Biology 559R: Introduction to Phylogenetic Comparative Methods

Topics for this week (Mar 3 & 5):

- Ancestral state reconstruction (continuous)
- Tree simulations
- Ancestral state reconstruction:
  Threshold models
Ancestral reconstruction using ‘geiger’

• For any comparative analyses we need the following

1) Our phylogeny (we got that)

2) Our data (we got that)

3) These data and phylogeny are concordant (we tested that already)

4) A model of evolutionary change or trait evolution along each branch of the phylogeny

• This last step is fundamental and this bring us find a way to decide how to choose the best possible model given our data and phylogeny.

• Similar to fitDiscrete, fitContinuous is a very flexible function of ‘geiger’. This method provides a set of statistics that allows us to decide which model best describes the trait that we want to reconstruct.
Ancestral reconstruction using ‘geiger’

• Let’s talk about some of the models implemented:

**Brownian model (BM):** This is the most basic model where trait evolves by small increments over time in a random way. At any given time, there is an equal probability for increase or decrease in this trait value.

Only upon an speciation event (tree node), each species starts evolving independently, starting from the same value at the node (MRCA).

\[ dX(t) = \sigma dB(t) \]

- \( dX(t) \) is the change in the character \( X \) over the time interval from \( t \) to \( t + dt \).
- \( dB(t) \) is "white noise“ a set of iid normal random variables with mean 0 and variance proportional to \( \sigma \)
- \( \sigma \) is a measure of the intensity of the random fluctuations in the evolutionary process.

**From Butler et al. 2008 (see website):** Applied to a phylogeny, the species are expected to covary in proportion to the amount of time they share in evolutionary history.

That is, they have only been evolving independently since they diverged from their most recent common ancestor. It is this covariance that methods such as independent contrasts, phylogenetic GLS, and other methods which assume BM seek to correct for.
Ancestral reconstruction using ‘geiger’

- Here is an example of a simulation of trait evolving under BM model for 10 taxa

```r
library(phytools)
tree<-pbtree(n=10)
bmPlot(tree)
```
Ancestral reconstruction using ‘geiger’

• Let’s talk about some of the models implemented:

**Ornstein-Uhlenbeck models (OU):** In this more complex model than BM and more real, the OU add extra element to the equation that allows to change the rate of the evolutionary process along branches of the tree by different adaptive regimes

\[ dX(t) = \alpha (\theta - X(t)) \, dt + \sigma \, dB(t) \]

\( dX(t) \) is the change in the character \( X \) over the time interval from \( t \) to \( t + dt \).
\( dB(t) \) is "white noise" a set of iid normal random variables with mean 0 and variance proportional to \( \sigma \)
\( \sigma \) is a measure of the intensity of the random fluctuations in the evolutionary process
\( \alpha \) measures the strength of selection, when \( \alpha \) is 0 then model is BM
Ancestral reconstruction using ‘geiger’

- Other models in geiger:
  
  **Early-burst model (EB):** This model determines where the rate of evolution increases or decreases exponentially through time.

  **White-noise model (non-phylogenetic):** This model assumes data come from a single normal distribution with no covariance structure among species.

  **Lambda based model:** Useful to test for phylogenetic signal using Pagel (1999) lambda that fits the extent to which the phylogeny predicts covariance among trait values for species.

  **Delta based model:** Useful to test if recent evolution has been comparatively slow or faster.
Ancestral reconstruction using ‘geiger’

• Other models in geiger:

**Early-burst model (EB):** This model determines where the rate of evolution increases or decreases exponentially through time.

**White-noise model (non-phylogenetic):** This model assumes data come from a single normal distribution with no covariance structure among species.

**Lambda based model:** Useful to test for phylogenetic signal using Pagel (1999) lambda that fits the extent to which the phylogeny predicts covariance among trait values for species.

**Delta based model:** Useful to test if recent evolution has been comparatively slow or faster.
Ancestral reconstruction using ‘geiger’

• We are going to prepare our data as before:

```r
log10_Mass_g_birds_geiger <- treedata(bird_pruned_tree, birds_traits_data)
log10_Mass_g_birds_geiger$phy # a class phylo
log10_Mass_g_birds_geiger$data # a matrix
```

• We are going to prepare our data as before, but we are going to use the modified code for the function that is present in the R_commands:

```r
phy <- log10_Mass_g_birds_geiger$phy
all_dat <- log10_Mass_g_birds_geiger$data
dat <- as.numeric(all_dat[,15])
names(dat) <- all_dat[,1]

Compare_models_fitContinuous: copy the function as is indicated below

• We run the function:

```r
compare_models_fitContinuous(phy, dat) # for log10_Mass_g
```
Ancestral reconstruction using ‘geiger’

• Here is the type of output that you will get:

# [1] "OK" ← The data is concordant.

# *** BROWNIAN MOTION -- BM: fitting BM with SE ***
#GEIGER-fitted comparative model of continuous data
# fitted ‘BM’ model parameters:
#   sigsq = 0.006036
#   SE = 0.000000
#   z0 = 2.814478

# model summary:
#   log-likelihood = -183.527424
#   AIC = 373.054848
#   AICc = 373.108539
#   free parameters = 3

#Convergence diagnostics:
#   optimization iterations = 100
#   failed iterations = 0
#   frequency of best fit = 0.31

# object summary:
#   'lik' -- likelihood function
#   'bnd' -- bounds for likelihood search
#   'res' -- optimization iteration summary
#   'opt' -- maximum likelihood parameter estimates
Ancestral reconstruction using ‘geiger’

- The model comparison based on AIC and summary statistics:

```r
# *** MODEL COMPARISON no measurement error -- log10_Mass_g ***
# delta-AIC values for models assuming no measurement error
# zero indicates the best model

#
# delta_aicc  BM   OU   EB   white   lambda   kappa   delta
# 59.282  59.092 61.31  719.142   0  6.169   60.808
# significance *** *** *** *** ns ** ***

# *** MODEL COMPARISON with measurement error -- log10_Mass_g ***
# delta-AIC values for models estimating SE
# zero indicates the best model

#
# delta_aicc.se  BM   OU   EB   white   lambda   kappa   delta
# 0.623   2.659 0.141  721.792   2.659  0.634        0
# significance_SE ns   ns   ns   ***   ns   ns       ns
```
### Ancestral reconstruction using ‘geiger’

- The model comparison based on AIC and summary statistics:

```r
# *** MODEL COMPARISON no measurement error --log10_Mass_g ***
# delta-AIC values for models assuming no measurement error
# zero indicates the best model
#
# delta_aicc      BM  OU  EB  white lambda  kappa  delta
# 59.282 59.092 61.31 719.142 0 6.169 60.808
# significance   *** *** *** ***  ns  **  ***

# *** MODEL COMPARISON with measurement error -- log10_Mass_g ***
# delta-AIC values for models estimating SE
# zero indicates the best model
#
# delta_aicc.se    BM  OU  EB  white lambda  kappa  delta
# 0.623 2.659 0.141 721.792 2.659 0.634 0
# significance_se ns  ns  ns  ***  ns  ns  ns

# AICc            371.0548 370.8381 373.0556 1030.9143 311.7458 317.9148 372.5542
# dAICc           59.282 59.092 61.31 719.142 0 6.169 60.808
# significance   *** *** *** ***  ns  **  ***
# AICc_SE         311.7458 313.7458 311.2271 1032.9143 313.7458 311.7201 311.0864
# dAICc_SE        0.623 2.659 0.141 721.792 2.659 0.634 0
# significance_SE ns  ns  ns  ***  ns  ns  ns
```
Ancestral reconstruction using ‘geiger’

- The model comparison based on AIC and summary statistics (parameters):

```plaintext
*** MODEL COMPARISON PARAMETERS -- without SE ***

<table>
<thead>
<tr>
<th></th>
<th>alpha</th>
<th>EB-a</th>
<th>lambda</th>
<th>kappa</th>
<th>delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM</td>
<td>------</td>
<td></td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>OU</td>
<td>0.0031</td>
<td></td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>EB</td>
<td>------</td>
<td>0</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>white</td>
<td>------</td>
<td></td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>lambda</td>
<td>------</td>
<td></td>
<td>0.9903</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>kappa</td>
<td>------</td>
<td></td>
<td>------</td>
<td>0.6987</td>
<td>------</td>
</tr>
<tr>
<td>delta</td>
<td>------</td>
<td></td>
<td>------</td>
<td>------</td>
<td>1.2214</td>
</tr>
</tbody>
</table>

*** MODEL COMPARISON PARAMETERS -- with SE ***

<table>
<thead>
<tr>
<th></th>
<th>alpha</th>
<th>EB-a</th>
<th>lambda</th>
<th>kappa</th>
<th>delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM</td>
<td>------</td>
<td></td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>OU</td>
<td>0</td>
<td></td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>EB</td>
<td>------</td>
<td>-0.0082</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>white</td>
<td>------</td>
<td></td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>lambda</td>
<td>------</td>
<td></td>
<td>0.9903</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>kappa</td>
<td>------</td>
<td></td>
<td>------</td>
<td>0.8636</td>
<td>------</td>
</tr>
<tr>
<td>delta</td>
<td>------</td>
<td></td>
<td>------</td>
<td>0.4981</td>
<td>------</td>
</tr>
</tbody>
</table>
```
Ancestral reconstruction using ‘geiger’

- The best fitting model was lambda assuming no measurement error

```r
# *** PAGEL'S LAMBDA WITH NO ME : fitting lambda ***

# GEIGER-fitted comparative model of continuous data
# fitted 'lambda' model parameters:
#   lambda = 0.990334  <- Notice the high lambda value
#   sigsq = 0.004739
#   z0 = 2.812349

# model summary:
#   log-likelihood = -152.872911
#   AIC = 311.745822
#   AICc = 311.799513
#   free parameters = 3

# Convergence diagnostics:
#   optimization iterations = 100
#   failed iterations = 0
#   frequency of best fit = 0.12

# object summary:
#   'lik' -- likelihood function
#   'bnd' -- bounds for likelihood search
#   'res' -- optimization iteration summary
#   'opt' -- maximum likelihood parameter estimates
```
Phylogenetic Signal for Continuous Characters

• We can use Pagel's lambda to assess the phylogenetic signal (tendency of closely related species to resemble each other for a specific trait).


• Lambda parameter is also defined as a tree transformation parameter that multiply the internal branches of the tree by values between 0 and 1.

A lambda of value equal or near 1 indicates strong phylogenetic signal (traits evolve in close association with the phylogeny)

A lambda of value equal or near 0 indicates no signal (traits are independent of the phylogeny)

• In consequence, we can determine the magnitude of the signal by estimating the lambda parameter in our tree and then compare the model fit using log-lik with that of model where the tree was transformed to a polytomy (i.e., branches multiplied by lambda = 0)

• This is similar to fitDiscrete procedure
• Let's estimate our lambda 0 tree

```r
bird_pruned_tree_lambda_0 <- rescale(bird_pruned_tree, model = "lambda", 0)
```

• Let's plot this tree and the original bird tree

```r
par(mfrow=c(1,2))
plot(bird_pruned_tree, edge.width = 1, show.tip.label = FALSE) #smaller tree
add.scale.bar(cex = 0.7, font = 2, col = "red")
plot(bird_pruned_tree_lambda_0, edge.width = 1, show.tip.label = FALSE) #smaller tree
add.scale.bar(cex = 0.7, font = 2, col = "red")
```
Phylogenetic Signal for Continuous Characters

Our tree      No signal (lambda = 0)
Phylogenetic Signal for Continuous Characters

• Let's fit the lambda transformation to determine if there is phylogenetic signal

```r
our_data_fitContinuous <- fitContinuous(phy,dat,SE=0, model="lambda",bounds=list(SE=c(0,0.5)))
our_data_fitContinuous

#GEIGER-fitted comparative model of continuous data
# fitted 'lambda' model parameters:
#   lambda = 0.990330
#   sigsq = 0.004739
#   z0 = 2.812348

# model summary:
#   log-likelihood = -152.872911
#   AIC = 313.745822
#   AICc = 313.835508
#   free parameters = 4

# Convergence diagnostics:
#   optimization iterations = 100
#   failed iterations = 0
#   frequency of best fit = 0.01

# object summary:
#   'lik' -- likelihood function
#   'bnd' -- bounds for likelihood search
#   'res' -- optimization iteration summary
#   'opt' -- maximum likelihood parameter estimates
```
Phylogenetic Signal for Continuous Characters

• Let's fit the lambda transformation to determine if there is signal

```r
# Fit the lambda 0 tree (no signal)

our_data_fitContinuous_lambda_0 <- fitContinuous(bird_pruned_tree_lambda_0,dat,SE=0,
model="lambda",bounds=list(SE=c(0,0.5)))
our_data_fitContinuous_lambda_0

# GEIGER-fitted comparative model of continuous data
# fitted 'lambda' model parameters:
#   lambda = 0.000000
#   sigsq = 0.004776
#   z0 = 1.938157

# model summary:
#   log-likelihood = -513.457139
#   AIC = 1034.914278
#   AICc = 1035.003965
#   free parameters = 4

# Convergence diagnostics:
#   optimization iterations = 100
#   failed iterations = 0
#   frequency of best fit = 0.69

# object summary:
#   'lik' -- likelihood function
#   'bnd' -- bounds for likelihood search
#   'res' -- optimization iteration summary
#   'opt' -- maximum likelihood parameter estimates
```
Phylogenetic Signal for Continuous Characters

• For model comparison, the model with the lowest AIC score is preferred

```
our_tree-versus_lambda_0_aicc <- abs(our_data_fitContinuous_lambda_0$opt$aicc - 
our_data_fitContinuous$opt$aicc)
our_tree-versus_lambda_0_aicc # 721.1685 -- lambda 0 tree (no signal) is very unlikely

##### our tree versus lambda 0

d_our_tree-versus_lambda_0 <- abs(2*(our_data_fitContinuous_lambda_0$opt$lnL-
our_data_fitContinuous$opt$lnL))
d_our_tree-versus_lambda_0

# [1] 721.1685

p_our_tree-versus_lambda_0 <- pchisq(d_our_tree-versus_lambda_0, 1, lower.tail=FALSE)
p_our_tree-versus_lambda_0

#[1] 7.457298e-159 -- Rejects no signal model
```
Tree Simulations

• Make a working directory (e.g., tree simulations) and select it as your working.

  setwd("/Users/jcsantos/Desktop/R_class_winter_2015_home/0_Bio559R_course_final_files/week_9_office/data")

• Several packages and functions are available for this purpose.

  library(ape)
  library(geiger)
  install.packages("TreeSim")
  library(TreeSim)

• Load our trees and data from last class and check concordance:

  bird_pruned_tree <- read.tree("bird_pruned_MB.newick")
  bird_pruned_tree
  birds_traits_data <- read.table("bird_data_pruned.txt", header = TRUE, sep = "\t")
  birds_traits_data

• Use the 'name.check' function to determine if we have concordance

  name.check (bird_pruned_tree, birds_traits_data) # we check the concordance between a data file and a phylogenetic tree.
  # [1] "OK" # we have concordance (i.e., we the same number of tips and data per tip)
Tree Simulations in ape

• Generates Random Trees. We are going to use our pruned tree as reference

bird_pruned_tree

# Phylogenetic tree with 451 tips and 450 internal nodes.
#
# Tip labels:
#   Nothoprocta_perdicaria, Apteryx_australis, Apteryx_haastii, Apteryx_owenii, Dromaius_novaehollandiae, Struthio_camelus, ...

# Rooted; includes branch lengths.

• We will use the following functions: rtree

# basic random trees:

# n is number of tips same number of our tree: length(bird_pruned_tree$tip.label)
# rooted a logical (TRUE or FALSE)
# tip.label provide a list of species names: bird_pruned_tree$tip.label

bird_pruned_random_tree <- rtree(n = length(bird_pruned_tree$tip.label), rooted = TRUE, tip.label = bird_pruned_tree$tip.label)

is.ultrametric(bird_pruned_random_tree) # [1] FALSE
• We will use the following functions: rcoal

• Random coalescent tree: branching times determined by coalescent process (i.e., ultrametric and nodes pushed towards tips)

```r
bird_pruned_random_coalescent_tree <- rcoal(n = length(bird_pruned_tree$tip.label), rooted = TRUE, tip.label = bird_pruned_tree$tip.label)
is.ultrametric(bird_pruned_random_coalescent_tree) # [1] TRUE
```

# generate for random ultrametric trees:

```r
for (i in 1:4) { # generate 4 random trees, save and append to tree file
tree <- rcoal(n = length(bird_pruned_tree$tip.label), rooted = TRUE, tip.label = bird_pruned_tree$tip.label)
write.tree(tree, file = "rcol_random_trees.tre", append = TRUE, digits = 10, tree.names = FALSE)
}
```
Tree Simulations in ape

- Plot trees simulated with rtree and rcoal

```r
par(mfrow=c(1,2))
plot(bird_pruned_random_tree, show.tip.label = FALSE)
title("Random tree")
add.scale.bar(cex = 0.7, font = 2, col = "red")
plot(bird_pruned_random_coalescent_tree, show.tip.label = FALSE)
title("Random coalescent tree")
add.scale.bar(cex = 0.7, font = 2, col = "red")
```
Tree Simulations in ape

• Generates Systematic Regular Trees using: 'stree'

```r
# tip.label provide a list of species names: bird_pruned_tree$tip.label

# n is number of tips same number of our tree: length(bird_pruned_tree$tip.label)
# type: “star” a star (or comb) tree with a single internal node.
# “balanced” a fully balanced dichotomous rooted tree; n must be a power of 2
# “left” a fully unbalanced rooted tree where the largest clade is on the left-hand side
# “right” a fully unbalanced rooted tree where the largest clade is on the right-hand side

random_fan_tree <- stree(length(bird_pruned_tree$tip.label), type = "star", tip.label = bird_pruned_tree$tip.label)
random_left_tree <- stree(length(bird_pruned_tree$tip.label), type = "left", tip.label = bird_pruned_tree$tip.label)

par(mfrow=c(1,2))
plot(random_fan_tree, show.tip.label = FALSE)
title("Random fan tree")
add.scale.bar(cex = 0.7, font = 2, col = "red")
plot(random_left_tree, show.tip.label = FALSE)
title("Random left tree")
add.scale.bar(cex = 0.7, font = 2, col = "red")
```
Tree Simulations in ape

- Generates Systematic Regular Trees using: 'stree'
Tree Simulations in geiger

• Tree Simulations Under the Time-Dependent Birth–Death Models: sim.bdtree

# sim.bdtree(b=1, d=0, stop=c("taxa", "time"), n=100, t=4, seed=0, extinct=TRUE)

# b per-lineage birth (speciation) rate
# d per-lineage death (extinction) rate

# stop provided stopping criterion
# n maximum number of taxa in simulation
# t maximum time steps of simulation
# seed random number seed (default is to seed based on the clock)
# extinct whether to return trees where all lineages have gone extinct (see Details)

# We will use 'drop.fossil' function from ape to drop extinct tips is a utility function
# to remove the extinct species.
• Plot tree simulations and plots

```r
# Yule process (pure-birth i.e. no extinction)
# birth rate: 0.1
# death rate: 0

simul_yule_process_lambda_0 <- sim.bdtree(b=0.1, d=0, stop=c("taxa"), n=length(bird_pruned_tree$tip.label), t=4, seed=0, extinct=TRUE)

simul_yule_process_lambda_0$tip.label <- bird_pruned_tree$tip.label # give the same names as in the bird_pruned_tree
plot(simul_yule_process_lambda_0, show.tip.label = FALSE)
title("Tree simulated under the pure-birth (Yule process)"

simul_yule_process_lambda_0

# Phylogenetic tree with 451 tips and 450 internal nodes. ← This is the same number of tips
Tree Simulations in geiger

• Plot tree simulations and plots

# Yule process (pure-birth i.e. no extinction)
# birth rate: 0.1
# death rate: 0
Tree Simulations in geiger

- Plot tree simulations and plots

```r
# simple birth-death process
# birth rate: 0.1
# death rate: 0.05

simul_birth_0_1_death_0_05 <- sim.bdtree(b=0.1, d=0.05, stop=c("taxa"), n=length(bird_pruned_tree$tip.label), t=4, seed=0, extinct=TRUE)
simul_birth_0_1_death_0_05

# Phylogenetic tree with 883 tips and 882 internal nodes. ← this way more than the tips that we have in our tree

simul_birth_0_1_death_0_05_no_extinct <- drop.fossil(simul_birth_0_1_death_0_05, tol = 1e-8)
simul_birth_0_1_death_0_05_no_extinct

# Phylogenetic tree with 160 tips and 159 internal nodes. ← number of extant taxa (no extinct), way lower

par(mfrow=c(1,2))
plot(simul_birth_0_1_death_0_05, show.tip.label = FALSE)
title("Tree simulated under the birth-death process")
add.scale.bar(cex = 0.7, font = 2, col = "red")
plot(simul_birth_0_1_death_0_05_no_extinct, show.tip.label = FALSE)
title("The same tree without extinct taxa")
add.scale.bar(cex = 0.7, font = 2, col = "red")
```
Tree Simulations in geiger

• Plot tree simulations and plots

```r
# simple birth-death process
# birth rate: 0.1
# death rate: 0.05
```

Tree simulated under the birth-death process

The same tree without extinct taxa
• Tree Simulations Under the Time-Dependent Birth–Death Models with a given age on a fixed number of extant taxa.

```r
# sim.bd.taxa.age(n, numbsim, lambda, mu, frac = 1, age, mrca = FALSE)
```

# n Number of extant sampled tips.
# numbsim Number of trees to simulate.
# lambda Speciation rate.
# mu Extinction rate.
# frac Each tip is included into the final tree with probability frac.
# age The time since origin / most recent common ancestor.
# mrca If mrca = FALSE: The time since the origin of the process.
# If mrca = TRUE: The time since the most recent common ancestor of the sampled species.

• We will use the following functions: rtree

```r
install.packages("phyclust")
library(phyclust)

tree_height <- get.rooted.tree.height(bird_pruned_tree)
tree_height
# [1] 119.4931  # We can use this value to rescale our tree later
```
• Tree simulation using: sim.bd.taxa.age

```r
tree_sim_bd <- sim.bd.taxa.age(length(bird_pruned_tree$tip.label), numbsim = 1, lambda = 0.1, mu = 0.05, frac = 1, age = tree_height, mrca = FALSE)
```

# It will return a list

```r
simul_birth_0_1_death_0_05_tree_sim <- tree_sim_bd [[1]]
simul_birth_0_1_death_0_05_tree_sim$tip.label <- bird_pruned_tree$tip.label # give the same names as in the bird_pruned_tree
simul_birth_0_1_death_0_05_tree_sim
```

# Phylogenetic tree with 451 tips and 450 internal nodes.

• Simulate multiple trees

# generate for random ultrametric trees:

```r
for (i in 1:4) { # generate 4 random trees, save and append to tree file
tree_sim_bd <- sim.bd.taxa.age(length(bird_pruned_tree$tip.label), numbsim = 1, lambda = 0.1, mu = 0.05, frac = 1, age = tree_height, mrca = FALSE)
simul_tree_sim <- tree_sim_bd [[1]]
write.tree(tree, file = "simul_birth_0_1_death_0_05_tree_sim_trees.tre", append = TRUE, digits = 10, tree.names = FALSE)
}
```
• Read the trees and plot the tree

```r
bird_pruned_tree <- read.tree("bird_pruned_MB.newick")
simul_birth_0_1_death_0_05_tree_sim_trees <- read.tree
("simul_birth_0_1_death_0_05_tree_sim_trees.tre")

# Plot the trees
par(mfrow=c(1,2))
plot(bird_pruned_tree, show.tip.label = FALSE)
title("Our bird_pruned_tree")
add.scale.bar(cex = 0.7, font = 2, col = "red")
plot(simul_birth_0_1_death_0_05_tree_sim_trees[[1]], show.tip.label = FALSE)
title("One of simulated under the birth-death process with TreeSim")
add.scale.bar(cex = 0.7, font = 2, col = "red")
```
Tree Simulations in TreeSim

- Read the trees and plot the tree
  
  Our bird_pruned_tree

One of simulated under the birth-death process with TreeSim
Ancestral Reconstruction using Threshold Models

- We already review some methods for character reconstruction such as Pagel (1994). This method allows inference of whether change is correlated, on a known tree.

- A newer method is based on the threshold model invented by Sewall Wright (1934).

- This model based on the idea that any discrete (or continuous) character is actually based on an underlying, unobserved “master” continuous trait called “liability” (Felsenstein 2012).

- The threshold model refers to liability exceeding a fixed threshold (a scaleless value, given that we have not observed or measured the liability trait) and then the character governed by liability will change character states or values.
What is Liability?

• Liability is a latent (unobserved) and scaleless variable that govern the change of the observed variables of our analyses, such as:

1) A concrete variable -- not measured: for example, hormone level (L. Revell example) and threshold level of blood hormone, when exceeded, could cause, the expression or repression of a discrete character

2) A proxy for the many and varied changes in organismal genotype and phenotype (complex phenotype) that underlie, for instance, a discrete shift in habitat use or life history.

• Changes in liability take place on a continuous trait axis, and for a lineage initially in state A to change to state B will require an underlying specific amount of evolutionary change of the underlying liability before or in parallel that will surpass a threshold that is expressed in the state change from A to B

• This predicts a period of polymorphism in a lineage as populations progresses in liability space to cross a threshold back and for over a period of time

• However, the model is problematic if character ordering is uncertain.
Ancestral Reconstruction using Threshold Models

- This figure illustrates the relationship between liability and an observed trait evolution on a 40-species phylogenetic tree under the threshold model. (From Revell, 2014; see course website):

Y-axis gives time since the root of the tree,
Color is the observed threshold character
X-axis gives our (unobserved) liability trait

Change in liability:
-2.0 \rightarrow 0.0 cause a change in state from \textbf{black} to \textbf{red}
0.0 \rightarrow 3.0 causes a change in state from \textbf{red} to \textbf{blue}
0.0 \rightarrow 3.5, causes a change in state from \textbf{red} to \textbf{green}
Ancestral Reconstruction using Threshold Models

- This figure illustrates the relationship between liability and an observed trait evolution on a 40-species phylogenetic tree under the threshold model. (From Revell, 2014; see course website):

Y-axis gives time since the root of the tree,
Color is the observed threshold character
X-axis gives our (unobserved) liability trait

Change in liability:
-2.0 $\rightarrow$ 0.0 cause a change in state from **black** to **red**
0.0 $\rightarrow$ 3.0 causes a change in state from **red** to **blue**
0.0 $\rightarrow$ 3.5,causes a change in state from **red** to **green**