa)
```

the base frequencies over A, C, G, and T

```
bf ~ dnDirichlet([1,1,1,1])
moves[mvi++] = mvSimplexElementScale(bf, alpha=10, weight=2)
```

then using the base frequencies and TsTv rate ratio to build the matrix

```
Q_mol := fnHKY(kappa, bf)
```

Next, we'll create a +Γ4 across site rate variation model. This requires a parameter to control how much site rate heterogeneity there is.

```
alpha ~ dnUniform(0,50)
moves[mvi++] = mvScale(alpha)
```

and a discretized Gamma distribution with 4 categories

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

When `alpha` is large, then the Gamma distribution centers its density around the rate multiplier of 1, meaning that all sites evolve at similar rates. When `alpha` is small, the Gamma distribution presents more site rate heterogeneity.

Finally, we'll create our molecular model of substitution

```
seq_mol ~ dnPhyloCTMC(Q=Q_mol, tree=phy, branchRates=branch_rates, siteRates=site_rates,
    type="DNA", nSites=dat_mol.nchar())
```

and attach the ETS dataset

```
seq_mol.clamp(dat_mol)
```

4.0.3 The biogeographic model

Dispersal rates might make use of some extrinsic information, such as geographical distances between areas (**??**). We model this as $d_{ij} = ae^{-bg_{ij}}$ where g_{ij} is the geographical distance between areas i and j and a and b are parameters that scale distance on linear and exponential scales, respectively. Note that all dispersal rates equal a when $b = 0$.

```
dispersal_rate ~ dnExp(1)
dispersal_rate.setValue(0.1)
moves[mvi++] = mvScale(a)
b ~ dnUniform(-5,5)
b.setValue(0.01)
moves[mvi++] = mvSlide(b, delta=0.1)
```

Now we can assign rates that are functions of distance between all pairs of areas

```
for (i in 1:n_epochs) {
    for (j in 1:n_areas) {
        for (k in 1:n_areas) {
            dr[i][j][k] <- abs(0)
            if (connectivity[i][j][k] > 0) {
                dr[i][j][k] := a * exp( -b * distances[j][k] )
            }
        }
    }
}
```

It is unlikely that widespread ranges persist across disjunct areas for long periods of time. Extirpation is more likely to occur in fragmented ranges than well-connected ranges, where peripheral populations are continuously reinforced from the center.

```
e ~ dnExp(1)
moves[mvi++] = mvScale(e)
```

```
for (i in 1:n_epochs) {
    for (j in 1:n_areas) {
        for (k in 1:n_areas) {
            er[i][j][k] <- abs(0)
            if (j == k) {
                er[i][j][k] := e
            }
        }
    }
}
```

Treat epoch times as random variables. The present is always the present.

```
for (i in 1:n_epochs) {
    time_max[i] <- time_bounds[i][1]
    time_min[i] <- time_bounds[i][2]
    if (i != n_epochs) {
        epoch_times[i] ~ dnUniform(time_min[i], time_max[i])
        moves[mvi++] = mvSlide(epoch_times[i], delta=0.5)
    } else {
        epoch_times[i] <- 0.0
    }
}
```

Build a rate matrix for each time interval

```
for (i in 1:n_epochs) {
    Q[i] := fnDECRateMatrix(dispersalRates=connectivity[i],
                            extirpationRates=er,
                            maxRangeSize=max_areas)
}
```

Create the epoch rate generator object

```
Q_epoch := fnDECCladoProbs(
```

Here, we treat the probability of different types of cladogenetic events as a random variable to be estimate.

```
clado_event_types <- [ "s", "a" ]
clado_type_probs ~ dnDirichlet( [1,1] )
moves[++mi] = mvSimplexElementScale(clado_type_probs, alpha=10)
P_clado := fnDECCladoProbs(eventProbs=clado_type_probs,
                           eventTypes=clado_event_types,
                           numCharacters=n_areas,
                           maxRangeSize=max_areas)
```

For this dataset, we assume cladogenetic probabilities are constant with respect to geological time.

```
rate_bg <- 1
```

```
rf_DEC <- rep(0, n_states)
rf_DEC[2] <- 1
rf_DEC <- simplex(rf_DEC)
```

Create the phylogenetic model

```
ctmc_bg ~ dnCTMCClado(tree=phy,
                      Q=Q_DEC_epoch,
                      cladoProbs=P_DEC,
                      branchRates=rate_bg,
                      rootFrequencies=rf_DEC,
                      type="NaturalNumbers",
                      nSites=1)
```

Attach the dataset

```
ctmc_bg.clamp(dat_range_n)
```

And the rest we've done before...

```
# make the model
mdl = model(m)

# make the monitors
mn[1] = mnScreen(dispersal_rate, distance_scale, extirpation_rate, printgen=1000)
mn[2] = mnModel(file=out_fn+".params.txt", printgen=100)
```

```
mn[3] = mnJointConditionalAncestralState(tree=psi, ctmc=m, filename=out_fn+".states.txt
    ", type="NaturalNumbers", printgen=100, withTips=true, withStartStates=true)

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

Finally, let's look at the trees and posterior distribution.

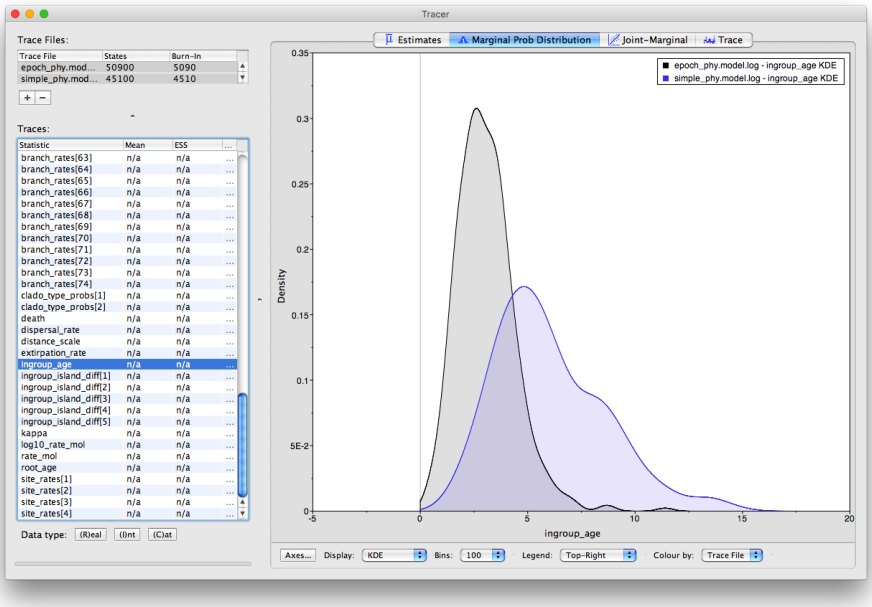

Figure 6: Using the paleogeographic model largely prefers that silverswords originated after the formation of Kauai. Without paleogeography, silverswords originated as long as 15 Ma ago.

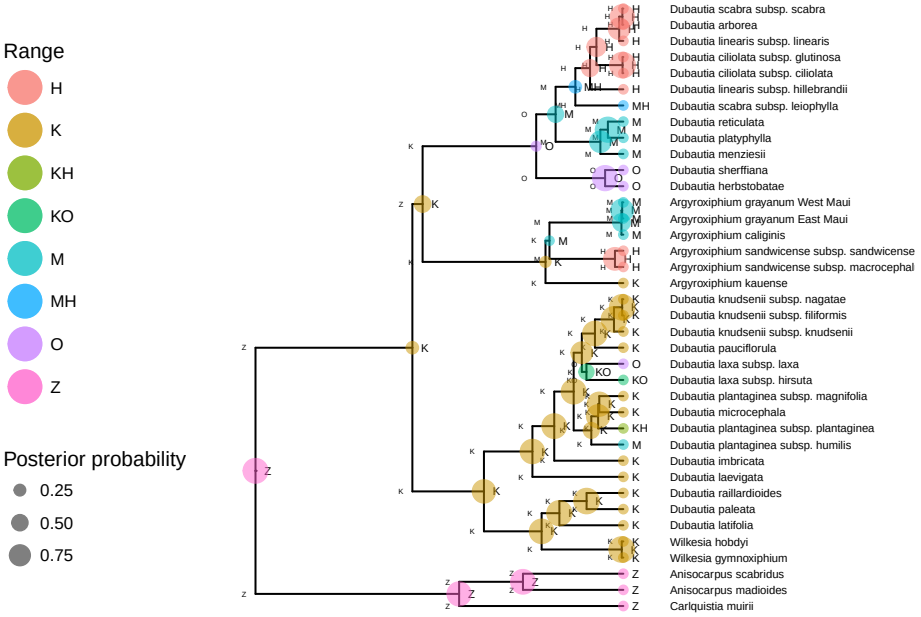

Figure 7: Joint estimate of phylogeny and biogeography while incorporating paleogeographic information.

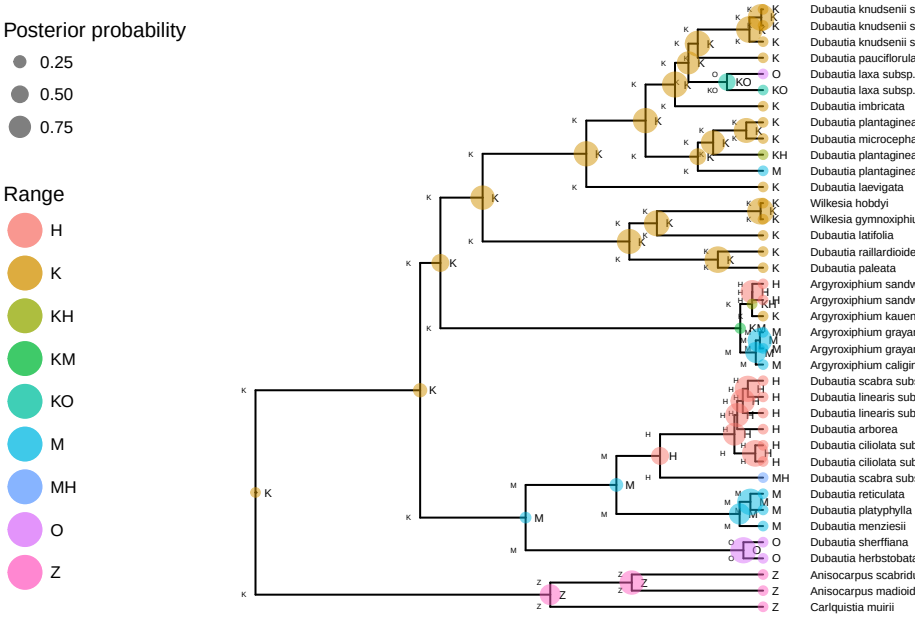

Figure 8: Joint estimate of phylogeny and biogeography ignoring paleogeographic information. This model has the tarweed+silversword clade originating in Kauai at a time when the island did not exist.

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.

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

Höhna, S. and A. J. Drummond. 2012. Guided Tree Topology Proposals for Bayesian Phylogenetic Inference. Systematic Biology 61:1–11.

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

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. 2012. Founder-event speciation in biogeobears package dramatically improves likelihoods and alters parameter inference in dispersal–extinction–cladogenesis dec analyses. Frontiers of Biogeography 4:210.

Nielsen, R. 2002. Mapping mutations on phylogenies. Systematic Biology 51:729–739.

Ree, R. H., B. R. Moore, C. O. Webb, M. J. Donoghue, and K. Crandall. 2005. A likelihood framework for inferring the evolution of geographic range on phylogenetic trees. Evolution 59:2299–2311.

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: December 16, 2016