Lab 5 R Source Code

Code:

```r
#
# Shawn Schwartz, 2019
# EEB C174 UCLA Spring 2019
# Lab 5 HW - Evolution of Grunts
#

rm(list=ls())

#imports
library(phytools)
library(geiger)
library(ape)

cwd <- "~/Developer/EEB-C174-Labs/Lab5"
resources_path <- "resources/"
output_path <- "output/"

setwd(cwd)

#### read in tree and data ####
gruntTree <- read.tree(paste0(resources_path,"grunts.tre"))
gruntTree <- ladderize(gruntTree)
plot(gruntTree, cex = 0.4)
axisPhylo()

gruntData <- read.csv(paste0(resources_path,"grunts.csv"), stringsAsFactors = F)
gruntData

cbind(gruntData$species, gruntTree$tip.label)
rownames(gruntData) <- gruntData$species
gruntData <- gruntData[match(gruntTree$tip.label, rownames(gruntData)),]
cbind(gruntData$species, gruntTree$tip.label)

Habitat <- gruntData$habitat
names(Habitat) <- gruntData$species
Habitat <- as.factor(Habitat)
```

```
Habitat

fitER <- fitDiscrete(phy = gruntTree, dat = Habitat, model = "ER")
fitARD <- fitDiscrete(phy = gruntTree, dat = Habitat, model = "ARD")

fitER
fitARD

#select model using AIC
reefs <- c(fitER$opt$aic, fitARD$opt$aic)
names(reefs) <- c("Equal Rates", "All Rates Different")
reefs

##Marginal ancestral statereconstruction
marginal_ER_fit <- rerootingMethod(tree = gruntTree, x = Habitat, model = "ER")
marginal_ARD_fit <- rerootingMethod(tree = gruntTree, x = Habitat, model = "ARD")

#Plot the estimated marginal ancestral states on the tree
plot(gruntTree, show.tip.label = F)
tiplabels(gruntTree$tip.label, adj = -0.5, frame = "none")
nodelabels(node = as.numeric(rownames(marginal_ER_fit$marginal.anc)), pie = marginal_ER_fit$marginal.anc,
           piecol = c("black", "red"), cex = 0.6)
tiplabels(pie = to.matrix(Habitat, sort(unique(Habitat))), piecol = c("black", "red"), cex = 0.3)

plot(gruntTree, show.tip.label = F)
tiplabels(gruntTree$tip.label, adj = -0.5, frame = "none")
nodelabels(node = as.numeric(rownames(marginal_ARD_fit$marginal.anc)), pie = marginal_ARD_fit$marginal.anc,
           piecol = c("black", "red"), cex = 0.6)
tiplabels(pie = to.matrix(Habitat, sort(unique(Habitat))), piecol = c("black", "red"), cex = 0.3)

#### Stochastic Character Mapping ####
mtrees <- make.simmap(tree = gruntTree, x = Habitat, model = "ARD", nsim = 100)
ER_mtrees <- make.simmap(tree = gruntTree, x = Habitat, model = "ER", nsim = 100)
```

```r
cols <- setNames(object = palette()[1:length(unique(Habitat))], nm = sort(unique(Habitat)))
cols

par(mfrow = c(10,10))
null <- sapply(X = mtrees, FUN = plotSimmap, colors = cols, lwd = 1, ftype = "off")
null <- sapply(X = ER_mtrees, FUN = plotSimmap, colors = cols, lwd = 1, ftype = "off")


pd <- describe.simmap(tree = mtrees, plot = FALSE)
pd

par(mfrow = c(1,1))
plot(pd)
add.simmap.legend(colors = cols, prompt = F, x = 0, y = 2, fsize = 0.8)
densityMap(mtrees)
```