Rarefying OTU Table

* Alpha\_rarefaction.py
  + Gave us the rarefaction plots that allowed us to pick # of sequences
* Single\_rarefaction.py
  + Gave us the actual rarefied biom table called Singlerarefactionpyresult.biom
  + Later converted to Singlerarefactionpyresult.txt
* Use this output for alpha and beta diversity measures

Filtering OTU Table

* Filter\_samples\_from\_otu\_table.py
  + Gave us Filtersamplesresult.biom and Filtersamplesresult.txt

Alpha Diversity

* Alpha\_diversity.py
  + Gave us alphadiversityprresult.txt

Beta Diversity

* Beta\_diversity\_through\_plots.py
  + #!/bin/bash -l

#PBS -l nodes=1:ppn=16,mem=2Gb,walltime=2:00:00

#PBS -m abe

#PBS -M your\_email

#PBS -o job\_name\_stdout

#PBS -e job\_name\_stderr

module load qiime/1.8.0

beta\_diversity\_through\_plots.py

-i file/path/to/otu\_table.biom

-o file/path/to/yourhomedir/output\_name

-t /home/biol1961/shared/97\_otus.tree

-m /home/biol1961/shared/map.txt

-p /home/biol1961/shared/parameters.txt

* + Will get Betadivpyresult2 folder of PcoA plots

Summarizing Taxa

* Summarize\_taxa\_through\_plots.py
  + Will get a set of plots and taxa summary tables in Summarizetaxathroughplotsresults

Loading tables in R

* #Load the rarefied OTU table

otu <- read.table("Singlerarefactionpyresult.txt",

comment="",

header=TRUE,

sep="\t",

skip=1,

as.is=TRUE,

check.names=F,

row=1)

* #Load the filtered OTU table

otu\_filt <- read.table("Filtersamplesresult.txt",

comment="",

header=TRUE,

sep="\t",

skip=1,

as.is=TRUE,

check.names=F,

row=1)

* # Read in the alpha diversity table

alpha <- read.table("alphadiversityprresult.txt",

sep='\t',

header=TRUE,

as.is=TRUE,

check.names=FALSE,

row=1)

* # Load the beta diversity matrix, notice that we use read.table(),

# but then change from a dataframe to a matrix with as.matrix()

betaunweighted <- as.matrix(read.table("unweighted\_unifrac\_dm.txt",

sep = "\t",

header=T,

row = 1,

as.is = T,

check.names = F))

* betaweighted <- as.matrix(read.table("weighted\_unifrac\_dm.txt",

sep = "\t",

header=T,

row = 1,

as.is = T,

check.names = F))

* betabraycurtis <- as.matrix(read.table("bray\_curtis\_dm.txt",

sep = "\t",

header=T,

row = 1,

as.is = T,

check.names = F))

Metadata File

* metadata <- read.table('Gevers\_mapping\_file.txt',

header=T,

sep='\t',

check.names=F,

comment='',

row=1)

Formatting Data

# First, define all the samples in the OTU table.

# Remember, when we load in the OTU table, samples are columns

# Remember, the last column in the OTU table is taxonomy (don't use it)

samples1 <- colnames(otu)[1:(ncol(otu)-1)]

samples2 <- colnames(otu\_filt)[1:(ncol(otu\_filt)-1)]

# Now let's see what the intersect with the metadata row names are

IDs\_Keep <- intersect(samples1, rownames(metadata))

IDs\_Keep2 <- intersect(samples2, rownames(metadata))

# Now let's filter the metadata to keep only those samples

# We tell R to make a new data frame that only has the rows we want

metadata <- metadata[IDs\_Keep,]

metadata2 <- metadata[IDs\_Keep2,]

# Now let's filter the OTU table to keep just the intersecting samples

# We will store it as a new otu table (incase we need the old one)

# Remember, OTU table has columns as samples!

# This will also remove the taxonomy, because it's not a sample ID we want

otu2 <- otu[,IDs\_Keep]

#for rarefied

otu\_filt2 <- otu\_filt[,IDs\_Keep2]

#for low depth removed (Doesn't work??)

# To add the taxonomy back, we can use the taxonomy info from

# the orignal table

otu2$taxonomy <- otu$taxonomy

otu\_filt2$taxonomy <- otu\_filt$taxonomy #(Also doesn't work)

# Now let's filer the alpha diversity table to keep those samples too

# Alpha diversity has the samples as row names

alpha <- alpha[IDs\_Keep, ]

# Now let's filter the beta diversity table to keep those samples too

# Beta diversity has the samples as row names AND column names

# We must filter both the rows and columns

betaunweighted <- betaunweighted[IDs\_Keep,IDs\_Keep]

betaweighted <- betaweighted[IDs\_Keep,IDs\_Keep]

betabraycurtis <- betabraycurtis[IDs\_Keep,IDs\_Keep]

#Let's check to make sure the samples match

as.character(rownames(metadata)) == colnames(otu2)[1:(ncol(otu2)-1)]

Plotting in R

library(ggplot2)

# We will make a copy of our metadata to work with

combined\_alphadata <- metadata

# Because our sample order is the same, we can make a new column in the table

# This column will contain all the Shannon index measurements for the samples

combined\_alphadata$shannon <- alpha$shannon

#Making a boxplot

ggplot(data=combined\_alphadata, aes(x= SMOKING, y= shannon)) + geom\_boxplot() +

labs(x=”Smoking Status”, y=”Shannon Index”, title=”Shannon Index Values for Different Smoking Statuses”)

theme\_bw() +

geom\_jitter(data=combined\_alphadata,width= 0.1, aes(x=SMOKING, y=shannon, color=SMOKING))

Testing for Normality

# We will find the samples that are current smokers in the metadata

Current.ix <- metadata$SMOKING == 'Current'

# And subset the alpha table to include only those, and store it as 'Current'

Current <- alpha[Current.ix,]

#Same for the rest

Former.ix <- metadata$SMOKING == 'Former'

Former <- alpha[Former.ix,]

Never.ix <- metadata$SMOKING == 'Never'

Never <- alpha[Never.ix,]

#Plotting the histograms

hist(Current$shannon, xlab="Alpha Diversity", main="Current Smokers")

hist(Former$shannon, xlab="Alpha Diversity", main="Former Smokers")

hist(Never$shannon, xlab="Alpha Diversity", main="Never Smokers")

#Shapiro-Wilk Normality Test

#P-values less than .1 tell us the distribution is not normal

shapiro.test(Current$shannon)

#p=.03392

shapiro.test(Former$shannon)

#p=.003078

shapiro.test(Never$shannon)

#p=3.888e-05

#Because the pvalues are less than .1 we need to do a test that does not require a normal

#distribution. For example a wilcox test. Because we have multiple groups we need to use a loop

#to look at all the unique variations

# First let's set all the groups available for the variable we care about

# In this case we will use Smoking Status instead of what we set as cov1

groups <- unique(metadata$SMOKING)

# We create empty vectors to store the pair-wise pvalues and the

# groups tested (names)

pw.pvalues <-NULL

pw.names <- NULL

# We set two counters, 'i' starts at 1 and goes until one less than

# the number of groups. 'j' will start at 2, and go until the full

# number of groups. This will end up comparing: 1 vs 2, 2 vs 3,

for(i in 1:(length(groups) - 1)){

for (j in (i+1):length(groups)){

#we use this to pick the groups assigned to 'i'

ix.metric.i <- metadata$SMOKING == groups[i]

#and this for 'j'

ix.metric.j <- metadata$SMOKING == groups[j]

#this stores the pvalue from the test

pvalue <- wilcox.test(alpha[ix.metric.i,"shannon"],

alpha[ix.metric.j,"shannon"])$p.value

#appends the new p-value to the list

pw.pvalues <- c(pw.pvalues, pvalue)

#sets the names of the groups tested

test.name <- paste(groups[i], "\_vs\_", groups[j],sep="")

#appends the names of the groups tested to the list

pw.names <- c(pw.names, test.name)

}

}

names(pw.pvalues) <- pw.names

pw.pvalues

# We will correct using 'fdr', which is the false discovery rate, i.e. correcting for the

#false positive

fdr.pvalues <- p.adjust(pw.pvalues,'fdr')

fdr.pvalues

#We can see that the Current\_vs\_Former .07 p-value was a false positive

# sink() will write whatever is listed below it to a file.

# You close that file by listing sink() again.

sink("alpha\_stats.txt")

cat("\nNumber of samples in each group:\n")

print(table(metadata$SMOKING))

#This prints a table of the number of samples at each body site

cat("\nMean Alpha Diversity:\n")

print(tapply(alpha$shannon, metadata$SMOKING, mean))

# This will get the mean of alpha diversity at each body site

# by using tapply() to apply the mean function across the alpha

# table (subsetted into smoking groups)

cat("\nMedian Alpha Diversity:\n")

print(tapply(alpha$shannon, metadata$SMOKING, median))

# This will get the median of alpha diversity in each smoking group

cat("\nStandard Deviation:\n")

print(tapply(alpha$shannon, metadata$SMOKING, sd))

# This will get the standard deviations of alpha diversity for each group

cat("\nPairwise Mann-Whitney-Wilcoxon Tests were performed.\n")

cat("Pairwise p-values are:\n")

print(pw.pvalues)

cat("\nFDR-corrected pairwise p-values are:\n")

print(p.adjust(pw.pvalues,'fdr'))

sink()

#Plotting this data

alpha2 <- alpha

alpha2$SMOKING <- metadata$SMOKING

ggplot(data=alpha2, aes(x=SMOKING, y= shannon)) +

scale\_color\_manual(values=c("red","blue","gold"))+

geom\_boxplot() +

labs(title="Shannon Index Values for Smoking Types") +

geom\_jitter(width= 0.1, aes(color=SMOKING)) +

theme\_bw()

PCOA in R

library(ape)

library(vegan)

library(ggplot2)

# Run the pcoa() function on the beta diversity table,

# and store the vectors generated as a dataframe

PCOA <- data.frame(pcoa(betaunweighted)$vectors)

# If you look at the PCOA table, you'll see the column names

# are the 'axes' and the row names are sample IDs. We want them to

# be labeled "PC" instead of "axis"

# We will make a vector with place holders

new\_names <- rep("", ncol(PCOA))

# Fill in first with PC followed by the number (e.g. PC1, PC2, PC3...)

for(i in 1:ncol(PCOA)){ new\_names[i] <- paste("PC",i, sep="") }

# Create a column that is SampleIDS for PCOA

PCOA$SampleID <- rownames(PCOA)

#Create a column that is SampleIDs for the metadata

metadata$SampleID <- rownames(metadata)

# Merge the metadata and beta diversity

PCOA <- merge(PCOA, metadata, by = "SampleID")

Graphing the PCOA

library(ape)

library(vegan)

library(ggplot2)

# Run the pcoa() function on the beta diversity table,

# and store the vectors generated as a dataframe

PCOA <- data.frame(pcoa(betaunweighted)$vectors)

# If you look at the PCOA table, you'll see the column names

# are the 'axes' and the row names are sample IDs. We want them to

# be labeled "PC" instead of "axis"

# We will make a vector with place holders

new\_names <- rep("", ncol(PCOA))

# Fill in first with PC followed by the number (e.g. PC1, PC2, PC3...)

for(i in 1:ncol(PCOA)){

new\_names[i] <- paste("PC",i, sep="") }

# Replace the column names of PCOA names

names(PCOA) <- new\_names

# Create a column that is SampleIDS for PCOA

PCOA$SampleID <- rownames(PCOA)

#Create a column that is SampleIDs for the metadata

metadata$SampleID <- rownames(metadata)

# Merge the metadata and beta diversity

PCOA <- merge(PCOA, metadata)

# Note that geom\_point() makes it a scatter plot where the points

# are colored according to Smoking Status

ggplot(PCOA) +

geom\_point(aes(x = PC1, y = PC2, color = SMOKING))

# Now let's add some clusters. This makes it look great, but can   
# also be misleading and make us think there are groups when there   
# aren't. Note that we are using BodySite to color the points and body   
# AREA to fill the clusters   
ggplot(PCOA) +

scale\_color\_manual(values=c("red","blue","goldenrod1"))+

geom\_point(aes(x = PC1, y = PC2, color = SMOKING)) +

labs(title="PCoA and Clusters") +

stat\_ellipse(alpha = 0.3, geom="polygon", linetype="blank",

aes(x = PC1, y = PC2, fill=SMOKING))+

scale\_fill\_manual(values=c("red","blue","goldenrod1"))

levels(PCOA$SMOKING)

Taxa Summaries in R

#Levels: 1 = kingdom 2 = phylum 3 = class 4 = order 5 = family 6 = genus 7 = species

#Picking a level

level=2

# First we make an empty table (array) for our new names

# The array will have the number of rows equal to the number of OTUs in the table

# and one column for each taxonomy level

names\_split <- array(dim=c(length(otu\_filt2$taxonomy), level))

# We will store our taxonomy as a list of names

otu\_names <- as.character(otu\_filt2$taxonomy)

# Then we run through each name and split based on the level we are

# interested in. We make a for loop to split every name stored in

# otu\_names. strsplit() splits the string (otu\_names[i]) at ";".

# This retains all the levels as separate strings. head() takes the

# first items (the total will be the number you specified with level)

# from the string split output and stores it in the names\_split

# array at the specied row.

for (i in 1:length(otu\_names)){

names\_split[i,] <- head(strsplit(otu\_names[i], "; ", fixed=T)[[1]], n=level)

}

# Now we will collapse the strings together into one string

otu\_names <- apply(names\_split, 1, function(x) paste(x[1:level], sep = "", collapse = ";"))

# Replace the old taxonomy with the truncated version, as a new table

otu\_filt3 <- otu\_filt2

otu\_filt3$taxonomy <- otu\_names

# Get the number of samples (the last column is taxonomy)

sample\_no <- ncol(otu\_filt3)-1

# Collapse the otu table and save it as a new table

otu\_filt3 <- aggregate(otu\_filt3[,1:sample\_no], by=list(otu\_filt3$taxonomy), FUN=sum)

# Name the first column taxonomy because R stores the column

# we told it to aggregate by as the first column

names(otu\_filt3)[1] <- "taxonomy"

# We can see that the consolidating worked by checking how many rows we

# now have - that's how many phyla there are (level=2)

nrow(otu\_filt3)

# Set rownames as taxonomy

rownames(otu\_filt3) <- otu\_filt3$taxonomy

# Keep all columns in the otu table that do NOT (!) have the column

# header "taxonomy"

otu\_filt3 <- otu\_filt3[,!names(otu\_filt3) == "taxonomy"]

Filtering OTUs and Samples

# Filter OTUs that are in low abundance

# Change those less than 1/1 millionth of read depth to 0

# colSums(otu\_filt3) takes the column sums of each column (sample)

# sum(colSums(otu\_filt3)) takes the sum off all the columns (total counts)

# sum(colSums(otu\_filt3))/1000000 divides by 1 million

# otu\_filt3[otu\_filt3 < sum(colSums(otu\_filt3))/1000000] will set the values less than that number to TRUE

# otu\_filt3[otu\_filt3 < sum(colSums(otu\_filt3))/1000000] <- 0 will set the TRUEs to 0

otu\_filt3[otu\_filt3 < sum(colSums(otu\_filt3))/1000000] <- 0

# Change singletons to 0 (needed for low depth OTU tables)

# otu\_filt3 < 2 sets all values less than 2 to TRUE

# otu\_filt3[otu\_filt3 < 2] <- 0 sets all the TRUEs to 0

otu\_filt3[otu\_filt3 <2] <- 0

# Filter the OTU table to keep OTUs in at least 5% of samples

# otu\_filt3 > 0 sets the values greater than 0 to TRUE

# rowSums(otu\_filt3 > 0) sums the TRUEs in each row (number of samples OTU is in)

# ncol(otu\_filt3) find the number of columns

# (0.05\*ncol(otu\_filt3)) multiple that by 0.05

# otu\_filt3[rowSums(otu\_filt3 > 0) > (0.05\*ncol(otu\_filt3)),] keep only rows where occurance is greater than 5% of samples

otu\_filt3 <- otu\_filt3[rowSums(otu\_filt3 > 0) > (0.05\*ncol(otu\_filt3)),]

# We want to find relative abundance and for that we need to use all the columns (since we already took out taxonomy)

for(i in 1:ncol(otu\_filt3)){ otu\_filt3[,i] <- otu\_filt3[,i]/sum(otu\_filt3[,i])

}

#Now we need to make this fit with the metadata

# Transpose as a data frame

otu\_filt3 <- data.frame(t(otu\_filt3))

# Make a column that is the Sample IDs (which are the rownames)

otu\_filt3$SampleID <- rownames(otu\_filt3)

# Let's save a backup of this filtered OTU table

otu\_backup <- otu\_filt3

#Now we will convert the data frame into 3 columns: sampleIDs, taxaIDs, and rel. abund

#using the fxn melt

library(reshape2)

library(plyr)

otu\_filt3 <- melt(otu\_filt3, id.vars = "SampleID",

variable.name = "Taxa",

value.name = "RelativeAbundance")

library(ggplot2)

# This will make a plot with the OTU table (otu), using the column

# headers specified. Geom bar makes it a bar plot. labels-NULL takes off the x labels.

ggplot(otu\_filt3, aes(x = SampleID, y = RelativeAbundance, fill= Taxa)) +

geom\_bar(stat = "identity", position="fill") +

scale\_x\_discrete(labels = NULL)

#That gives us a really messy taxa summary plot

Adding Metadata to Taxa Summaries

#Now we need to add metadata information

otu\_filt3 <- otu\_backup

otu\_filt3 <- melt(otu\_filt3, id.vars = "SampleID", variable.name = "Taxa", value.name = "RelativeAbundance")

#This will keep only the columns with the headers we want

colnames(metadata)

columns\_keep <- c("SMOKING", "SampleID")

metadata3 <- metadata[,columns\_keep]

# Now we merge covariates to sample ids

# First we need to make a column that is the sample IDs

metadata3$SampleID <- rownames(metadata)

# This will drop any samples in the mapping file that aren't in the OTU table

otu\_filt3 <- merge(otu\_filt3, metadata3, by="SampleID")

#Plot, using position="fill" makes sure it sums to 1

ggplot(otu\_filt3, aes(x=SMOKING, y=RelativeAbundance, fill=Taxa)) +

geom\_bar(stat ="identity", position="fill") +

labs(title="Taxa Summary for Current, Former and Never Smokers", x="Smoking Status", y="Relative Abundance")

Plotting Specific Taxa

# If we don't remember the spelling, we can print all the taxa and

# copy and paste:

unique(otu\_filt3$Taxa)

# Let's subset to just Firmicutes and Actinobacteria

taxaList <- c("k\_\_Bacteria.p\_\_Firmicutes", "k\_\_Bacteria.p\_\_Actinobacteria" )

# Let's make a new subsetted table that is just those phyla

filtered <- subset(otu\_filt3, is.element(otu\_filt3$Taxa, taxaList))

ggplot(filtered, aes(x = SMOKING, y = RelativeAbundance, fill=Taxa)) +

geom\_bar(stat="identity") +

labs(y = "Relative Abundance", x="Smoking Status", title="Relative Abundances of Specific Taxa in Current, Former and Never Smokers") +

scale\_fill\_discrete(labels = c("Actinobacteria", "Bacteroidetes"))

Testing for Abundance Differences

#First we need to take the taxonomy out of the table, filter low abundant OTUs

#and low occuring OTUs

# We can store taxonomy and which OTUs they are to use for later

# drop=F makes sure it stays as a table

taxonomy\_table <- otu\_filt2[,"taxonomy",drop=F]

#Keep only the samples, drop taxonomy from table

otu\_filt3 <- otu\_filt2[, ! names(otu\_filt2) =="taxonomy"]

#Filter OTUs that are in low abundance

#Change those less than 1/1 millionth of read depth to 0

otu\_filt3[otu\_filt3 < sum(colSums(otu\_filt3))/1000000] <- 0

#Change singletons to 0 (needed for low depth OTU tables)

otu\_filt3[otu\_filt3 < 2] <- 0

#Filter the OTU table to keep OTUs in at least 5% of samples

otu\_filt3 <- otu\_filt3[rowSums(otu\_filt3 > 0) > (0.05\*ncol(otu\_filt3)),]

#Now we need to transform the data using a centered log-ratio transformation

library(robCompositions)

#Convert any 0 to 0.65 to allow for CLR transform

otu\_filt3[otu\_filt3 == 0] <- 0.65

#CLR transform

#convert to samples as rows

otu\_table <- t(otu\_filt3)

#Centered log-ratio tranform the data

otu\_table <- cenLR(otu\_table)$x.clr

# Let's test the first OTU (first column) in the OTU table

# What is the name of this OTU? We can look it up in our table

# We pick the row we want using the otu id in the column

this\_taxa <- taxonomy\_table[colnames(otu\_table)[1],"taxonomy"]

this\_taxa

# Now lets run the test using the first column and according to smoking statuses in the metadata

aov\_test <- aov(otu\_table[,1] ~ metadata2$SMOKING)

summary(aov\_test)

#Shows us only the p-value

summary(aov\_test)[[1]][1,5]

# We can plot this. Because ggplot likes to have all the data in one table, let's make a new ta ble to plot with

plot\_table <- data.frame(otu\_table)

#Note that this will store and x infront of all the numerical column names

plot\_table$SMOKING <- metadata2$SMOKING

#store which column (header) you want to plot

this\_otu <- colnames(plot\_table)[1]

# We can also store its name

# will split the taxonomy based on the ";"

# Then take the last two values (genus and species) to shorten the name

name = strsplit(this\_taxa, ";", fixed=T)[[1]]

names\_tail = tail(name, n=2)

# This will plot the transformed abundances for each body site

# Note that we have to use aes\_string() because we are filling in the y colum n header with a string

ggplot(plot\_table) +

geom\_boxplot(aes\_string(x="SMOKING", y=this\_otu, fill="SMOKING"))+

scale\_fill\_manual(values = c("red", "blue", "gold"))+

labs(y=names\_tail, x="Smoking Status", title="Abundance of Specific Species for Current, Former and Never Smokers")+

theme\_bw()

Testing All Taxa

#The first step is to make an empty vector that will store our p-values.

pvals <- c()

#Remake the plot\_table

plot\_table$SMOKING <- metadata2$SMOKING

#Loop through each column except the last (because it's smoking status)

for(i in 1:(ncol(plot\_table)-1)){

aov\_out <- aov(plot\_table[,i] ~ plot\_table$SMOKING)

pvals[i] <- summary(aov\_out)[[1]][1,3]

}

#We can check how many significant pvals we have

sum(pvals < 0.05)

#Plotting the significant pvals

# Index just the first three significantly different OTUs

# which() tells us the position of the values that are true (< 0.05), and [1: 3]

# takes the first 3.

first\_three <- which(pvals < 0.05)[1:3]

# This loops through the significant OTUs, stores their name

# and makes a box plot of the transformed abundances of the taxa

# We then store the plots in a list

plot\_list <- list()

for(i in 1:length(first\_three)){

index <- first\_three[[i]]

this\_otu <- colnames(plot\_table)[i]

this\_taxa <- taxonomy\_table[i,"taxonomy"]

name <- strsplit(this\_taxa, ";", fixed=T)[[1]]

taxon <- paste(name[4], name[5], name[6], name[7], sep=" ")

plot\_out <- ggplot(plot\_table) +

geom\_boxplot(aes\_string(x="SMOKING", y=this\_otu, fill="SMOKING")) +

scale\_fill\_manual(values = c("tomato", "darkorchid4", "gold")) +

labs(y=taxon)

plot\_list[[i]] <- plot\_out

}

# Now lets print the three plots to a pdf

# each plot will be a new page in the pdf

pdf("Diff\_taxa.pdf", height=4, width=6)

for(i in 1:length(plot\_list)){

plot(plot\_list[[i]])

}

dev.off()