*### Essential AA LDA Demo*

*# start by loading the necessary libraries*

*library(MASS)*

*library(stats)*

*library(car)*

*library(ellipse)*

*library(ggplot2)*

*library(ggnewscale)*

*# lets make sure everything has loaded correctly*

*# running these lines of code should not produce any error messages*

*?lda()*

*?predict()*

*?ellipse()*

*?shapiro.test()*

*?leveneTest()*

*# set your working directory*

*setwd("/Users/AlexiBesser/Desktop")*

*# read in your data!*

*# The producer file will be entitled “Producer\_Library.csv”*

*# read in the producer data csv file and name it "prod" within R*

*prod <- read.csv("Producer\_Library.csv")*

# let’s also make sure all columns are in the right format.

# All AA d13C data should be numeric, all else characters "chr"

str(prod)

*# let's practice subsetting our data by creating separate dataframes.*

*# Note that if you DO NOT want data for all 12 amino acids, you will need to # specify which variables you want to pull out.*

*# examples for a single group:*

*kelp <- subset(prod,*

*Type == "Kelp",*

*select = c("ID", "Ile13C", "Leu13C", "Lys13C", "Phe13C”))*

*fito <- subset(prod,*

*Type == "Marine\_POM",*

*select = c("ID", "Ile13C", "Leu13C", "Lys13C", "Phe13C”))*

*green <- subset(prod,*

*Type == "Marine\_green\_algae",*

*select = c("ID", "Ile13C", "Leu13C", "Lys13C", "Phe13C”))*

# We can also merge these back together

kelp\_fito <- rbind(fito, *kelp*)

*# we can also do this with multiple groups at one time:*

*kelp\_fito\_C3 <- subset(prod,*

*Type == "Kelp" | Type == "Marine\_POM" | Type == "C3 Plants",*

*select = c("ID", "Ile13C", "Leu13C", "Lys13C", "Phe13C"))*

*# Before running an LDA, it might be helpful to calculate summary statistics*

*# Here we calculate the mean and standard deviations for a few essential AA*

*kelp.ile.mean <- mean(kelp$Ile13C) # for Isoleucine*

*kelp.ile.sd <- sd(kelp$Ile13C)*

*fito.ile.mean <- mean(fito$Ile13C)*

*fito.ile.sd <- sd(fito$Ile13C)*

# NOTE --- WE ARE GOING TO SKIP CHECKING ASSUMPTIONS OF NORMALITY AND

# HOMOGENEITY OF VARIANCE.

# BUT YOU WILL NEED TO DO THIS IF YOU RUN THESE MODELS WITH YOUR OWN DATA.

# EXAMPLE CODE PROVIDED AT THE BOTTOM OF THIS DOCUMENT

*# make and save pdfs of box plots of producer Ile d13C values*

*pdf("ProducerIled13CBoxplot.pdf", width = 12, height = 8)*

*boxplot(kelp$Ile13C,*

*fito$Ile13C,*

*green$Ile13C,*

*names = c("Kelp", "POM", "Green"),*

*xlab = "Primary Producer", ylab = "Ile d13C")*

*dev.off()*

*# Okay! So now let's try our LDA model*

*# we'll start by creating an LDA with our producer essential AA data*

*prod\_LDA <- lda(Type ~ Ile13C + Lys13C + Val13C + Thr13C + Phe13C + Leu13C, data = prod)*

*# let's look at which AA are driving the patterns!*

*prod\_LDA*

*# A couple of important notes about this output:*

*# (1) the 'Prior probabilities of groups' represents how many individuals of # each species are present (e.g., 26% of our samples are kelps)*

*# (2) the* ***'Coefficients of linear discriminants'*** *gives you information on how # important each AA is for*

*# distinguishing among groups along each linear discriminant axis –*

*#* ***Larger absolute values mean those AA are important in driving that LD***

*# (3) the 'Proportion of trace' tells you how much variation is explained by # each LD axis*

*# now, let's calculate an error rate for our producer data set*

*# this is done with a cross validation approach*

*# in this function, the line 'CV = TRUE' makes the LDA perform a*

*# jacknifed (leave one out) model fit*

*prod\_LDA\_error <- lda(Type ~ Ile13C + Lys13C + Val13C + Thr13C + Phe13C + Leu13C, data = prod, CV = TRUE)*

*# Now let's create a table which compares the classification of the LDA model # to the actual species*

*# prod$Type is original grouping, prod\_LDA\_error$class is the LDA assignment*

*ct.prod <- table(prod$Type, prod\_LDA\_error$class)*

*ct.prod*

*# the total percent of samples correctly classified is the sum of the diagonal of this table*

*sum(diag(prop.table(ct.prod)))*

*# this line of code tells us what % of each group is correctly classified*

*diag(prop.table(ct.prod, 1))*

*# Let's plot these data!!!*

*# To do this, we need to export the new values of our individuals along the*

*# LD coordinates*

*# we could calculate this ourselves using the information in the LDA function*

*# but the predict() function in base R will do this for us! Woohoo!*

*# here we create a dataframe which contains these coordinates*

*datPred <- data.frame(Group = prod$Type, predict(prod\_LDA)$x)*

*as.factor(datPred$Group)*

*str(datPred)*

*# PLOTTING THE DATA*

*ggplot(datPred, aes(x = LD1, y = LD2, color = Group)) +*

*geom\_point(size = 6) +*

*theme\_classic() +*

*stat\_ellipse(geom="path", aes(color = Group),*

*alpha = 1,*

*show.legend = FALSE,*

*level = 0.95, size =1)*

*# Let’s adjust the colors!*

*ggplot(datPred, aes(x = LD1, y = LD2, color = Group)) +*

*geom\_point(size = 6) +*

*theme\_classic() +*

*scale\_color\_manual(values = c('#56B4E9', '#009E73', '#999999', '#D55E00',*

*'#000000', '#FFC000', '#0072B2', '#CC79A7', ‘purple’))+*

*stat\_ellipse(geom="path", aes(color = Group),*

*alpha = 1,*

*show.legend = FALSE,*

*level = 0.95, size =1)*

*ggsave("LDADemoEE.pdf", plot = last\_plot(), width = 10, height = 10)*

*#############################################################################*

*#*

*##*

*#### Other code that may be useful:*

*##*

*#*

*# normality and homogeneity of variance checks:*

*# test for normality (p-values above 0.05 indicate normality)*

*shapiro.test(kelp$Ile13C)*

*shapiro.test(fito$Ile13C)*

*# test for homogeneity of variance*

*#(p-values above 0.05 indicate homogeneity of variance)*

*leveneTest(Ile13C ~ Type, data = prod)*

*# perform an ANOVA to look for differences in ile d13C values among species*

*ile.aov <- aov(Ile13C ~ Type, data = prod)*

*summary(ile.aov)*

*# perform pairwise comparisons using Tukey Honest Significant Differences*

*TukeyHSD(ile.aov)*

*# CLASSIFYING UNKNOWN OR CONSUMER SAMPLES:*

*# now let's classify the consumers based on their essential AA d13C values*

*# read in consumer files:*

*con <- read.csv(YOUR DATA HERE.csv)*

*# using the predict function, and specifying variables of interest*

*# i.e., which AA), perform LDA to classify unknown samples.*

*consumers <- predict(prod\_LDA, con[,c(2:7)]) # here we are pulling out EAA*

*# put LDA coordinates and classifications into a dataframe*

*datPred2 <- data.frame(Group = con$spp, consumers$x)*

*# we can look at how many individuals from each species were classified*

*# with each producer group*

*ct.consumers <- table(con$spp,consumers$class)*

*ct.consumers*

*# From the example data – specifying a smaller number of consumers*

*datPred3 <- datPred2[!(datPred2$Group == "ACH"|*

*datPred2$Group == "ENI"|*

*datPred2$Group == "TNI"|*

*datPred2$Group == "HHE"|*

*datPred2$Group == "COR"|*

*datPred2$Group == "PPU"|*

*datPred2$Group == "TAT"|*

*datPred2$Group == "CCH"|*

*datPred2$Group == "ZOOP"|*

*datPred2$Group == "film"),]*

*# plot the consumers (datPred2 or datPred3) on top of the producers (datPred)*

*ggplot(datPred, aes(x = LD1, y = LD2, color = Group)) +*

*geom\_point(size = 6, alpha = 0) +*

*theme\_classic() +*

*stat\_ellipse(geom="path", aes(color = Group),*

*alpha = 1,*

*show.legend = FALSE,*

*level = 0.95, size = 2) +*

*geom\_point(data = datPred3, size = 6, aes(color = Group)) +*

*scale\_color\_manual(values = c('#000000', '#FFC000', '#0072B2', '#CC79A7', '#56B4E9', '#009E73', '#C69F78', '#92D050', '#999999', '#D55E00'))*