Microbiome analysis

Dattatray S. Mongad

25/04/2020

setwd('/BioVolume/Dropbox/Book\_chapter\_microbiome\_datta\_nikki/data/analysis/')  
library(phyloseq)  
# DADA2  
ps.dada2 <- readRDS(file = '../dada2/ps\_dada2.rds')  
ps.dada2

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 232 taxa and 19 samples ]  
## sample\_data() Sample Data: [ 19 samples by 2 sample variables ]  
## tax\_table() Taxonomy Table: [ 232 taxa by 7 taxonomic ranks ]  
## phy\_tree() Phylogenetic Tree: [ 232 tips and 230 internal nodes ]  
## refseq() DNAStringSet: [ 232 reference sequences ]

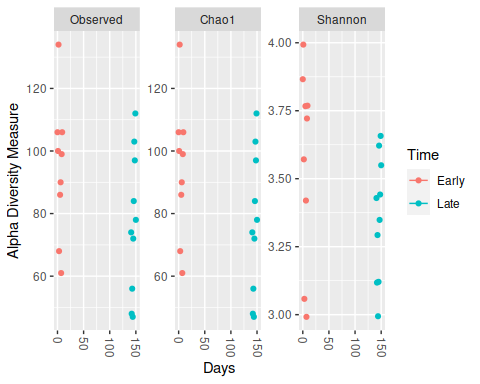
# QIIME1  
ps.qiime1 <- readRDS(file = '../qiime1/ps\_qiime1.rds')  
ps.qiime1

## phyloseq-class experiment-level object  
## otu\_table() OTU Table: [ 779 taxa and 19 samples ]  
## sample\_data() Sample Data: [ 19 samples by 5 sample variables ]  
## tax\_table() Taxonomy Table: [ 779 taxa by 7 taxonomic ranks ]  
## phy\_tree() Phylogenetic Tree: [ 779 tips and 778 internal nodes ]  
## refseq() DNAStringSet: [ 779 reference sequences ]

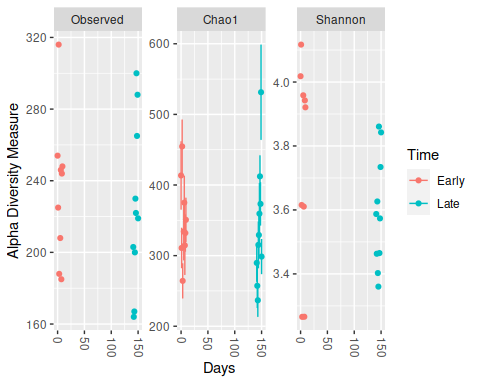
p1 <- plot\_richness(physeq = ps.dada2,x = 'Days',color = 'Time',measures = c('Observed','Chao1','Shannon'))

## Warning in estimate\_richness(physeq, split = TRUE, measures = measures): The data you have provided does not have  
## any singletons. This is highly suspicious. Results of richness  
## estimates (for example) are probably unreliable, or wrong, if you have already  
## trimmed low-abundance taxa from the data.  
##   
## We recommended that you find the un-trimmed data and retry.

print(p1)



p2 <- plot\_richness(physeq = ps.qiime1,x = 'Days',color = 'Time',measures = c('Observed','Chao1','Shannon'))  
print(p2)



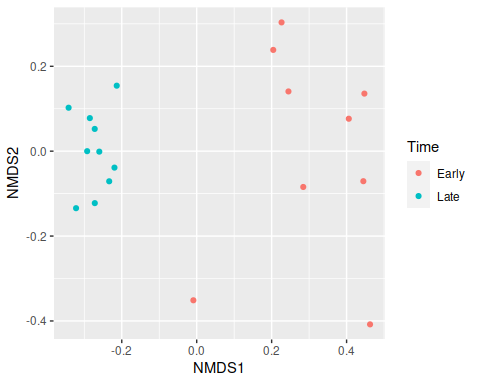
# Transform data to relative proportions  
ps.dada2.prop <- transform\_sample\_counts(physeq = ps.dada2,function(x) x/sum(x))  
ps.qiime1.prop <- transform\_sample\_counts(ps.qiime1,function(x) x/sum(x))  
  
# Ordination using bray-curtis distance metric  
ps.dada2.ord <- ordinate(physeq = ps.dada2.prop,method = 'NMDS',distance = 'bray')

## Run 0 stress 0.08072627   
## Run 1 stress 0.1232732   
## Run 2 stress 0.08602659   
## Run 3 stress 0.08602578   
## Run 4 stress 0.1214508   
## Run 5 stress 0.08071735   
## ... New best solution  
## ... Procrustes: rmse 0.001603428 max resid 0.004828601   
## ... Similar to previous best  
## Run 6 stress 0.08602697   
## Run 7 stress 0.0860258   
## Run 8 stress 0.1320382   
## Run 9 stress 0.08072352   
## ... Procrustes: rmse 0.001107197 max resid 0.003310325   
## ... Similar to previous best  
## Run 10 stress 0.08602578   
## Run 11 stress 0.1214508   
## Run 12 stress 0.08602578   
## Run 13 stress 0.09462684   
## Run 14 stress 0.09462676   
## Run 15 stress 0.08072439   
## ... Procrustes: rmse 0.00126025 max resid 0.003777451   
## ... Similar to previous best  
## Run 16 stress 0.09462673   
## Run 17 stress 0.09462672   
## Run 18 stress 0.08995025   
## Run 19 stress 0.08995063   
## Run 20 stress 0.08995026   
## \*\*\* Solution reached

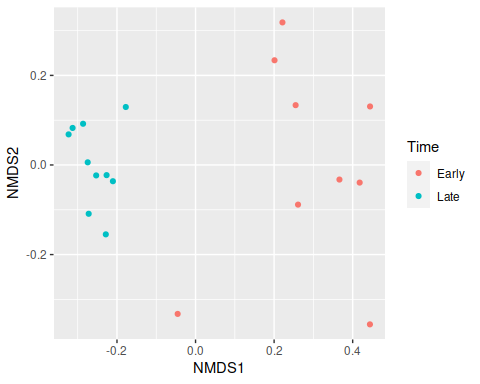
ps.qiime1.ord <- ordinate(physeq = ps.qiime1.prop,method = 'NMDS',distance = 'bray')

## Run 0 stress 0.07454608   
## Run 1 stress 0.07819846   
## Run 2 stress 0.07454608   
## ... Procrustes: rmse 3.73751e-06 max resid 8.889908e-06   
## ... Similar to previous best  
## Run 3 stress 0.1194227   
## Run 4 stress 0.147418   
## Run 5 stress 0.07819846   
## Run 6 stress 0.0926111   
## Run 7 stress 0.07454608   
## ... Procrustes: rmse 2.797258e-06 max resid 6.774762e-06   
## ... Similar to previous best  
## Run 8 stress 0.07454608   
## ... Procrustes: rmse 2.367982e-06 max resid 5.662184e-06   
## ... Similar to previous best  
## Run 9 stress 0.07454608   
## ... Procrustes: rmse 4.160688e-06 max resid 8.833841e-06   
## ... Similar to previous best  
## Run 10 stress 0.09380748   
## Run 11 stress 0.142169   
## Run 12 stress 0.09381271   
## Run 13 stress 0.07819846   
## Run 14 stress 0.09381258   
## Run 15 stress 0.09380999   
## Run 16 stress 0.07454608   
## ... New best solution  
## ... Procrustes: rmse 1.658545e-06 max resid 4.456316e-06   
## ... Similar to previous best  
## Run 17 stress 0.07819846   
## Run 18 stress 0.09381185   
## Run 19 stress 0.07831371   
## Run 20 stress 0.07454608   
## ... New best solution  
## ... Procrustes: rmse 1.120042e-06 max resid 1.914227e-06   
## ... Similar to previous best  
## \*\*\* Solution reached

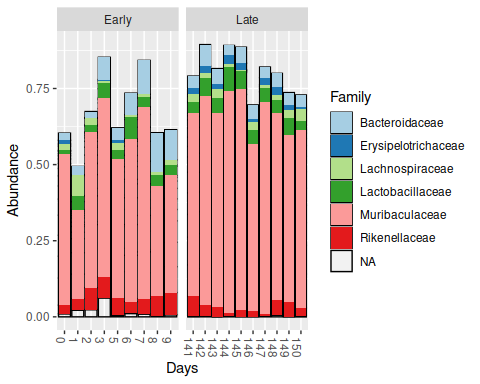
# ordination plots  
p1 <- plot\_ordination(physeq = ps.dada2.prop,ordination = ps.dada2.ord,color = 'Time')  
print(p1)



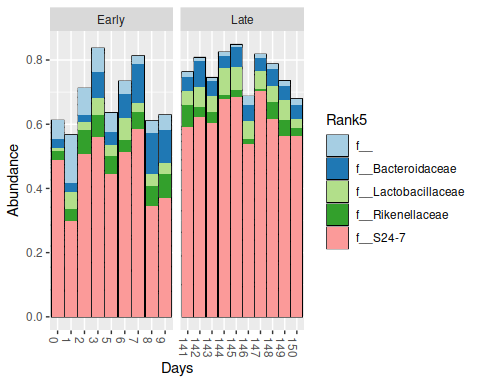
p2 <- plot\_ordination(physeq = ps.qiime1.prop,ordination = ps.qiime1.ord,color = 'Time')  
print(p2)



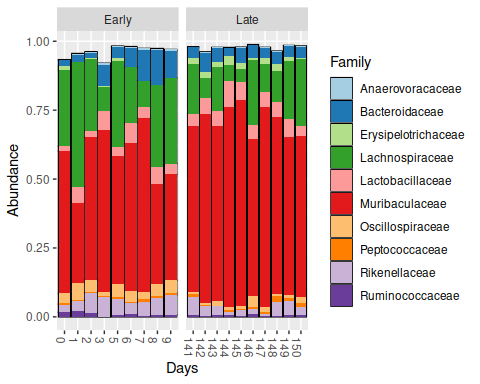
# Change type of data and levels for Days column in sample metadata  
# This step is just to change the order of Days in sample metadata. If you don't want to change the order, you can skip this step  
sample\_data(ps.dada2.prop)$Days <- factor(as.character(sample\_data(ps.dada2.prop)$Days),levels = c("0","1","2","3","5","6","7","8","9","141","142","143","144","145","146","147","148","149","150"))  
sample\_data(ps.qiime1.prop)$Days <- factor(as.character(sample\_data(ps.qiime1.prop)$Days),levels = c("0","1","2","3","5","6","7","8","9","141","142","143","144","145","146","147","148","149","150"))  
  
# plotting abundance of top20 OTUs/ASVs  
library(ggplot2)  
top20 <- names(sort(taxa\_sums(ps.dada2.prop),decreasing = T))[1:20]  
ps.dada2.top20 <- prune\_taxa(top20,ps.dada2.prop)  
p1 <- plot\_bar(physeq = ps.dada2.top20,x = 'Days',fill = 'Family') + facet\_wrap(~Time,scales = 'free\_x') + geom\_bar(stat='identity') + scale\_fill\_brewer(palette = 'Paired')  
print(p1)



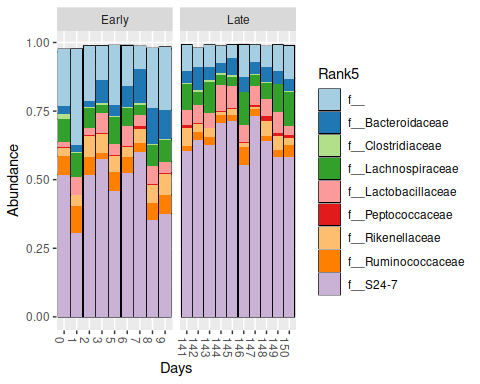
top20 <- names(sort(taxa\_sums(ps.qiime1.prop),decreasing = T))[1:20]  
ps.qiime1.top20 <- prune\_taxa(top20,ps.qiime1.prop)  
#sample\_data(ps.qiime1.top20)$Days <- as.character(sample\_data(ps.qiime1.top20)$Days)  
p2 <- plot\_bar(physeq = ps.qiime1.top20,x = 'Days',fill = 'Rank5') + facet\_wrap(~Time, scales="free\_x") + geom\_bar(stat = 'identity') + scale\_fill\_brewer(palette = 'Paired')  
print(p2)



# Here we will consolidate the abundances of each OTUs/ASVs at Family level and then plot top10 Families.  
ps.dada2.prop.glom <- tax\_glom(ps.dada2.prop,taxrank = 'Family')  
top10 <- names(sort(taxa\_sums(ps.dada2.prop.glom),decreasing = T))[1:10]  
ps.dada2.top10 <- prune\_taxa(top10,ps.dada2.prop.glom)  
p1 <- plot\_bar(physeq = ps.dada2.top10,x = 'Days',fill = 'Family') + facet\_wrap(~Time,scales = 'free\_x') + geom\_bar(stat='identity') + scale\_fill\_brewer(palette = 'Paired')  
print(p1)



ps.qiime1.prop.glom <- tax\_glom(ps.qiime1.prop,taxrank = 'Rank5')  
top10 <- names(sort(taxa\_sums(ps.qiime1.prop.glom),decreasing = T))[1:10]  
ps.qiime.top10 <- prune\_taxa(top10,ps.qiime1.prop.glom)  
p2 <- plot\_bar(physeq = ps.qiime.top10,x = 'Days',fill = 'Rank5') + facet\_wrap(~Time,scales = 'free\_x') + geom\_bar(stat='identity') + scale\_fill\_brewer(palette = 'Paired')  
print(p2)



library(ALDEx2)  
# DADA2  
otu.tab <- t(otu\_table(ps.dada2))  
conds <- sample\_data(ps.dada2)$Time  
# Run aldex function  
x.all <- aldex(reads = otu.tab,conditions = conds)

## aldex.clr: generating Monte-Carlo instances and clr values

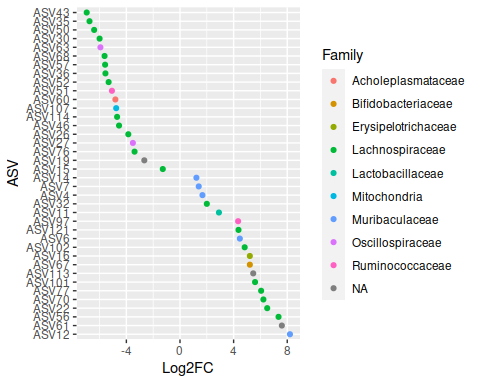
## operating in serial mode

## computing center with all features

## aldex.ttest: doing t-test

## aldex.effect: calculating effect sizes

x.all <- x.all[x.all$wi.eBH<=0.05,] # filter significant ASVs  
# add taxonomy to result table  
x.all <- merge(x.all,tax\_table(ps.dada2),by=0)  
x.all <- x.all[order(x.all$diff.btw,decreasing = T),] # sort table by log2FC  
x.all$Row.names <- factor(x = x.all$Row.names,levels = x.all$Row.names)  
#plot  
p1 <- ggplot(data = x.all,mapping = aes(x = diff.btw,y = Row.names, color=Family)) + geom\_point() + xlab('Log2FC') + ylab('ASV')  
print(p1)



# QIIME1  
otu.tab <- otu\_table(ps.qiime1)  
conds <- sample\_data(ps.qiime1)$Time  
# Run aldex function  
x.all <- aldex(reads = otu.tab,conditions = conds)

## aldex.clr: generating Monte-Carlo instances and clr values

## operating in serial mode

## computing center with all features

## aldex.ttest: doing t-test

## aldex.effect: calculating effect sizes

x.all <- x.all[x.all$wi.eBH<=0.05,] # filter significant ASVs  
# add taxonomy to result table  
x.all <- merge(x.all,tax\_table(ps.qiime1),by=0)  
x.all <- x.all[order(x.all$diff.btw,decreasing = T),] # sort table by log2FC  
x.all$Row.names <- factor(x = x.all$Row.names,levels = x.all$Row.names)  
#plot  
p2 <- ggplot(data = x.all,mapping = aes(x = diff.btw,y = Row.names, color=Rank5)) + geom\_point() + xlab('Log2FC') + ylab('OTU')  
print(p2)

