##################################################################

#

# R script for the data analysis as presented in

# Medina Paz et al. (2021) in International Journal of Molecular Sciences

#

###

# for more information:

# Francisco Medina Paz, francisco.medina@cinvestav.mx

#

# 11-13-2021

#

###################################################################

#################### Prepare R Environment ####################

rm(list=ls())

setwd (“~/Documents/git/PAPER\_Medina\_2021\_AllSetBeforeFlowering“

options(max.print=999999, scipen=3, digits=5)

######### load R packages

library (phyloseq)

library (ggplot2)

library (data.table)

library (plyr)

library (scales)

library (grid)

library (gplots)

library (VennDiagram)

library (biomformat)

library (vegan)

library (ggsignif)

library (edgeR)

if (Sys.getenv("JAVA\_HOME")!="")

Sys.setenv(JAVA\_HOME="")

library (xlsx)

#################### Import Data ####################

mapFile <- "map.txt"

map <- import\_qiime\_sample\_data(mapfilename=mapFile)

mapTab <- read.table(mapFile, header = TRUE, as.is=TRUE, sep="\t", comment.char="", row.names=1)

otus<-import\_biom("new\_otu\_table\_mod\_final.biom", parseFunction= parse\_taxonomy\_greengenes)

otus

real\_data<-merge\_phyloseq(map,otus)

real\_data<-filter\_taxa(real\_data, function(x) (sum(x) > 0), TRUE)

real\_data

################### Defining parameters ############

topN=10

distance <- "bray" ### It could be replace by unifrac, wunifrac dcpoa and jsd ##

Count\_threshold<-10

tax\_level <- c(2:6) ### 1 == "Kingdom", 2 == "Phylum", 3 == "Class", 4 == "Order", 5 == "Family", 6 == "Genus"

FDR\_thres <- 0.05

logFC\_thres <- 2

################### Emule colors for ggplots2 ######

gg\_color\_hue <- function(n) {

hues = seq(15, 375, length=n+1)

hcl(h=hues, l=65, c=100)[1:n]

}

################### Calculating Diversity Indices ###################

index\_list<- c("Shannon", "Simpson", "Evenness", "Observed", "Chao1")

Diversity\_Indices<- estimate\_richness(real\_data,split= TRUE, measures= index\_list)

evenness<-c()

for (i in (1:nsamples(real\_data))){

evenness[i]<-c((Diversity\_Indices[i,"Shannon"])/(log(Diversity\_Indices[i,"Observed"],base = exp(1))))

}

Diversity\_Indices[,"Evenness"]<-evenness

Diversity\_Indices[,"SampleNames"]<-sample\_data(real\_data)[,4]

Diversity\_Indices[,"Compartment"]<-sample\_data(real\_data)[,5]

Diversity\_Indices[,"DevStage"]<-sample\_data(real\_data)[,6]

Diversity\_Indices

################### Pairwise t-student test of Diversity indices #############

pairwise.t.test(Diversity\_Indices$Shannon, Diversity\_Indices$Compartment, p.adjust="bonferroni") #http://www.stat.columbia.edu/~martin/W2024/R3.pdf

################### Boxplot of Shannon Index per sample Compartment #################

ggplot(Diversity\_Indices, aes(x=Compartment, y=Shannon, fill= Compartment)) + geom\_boxplot() + theme\_bw() + labs(title="Shannon Diversity Index", x="", y = "Shannon Index") + theme(legend.text = element\_text (size=18), plot.title=element\_text (size=22, hjust =0.5), axis.text = element\_text(size=18, colour = "black"), axis.title=element\_text(size=18), legend.title =element\_text(size=18), axis.title.y = element\_text(vjust = 2.5))+

geom\_signif(textsize = 8, comparisons = list(c("Endosphere", "Rhizosphere")), annotations = c ("\*\*\*"), inherit.aes = T, map\_signif\_level=TRUE, y\_position = c(7.3), test= t.test) + ylim(c(3,7.8)) + geom\_jitter(alpha= 0.3, width = 0.1, size= 3) + scale\_fill\_manual (values=c("#ED8141", "#69b3a2")) + scale\_x\_discrete(limits = rev(levels(factor(sample\_data(real\_data)$Compartment)))) + guides(fill = guide\_legend(reverse=TRUE))

################### Principal Coordinate Analysis using Bray-Curtis distance #####

title= paste0 ("PCoA All Samples")

real\_data.ord<-phyloseq::ordinate(real\_data, method= "PCoA", distance= distance)

plot\_ordination (real\_data, real\_data.ord,type = "samples", color = "Compartment", shape="DevStage", title = title, axes=c(1,2)) + theme\_bw() +

geom\_point(size= 7) + theme(plot.title = element\_text(size =18, hjust = 0.5),axis.title=element\_text(size=18, vjust=-0.5), axis.text = element\_text (size = 18, colour="black"), legend.title = element\_text(size = 18), legend.text=element\_text(size=18))+

scale\_color\_manual(values = c( "#ED8141", "#69b3a2")) + guides(color = guide\_legend(order = 1, reverse = TRUE), shape = guide\_legend(order = 2))

############################ OTU Relative Abundance stacked barplot ###################

###################### Grouping samples ############

grouped\_data<-merge\_samples(real\_data, group="Description")

mapFile <- "map\_descriptions.txt"

map <- import\_qiime\_sample\_data(mapfilename=mapFile)

rownames(map)<-rownames(sample\_data(grouped\_data))

sample\_data(grouped\_data)<-map

sample\_data(grouped\_data)

################### Calculating Relative Abundance ##################

grouped\_data\_rel<-transform\_sample\_counts(grouped\_data, function(OTU) OTU / sum(OTU))

grouped\_data\_rel

################### Calculating Top 10 Most Abundant Taxa per Compartment ##################

Tax\_level<- rank\_names(grouped\_data)[c(tax\_level)]

samp\_rel <- list()

i<-1

for (n in levels(factor(sample\_data(grouped\_data)$Compartment))){

samp\_rel[[n]] <- assign(n,subset\_samples(grouped\_data\_rel,Compartment==(levels(factor(sample\_data(grouped\_data)$Compartment)))[i]))

i<-(i+1)

}

top\_samp <- paste0("top\_",levels(factor(sample\_data(grouped\_data)$Compartment)))

top\_samp\_topN <- paste0(top\_samp,"\_",topN)

TopOTUs\_Compartments <- paste0 ("TopOTUs\_", levels(factor(sample\_data(grouped\_data)$Compartment)))

top\_samp\_rel <- list()

topN\_samp <- list()

Compartments\_TopOTUs <- list ()

i=1 ;j=1 ;k=1; b=1; c=1; h=1; d=1; q=1;

for (n in Tax\_level){

for (m in top\_samp){

top\_samp\_rel[[m]] <- assign(m,tax\_glom(samp\_rel[[i]],n, NArm=FALSE))

i=i+1

}

for (c in top\_samp\_rel){

for (d in (1:(length(tax\_table(c)[,n])))){

if (is.na (tax\_table(c)[d,n]) == TRUE){

for (j in c(1:6)){

tax\_table(c)[d,c(j:(grep(n,rank\_names(c))))] <- ("Unclassified")

}

}

}

top\_samp\_rel[[top\_samp[h]]] <- assign(top\_samp[h],tax\_glom(c, n, NArm=FALSE))

h=h+1

}

print (top\_samp\_rel)

for (l in (1:length(top\_samp\_rel))){

if (any(names((sort(tapply(taxa\_sums(top\_samp\_rel[[l]]), tax\_table(top\_samp\_rel[[l]])[, n], sum),TRUE)[seq\_len(topN)])/nsamples(grouped\_data)) == "Unclassified") == TRUE)

{

Compartments\_TopOTUs[[l]] <- (sort(taxa\_sums(top\_samp\_rel[[l]]),TRUE)[seq\_len(topN+1)])/(nsamples(samp\_rel[[l]]))

topN\_samp[[l]] <- assign(top\_samp\_topN[l],prune\_taxa(names(Compartments\_TopOTUs[[l]]),top\_samp\_rel[[l]]))

color\_var <- sort(names ((sort(tapply(taxa\_sums(top\_samp\_rel[[l]]), tax\_table(top\_samp\_rel[[l]])[, n], sum),TRUE)[seq\_len(topN+1)])/nsamples(grouped\_data)))

} else if (any(names((sort(tapply(taxa\_sums(top\_samp\_rel[[l]]), tax\_table(top\_samp\_rel[[l]])[, n], sum),TRUE)[seq\_len(topN)])/nsamples(grouped\_data)) == "Unclassified") == FALSE)

{

Compartments\_TopOTUs[[l]] <- (sort(taxa\_sums(top\_samp\_rel[[l]]),TRUE)[seq\_len(topN)])/(nsamples(samp\_rel[[l]]))

topN\_samp[[l]] <- assign(top\_samp\_topN[l],prune\_taxa(names(Compartments\_TopOTUs[[l]]),top\_samp\_rel[[l]]))

color\_var <- sort(names ((sort(tapply(taxa\_sums(top\_samp\_rel[[l]]), tax\_table(top\_samp\_rel[[l]])[, n], sum),TRUE)[seq\_len(topN)])/nsamples(grouped\_data)))

}

}

for (a in (1:length(top\_samp\_rel))){

for (k in (1:length(tax\_table(top\_samp\_rel[[a]])[,n]))){

if (any((as.vector(tax\_table(top\_samp\_rel[[a]])[k,n]) == as.vector(tax\_table(topN\_samp[[a]])[,n]))) == F){

if (as.vector(tax\_table(top\_samp\_rel[[a]])[k,n]) == "Unclassified"){} else{

for (o in (1:length(tax\_table(topN\_samp[[a]])[,n]))){

if (as.vector(tax\_table(top\_samp\_rel[[a]])[k,n]) != as.vector(tax\_table(topN\_samp[[a]])[o,n])){

for (p in c(1:6)){

tax\_table(top\_samp\_rel[[a]])[k,c(p:grep(n,rank\_names(grouped\_data)))] <- ("Other")

}

break }

}

}

}

}

}

# }

for (m in top\_samp){

top\_samp\_rel[[m]] <- assign(m,tax\_glom(top\_samp\_rel[[q]],n, NArm=FALSE))

q=q+1

}

for (l in (1:length(top\_samp\_rel))){

if (any(names((sort(tapply(taxa\_sums(top\_samp\_rel[[l]]), tax\_table(top\_samp\_rel[[l]])[, n], sum),TRUE)[seq\_len(topN)])/nsamples(grouped\_data)) == "Other", na.rm=T) == TRUE)

{

Compartments\_TopOTUs[[l]] <- (sort(taxa\_sums(top\_samp\_rel[[l]]),TRUE)[seq\_len(topN+2)])/(nsamples(samp\_rel[[l]]))

print ((sort(tapply(taxa\_sums(top\_samp\_rel[[l]]), tax\_table(top\_samp\_rel[[l]])[, n], sum),TRUE)[seq\_len(topN+1)])/(nsamples(samp\_rel[[l]])))

topN\_samp[[l]] <- assign(top\_samp\_topN[l],prune\_taxa(names(Compartments\_TopOTUs[[l]]),top\_samp\_rel[[l]]))

color\_var <- sort(names ((sort(tapply(taxa\_sums(top\_samp\_rel[[l]]), tax\_table(top\_samp\_rel[[l]])[, n], sum),TRUE)[seq\_len(topN+2)])/nsamples(grouped\_data)))

} else if (any(names((sort(tapply(taxa\_sums(top\_samp\_rel[[l]]), tax\_table(top\_samp\_rel[[l]])[, n], sum),TRUE)[seq\_len(topN+1)])/nsamples(grouped\_data)) == "Other", na.rm=T) == FALSE)

{

Compartments\_TopOTUs[[l]] <- (sort(taxa\_sums(top\_samp\_rel[[l]]),TRUE)[seq\_len(topN)])/(nsamples(samp\_rel[[l]]))

print ((sort(tapply(taxa\_sums(top\_samp\_rel[[l]]), tax\_table(top\_samp\_rel[[l]])[, n], sum),TRUE)[seq\_len(topN)])/(nsamples(samp\_rel[[l]])))

topN\_samp[[l]] <- assign(top\_samp\_topN[l],prune\_taxa(names(Compartments\_TopOTUs[[l]]),top\_samp\_rel[[l]]))

color\_var <- sort(names ((sort(tapply(taxa\_sums(top\_samp\_rel[[l]]), tax\_table(top\_samp\_rel[[l]])[, n], sum),TRUE)[seq\_len(topN)])/nsamples(grouped\_data)))

print (top\_samp\_rel[[l]])

}

}

for (o in (1:length(topN\_samp))){

if (n == "Kingdom"){title <- paste0("Top ", topN," ","Kingdoms ",levels(factor(sample\_data(grouped\_data)$Compartment))[o], " Samples")} else if (n == "Phylum"){title <- paste0("Top ", topN," ","Phyla ",levels(factor(sample\_data(grouped\_data)$Compartment))[o], " Samples")}

else if (n == "Class"){title <- paste0("Top ", topN," ","Classes ",levels(factor(sample\_data(grouped\_data)$Compartment))[o], " Samples")} else if (n == "Order"){title <- paste0("Top ", topN," ","Orders ",levels(factor(sample\_data(grouped\_data)$Compartment))[o], " Samples")}

else if (n == "Family"){title <- paste0("Top ", topN," ","Families ",levels(factor(sample\_data(grouped\_data)$Compartment))[o], " Samples")} else if (n == "Genus"){title <- paste0("Top ", topN," ","Genera ",levels(factor(sample\_data(grouped\_data)$Compartment))[o], " Samples")}

else {title <- paste0("Top ", topN," ","Species ",levels(factor(sample\_data(grouped\_data)$Compartment))[o], " Samples")}

color\_pal<- gg\_color\_hue(length(color\_var))

names(color\_pal)<-c(color\_var)

color\_pal["Unclassified"] <- "#999999"; color\_pal["RF3"]<- "olivedrab2"; color\_pal["Thaumarchaeota"] <- "#B385FF";color\_pal["TM7"] <- "gold"; color\_pal["Planctomycetes"] <- "darkslateblue"

color\_pal["Aurantimonadaceae"] <- "#F8766D";color\_pal["Chitinophagaceae"] <- "#DB8E00";color\_pal["Cytophagales"] <- "#AEA200"; color\_pal["Burkholderiales"]<- "#64B200";

color\_pal["Cytophagaceae"] <- "#AEA200"; color\_pal["Nitrososphaeraceae"] <- "coral" ;color\_pal["Paenibacillaceae"] <- "#00BD5C";color\_pal["Myxococcales"]<- "gold";

color\_pal["Rickettsiales"]<- "goldenrod4";color\_pal["Bacillales"]<- "darkkhaki";color\_pal["Enterobacteriales"]<- "darkolivegreen3";color\_pal["Pseudomonadales"]<- "cornflowerblue";

color\_pal["Rhizobiaceae"] <- "#00C1A7"; color\_pal["Sinobacteraceae"] <- "#00A6FF";color\_pal["Sphingomonadaceae"] <- "#B385FF";color\_pal["Comamonadaceae"] <- "tomato2";

color\_pal["Bacillaceae"] <- "skyblue1";color\_pal["Enterobacteriaceae"] <- "moccasin";color\_pal["Planococcaceae"] <- "mistyrose3";color\_pal["Pseudomonadaceae"] <- "orange4";

color\_pal["Staphylococcaceae"] <- "red4";color\_pal["mitochondria"] <- "seagreen4"; color\_pal["Streptomycetaceae"] <- "sienna1";

color\_pal["Caulobacteraceae"] <- "#00BADE";color\_pal["Hyphomonadaceae"] <- "#EF67EB";color\_pal["Syntrophobacteraceae"] <- "#F8766D";color\_pal ["Pseudomonas"]<- "darkkhaki";

color\_pal["Rhodospirillaceae"] <- "gold2"; color\_pal["ML615J-28"]<- "darkkhaki"; color\_pal["Nitrososphaerales"]<- "coral"; color\_pal["[Saprospirales]"]<-"skyblue" ;

color\_pal["Betaproteobacteria"] <- "darkkhaki"; color\_pal["Deltaproteobacteria"]<- "coral"; color\_pal ["[Saprospirae]"]<-"skyblue";color\_pal["Sphingomonas"]<-"firebrick1"

color\_pal["Proteobacteria"] <- "#B385FF"; color\_pal["Nitrospirae"] <- "#FF6A98"; color\_pal["Firmicutes"] <- "deepskyblue"; color\_pal["Chloroflexi"] <- "darkkhaki"; color\_pal["Verrucomicrobia"]<- "skyblue";

color\_pal["Other"] <- "lemonchiffon2";

color\_pal["Paenibacillus"] <- "#00BFC4";

color\_pal["Asticcacaulis"] <- "gold2" ; color\_pal["Gemmatimonadetes"] <- "#00A6FF";color\_pal["Gammaproteobacteria"]<-"cadetblue"

color\_pal["Bradyrhizobiaceae"]<- "#00B0F6"; color\_pal["Syntrophobacteraceae"]<- "gold"; color\_pal["Skermanella"]<- "darkslateblue";

color\_pal["Bacilli"]<- "#AEA200"; color\_pal["S085"]<- "firebrick3"; color\_pal["Flavobacteriia"]<- "darkslateblue"; color\_pal["TM7-3"]<- "gold"; color\_pal["Tenericutes"]<- "darkslateblue";

color\_pal["Saccharibacillus"]<- "tan3";color\_pal["Sporosarcina"]<- "gold";color\_pal["Staphylococcus"]<- "turquoise4"; color\_pal["Acidobacteria"]<- "gold";

color\_pal["Brucellaceae"]<-"forestgreen";color\_pal["Lupinus"]<- "darkolivegreen1";color\_pal["Azospirillum"]<- "darkseagreen1";color\_pal["Alcaligenaceae"]<- "dodgerblue";color\_pal["Burkholderiaceae"]<- "darkorchid1";

color\_pal["Methylobacteriaceae"]<- "darkslateblue";color\_pal["Methylobacterium"]<- "goldenrod2";color\_pal["Burkholderia"]<- "darkslateblue";color\_pal["Moraxellaceae"]<- "lightsalmon";color\_pal["Dickeya"]<- "mediumpurple4";

color\_pal["Acinetobacter"]<- "dodgerblue3";color\_pal["Streptococcus"]<- "goldenrod2";

color\_pal["Rhizobium"] <- "#9590FF";color\_pal["Agrobacterium"] <- "lightslateblue";color\_pal["Ochrobactrum"] <- "blueviolet";color\_pal["Kaistobacter"] <- "mediumpurple4";color\_pal["Steroidobacter"] <- "purple";

color\_pal["Cellvibrio"]<- "darkviolet";color\_pal["Sinorhizobium"] <- "mediumorchid2";color\_pal["Didymeles"]<- "violet";color\_pal ["Pseudomonas"]<- "darkslateblue";color\_pal["Balneimonas"] <- "purple4";

color\_pal["Bacillus"] <- "deepskyblue";

color\_pal["Nitrospira"]<- "pink";

color\_pal["Chryseobacterium"]<- "darkgoldenrod";color\_pal["Flavisolibacter"] <- "darkgoldenrod3";

color\_pal["Candidatus Nitrososphaera"] <- "mediumseagreen";

color\_pal["Streptomyces"]<- "yellow4";color\_pal["Rubrobacter"] <- "gold";

mdf <- psmelt(topN\_samp[[o]])

order <- as.character(unique(mdf[,n]))

if (any(order == "Unclassified") == TRUE)

{

order<- order[-grep("Unclassified", order)]

}

if (any(order == "Other") == TRUE)

{

order <- order [-grep ("Other", order)]

order <- sort(order,F)

order <- append(order, "Other")

order <- append(order, "Unclassified")

} else if (any (order == "Other") == FALSE)

{

order <- sort(order,F)

order <- append(order, "Unclassified")

}

mdf$Names <- factor(mdf[,n], levels = c(order))

addline\_format <- function(x,...){

gsub('\\s','\n',x)

}

labels=addline\_format(mdf$SampleName)

color\_pal<-color\_pal[order]

p<-ggplot(mdf, aes\_string(x = 'DevStage', y = "Abundance")) + theme\_bw()+

theme(axis.text.x = element\_text(size = 18,angle = 0, hjust = 0.5, colour = "black"), axis.title.x = element\_text(size = 18,hjust=0.5,vjust=-0.5),axis.title.y = element\_text(size = 18,vjust=2, hjust=0.5)) + ggtitle(title) +

geom\_bar (aes(fill = factor(mdf[,n],levels=order)), stat = 'identity', position = 'stack') +

ylab("Relative Abundance") + theme(plot.title=element\_text(size=18, hjust=0.5), title=element\_text(size=18, hjust= 0.38),axis.text.y=element\_text(size=18, colour = "black"), legend.text=element\_text(size=18), legend.title=element\_text(size=18) + scale\_y\_continuous(limit = c(0:1)))+

scale\_colour\_manual(values = color\_pal) + scale\_fill\_manual(values = color\_pal, name = n)

print (p)

}

i=1;j=1 ;k=1; b=1; c=1; h=1; d=1; q=1;

}

######################## Venn Diagrams ##################

samp <- list()

for (n in unique(sample\_data(real\_data)$Compartment)){

samp[[n]] <- assign(n,subset\_samples(real\_data,Compartment==n))

}

Tax\_level<- rank\_names(real\_data)[c(tax\_level)]

taxa\_Compartments <- paste0(unique(sample\_data(real\_data)$Compartment),"\_taxa")

taxa\_Compartments\_pruned <- paste0(taxa\_Compartments,"\_pruned")

taxa\_Compartments\_names <- paste0(taxa\_Compartments\_pruned,"\_names")

Compartments\_venn<-paste0(unique(sample\_data(real\_data)$Compartment),"\_venn")

Compartments\_taxa<- list()

pruned\_Compartments\_taxa<- list()

names\_Compartments\_taxa<- list()

venn\_list <- list()

venn\_Compartments<-list()

universe <- list()

i=1; j=1; k=1; a=1; b=1; c=1

for (n in Tax\_level){

if (n == "Kingdom"){t <- 1} else if (n == "Phylum"){t <- 2} else if (n == "Class"){t <- 3} else if (n == "Order"){t <- 4}

else if (n == "Family"){t <- 5} else if (n == "Genus"){t <- 6} else {t <- 7}

for (m in taxa\_Compartments){

Compartments\_taxa[[m]] <- assign(m,tax\_glom(samp[[i]],rank\_names(samp[[i]])[t]))

i=i+1

}

for (l in taxa\_Compartments\_pruned){

pruned\_Compartments\_taxa[[l]] <- assign(l,prune\_taxa(taxa\_sums(Compartments\_taxa[[j]]) > Count\_threshold, Compartments\_taxa[[j]]))

j=j+1

}

for (o in taxa\_Compartments\_names){

names\_Compartments\_taxa[[o]] <- assign(o,names(sort(tapply(taxa\_sums(pruned\_Compartments\_taxa[[k]]), tax\_table(pruned\_Compartments\_taxa[[k]])[, n], sum),TRUE)))

k=k+1

}

for (p in names\_Compartments\_taxa){

venn\_list[[unique(sample\_data(real\_data)$Compartment)[a]]]<-(p)

if (a == 1){

universe <- names\_Compartments\_taxa[[a]]

}

else{

universe <- append(universe, names\_Compartments\_taxa[[a]])

}

a=a+1

}

universe <- unique(universe)

for (r in Compartments\_venn){

venn\_Compartments[[r]]<-assign(r, universe %in% names\_Compartments\_taxa[[b]])

b=b+1

}

if (n == "Kingdom"){title<- "Kingdoms All Samples"} else if (n == "Phylum"){title<- "Phyla All Samples"}

else if (n == "Class"){title<- "Classses All Samples"} else if (n == "Order"){title<- "Orders All Samples"}

else if (n == "Family"){title<- "Families All Samples"} else if (n == "Genus"){title<- "Genera All Samples"}

else{title<- "Species All Samples"}

p <- venn.diagram (venn\_list, height = 100, width = 100, filename=NULL,cex = 1.5 ,cat.cex = 1.3, fill=rainbow(2),main.cex = 1.5,main = title, category.names = c("Endosphere", "Rhizosphere"))

grid.newpage()

grid.draw(p)

Endosphere\_and\_Rhizosphere\_shared <- universe [Rhizosphere\_venn & Endosphere\_venn]

Only\_Rhizosphere <- universe [Rhizosphere\_venn & !Endosphere\_venn]

Only\_Endosphere <- universe [!Rhizosphere\_venn & Endosphere\_venn]

i=1;j=1;k=1;a=1;b=1;c=1

max.len= max(length( Endosphere\_and\_Rhizosphere\_shared ), length(Only\_Rhizosphere), length(Only\_Endosphere))

Endosphere\_and\_Rhizosphere\_shared = c(Endosphere\_and\_Rhizosphere\_shared, rep(NA, max.len - length(Endosphere\_and\_Rhizosphere\_shared)))

Only\_Rhizosphere = c(Only\_Rhizosphere, rep(NA, max.len - length(Only\_Rhizosphere)))

Only\_Endosphere = c(Only\_Endosphere, rep(NA, max.len - length(Only\_Endosphere)))

# matrix <- list( All\_shared= All\_shared, Endosphere\_and\_soil\_shared= Endosphere\_and\_Rhizosphere\_shared , Only\_Rhizosphere= Only\_Rhizosphere, Only\_Endosphere=Only\_Endosphere)

# matrix\_filename <- paste0("matrix\_",n,".xlsx")

# write.xlsx(matrix, matrix\_filename)

}

####################### PERMANOVA using adonis and ANOSIM (vegan package) ##################

set.seed (42)

BC <- phyloseq::distance(real\_data, distance)

map <- data.frame(sample\_data(real\_data))

adonis(formula=BC~Compartment,data=map, permutations = 999, method = distance)

BC.ano<-anosim(BC, data.frame(map)$Compartment, permutations = 999, distance , strata = NULL, parallel = getOption("mc.cores"))

BC.ano

BC <- phyloseq::distance (subset\_samples(real\_data,Compartment != c("Endosphere")),distance)

map <- data.frame(sample\_data(subset\_samples(real\_data,Compartment != c("Endosphere"))))

adonis(formula=BC~DevStage,data=map,permutations = 999, method = distance)

BC.ano<-anosim(BC, data.frame(map)$DevStage, permutations = 999, distance , strata = NULL, parallel = getOption("mc.cores"))

BC.ano

BC <- phyloseq::distance (subset\_samples(real\_data,Compartment != c("Rhizosphere")),distance)

map <- data.frame(sample\_data(subset\_samples(real\_data,Compartment != c("Rhizosphere"))))

adonis(formula=BC~DevStage,data=map,permutations = 999, method = distance)

BC.ano<-anosim(BC, data.frame(map)$DevStage, permutations = 999, distance , strata = NULL, parallel = getOption("mc.cores"))

BC.ano

############## Differential Abundance Analysis Developmental Stage #####################

real\_data<-merge\_phyloseq(map,otus)

otutable <- data.frame(otu\_table(subset\_samples(real\_data,Compartment == ("Rhizosphere"))))

head(otutable)

otutable <- otutable[rowSums(cpm(otutable) > 0) >= 2,]

dim(otutable)

head(otutable)

grp<- as.factor(paste(sample\_data(subset\_samples(real\_data,Compartment == ("Rhizosphere")))$Compartment,sample\_data(subset\_samples(real\_data,Compartment == ("Rhizosphere")))$DevStage,sep="."))

grp

dge <- DGEList(counts = otutable, group = grp)

dge <- calcNormFactors(dge)

dge <- estimateCommonDisp(dge)

dge

de12 <- exactTest(dge, pair=c(1,2) , dispersion = dge$common.dispersion)

de13 <- exactTest(dge, pair=c(1,3) , dispersion = dge$common.dispersion)

de23 <- exactTest(dge, pair=c(2,3) , dispersion = dge$common.dispersion)

de <-list(de12, de13, de23)

for (a in de) {

b <- a$comparison

de\_tab <- topTags(a, n=Inf)$table

de\_tab <- signif(de\_tab, digits = 3)

de\_tab$FDR <- p.adjust(de\_tab$PValue, method="BH")

head(de\_tab)

deOTUs <- rownames(de\_tab)[de\_tab$FDR < FDR\_thres & abs(de\_tab$logFC) > logFC\_thres]

pos\_de\_tab <- de\_tab[(de\_tab$FDR < FDR\_thres & abs(de\_tab$logFC) >= logFC\_thres & de\_tab$logFC > 0),]

if (nrow(pos\_de\_tab) != 0){

v<-rownames(tax\_table(prune\_taxa(rownames(pos\_de\_tab), real\_data)))

pos\_de\_tab<-as.data.frame(pos\_de\_tab)[v,]

pos\_de\_tab <- cbind (pos\_de\_tab,tax\_table(prune\_taxa(rownames(pos\_de\_tab), real\_data)), stringsAsFactors = FALSE)

pos\_de\_tab <- cbind (pos\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(pos\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[1],start=13,stop=nchar(b[1]))))))), stringsAsFactors = FALSE)

pos\_de\_tab <- cbind (pos\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(pos\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[2],start=13,stop=nchar(b[2]))))))), stringsAsFactors = FALSE)

pos\_de\_tab <- cbind (pos\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(pos\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[1],start=13,stop=nchar(b[1])))))))/sum(otu\_table((subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[1],start=13,stop=nchar(b[1]))))))), stringsAsFactors = FALSE)

pos\_de\_tab <- cbind (pos\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(pos\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[2],start=13,stop=nchar(b[2])))))))/sum(otu\_table((subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[2],start=13,stop=nchar(b[2]))))))), stringsAsFactors = FALSE)

colnames(pos\_de\_tab)[12]<-paste0("Abundance\_Rhiz\_",c(substr(b[1],start=13,stop=nchar(b[1]))))

colnames(pos\_de\_tab)[13]<-paste0("Abundance\_Rhiz\_",c(substr(b[2],start=13,stop=nchar(b[2]))))

colnames(pos\_de\_tab)[14]<-paste0("Abundance\_Rhiz\_","Pod-filling stage")

colnames(pos\_de\_tab)[15]<-paste0("Abundance\_Rhiz\_rel\_",c(substr(b[1],start=13,stop=nchar(b[1]))))

pos\_de\_tab<- pos\_de\_tab[with(pos\_de\_tab, order(-pos\_de\_tab$logFC)), ]

print (pos\_de\_tab)} else {pos\_de\_tab<- "None"

print(pos\_de\_tab)}

neg\_de\_tab <- de\_tab[(de\_tab$FDR < FDR\_thres & abs(de\_tab$logFC) >= logFC\_thres & de\_tab$logFC < 0),]

if (nrow(neg\_de\_tab) != 0){

v<-rownames(tax\_table(prune\_taxa(rownames(neg\_de\_tab), real\_data)))

neg\_de\_tab<-as.data.frame(neg\_de\_tab)[v,]

neg\_de\_tab <- cbind (neg\_de\_tab,tax\_table(prune\_taxa(rownames(neg\_de\_tab), real\_data)), stringsAsFactors = FALSE)

neg\_de\_tab <- cbind (neg\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(neg\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[1],start=13,stop=nchar(b[1]))))))), stringsAsFactors = FALSE)

neg\_de\_tab <- cbind (neg\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(neg\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[2],start=13,stop=nchar(b[2]))))))), stringsAsFactors = FALSE)

neg\_de\_tab <- cbind (neg\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(neg\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[1],start=13,stop=nchar(b[1])))))))/sum(otu\_table((subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[1],start=13,stop=nchar(b[1]))))))), stringsAsFactors = FALSE)

neg\_de\_tab <- cbind (neg\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(neg\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[2],start=13,stop=nchar(b[2])))))))/sum(otu\_table((subset\_samples(subset\_samples(real\_data,Compartment == c("Rhizosphere")),DevStage == c(substr(b[2],start=13,stop=nchar(b[2]))))))), stringsAsFactors = FALSE)

colnames(neg\_de\_tab)[12]<-paste0("Abundance\_Rhiz\_",c(substr(b[1],start=13,stop=nchar(b[1]))))

colnames(neg\_de\_tab)[13]<-paste0("Abundance\_Rhiz\_",c(substr(b[2],start=13,stop=nchar(b[2]))))

colnames(neg\_de\_tab)[14]<-paste0("Abundance\_Rhiz\_","Pod-filling stage")

colnames(neg\_de\_tab)[15]<-paste0("Abundance\_Rhiz\_rel\_",c(substr(b[1],start=13,stop=nchar(b[1]))))

neg\_de\_tab<- neg\_de\_tab[with(neg\_de\_tab, order(neg\_de\_tab$logFC)), ]

print (neg\_de\_tab)} else {neg\_de\_tab<- "None"

print(neg\_de\_tab)}

# matrix <- list(pos\_de\_tab)

# matrix\_filename <- paste0("Rhiz\_",c(substr(b[1],start=13,stop=nchar(b[1]))),"\_vs\_","Rhiz\_",c(substr(b[2],start=13,stop=nchar(b[2]))),".xlsx")

# write.xlsx(matrix, matrix\_filename, sheetName = paste0("Rhiz\_",c(substr(b[1],start=13,stop=nchar(b[1]))),"\_enriched"))

# matrix<- list(neg\_de\_tab)

# write.xlsx (matrix, matrix\_filename, sheetName = paste0("Rhiz\_",c(substr(b[2],start=13,stop=nchar(b[2]))),"\_enriched"), append = T)

plotSmear(dge, pair=c(b[[1]],b[[2]]), de.tags=deOTUs, cex = 0.65, ylab= "logFC")

title <- paste0("LogFC:Rhiz", c(substr(b[1],start=13,stop=nchar(b[1]))), "-Rhiz.", c(substr(b[2],start=13,stop=nchar(b[2]))))

plotSmear(a, de.tags=deOTUs, ylab = title, cex = 0.65)

}

otutable <- data.frame(otu\_table(subset\_samples(real\_data,Compartment == ("Endosphere"))))

head(otutable)

otutable <- otutable[rowSums(cpm(otutable) > 0) >= 2,]

dim(otutable)

head(otutable)

grp<- as.factor(paste(sample\_data(subset\_samples(real\_data,Compartment == ("Endosphere")))$Compartment,sample\_data(subset\_samples(real\_data,Compartment == ("Endosphere")))$DevStage,sep="."))

grp

dge <- DGEList(counts = otutable, group = grp)

dge <- calcNormFactors(dge)

dge <- estimateCommonDisp(dge)

dge

de12 <- exactTest(dge, pair=c(1,2) , dispersion = dge$common.dispersion)

de13 <- exactTest(dge, pair=c(1,3) , dispersion = dge$common.dispersion)

de23 <- exactTest(dge, pair=c(2,3) , dispersion = dge$common.dispersion)

de <-list(de12, de13, de23)

for (a in de) {

b <- a$comparison

de\_tab <- topTags(a, n=Inf)$table

de\_tab <- signif(de\_tab, digits = 3)

de\_tab$FDR <- p.adjust(de\_tab$PValue, method="BH")

head(de\_tab)

deOTUs <- rownames(de\_tab)[de\_tab$FDR < FDR\_thres & abs(de\_tab$logFC) > logFC\_thres]

pos\_de\_tab <- de\_tab[(de\_tab$FDR < FDR\_thres & abs(de\_tab$logFC) >= logFC\_thres & de\_tab$logFC > 0),]

if (nrow(pos\_de\_tab) != 0){

v<-rownames(tax\_table(prune\_taxa(rownames(pos\_de\_tab), real\_data)))

pos\_de\_tab<-as.data.frame(pos\_de\_tab)[v,]

pos\_de\_tab <- cbind (pos\_de\_tab,tax\_table(prune\_taxa(rownames(pos\_de\_tab), real\_data)), stringsAsFactors = FALSE)

pos\_de\_tab <- cbind (pos\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(pos\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[1],start=12,stop=nchar(b[1]))))))), stringsAsFactors = FALSE)

pos\_de\_tab <- cbind (pos\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(pos\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[2],start=12,stop=nchar(b[2]))))))), stringsAsFactors = FALSE)

pos\_de\_tab <- cbind (pos\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(pos\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[1],start=12,stop=nchar(b[1])))))))/sum(otu\_table((subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[1],start=12,stop=nchar(b[1]))))))), stringsAsFactors = FALSE)

pos\_de\_tab <- cbind (pos\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(pos\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[2],start=12,stop=nchar(b[2])))))))/sum(otu\_table((subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[2],start=12,stop=nchar(b[2]))))))), stringsAsFactors = FALSE)

colnames(pos\_de\_tab)[12]<-paste0("Abundance\_Endo\_",c(substr(b[1],start=12,stop=nchar(b[1]))))

colnames(pos\_de\_tab)[13]<-paste0("Abundance\_Endo\_",c(substr(b[2],start=12,stop=nchar(b[2]))))

colnames(pos\_de\_tab)[14]<-paste0("Abundance\_Endo\_rel\_",c(substr(b[1],start=12,stop=nchar(b[1]))))

colnames(pos\_de\_tab)[15]<-paste0("Abundance\_Endo\_rel\_",c(substr(b[2],start=12,stop=nchar(b[2]))))

pos\_de\_tab<- pos\_de\_tab[with(pos\_de\_tab, order(-pos\_de\_tab$logFC)), ]

print (pos\_de\_tab)} else {pos\_de\_tab<- "None"

print(pos\_de\_tab)}

neg\_de\_tab <- de\_tab[(de\_tab$FDR < FDR\_thres & abs(de\_tab$logFC) >= logFC\_thres & de\_tab$logFC < 0),]

if (nrow(neg\_de\_tab) != 0){

v<-rownames(tax\_table(prune\_taxa(rownames(neg\_de\_tab), real\_data)))

neg\_de\_tab<-as.data.frame(neg\_de\_tab)[v,]

neg\_de\_tab <- cbind (neg\_de\_tab,tax\_table(prune\_taxa(rownames(neg\_de\_tab), real\_data)), stringsAsFactors = FALSE)

neg\_de\_tab <- cbind (neg\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(neg\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[1],start=12,stop=nchar(b[1]))))))), stringsAsFactors = FALSE)

neg\_de\_tab <- cbind (neg\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(neg\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[2],start=12,stop=nchar(b[2]))))))), stringsAsFactors = FALSE)

neg\_de\_tab <- cbind (neg\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(neg\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[1],start=12,stop=nchar(b[1])))))))/sum(otu\_table((subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[1],start=12,stop=nchar(b[1]))))))), stringsAsFactors = FALSE)

neg\_de\_tab <- cbind (neg\_de\_tab,rowSums(otu\_table(prune\_taxa(rownames(neg\_de\_tab), subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[2],start=12,stop=nchar(b[2])))))))/sum(otu\_table((subset\_samples(subset\_samples(real\_data,Compartment == c("Endosphere")),DevStage == c(substr(b[2],start=12,stop=nchar(b[2]))))))), stringsAsFactors = FALSE)

colnames(neg\_de\_tab)[12]<-paste0("Abundance\_Endo\_",c(substr(b[1],start=12,stop=nchar(b[1]))))

colnames(neg\_de\_tab)[13]<-paste0("Abundance\_Endo\_",c(substr(b[2],start=12,stop=nchar(b[2]))))

colnames(neg\_de\_tab)[14]<-paste0("Abundance\_Endo\_rel\_",c(substr(b[1],start=12,stop=nchar(b[1]))))

colnames(neg\_de\_tab)[15]<-paste0("Abundance\_Endo\_rel\_",c(substr(b[2],start=12,stop=nchar(b[2]))))

neg\_de\_tab<- neg\_de\_tab[with(neg\_de\_tab, order(neg\_de\_tab$logFC)), ]

print (neg\_de\_tab)} else {neg\_de\_tab<- "None"

print(neg\_de\_tab)}

# matrix <- list(pos\_de\_tab)

# matrix\_filename <- paste0("Endo\_",c(substr(b[1],start=12,stop=nchar(b[1]))),"\_vs\_","Endo\_",c(substr(b[2],start=12,stop=nchar(b[2]))),".xlsx")

# write.xlsx (matrix, matrix\_filename, sheetName = paste0("Endo\_",c(substr(b[1],start=12,stop=nchar(b[1]))),"\_enriched"))

# matrix<- list(neg\_de\_tab)

# write.xlsx (matrix, matrix\_filename, sheetName = paste0("Endo\_",c(substr(b[2],start=12,stop=nchar(b[2]))),"\_enriched"), append = T)

plotSmear(dge, pair=c(b[[1]],b[[2]]), de.tags=deOTUs, cex = 0.65, ylab= "logFC")

title <- paste0("LogFC:Endo.", c(substr(b[1],start=12,stop=nchar(b[1]))), "-Endo.", c(substr(b[2],start=12,stop=nchar(b[2]))))

plotSmear(a, de.tags=deOTUs, ylab = title, cex = 0.65)

}

######################### Rarefaction curves ######################

sample\_sums(real\_data)

set.seed(42)

calculate\_rarefaction\_curves <- function(real\_data, measures, depths) {

require('plyr') # ldply

require('reshape2') # melt

estimate\_rarified\_richness <- function(real\_data, measures, depth) {

if(max(sample\_sums(real\_data)) < depth) return()

real\_data <- prune\_samples(sample\_sums(real\_data) >= depth, real\_data)

rarified\_real\_data <- rarefy\_even\_depth(real\_data, depth, verbose = FALSE)

alpha\_diversity <- estimate\_richness(rarified\_real\_data, measures = measures)

# as.matrix forces the use of melt.array, which includes the Sample names (rownames)

molten\_alpha\_diversity <- melt(as.matrix(alpha\_diversity), varnames = c('Sample', 'Measure'), value.name = 'Alpha\_diversity')

molten\_alpha\_diversity

}

names(depths) <- depths # this enables automatic addition of the Depth to the output by ldply

rarefaction\_curve\_data <- ldply(depths, estimate\_rarified\_richness, real\_data = real\_data, measures = measures, .id = 'Depth', .progress = ifelse(interactive(), 'text', 'none'))

# convert Depth from factor to numeric

rarefaction\_curve\_data$Depth <- as.numeric(levels(rarefaction\_curve\_data$Depth))[rarefaction\_curve\_data$Depth]

#original rarefaction\_curve\_data$Depth <- as.numeric(unique(rarefaction\_curve\_data$Depth))[rarefaction\_curve\_data$Depth]

rarefaction\_curve\_data

}

rarefaction\_curve\_data <- calculate\_rarefaction\_curves(real\_data,c('Observed'),rep(c(1000,1:110\*10000), each=3))

summary(rarefaction\_curve\_data)

rarefaction\_curve\_data\_summary <- ddply(rarefaction\_curve\_data, c('Depth', 'Sample', 'Measure'), summarise, Alpha\_diversity\_mean = mean(Alpha\_diversity), Alpha\_diversity\_sd = sd(Alpha\_diversity))

head(rarefaction\_curve\_data\_summary$Sample)

head(rarefaction\_curve\_data\_summary)

sample\_data(real\_data)$X.SampleID

rarefaction\_curve\_data\_summary$Sample<-sub("\\.","-",data.frame(rarefaction\_curve\_data\_summary)$Sample)

rarefaction\_curve\_data\_summary\_verbose <- merge(rarefaction\_curve\_data\_summary, data.frame(sample\_data(real\_data)), by.x= "Sample", by.y = "X.SampleID",all=T)

colnames(rarefaction\_curve\_data\_summary\_verbose)[4]<-"Richness"

head(rarefaction\_curve\_data\_summary\_verbose)

ggplot(data = rarefaction\_curve\_data\_summary\_verbose,mapping = aes(x = Depth,y = Richness, ymin = Richness - Alpha\_diversity\_sd, ymax = Richness + Alpha\_diversity\_sd, colour = Compartment, group = Sample)) + theme\_bw() +

geom\_line() + theme(legend.text = element\_text(size=18),axis.title.x=element\_text(size=18),axis.text.x=element\_text(size=18, colour = "black"),axis.title.y=element\_text(size=18),axis.text.y=element\_text(size=18, colour = "black"),plot.title = element\_text(hjust=0.475), title = element\_text(size=18), legend.title = element\_text(size=18)) +

ggtitle ("Rarefaction Curve") + geom\_pointrange() + facet\_wrap(facets = ~ Measure, scales = 'free\_y')+ scale\_color\_manual(values=c("#ED8141","#69b3a2")) + theme(strip.background = element\_blank(),strip.text= element\_blank())+ guides(colour = guide\_legend(reverse=TRUE))