Biology 559R: Introduction to Phylogenetic Comparative Methods

Topics for this week (Feb 24 & 26):

- Ancestral state reconstruction (discrete)
- Ancestral state reconstruction (continuous)
Testing if rates have increased/decreased Discrete Characters

• We can test whether rates of change have increased or slowed over evolutionary time using 'fitDiscrete'.

• For this purpose, we use the delta model.

The delta model fits the relative contributions of early versus late evolution of a character state in our tree to the covariance of species trait values.

If delta > 1, recent evolution has been relatively fast
If delta < 1, recent evolution has been relatively slow

If rates decrease over time it is a signature of adaptive radiation if the traits are ecologically relevant.

```r
food_bird_ace_ARD_geiger_delta <- fitDiscrete(bird_pruned_tree, Food_birds_geiger$data [,13], type="discrete", model = "ARD", transform = "delta")
```
Testing if rates have increased/decreased Discrete Characters

• Let’s explore the results

```r
food_bird_ace_ARD_geiger_delta <- fitDiscrete(bird_pruned_tree, Food_birds_geiger$data [,13],
type="discrete", model = "ARD", transform = "delta")
```

#GEIGER-fitted comparative model of discrete data
# fitted Q matrix:
#       carnivore   omnivore   vegetarian
# carnivore   -0.0018102415  0.001810241  2.114637e-38
# omnivore    0.0120647007 -0.022792976  1.072828e-02
# vegetarian  0.0004968849  0.003132820 -3.629705e-03
#
# fitted ‘delta’ model parameter:
#       delta = 1.691392
#
# model summary:
# log-likelihood = -303.756563
# AIC = 621.513126
# AICc = 621.765947
# free parameters = 7
#
# Convergence diagnostics:
# optimization iterations = 100
# failed iterations = 0
# frequency of best fit = 0.01
Testing if rates have increased/decreased Discrete Characters

• We compare the AICc values

```
# AICc = 621.765947 for delta
# AICc = 654.756994 for equal rates
```

# For model comparison, the model with the lowest AIC score is preferred
# difference in AICc is 32.991047. Then delta model is supported

# fitted ‘delta’ model parameter:
#   delta = 1.691392

The delta > 1 suggest that recent evolution has been relatively fast
Exploring Continuous Characters

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

R top menus: Misc>Change Working Directory (select the directory that has our file)

    setwd("/Users/jcsantos/Desktop/R_class_winter_2015_home/0_Bio559R_course_final_files/week_8/data/")

• We need to download these files from the course website into our working directory:

bird_data_pruned.txt
bird_pruned_MB.newick
tyrannidae_reduced_data.txt
tyrannidae_reduced_tree.newick

• read 'ape' and 'geiger' package

    library(ape)
    library(geiger)
Exploring Continuous Characters

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

```r
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
```

using the 'read.tree' function of ape:

```r
bird_trees <- read.tree(file = "bird_tree.tree")
bird_trees #5 trees in loaded
bird_trees [[1]] # first tree a very large tree
```

```r
bird_tree_1 <- bird_trees [[1]] # assign first tree to object class phylo
```

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

```r
name.check (bird_pruned_tree, birds_traits_data)
```

# [1] "OK" # we have concordance (i.e., we the same number of tips and data per tip)
Exploring Continuous Characters

• We need to make sure that our tree is ultrametric tree and preferably bifurcating

is.ultrametric(bird_pruned_tree) # TRUE

is.binary.tree(bird_pruned_tree)
# TRUE all nodes (including the root node) have exactly two descendant nodes
Let's explore our continuous data for both all birds by Order

```r
library(ggplot2)

ggplot(birds_traits_data, aes(x = Order, y = Mass_g/1000, fill = Order)) + geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Aves -- Order", y = "Body Mass (kg)", fill = NULL)
```
Exploring Continuous Characters

- Let's explore our continuous data for both all birds by Order

```r
ggplot(birds_traits_data, aes(x = Order, y = log10(Mass_g), fill = Order)) + geom_boxplot() + theme(axis.text.x = element_text(angle = 90, hjust = 1)) + labs(x = "Aves -- Order", y = "Log10 Body Mass (g)", fill = NULL)
```
Let's explore our continuous data for both all birds by Family

ggplot(birds_traits_data, aes(x = Family, y = Mass_g/1000, fill = Family)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Aves -- Family", y = "Body Mass (kg)", fill = NULL)
Let's explore our continuous data for both all birds by Family

ggplot(birds_traits_data, aes(x = Family, y = \text{log10}(\text{Mass}_g), fill = Family)) + geom_boxplot() +
theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
labs(x = "Aves -- Family", y = "Log10 Body Mass (g)", fill = NULL)
Exploring Continuous Characters

• It is apparent that at the Aves level the data is not normal

```r
# Let's test for normality

shapiro.test(birds_traits_data$Mass_g)# Aves level

#    Shapiro-Wilk normality test
#    data:  birds_traits_data$Mass_g
#    W = 0.1023, p-value < 2.2e-16
```

• Let's create a variable that has the log10 transformed body mass data

```r
birds_traits_data$log10_Mass_g <- log10(birds_traits_data$Mass_g)
head(birds_traits_data)

shapiro.test(birds_traits_data$log10_Mass_g)# Aves level

#    Shapiro-Wilk normality test
#    data:  birds_traits_data$log10_Mass_g
#    W = 0.9479, p-value = 1.69e-11
```

• Let's also test using QQ plots (Quantile-Quantile Plots)
Exploring Continuous Characters

* Let's plot the QQ plots, Quantile-Quantile Plots

```r
install.packages("nortest") # test for normality
library(nortest)

par(mfrow=c(1,2))
qqnorm(birds_traits_data$Mass_g, main = "Normal Q-Q Plot -- Aves -- Body Mass")
qqnorm(birds_traits_data$log10_Mass_g, main = "Normal Q-Q Plot -- Aves -- log10 Body Mass")
```

![QQ plots](image)
Exploring Continuous Characters

- Let's plot the QQ plots, Quantile-Quantile Plots

  The log transformation has improved the distribution. It is not normal, but given the large sample size, it is not that important for regression analyses.

  Other transformations are possible (specially BoxCox), but the interpretation of such transformed data will be even harder. It will not change the results if we use robust statistical methods.

Let’s plot the QQ plots, Quantile-Quantile Plots

The log transformation has improved the distribution. It is not normal, but given the large sample size, it is not that important for regression analyses.

Other transformations are possible (specially BoxCox), but the interpretation of such transformed data will be even harder. It will not change the results if we use robust statistical methods.

Exploring Continuous Characters

• Let’s plot data for bird tree for continuous variable

```r
# We need to sort variable values to match phylogeny names order

match_birds_data <- match(bird_pruned_tree$tip.label, rownames(birds_traits_data)) # use match function
sorted_birds_data <- as.data.frame(birds_traits_data[match_birds_data,])

# print phylogeny next to transform data

matrix_graph <- matrix(c(1,2),1) # matrix to locate plots, plot 1 -- left; plot 2 -- right
layout(matrix_graph, c(0.6,0.4))
par(mar=c(4.1,0,1,1)) #A numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin
plot(bird_pruned_tree, show.tip.label=FALSE) #Plot a tree that leaves some room between the tree tips and taxon labels so that we can plot habitat use in this space
par(mar=c(4.1,1,1,1)) #A numerical vector of the form c(bottom, left, top, right) which gives the number of lines of margin
barplot(sorted_birds_data$log10_Mass_g, horiz=TRUE, width=1, space=0, names="")
```
Exploring Continuous Characters

- Let’s plot data for bird tree for continuous variable
Ancestral reconstruction using ‘ape’

• The ‘ace’ allows to fit models for continuous data. This function returns parameter estimates and the likelihood of the model for a single trait (univariate) datasets.

```r
# We will use the ace (Ancestral Character Estimation) function of ape to reconstruct ancestral character states using maximum likelihood:

# ace(x, phy, type = "continuous",

# method = "REML" # first estimates the ancestral value at the root (the phylogenetic mean), then the variance of the Brownian motion process is estimated # by optimizing the residual log-likelihood. The ancestral values are # finally inferred from the likelihood function giving these two parameters.

# method = "pic" # the confidence intervals are computed using the expected variances under the model, so they depend only on the tree.

# method = "GLS" # similar to "pic"
```

```r
#    CI = TRUE,     # 95% confidence intervals
#    model = "BM",     # Only Brownian motion option
#    scaled = TRUE,
#    kappa = 1,           # if branch lengths needs transformation,
#    corStruct = NULL,
#    ip = 0.1,
#    use.expm = FALSE,
#    use.eigen = TRUE,
#    marginal = FALSE) # returns the node marginal reconstruction (
Ancestral reconstruction using ‘ape’

• Let’s explore the model "REML”. This is the Brownian motion-based maximum likelihood (ML) estimator of Schluter et. al. (1997). We use REML to provide unbiased estimates (compared to "ML") of the variance of the Brownian motion process

```r
log10_Mass_bird_ace_REML <- ace(birds_traits_data$log10_Mass_g, bird_pruned_tree,
  type="continuous", model = "REML", marginal = TRUE)
log10_Mass_bird_ace_REML
#
# Ancestral Character Estimation
# Call: ace(x = birds_traits_data$log10_Mass_g, phy = bird_pruned_tree,
#   type = "continuous", model = "REML", marginal = TRUE)
#
# Residual log-likelihood: 1757.884 ← The log-likelihood of the most likely reconstruction
#
# $ace ← ace These form a vector of reconstructed node values
#
# 452 453 454 455 456 457 458 459 460
# 3.0204793 3.2499122 3.0810056 3.3392881 3.7115866 3.5490073 2.9503006 2.9305053 2.6328168
# ...
#
# $sigma2 ← A two element vector of the rate estimate and its standard error
# [1] 0.0094210721 0.0006407525
#
# $CI95 ← The 95% confidence intervals around the vector of reconstructed node values
# [,1] [,2]
# 452 2.1130802 3.927878
```
Ancestral reconstruction using ‘ape’

- Let's explore the method "pic". This is also a Brownian-motion based model used Felsenstein's (1985) phylogenetic independent contrasts (pic). However, it only takes descendants of each node into account in reconstructing the state at that node (i.e., more basal nodes are ignored).

```r
log10_Mass_bird_ace_pic <- ace(birds_traits_data$log10_Mass_g, bird_pruned_tree, type="continuous", model = "pic", marginal = TRUE)
log10_Mass_bird_ace_pic

# Ancestral Character Estimation
# Call: ace(x = birds_traits_data$log10_Mass_g, phy = bird_pruned_tree,
#          type = "continuous", model = "pic", marginal = TRUE)
#
# Residual log-likelihood: 1757.884 ← The log-likelihood of the most likely reconstruction
#
# $ace ← ace These form a vector of reconstructed node values
#
#   452 453 454 455 456 457 458 459 460
# 3.0204793 3.2499122 3.0810056 3.3392881 3.7115866 3.5490073 2.9503006 2.9305053 2.6328168
#
# $sigma2 ← A two element vector of the rate estimate and its standard error
# [1] 0.0094210721 0.0006407525
#
# $CI95 ← The 95% confidence intervals around the vector of reconstructed node values
#      [,1]     [,2]
# 452 2.1130802 3.927878
```
Ancestral reconstruction using ‘ape’

• Let's plot our reconstructions in our tree

plot(bird_pruned_tree, type = "fan", show.tip.label=FALSE)
tiplabels(pch = 21, cex=birds_traits_data$log10_Mass_g/2)
nodelabels(pch = 21, cex=log10_Mass_bird_ace_pic$ace/2)
axisPhylo()
Ancestral reconstruction using ‘phytools’

- For this package, we need to isolated our trait of interest and then give the corresponding names in the phylogeny

```r
birds_traits_data_log10_Mass_g <- birds_traits_data$log10_Mass_g
names(birds_traits_data_log10_Mass_g) <- birds_traits_data$Genus_Species
```

- Let's try the reconstructions using phytools.

```r
# fastAnc (from the manual)
#
# This function performs fast estimation of the ML ancestral states for a continuous trait
# by taking advantage of the fact that the state computed for the root node of the tree
# during Felsenstein's (1985) contrasts algorithm is also the maximum-likelihood estimation (MLE)
# of the root node. Thus, the function reroots the tree at all internal nodes and computes
# the contrasts state at the root each time.
```
Ancestral reconstruction using ‘phytools’

• Let's try the reconstructions using phytools.

```r
log10_Mass_bird_fastAnc <- fastAnc(bird_pruned_tree, birds_traits_data_log10_Mass_g, CI = TRUE)
log10_Mass_bird_fastAnc

# $ace ← ancestral state estimates
#      452       453       454       455       456       457       458       459       460
# 461  462  463  464  465  466  467  468
# 2.8144783 3.2564965 3.8268806 3.8844212 3.3854237 3.3461974 2.4683799 2.5775496 2.4386633
# 2.1295003 2.5220832 2.4198142 2.1598356 2.0405282 2.0243548 2.7157835 2.7011164
# ...
#
# $CI95 ← Upper and lower 95-percent confidence intervals
# [,1] [,2]
# 452 2.0892972 3.539659
# 453 2.5456000 3.967393
# 454 3.1860535 4.467708
# ...
```
Ancestral reconstruction using ‘phytools’

- Let's plot the fastAnc reconstructions on our tree.

**Notice:** You need to get the ‘contMap’ object first before changing to "fan" type phylogeny plot and it is particularly less flexible,

```r
log10_Mass_bird_fastAnc_contMap <- contMap(bird_pruned_tree, birds_traits_data_log10_Mass_g, spread.labels = TRUE)
plot(log10_Mass_bird_fastAnc_contMap, type = "fan", show.tip.label=FALSE)
```
Ancestral reconstruction using ‘phytools’

• Let’s try the reconstructions using ‘anc.ML’ function of phytools.

```
# anc.ML (from the manual)
#
# This function estimates the evolutionary parameters and ancestral states for Brownian
# evolution using likelihood. It is also possible (for model="BM") to allow for missing data
# for some tip taxa.

log10_Mass_bird_ancML_BM <- anc.ML(bird_pruned_tree, birds_traits_data_log10_Mass_g,
model="BM")
log10_Mass_bird_ancML_BM

# you can also use the OU model if supported (Ornstein–Uhlenbeck process – more on this later
# in geiger)

log10_Mass_bird_ancML_OU <- anc.ML(bird_pruned_tree, birds_traits_data_log10_Mass_g,
model="OU")
log10_Mass_bird_ancML_OU
```
Ancestral reconstruction using ‘phytools’

- Let's try the reconstructions using ‘anc.ML’ function of phytools under the BM model

```r
# The results should be similar to the fastAnc

# $sig2
# 0.003024751

# $ace
#       452       453       454       455       456       457       458       459       460
# 2.8144783 3.2564965 3.8268806 3.8844212 3.3854237 3.3461974 2.4683799 2.5775496 2.4386633
# ...

# $logLik
# [1] 522.9801

# $counts
# function gradient
#       21       21

# $convergence
# [1] 52

# $message
# [1] "ERROR: ABNORMAL_TERMINATION_IN_LNSRCH"

# $model
# [1] "BM"
# attr("class")
# [1] "anc.ML"
```
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 i.i.d 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)
```
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 i.i.d 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.
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**

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:

```r
# *** 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: 1 ***
# delta-AIC values for models assuming no measurement error
# zero indicates the best model
#
# BM   OU   EB    white    lambda    kappa
# daic 59.282 59.092 61.31 719.142 0 6.169
# *** *** *** *** ***
#
# *** MODEL COMPARISON: 1 ***
# delta-AIC values for models estimating SE
# zero indicates the best model
#
# BM   OU   EB    white    lambda    kappa
# daic.se 61.299 61.118 63.336 721.158 2.026 0
# *** *** *** *** ***

# AICc 371.081 370.891 373.109 1030.941 311.799 317.968
# AICc_SE 373.108 372.927 375.145 1032.968 313.836 311.810
# dAICc 59.282 59.092 61.31 719.142 0 6.169
# significance *** *** *** *** ***
# dAICc_SE 61.299 61.118 63.336 721.158 2.026 0
# significance *** *** *** *** ***
```
### Ancestral reconstruction using ‘geiger’

- The model comparison based on AIC and summary statistics:

```r
# *** MODEL COMPARISON: 1 ***
# delta-AIC values for models assuming no measurement error
# zero indicates the best model
#
#   BM       OU       EB    white    lambda    kappa
# daic 59.282 59.092 61.31 719.142    0    6.169
#   ***      ***      ***      ***      **
#
# *** MODEL COMPARISON: 1 ***
# delta-AIC values for models estimating SE
# zero indicates the best model
#
#   BM       OU       EB    white    lambda    kappa
# daic.se 61.299 61.118 63.336 721.158 2.026    0
#   ***      ***      ***      ***      **
```

### Model Comparison:

- **AICc**
  - `BM`: 371.081
  - `OU`: 370.891
  - `EB`: 373.109
  - `white`: 1030.941
  - `lambda`: 311.799
  - `kappa`: 317.968

- **AICc_SE**
  - `BM`: 373.108
  - `OU`: 372.927
  - `EB`: 375.145
  - `white`: 1032.968
  - `lambda`: 313.836
  - `kappa`: 311.810

- **dAICc**
  - `BM`: 59.282
  - `OU`: 59.092
  - `EB`: 61.31
  - `white`: 719.142
  - `lambda`: 0
  - `kappa`: 6.169

- **dAICc_SE**
  - `BM`: 61.299
  - `OU`: 61.118
  - `EB`: 63.336
  - `white`: 721.158
  - `lambda`: 2.026
  - `kappa`: 0
```````
Ancestral reconstruction using ‘geiger’

- The best fitting model was lambda

```r
# *** PAGEL'S LAMBDA: fitting lambda with SE ***
#GEIGER-fitted comparative model of continuous data
# fitted ‘lambda’ model parameters:
#  lambda = 0.990330
#  sigsq = 0.004739
#  SE = 0.000000
#  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

• We can use Pagel's lambda to assesses 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)
Phylogenetic Signal for Continuous Characters

• 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

our_data_fitContinuous <- fitContinuous(phy, dat, SE = NA, 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
# SE = 0.000000
# 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 = NA, 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
# SE = 0.000000
# 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

```r
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] 379.3937

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
```