W3. ITS analysis

Bora Kim

March 5, 2021

This is meant for sharing dataset and script of the publication "Strigolactone_structural specificity in microbiome recruitment in rice, 2021". This markdown contains the process of ITS amplicon sequencing from rhizosphere and roots of 16 rice genotypes grown on two natural soils for 31 days. To begin with, raw ITS amplicon sequence has been reposited in SRA database under accession number: In W3 workflow, there are 6 big main steps as: Processing sequencing data -> alpha diversity -> beta diversity -> exploring Glomeromycota -> effect of SLs on alpha diversity and beta diversity -> prepare datasets for correlation study with SLs (W4). Unfortunately, this markdown was written after processing raw sequencing data in the former expired sever from University of Amsterdam. Therefore, in this markdown, the code (this markdown), final outputs of DADA2 (R image "W3_ITS_analaysis_image.Rdata") are shared for sequencing processing step, but intermediate results won't be shown. Rest of data analysis parts with all intermediate objects and results after DADA2 step will be full shared.

1. Processing ITS amplicon sequencing (Illumina Miseq)

The primers used in this study amplified ITS region as below: Forward- CTTGGTCATTTAGAGGAAGTAA Reverse- GCTGCGTTCTTCATCGATGC

Because DNA are amplified after primer region, you would only find 5'-[your reads]-[R1 adapter, reverse-complement R2 primer, and etc]-3'. In my case, more specifically 5'-[reads]-[reverse-complement R2 primer]-[link]-[pad]-[index]-[i7 adapter]-3'. Therefore, I needed to remove reverse-complement primer and adapter at the 3'end.

Trimming ITS sequence Adapter TACTGACTGACT Read 1 trimming AdapterRead2 GCCTGCTCGACG Read 2 trimming Reverse-complement R2 primer GCATCGATGAAGAACGCAGC for R1 read trimming Reverse-complement R1 primer TTACTTCCTCTAAATGACCAAG for R2 read trimming

1.1. Remove primer and adapter sequence in raw sequencing data

I used software called 'Cutadapt' employed in linux environmnet with python. First I removed reverse-complement primer parts as:

```
for=(*R1_001.fastq.gz) #forward files
rev=(*R2_001.fastq.gz) #reverse files
for ((i=0; i<${#for[*]}; i++)) # iterates over the forward reads array
  do
  fullname=$(basename -- ${for[i]})
  sample="${fullname%_S[0-9+]*}"
  echo "processing" $sample
  trimmed_for="$sample""_R1_trim.fastq"
  trimmed_rev="$sample""_R2_trim.fastq"
  echo $trimmed_for
  cutadapt -a GCATCGATGAAGAACGCAGC -A TTACTTCCTCTAAATGACCAAG --no-indels -o $trimmed_for -p $trimmed_re
```

The outputs were moved to new directory and there, I trmmed again the adapter sequence as:

```
  for=(*R1_trim.fastq) #forward files
  rev=(*R2_trim.fastq) #reverse files
  for ((i=0; i<${#for[*]}; i++)) # iterates over the forward reads array
    do
    fullname=$(basename -- ${for[i]})
    sample="${fullname%_R[0-9+]*}"
    echo "processing" $sample
    trimmed_for="$sample""_R1_trim_trim.fastq"
    trimmed_rev="$sample""_R2_trim_trim.fastq"
    echo $trimmed_for
    cutadapt -a TACTGACTGACT -A GCCTGCTCGACG --no-indels -o $trimmed_for -p $trimmed_rev ${for[i]} ${re
    done
```

The outputs were used for next step.

1.2. Processing sequencing data using DADA2

DADA2 was performed in the same server from University of Amsterdam using R studio.

```
library("DADA2")
library("phyloseq")
```

First of all, I set the path containing trimmed files and inspect sequence quality by plotting them.

```
setwd("~/Ricebiome/rice_sequencing_process/Rice_ITS_Bora")
path<-"~/Ricebiome/rice_sequencing_process/Rice_ITS_Bora"
list.files(path)

fnFs <- sort(list.files(path, pattern="_R1_trim_trim.fastq", full.names = TRUE)) #sort forward and reve
fnRs <- sort(list.files(path, pattern="_R2_trim_trim.fastq", full.names = TRUE)) #sort forward and reve

plotQualityProfile(fnFs[4])
plotQualityProfile(fnRs[4])
```

In our study, expected amplicon length was 465bp (341-806), therefore merging forward and reverse shoul be more than 470 bp. We decided parameter for filtering (below) considering this fact and sequencing quality.

```
fnFs<-fnFs[-1] #remove negative control sample
fnRs<-fnRs[-1] #remove negative control sample

sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`, 1) # Extract sample names, assuming filename

filtFs <- file.path(path, "filtered", paste0(sample.names, "_F_filt.fastq.gz")) # Place filtered files
filtRs <- file.path(path, "filtered", paste0(sample.names, "_R_filt.fastq.gz")) # Place filtered files

names(filtFs) <- sample.names
names(filtRs) <- sample.names



out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs, maxN = 0, maxEE = c(1, 1),
                     truncQ = 2, minLen = 50, rm.phix = TRUE,
                     compress = TRUE, multithread = TRUE)  # on windows, set multithread = FALSE
```

Remove errors that was leaned based on most abundunt sequence error rate as maximum possible error rates(initial rartes for the input of machine-learning)

```
errF <- learnErrors(filtFs, multithread=TRUE)
errR <- learnErrors(filtRs, multithread=TRUE)

plotErrors(errF, nominalQ=TRUE)
plotErrors(errR, nominalQ=TRUE)

dadaFs <- dada(filtFs, err=errF, multithread=TRUE)
dadaRs <- dada(filtRs, err=errR, multithread=TRUE)

dadaFs[[1]]#inspecting dada-class object
```

Filtering low quality of sequence is done and now we merge pair-end reads. Chimera can occur duing merging, therefore remove them.

```
mergers <- mergePairs(dadaFs, filtFs, dadaRs, filtRs, verbose=TRUE)
head(mergers[[1]]) # Inspect the merger data.frame from the first sample

seqtab <- makeSequenceTable(mergers)# Construct sequence table
dim(seqtab)
table(nchar(getSequences(seqtab)))# Inspect distribution of sequence lengths

seqtab.nochim <- removeBimeraDenovo(seqtab, method="consensus", multithread=TRUE, verbose=TRUE)# remove
dim(seqtab.nochim)
sum(seqtab.nochim)/sum(seqtab)
```

Track the number of reads reads through the pipeline. By looking at it, you will see if you lost too many reads in which step.

```
getN <- function(x) sum(getUniques(x))
track <- cbind(out, sapply(dadaFs, getN), sapply(dadaRs, getN), sapply(mergers, getN), rowSums(seqtab.no
colnames(track) <- c("input", "filtered", "denoisedF", "denoisedR", "merged", "nonchim")
rownames(track) <- sample.names
head(track)
```

Everything looks okay, then assign sequence to taxonomy. Database for taxonomy annotation, I downloaded database from distributor.

```
taxa <- assignTaxonomy(seqtab.nochim,"~/Ricebiome/rice_sequencing_process/Rice_ITS_Bora/ITS_trim_primer
```

Afterwards, make small modification on sample names & taxa name, assign unique sequences to amplicon sequence variant (ASV), remove signletones.

```
sampleID<-rownames(seqtab.nochim)
sampleID <- sampleID %>% str_replace_all("-", "_") #want to change "-" in sample name to "_"
rownames(seqtab.nochim)<-sampleID

ps <- phyloseq(otu_table(seqtab.nochim, taxa_are_rows=FALSE), tax_table(taxa)) #incorporate all dataset
dna <- Biostrings::DNAStringSet(taxa_names(ps))
names(dna) <- taxa_names(ps)
ps <- merge_phyloseq(ps, dna)
taxa_names(ps) <- paste0("fASV", seq(ntaxa(ps))) # Give name to sequence as ASV__

tax <- data.frame(tax_table(ps))
for (i in 1:7){ tax[,i] <- as.character(tax[,i])}
tax[is.na(tax)] <- "Unknown" #fill missing taxa as unknown
tax_table(ps) <- as.matrix(tax)
```

```
tax.clean <- data.frame(row.names = row.names(tax),
                        Kingdom = str_replace(tax[,1], "k__",""),
                        Phylum = str_replace(tax[,2], "p__",""),
                        Class = str_replace(tax[,3], "c__",""),
                        Order = str_replace(tax[,4], "o__",""),
                        Family = str_replace(tax[,5], "f__",""),
                        Genus = str_replace(tax[,6], "g__",""),
                        Species = str_replace(tax[,7], "s__",""),
                        stringsAsFactors = FALSE)

tax_table(ps) <- as.matrix(tax.clean)
ps <- prune_taxa(taxa_sums(ps) > 1, ps) #remove singletone
```

Finally, obtaine data frame from phyloseq object: abaundance table of ASVs, taxa annotation, sequence of ASV

```
asv<-as.data.frame(otu_table(ps))
tax<-as.data.frame(tax_table(ps))
seq<-as.data.frame(refseq(ps))
```

Build essential datasets to be ready to go next section

```
SAM=sample_data(meta, errorIfNULL = T) #add metadata (that is same one used in W1. phenotype data) into
ps2 = merge_phyloseq(ps, SAM)
```

As I mentioned ealier, you can find final outputs in R work image "W3_ITS_analaysis_image.Rdata" named as ps2: phyloseq object that including all information meta: sample information, phenotype measurement (sample names on row, variables on colunm) asv: ASV abundance data frame (sample names on row, ASVs on colunm) tax: taxonomy annotation data frame (ASVs on row, taxanomic rank on colunm) seq: sequences that was assigned to each ASV (ASVs on row, sequence on colunm)

2. Getting started

Now the working envrionemnt changed from university sever to local computer.

Glimpse current datasets.

```
load("W3_ITS_analysis_image.Rdata") #To load this data image, the package 'phyloseq' is required
ps2
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 3430 taxa and 198 samples ]
## sample_data() Sample Data:       [ 198 samples by 13 sample variables ]
## tax_table()   Taxonomy Table:    [ 3430 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 3430 reference sequences ]
```

```
meta[1:5,1:5]
```

```
##          Genotype   Soil Compartment Soil_compartment Replicate
## R_A1_e     IAC165  Field        Root            Fi_RT         1
## R_A1_r     IAC165  Field Rhizosphere            Fi_RS         1
## R_A10_e    IAC165 Forest        Root            Fo_RT         5
## R_A10_r    IAC165 Forest Rhizosphere            Fo_RS         5
## R_A2_e     IAC165  Field        Root            Fi_RT         2
```

```
asv[1:5,1:5]
```

```
##         fASV1 fASV2 fASV3 fASV4 fASV5
## R_A2_e  3134    61  3537  2492     2
## R_A2_r  1089     0  2111   239     0
## R_A3_e  2241     6  2568  2718     0
## R_A3_r   723     0  1230   240     0
## R_A4_e  3149    13  2650  3252     0
```

```
tax[1:5,1:5]
```

```
##       Kingdom        Phylum            Class         Order               Family
## fASV1   Fungi     Ascomycota Sordariomycetes   Hypocreales          Nectriaceae
## fASV2   Fungi Basidiomycota          Unknown       Unknown              Unknown
## fASV3   Fungi     Ascomycota Sordariomycetes   Hypocreales          Nectriaceae
## fASV4   Fungi     Ascomycota Dothideomycetes Pleosporales              Unknown
## fASV5   Fungi Basidiomycota Tremellomycetes   Tremellales Trimorphomycetaceae
```

Load required packages

```
library(dplyr) #select, filter , join function
library(tibble) #select, filter , join function
library(phyloseq) # rarefying, PCoA plot
library(ranacapa) # rarecurve
library(ggplot2) # general plot
library(vegan) # measure alpha diversity, rarecurve, PERMANOVA, CAP, anova.cca
library(FSA)
library(rcompanion) #duun test
library(multcompView) #duun test
library(reshape2) #To melt dataframe
library(tidyr)
library(ggrepel)
library(lmPerm)
library(MASS)
```

Have a look at the data distribution of microbiome data.

```
min(colSums(asv))
```

```
## [1] 2
```

```
max(colSums(asv))
```

```
## [1] 470014
```

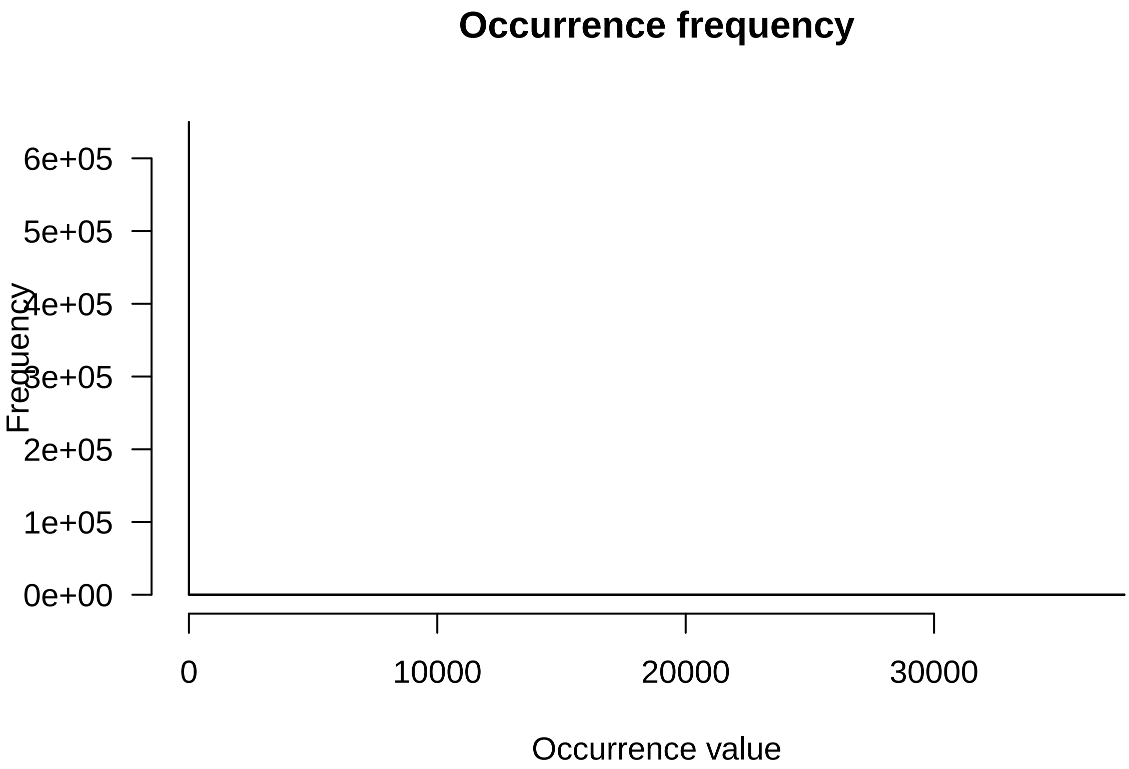
```
nsam<-dim(asv)[1] # number of samples
nvar<-dim(asv)[2] # nubmer of variables
sum(asv==0) #### Number of zeros
```

```
## [1] 650153
```

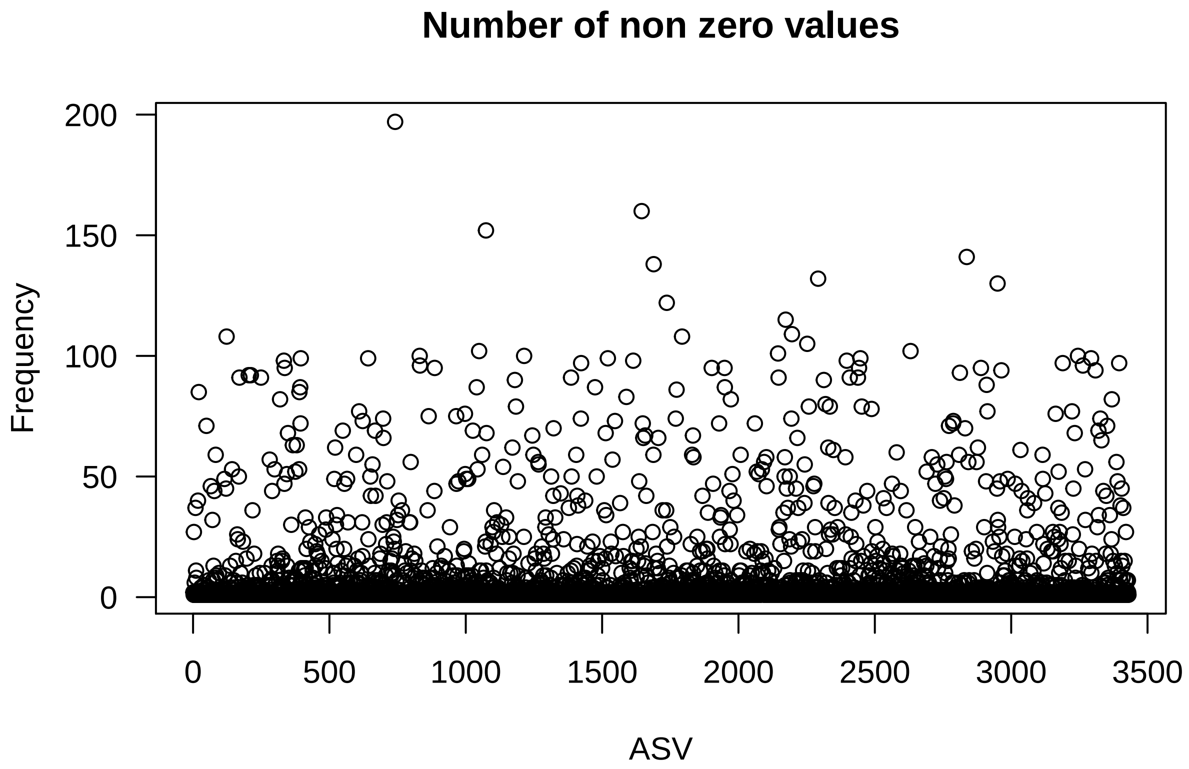
```
sum(asv==0)/(nvar*nsam)*100 #percentage of zeros
```

```
## [1] 95.73181
```

```
hist(as.matrix(asv), max(asv), right=FALSE, las=1,
     xlab = "Occurrence value", ylab = "Frequency", main = "Occurrence frequency")# Plot zeros
```



```
non_zero<-0*1:nvar
for (i in 1:nvar){non_zero[i]<-sum(asv[,i] != 0)}
plot(sample(non_zero), xlab = "ASV", ylab = "Frequency", main="Number of non zero values", las=1)# Plot
```



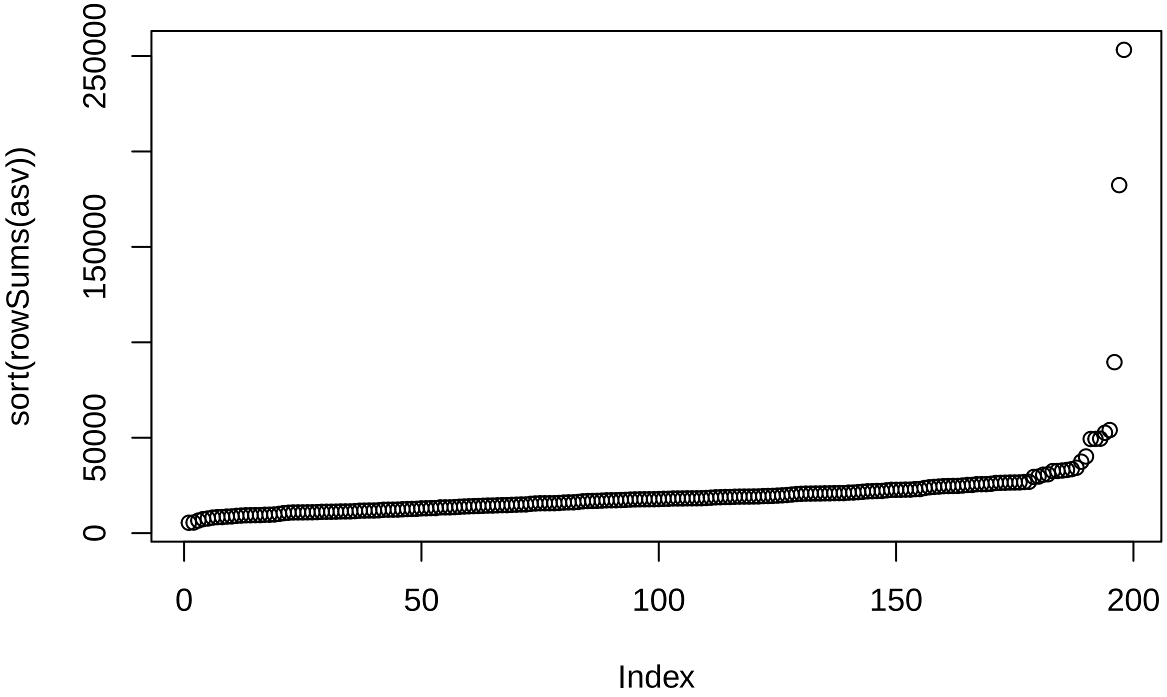
```
min(rowSums(asv)) # minimum sequencing depth in samples
```

```
## [1] 5452
```

```
max(rowSums(asv)) # maximum sequencing depth in samples
```

```
## [1] 253255
```

```
plot(sort(rowSums(asv))) #plot sequencing depth in samples
```



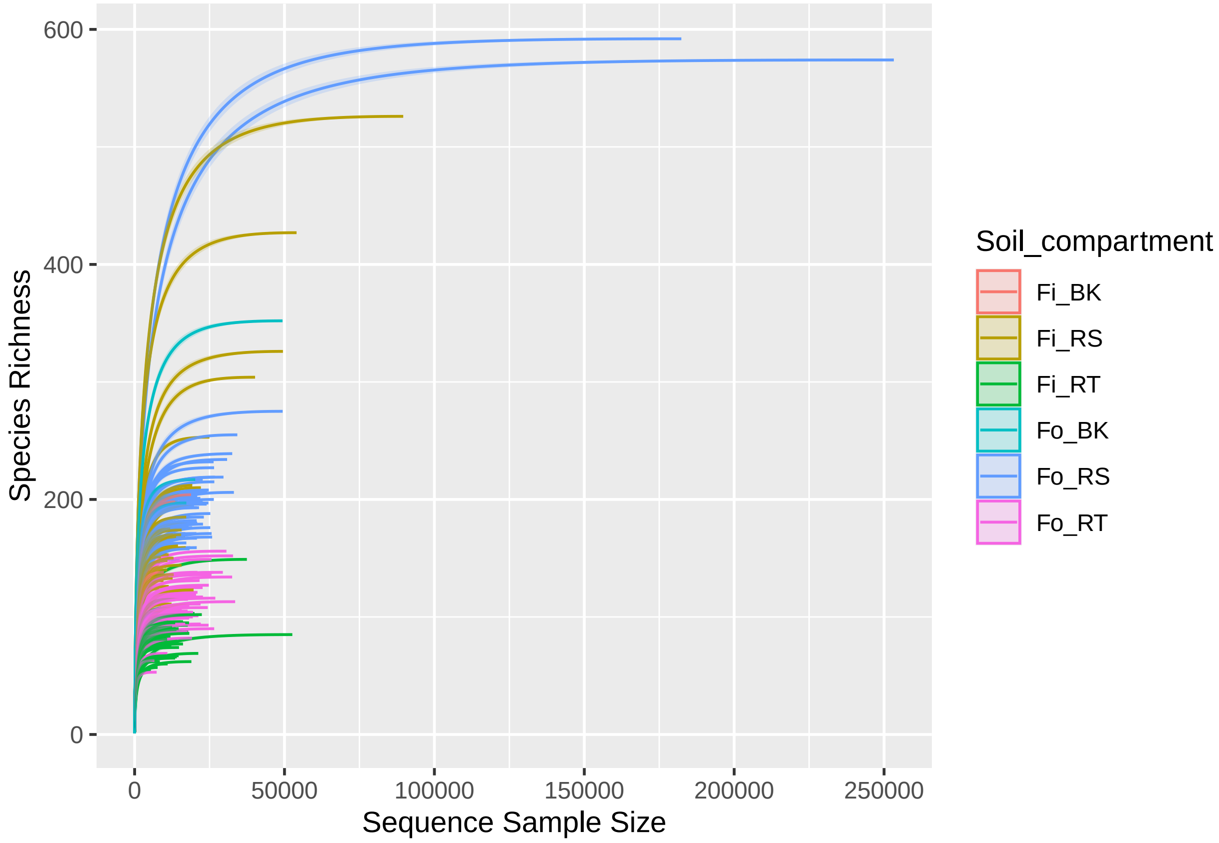
3. Alpha diversity of fungal community (Fig 2D, Fig S3)

3.1. Rarefaction curve

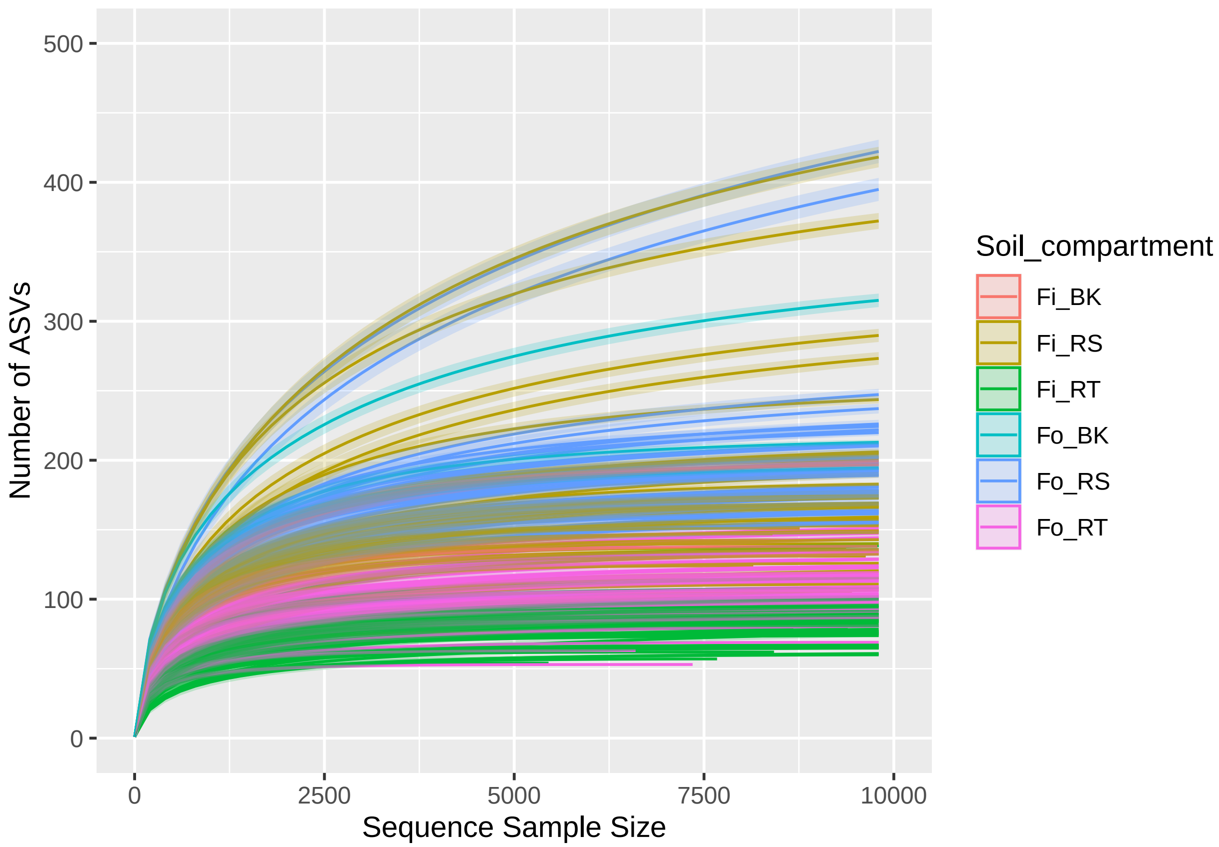
Check rarefaction curve to see if each sample reach saturated sequencing depth

```
p<- ggrare(ps2, step = 200, label = NULL, color = "Soil_compartment" ,se = TRUE)
```

```
## rarefying sample R_A2_e
## rarefying sample R_A2_r
## rarefying sample R_A3_e
## rarefying sample R_A3_r
## rarefying sample R_A4_e
## rarefying sample R_A4_r
## rarefying sample R_A7_e
## rarefying sample R_A7_r
## rarefying sample R_A8_e
## rarefying sample R_A8_r
## rarefying sample R_A9_e
## rarefying sample R_A9_r
## rarefying sample R_B2_e
## rarefying sample R_B2_r
## rarefying sample R_B3_e
## rarefying sample R_B3_r
## rarefying sample R_B4_e
## rarefying sample R_B4_r
## rarefying sample R_B7_e
## rarefying sample R_B7_r
## rarefying sample R_B8_e
## rarefying sample R_B8_r
## rarefying sample R_B9_e
## rarefying sample R_B9_r
## rarefying sample R_C2_e
## rarefying sample R_C2_r
## rarefying sample R_C3_e
## rarefying sample R_C3_r
## rarefying sample R_C4_e
## rarefying sample R_C4_r
## rarefying sample R_C7_e
## rarefying sample R_C7_r
## rarefying sample R_C8_e
## rarefying sample R_C8_r
## rarefying sample R_C9_e
## rarefying sample R_C9_r
## rarefying sample R_D10_e
## rarefying sample R_D10_r
## rarefying sample R_D2_e
## rarefying sample R_D2_r
## rarefying sample R_D3_e
## rarefying sample R_D3_r
## rarefying sample R_D4_e
## rarefying sample R_D4_r
## rarefying sample R_D8_e
## rarefying sample R_D8_r
## rarefying sample R_D9_e
## rarefying sample R_D9_r
## rarefying sample R_E2_e
## rarefying sample R_E2_r
## rarefying sample R_E3_e
## rarefying sample R_E3_r
## rarefying sample R_E5_e
## rarefying sample R_E5_r
## rarefying sample R_E7_e
## rarefying sample R_E7_r
## rarefying sample R_E8_e
## rarefying sample R_E8_r
## rarefying sample R_E9_e
## rarefying sample R_E9_r
## rarefying sample R_F10_e
## rarefying sample R_F10_r
## rarefying sample R_F2_e
## rarefying sample R_F2_r
## rarefying sample R_F3_e
## rarefying sample R_F3_r
## rarefying sample R_F4_e
## rarefying sample R_F4_r
## rarefying sample R_F7_e
## rarefying sample R_F7_r
## rarefying sample R_F9_e
## rarefying sample R_F9_r
## rarefying sample R_G10_e
## rarefying sample R_G10_r
## rarefying sample R_G2_e
## rarefying sample R_G2_r
## rarefying sample R_G3_e
## rarefying sample R_G3_r
## rarefying sample R_G5_e
## rarefying sample R_G5_r
## rarefying sample R_G8_e
## rarefying sample R_G8_r
## rarefying sample R_G9_e
## rarefying sample R_G9_r
## rarefying sample R_H2_e
## rarefying sample R_H2_r
## rarefying sample R_H3_e
## rarefying sample R_H3_r
## rarefying sample R_H4_e
## rarefying sample R_H4_r
## rarefying sample R_H7_e
## rarefying sample R_H7_r
## rarefying sample R_H8_e
## rarefying sample R_H8_r
## rarefying sample R_H9_e
## rarefying sample R_H9_r
## rarefying sample R_I2_e
## rarefying sample R_I2_r
## rarefying sample R_I3_e
## rarefying sample R_I3_r
## rarefying sample R_I4_e
## rarefying sample R_I4_r
## rarefying sample R_I7_e
## rarefying sample R_I7_r
## rarefying sample R_I8_e
## rarefying sample R_I8_r
## rarefying sample R_I9_e
## rarefying sample R_I9_r
## rarefying sample R_J2_e
## rarefying sample R_J2_r
## rarefying sample R_J3_e
## rarefying sample R_J3_r
## rarefying sample R_J4_e
## rarefying sample R_J4_r
## rarefying sample R_J7_e
## rarefying sample R_J7_r
## rarefying sample R_J8_e
## rarefying sample R_J8_r
## rarefying sample R_J9_e
## rarefying sample R_J9_r
## rarefying sample R_K2_e
## rarefying sample R_K2_r
## rarefying sample R_K3_e
## rarefying sample R_K3_r
## rarefying sample R_K4_e
## rarefying sample R_K4_r
## rarefying sample R_K7_e
## rarefying sample R_K7_r
## rarefying sample R_K8_e
## rarefying sample R_K8_r
## rarefying sample R_K9_e
## rarefying sample R_K9_r
## rarefying sample R_L2_e
## rarefying sample R_L2_r
## rarefying sample R_L3_e
## rarefying sample R_L3_r
## rarefying sample R_L4_e
## rarefying sample R_L4_r
## rarefying sample R_L7_e
## rarefying sample R_L7_r
## rarefying sample R_L8_e
## rarefying sample R_L8_r
## rarefying sample R_L9_e
## rarefying sample R_L9_r
## rarefying sample R_M2_e
## rarefying sample R_M2_r
## rarefying sample R_M3_e
## rarefying sample R_M3_r
## rarefying sample R_M4_e
## rarefying sample R_M4_r
## rarefying sample R_M7_e
## rarefying sample R_M7_r
## rarefying sample R_M8_e
## rarefying sample R_M8_r
## rarefying sample R_M9_e
## rarefying sample R_M9_r
## rarefying sample R_N1_e
## rarefying sample R_N1_r
## rarefying sample R_N3_e
## rarefying sample R_N3_r
## rarefying sample R_N5_e
## rarefying sample R_N5_r
## rarefying sample R_N7_e
## rarefying sample R_N7_r
## rarefying sample R_N8_e
## rarefying sample R_N8_r
## rarefying sample R_N9_e
## rarefying sample R_N9_r
## rarefying sample R_O10_e
## rarefying sample R_O10_r
## rarefying sample R_O2_e
## rarefying sample R_O2_r
## rarefying sample R_O3_e
## rarefying sample R_O3_r
## rarefying sample R_O5_e
## rarefying sample R_O5_r
## rarefying sample R_O8_e
## rarefying sample R_O8_r
## rarefying sample R_O9_e
## rarefying sample R_O9_r
## rarefying sample R_P2_e
## rarefying sample R_P2_r
## rarefying sample R_P3_e
## rarefying sample R_P3_r
## rarefying sample R_P5_e
## rarefying sample R_P5_r
## rarefying sample R_P7_e
## rarefying sample R_P7_r
## rarefying sample R_P8_e
## rarefying sample R_P8_r
## rarefying sample R_P9_e
## rarefying sample R_P9_r
## rarefying sample R_Q2_r
## rarefying sample R_Q3_r
## rarefying sample R_Q5_r
## rarefying sample R_Q7_r
## rarefying sample R_Q8_r
## rarefying sample R_Q9_r
```



```
p+ xlim(0, 10000)+ ylim(0, 500) + labs(y = "Number of ASVs") #adjusting x axis
```



3.2. Calculate alpha diversity indices.

```
shannon <- diversity(asv, index = "shannon") #shannon index
chaos <- as.data.frame(t(estimateR(asv)))
no.species<-chaos$S.obs
chao1<-chaos$S.chao1
evenness <- diversity(asv)/log(specnumber(asv))# Evenness index
fun_alpha<-as.data.frame(cbind(shannon, no.species, chao1, evenness, sample_data(ps2)))
fun_alpha$Compartment2<-factor(fun_alpha$Compartment,c("Bulksoil","Rhizosphere","Root"))
```

3.3. Kruskal-Wallis on alpha diversity indices

Check the effect of soil type, compartment (rhizosphere/root) on alpha diversity indices

```
indices=4 #number of alpha diversity indices that I am testing
soil.p<-0*1:indices
soil.cs<-0*1:indices
soil.df<-0*1:indices
soil.com.p<-0*1:indices
soil.com.cs<-0*1:indices
soil.com.df<-0*1:indices
names<-0*1:indices

for(i in 1:indices) {
  k<-kruskal.test(fun_alpha[,i]~fun_alpha$Soil, data=fun_alpha)
  soil.cs[i]<-k$statistic[[1]]
  soil.df[i]<-k$parameter[[1]]
  soil.p[i]<- k$p.value
  k<-kruskal.test(fun_alpha[,i]~fun_alpha$Soil_compartment, data=fun_alpha)
  soil.com.cs[i]<-k$statistic[[1]]
  soil.com.df[i]<-k$parameter[[1]]
  soil.com.p[i] <-k$p.value
  names[i]<-colnames(fun_alpha[i])}

soil.p<-p.adjust(soil.p, method = "BH")
soil.com.p<-p.adjust(soil.com.p, method = "BH")
KW.p<-cbind(names,soil.cs, soil.df, soil.p, soil.com.cs, soil.com.df, soil.com.p)
KW.p
```

```
##      names        soil.cs              soil.df soil.p
## [1,] "shannon"    "5.03355621080607"   "1"     "0.033147734492291"
## [2,] "no.species" "18.2354218031917"   "1"     "3.90428672813315e-05"
## [3,] "chao1"      "18.2354218031917"   "1"     "3.90428672813315e-05"
## [4,] "evenness"   "0.0481886014092652" "1"     "0.826245763099792"
##      soil.com.cs        soil.com.df soil.com.p
## [1,] "141.730159636567" "5"         "1.02281637074883e-28"
## [2,] "146.895304233137" "5"         "1.2226144831639e-29"
## [3,] "146.895304233137" "5"         "1.2226144831639e-29"
## [4,] "109.583231815644" "5"         "5.01848674896069e-22"
```

ake summary table of results

```
data=fun_alpha
by=data$Soil

st<-as.data.frame(matrix(NA, 2, indices))
for(i in 1:indices) {
  ag<-aggregate(data[,i]~ by, data, function(x) c(mean = mean(x), sd = sd(x)))
  agres<-as.data.frame(ag$`data[, i]`)
  agres$r.mean<-round(agres$mean,3)
  agres$r.sd<-round(agres$sd,3)
  agres$mean_sd<- paste(agres$r.mean, agres$r.sd, sep="±")
  st[,i]<-agres$mean_sd
  }

rownames(st)<-ag$by
colnames(st)<-colnames(data[1:indices])

sample_size<-as.data.frame(with(data, table(Soil)))

st$sample_size<-sample_size$Freq
st$name_size<-paste(rownames(st),st$sample_size, sep=",n=")
rownames(st)<-st$name_size

st2<-data.frame(t(st[,-(5:6)]))
st2$KW_adj.p<-KW.p[,7]
fun_alpha_summary_soil<-st2
fun_alpha_summary_soil
```

```
##                 Field.n.99   Forest.n.99             KW_adj.p
## shannon         3.195±0.58   3.378±0.472 1.02281637074883e-28
## no.species 130.838±72.396 161.96±80.306  1.2226144831639e-29
## chao1      130.838±72.396 161.96±80.306  1.2226144831639e-29
## evenness      0.666±0.077   0.673±0.056 5.01848674896069e-22
```

3.4. Dunn test on alpha diversity indices among soil_compartment group

Because index ‘se.chao1’ was not significantly different, it was excluded (otherwise, the loop stops)

```
Z<-as.data.frame(matrix(NA, 15, indices)) #results list =15
P.unadj<-as.data.frame(matrix(NA, 15, indices)) #results list =15
P.adj<-as.data.frame(matrix(NA, 15, indices)) #results list =15
Let<-as.data.frame(matrix(NA, 6, indices)) #results list =6

for(i in 1:indices) {
  PT<-dunnTest(fun_alpha[,i]~Soil_compartment, data=fun_alpha, method = "bh")
  Z[,i]<-PT$res$Z
  P.unadj[,i]<-PT$res$P.unadj
  P.adj[,i]<-PT$res$P.adj
  PT2<-PT$res
  cl<-cldList(comparison = PT2$Comparison,p.value = PT2$P.adj,threshold  = 0.05)
  Let[,i]<-cl$Letter
}

rownames(Z) <- PT$res$Comparison
colnames(Z) <- colnames(fun_alpha[1:indices])
rownames(P.unadj) <- PT$res$Comparison
colnames(P.unadj) <- colnames(fun_alpha[1:indices])
rownames(P.adj) <- PT$res$Comparison
colnames(P.adj) <- colnames(fun_alpha[1:indices])
rownames(Let) <- cl$Group
colnames(Let) <- colnames(fun_alpha[1:indices])

Let
```

```
##       shannon no.species chao1 evenness
## Fi_BK       a         ab    ab        a
## Fi_RS       a          a     a        a
## Fi_RT       b          c     c        b
## Fo_BK       a         ab    ab        a
## Fo_RS       a          b     b        a
## Fo_RT       c          d     d        b
```

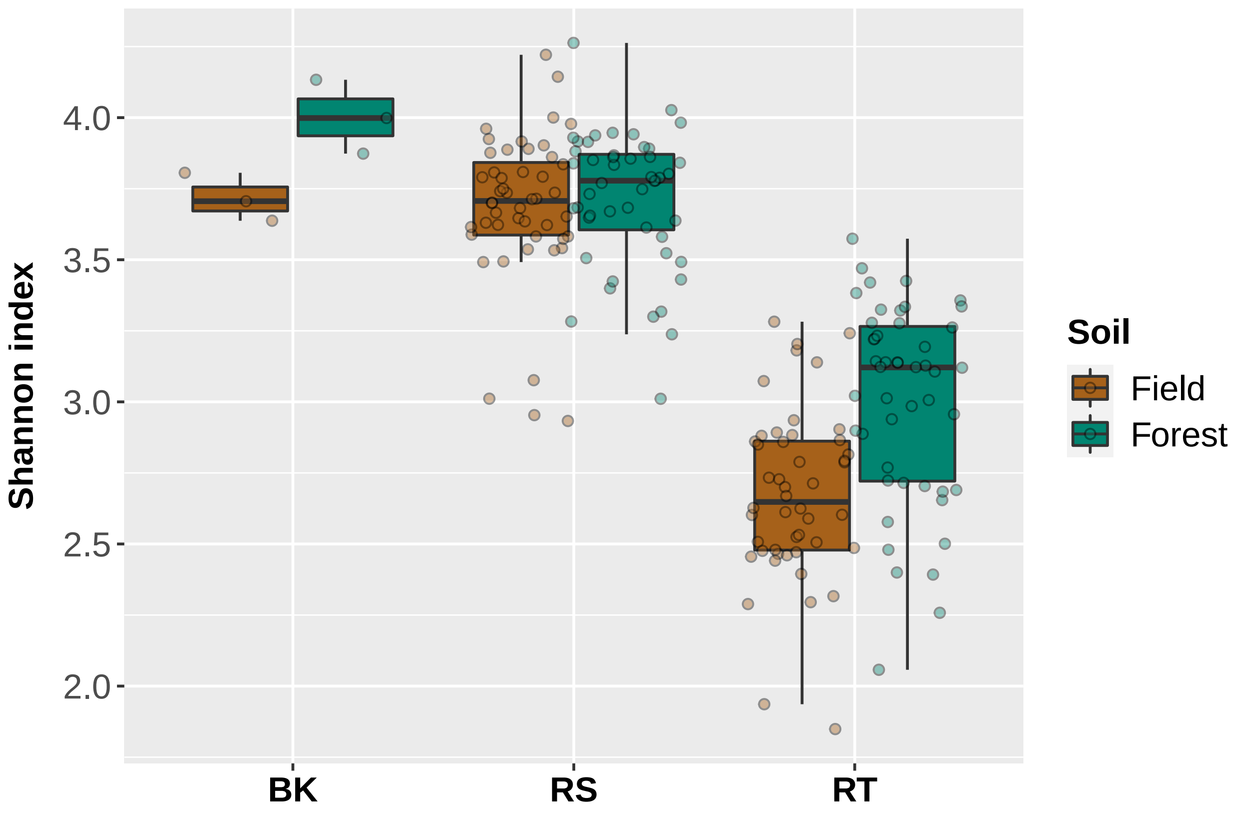
```
data=fun_alpha
by=data$Soil_compartment

st<-as.data.frame(matrix(NA, 6, indices))

for(i in 1:indices) {
  ag<-aggregate(data[,i]~ by, data, function(x) c(mean = mean(x), sd = sd(x)))
  agres<-as.data.frame(ag$`data[, i]`)
  agres$r.mean<-round(agres$mean,3)
  agres$r.sd<-round(agres$sd,3)
  agres$mean_sd<- paste(agres$r.mean, agres$r.sd, sep="±")
  st[,i]<-agres$mean_sd
  }
```

```r
st2<-as.data.frame(matrix(NA, 6, indices))
for(i in 1:indices) {
  st2[,i]<- paste(st[,i], Let[,i], sep=",")
  }

rownames(st2)<-ag$by
colnames(st2)<-colnames(data[1:indices])
sample_size<-as.data.frame(with(data, table(Soil_compartment)))
st2$sample_size<-sample_size$Freq
st2$name_size<-paste(rownames(st2),st2$sample_size, sep=",n=")
rownames(st2)<-st2$name_size
fun_alpha_summary_soil_com<-as.data.frame(t(st2[,-(5:6)]))

fun_alpha_summary_soil_com
```

```
##                Fi_BK,n=3       Fi_RS,n=48     Fi_RT,n=48         Fo_BK,n=3
## shannon    3.716±0.085,a    3.684±0.265,a  2.673±0.301,b      4.002±0.13,a
## no.species 180±36.497,ab 174.729±78.888,a 83.875±16.56,c 255.333±84.311,ab
## chao1      180±36.497,ab 174.729±78.888,a 83.875±16.56,c 255.333±84.311,ab
## evenness   0.718±0.019,a    0.724±0.041,a  0.605±0.058,b     0.727±0.027,a
##                  Fo_RS,n=48       Fo_RT,n=48
## shannon        3.715±0.24,a    3.002±0.348,c
## no.species 208.667±84.063,b 109.417±21.774,d
## chao1      208.667±84.063,b 109.417±21.774,d
## evenness      0.702±0.034,a    0.641±0.058,b
```

3.5. plot shannon index

```r
ggplot(fun_alpha) +
  geom_boxplot(aes(x=Compartment2, y=shannon, fill=Soil), outlier.colour = NA)+
  labs(x="", y = "Shannon index") + scale_fill_manual(values=c("#a6611a","#018571"))+
  geom_point(aes(x=Compartment2, y=shannon, fill=Soil), alpha = 0.4, shape = 21,
             position = position_jitterdodge())+
  scale_x_discrete(labels = c("BK", "RS", "RT"))+
  theme(axis.text.x = element_text(size = 13, face = "bold",colour = "black"),
        axis.text.y = element_text(size = 13),
        axis.title.y = element_text(face = "bold", size = 13, vjust = 3),
        legend.text = element_text(size = 13), legend.title = element_text(size = 13,face = "bold"))
```



4. Beta diversity (Fig 2E, Fig 2F)

4.1. Rarefying abundance table

```
set.seed(1234) # to have reproducible result
rar_ps =rarefy_even_depth(ps2, rngseed=T, replace = F)
```

4.2. Create sum of bacterial phylum table

Combine ASV table and taxonomic table

```
t.rar_asv<-as.data.frame(t(otu_table(rar_ps)))
t.rar_asv_rc<-rownames_to_column(t.rar_asv)
tax_rc<-rownames_to_column(tax)
phyla<-right_join(tax_rc, t.rar_asv_rc, by="rowname")
rownames(phyla)<-phyla$rowname
phyla<-phyla[,c(3,9:206)]
phyla[1:5,1:5]
```

```
##              Phylum R_A2_e R_A2_r R_A3_e R_A3_r
## fASV1    Ascomycota   1211    468    902    343
## fASV2 Basidiomycota     25      0      0      0
## fASV3    Ascomycota   1339    905    974    554
## fASV4    Ascomycota    946    100   1117    108
## fASV5 Basidiomycota      1      0      0      0
```

Create sum of phylum

```
phyla$Phylum <- droplevels(phyla)$Phylum
np = length(levels(phyla$Phylum)) #number of phylum
ns = 198 #number of sample
phyla_sum = data.frame(matrix(ncol=ns,nrow=np))

for(i in 1:ns){
  ag<-aggregate(phyla[,1+i] ~ Phylum, phyla, sum)
  phyla_sum[,i]<-ag[2]
}

rownames(phyla_sum)<-ag$Phylum
colnames(phyla_sum)<-colnames(phyla[,2:199])
phyla_sum[1:5,1:5]
```

```
##                     R_A2_e R_A2_r R_A3_e R_A3_r R_A4_e
## Ascomycota            4451   2750   3894   3211   4835
## Basidiobolomycota        0      0      0      0      0
## Basidiomycota          269    831     29    547     74
## Blastocladiomycota       0      0      0      0      0
## Chytridiomycota         15    333    683    499    100
```

To show major phylum on the plot later, we create 'others' by summing minor phlyum based on their percentage in community

```
phyla_sum$percentage<-rowSums(phyla_sum)/sum(rowSums(phyla_sum))*100
phyla_major<-subset(phyla_sum, percentage >=0.5)
phyla_minor<-subset(phyla_sum, percentage <0.5)
Others<-as.data.frame(colSums(phyla_minor))
colnames(Others)<-"Others"
phyla_test<-as.data.frame(cbind(t(phyla_major),Others))
phyla_test = phyla_test[!row.names(phyla_test)%in% "percentage",]# remove percentage row
phyla_test[1:5,1:10]
```

```
##        Ascomycota Basidiomycota Chytridiomycota Glomeromycota Mortierellomycota
## R_A2_e       4451           269              15           262                78
## R_A2_r       2750           831             333            27               763
## R_A3_e       3894            29             683           310                70
## R_A3_r       3211           547             499            11               865
## R_A4_e       4835            74             100           174                54
##        Mucoromycota Olpidiomycota Rozellomycota Unknown Others
## R_A2_e            0           136             0     241      0
## R_A2_r           19            11            31     684      3
## R_A3_e            0           243             4     219      0
## R_A3_r            3            50            42     224      0
## R_A4_e            0           165             0      50      0
```

4.3. Dunn test on phylum composition

First, transform phyla dataset into percentage unit

```
phyla_test_perc<-as.data.frame(matrix(NA,ns,10)) #sample =198, phyla=10

for (i in 1:ns){ #row
  for(j in 1:10) #column
    phyla_test_perc[i,j]<-phyla_test[i,j]/rowSums(phyla_test[i,1:10])*100
}
```

```
rownames(phyla_test_perc)<-rownames(phyla_test)
colnames(phyla_test_perc)<-colnames(phyla_test)
phyla_test_perc$Soil_compartment<-(sample_data(rar_ps))$Soil_compartment
phyla_test_perc[1:5,1:10]
```

```
##        Ascomycota Basidiomycota Chytridiomycota Glomeromycota Mortierellomycota
## R_A2_e   81.63977     4.9339692       0.2751284     4.8055759         1.4306676
## R_A2_r   50.44021    15.2421130       6.1078503     0.4952311        13.9948643
## R_A3_e   71.42333     0.5319149      12.5275128     5.6859868         1.2839325
## R_A3_r   58.89582    10.0330154       9.1526045     0.2017608        15.8657373
## R_A4_e   88.68305     1.3573001       1.8341893     3.1914894         0.9904622
##        Mucoromycota Olpidiomycota Rozellomycota    Unknown     Others
## R_A2_e   0.00000000     2.4944974    0.00000000  4.4203962 0.00000000
## R_A2_r   0.34849596     0.2017608    0.56859868 12.5458547 0.05502568
## R_A3_e   0.00000000     4.4570800    0.07336757  4.0168745 0.00000000
## R_A3_r   0.05502568     0.9170946    0.77035950  4.1085840 0.00000000
## R_A4_e   0.00000000     3.0264123    0.00000000  0.9170946 0.00000000
```

Run dunn test to compare composition among soil_compartment groups. In this test, phylum ‘others’ and ‘unknown’ were not included as its comparison is meaningless.

```
indices=8 #number of variable that I am testing
Z<-as.data.frame(matrix(NA, 15, indices)) #results list =15
P.unadj<-as.data.frame(matrix(NA, 15, indices)) #results list =15
P.adj<-as.data.frame(matrix(NA, 15, indices)) #results list =15
Let<-as.data.frame(matrix(NA, 6, indices)) #results list =6

for(i in 1:indices) {
  PT<-dunnTest(phyla_test_perc[,i]~Soil_compartment, data=phyla_test_perc, method = "bh")
  Z[,i]<-PT$res$Z
  P.unadj[,i]<-PT$res$P.unadj
  P.adj[,i]<-PT$res$P.adj
  PT2<-PT$res
  cl<-cldList(comparison = PT2$Comparison,p.value = PT2$P.adj,threshold  = 0.05)
  Let[,i]<-cl$Letter
}

rownames(Z) <- PT$res$Comparison
colnames(Z) <- colnames(phyla_test_perc[1:indices])
rownames(P.unadj) <- PT$res$Comparison
colnames(P.unadj) <- colnames(phyla_test_perc[1:indices])
rownames(P.adj) <- PT$res$Comparison
colnames(P.adj) <- colnames(phyla_test_perc[1:indices])
rownames(Let) <- cl$Group
colnames(Let) <- colnames(phyla_test_perc[1:indices])



Let
```

```
##       Ascomycota Basidiomycota Chytridiomycota Glomeromycota Mortierellomycota
## Fi_BK         ab           abc              ab             a                 a
## Fi_RS          a             a               a             a                 a
## Fi_RT          c             b               b             b                 b
## Fo_BK          b            ac               c             a                ac
## Fo_RS          b             c               c             a                 c
## Fo_RT          b             c               c             b                 b
##       Mucoromycota Olpidiomycota Rozellomycota
## Fi_BK          abc            ab            ab
## Fi_RS            a             a             c
## Fi_RT            b             b             a
## Fo_BK           de             c             d
## Fo_RS            d             c             d
## Fo_RT           ce             c             b
```

4.4. Phylum stack bar plot

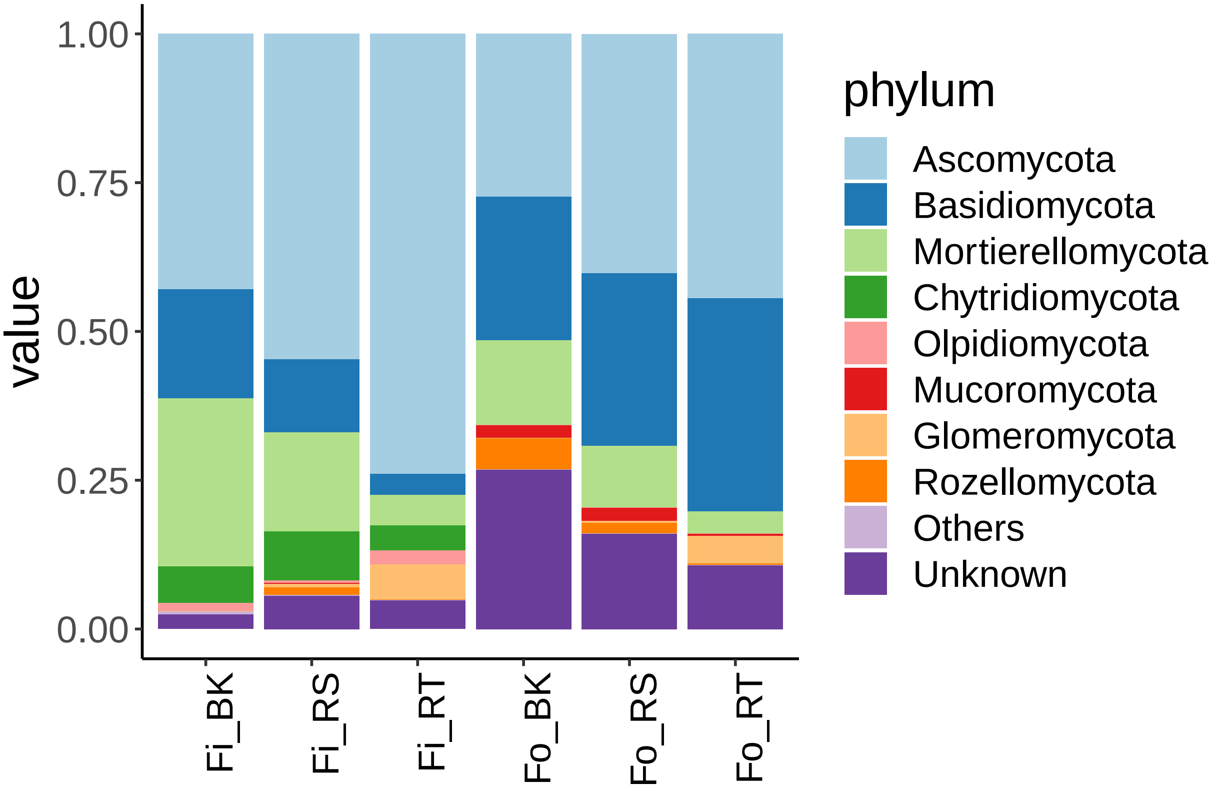
Prepare dataset for stack bar plot

```
np2=10  #number of phylums
nsoilcom=6 #number of factors in soil_com
phyla_soilcom<-matrix(NA,nsoilcom,np2)

for(i in 1:np2){
  a<-aggregate(phyla_test_perc[,i], by=list(Soil_compartment=phyla_test_perc$Soil_compartment), FUN=sum
  phyla_soilcom[,i]<-a$x
}

rownames(phyla_soilcom)<-a$Soil_compartment
colnames(phyla_soilcom)<-colnames(phyla_test_perc[,1:10])

phyla_soilcom_rc<-rownames_to_column(as.data.frame(t(phyla_soilcom)))
phyla_soilcom_rc_melt<-melt(phyla_soilcom_rc,
                            rowname=c("Fi_BK", "Fi_RS","Fi_RT","Fo_BK","Fo_RS", "Fo_RT"))

phyla_soilcom_rc_melt$phylum<-factor(phyla_soilcom_rc_melt$rowname,
                                     c("Ascomycota","Basidiomycota","Mortierellomycota","Chytridiomycot
                                       "Olpidiomycota","Mucoromycota","Glomeromycota","Rozellomycota","

phyla_soilcom_rc_melt[1:5,1:4]
```

```
##              rowname variable      value            phylum
## 1        Ascomycota    Fi_BK 128.668379        Ascomycota
## 2     Basidiomycota    Fi_BK  55.117388     Basidiomycota
## 3   Chytridiomycota    Fi_BK  18.305209   Chytridiomycota
## 4     Glomeromycota    Fi_BK   0.348496     Glomeromycota
## 5 Mortierellomycota    Fi_BK  84.666178 Mortierellomycota
```

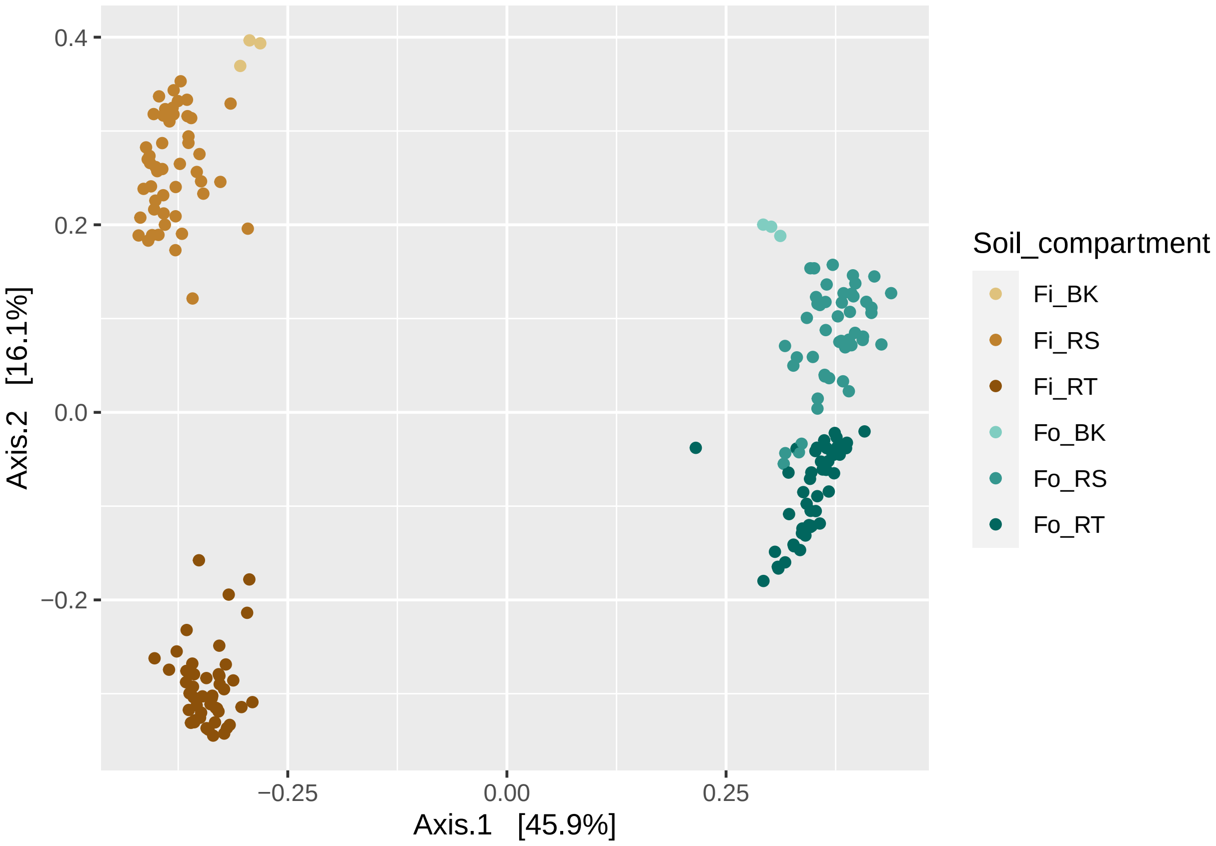
creat plot

```
cols<-c("#a6cee3", "#1f78b4", "#b2df8a", "#33a02c","#fb9a99",
        "#e31a1c", "#fdbf6f", "#ff7f00","#cab2d6","#6a3d9a")  #assign colors

p<-ggplot(phyla_soilcom_rc_melt, aes(variable, value, fill=phylum)) +
  geom_bar(stat="identity", position="fill")

p + scale_fill_manual(values=cols) + theme_classic() +
  theme(text = element_text(size=18), axis.text.x = element_text(angle=90, hjust=1, colour = "black"))+
```



4.5. Principle coordinate analysis (PCoA)

```
soil_com_colors<-c("#dfc27d","#bf812d","#8c510a","#80cdc1","#35978f","#01665e")
ord <- ordinate(rar_ps, "PCoA", "bray")
plot_ordination(rar_ps, ord, color="Soil_compartment") +
  scale_color_manual(values = soil_com_colors) + geom_point(size=1)
```



4.6. Permutational analysis of variance (PERMANOVA)

```
### PERMANOVA
dis<-phyloseq::distance(rar_ps, method = "bray")
sam<-as(sample_data(rar_ps),"data.frame")
perm<-adonis(dis ~Soil*Compartment*Genotype, data=sam, permutations = 9999)
perm.res<-as.data.frame(perm$aov.tab)
perm.res
```

```
##                            Df  SumsOfSqs     MeanSqs     F.Model         R2
## Soil                        1 25.5381485 25.53814847 403.6975944 0.45492568
## Compartment                 2  8.5362679  4.26813393  67.4690808 0.15206143
## Genotype                   15  2.2821998  0.15214665   2.4050780 0.04065413
## Soil:Compartment            2  7.4365219  3.71826093  58.7768919 0.13247103
## Soil:Genotype              15  2.2227260  0.14818173   2.3424020 0.03959469
## Compartment:Genotype       15  0.8777476  0.05851651   0.9250073 0.01563582
## Soil:Compartment:Genotype  15  0.8929583  0.05953055   0.9410369 0.01590678
## Residuals                 132  8.3503980  0.06326059          NA 0.14875043
## Total                     197 56.1369679          NA          NA 1.00000000
##                           Pr(>F)
## Soil                      0.0001
## Compartment               0.0001
## Genotype                  0.0001
## Soil:Compartment          0.0001
## Soil:Genotype             0.0001
## Compartment:Genotype      0.6207
## Soil:Compartment:Genotype 0.5935
## Residuals                      NA
## Total                          NA
```

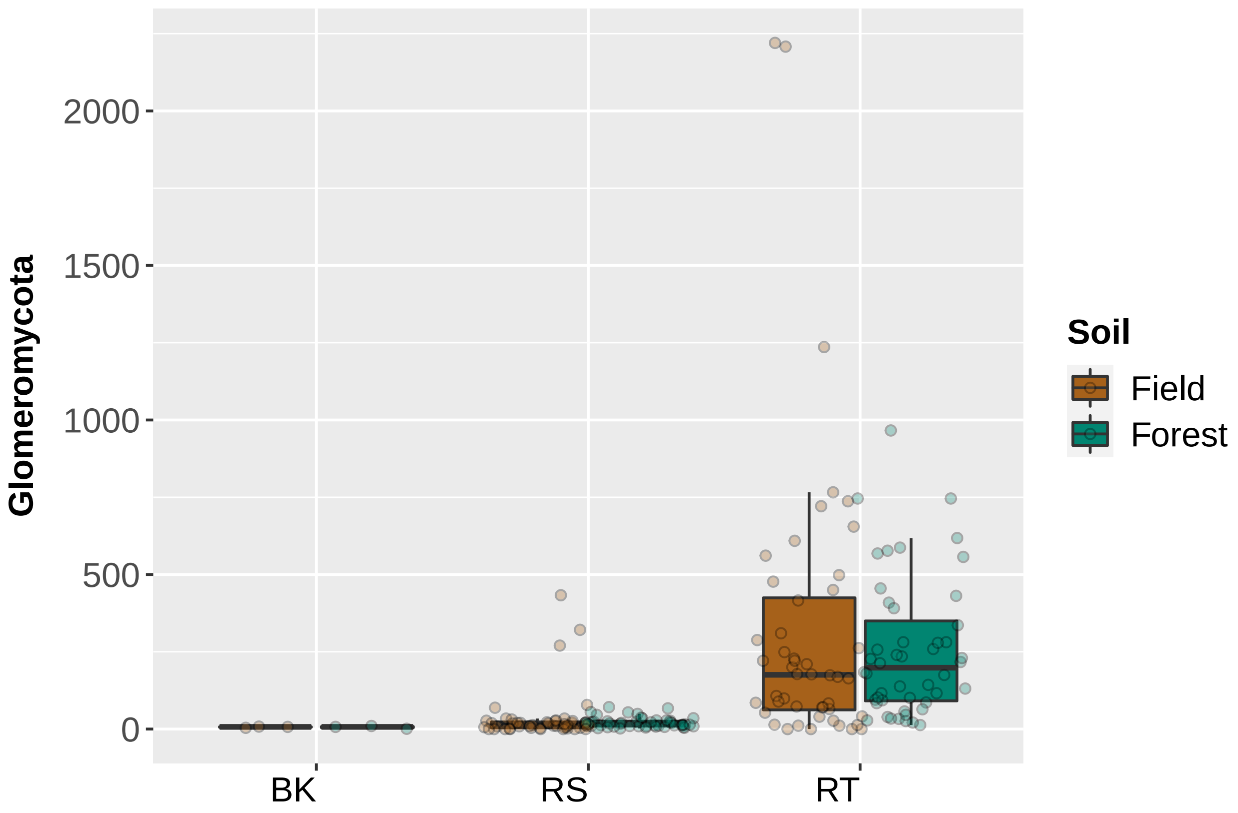
subsetting dataset for further use

```r
rar_ps_Fo_RS<-subset_samples(rar_ps, Soil_compartment=="Fo_RS" )
rar_ps_Fo_RT<-subset_samples(rar_ps, Soil_compartment=="Fo_RT" )
rar_ps_Fi_RT<-subset_samples(rar_ps, Soil_compartment=="Fi_RT" )
rar_ps_Fi_RS<-subset_samples(rar_ps, Soil_compartment=="Fi_RS" )

rar_Fo_RT<-cbind(sample_data(rar_ps_Fo_RT), otu_table(rar_ps_Fo_RT))
rar_Fo_RS<-cbind(sample_data(rar_ps_Fo_RS), otu_table(rar_ps_Fo_RS))
rar_Fi_RT<-cbind(sample_data(rar_ps_Fi_RT), otu_table(rar_ps_Fi_RT))
rar_Fi_RS<-cbind(sample_data(rar_ps_Fi_RS), otu_table(rar_ps_Fi_RS))
```

5. Exploring AMF distribution (correlation with biomass; Fig 3)

5.1. The relative abundance of Glomeromycota in each condition

```r
Glomeromycota<-data.frame(t(phyla_sum[6,-199]))
Glomeromycota<-cbind(Glomeromycota, (sample_data(rar_ps))[,1:4])
Glomeromycota$Compartment2<-factor(Glomeromycota$Compartment,c("Bulksoil","Rhizosphere","Root"))

ggplot(Glomeromycota) + #400x300
  geom_boxplot(aes(x=Compartment2, y=Glomeromycota, fill=Soil), outlier.colour = NA)+
  labs(x="", y = "Glomeromycota")+ scale_fill_manual(values=c("#a6611a","#018571"))+
  geom_point(aes(x=Compartment2, y=Glomeromycota, fill=Soil), alpha = 0.3, shape = 21,
             position = position_jitterdodge())+
  scale_x_discrete(labels = c("BK", "RS", "RT"))+
  theme(axis.text.x = element_text(size = 13, colour = "black",hjust=1),
        axis.text.y = element_text(size = 13),
        axis.title.y = element_text(face = "bold", size = 13, vjust = 3),
        legend.text = element_text(size = 13), legend.title = element_text(size = 13,face = "bold"))
```



5.2. Order ditribution in Glomeromycota

Make some of orders of Glomeromycota in each condition

```
order<-right_join(tax_rc, t.rar_asv_rc, by="rowname")
rownames(order)<-order$rowname
glome_order<-subset(order, Phylum=="Glomeromycota")
glome_order<-glome_order[,c(5,9:206)]
glome_order$Order <- droplevels(glome_order)$Order

np = length(levels(glome_order$Order)) #number of order
ns = 198 #number of sample
glome_order_sum = data.frame(matrix(ncol=ns,nrow=np))

for(i in 1:ns){
  ag<-aggregate(glome_order[,1+i] ~ Order, glome_order, sum)
  glome_order_sum[,i]<-ag[2]
}

rownames(glome_order_sum)<-ag$Order
colnames(glome_order_sum)<-colnames(glome_order[,2:199])
glome_order_sum[1:5,1:5]
```

```
##                   R_A2_e R_A2_r R_A3_e R_A3_r R_A4_e
## Archaeosporales        0      0      0      0      0
## Diversisporales        0      0      4      0      0
## Gigasporales           0      0      0      0      0
## Glomerales         260     27    306     11    174
## Paraglomerales       2      0      0      0      0
```

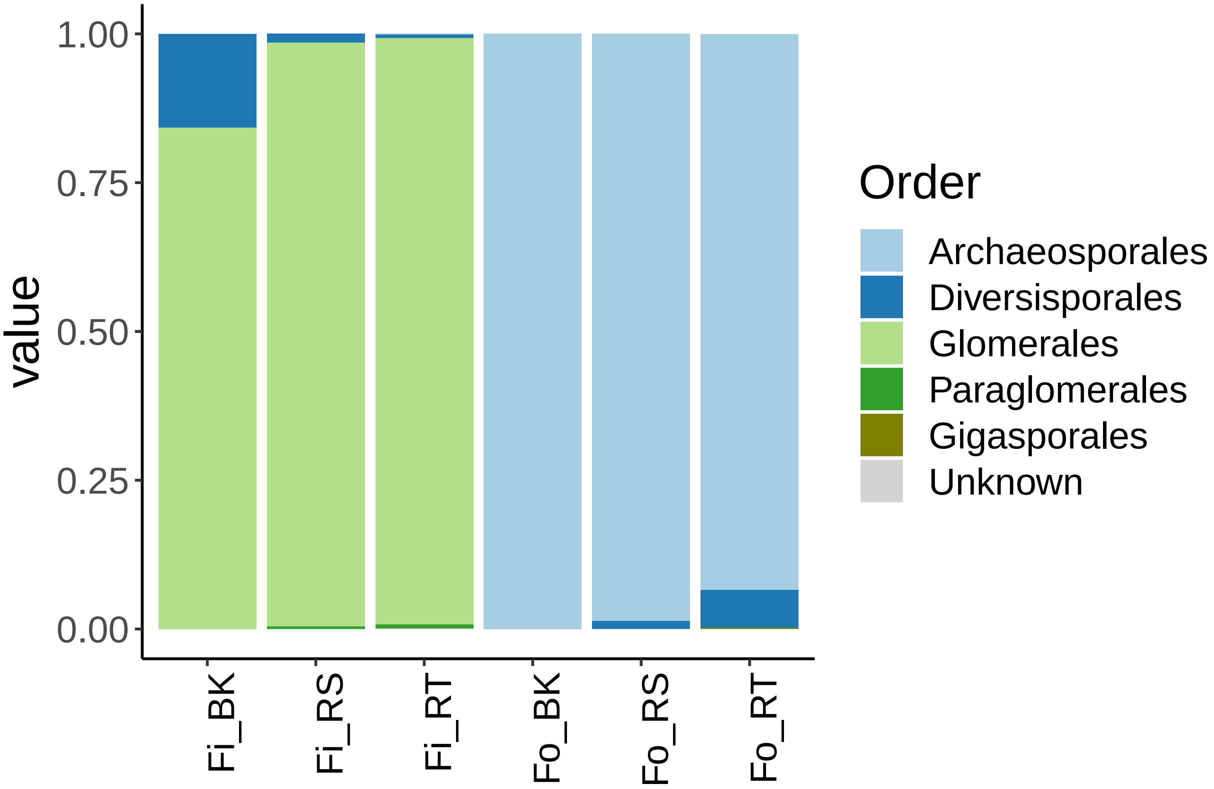
add soil_compartment group information

```
t_glome_order_sum<-as.data.frame(t(glome_order_sum))
t_glome_order_sum<-cbind(t_glome_order_sum, (sample_data(rar_ps))[,1:4])
t_glome_order_sum$Compartment2<-factor(t_glome_order_sum$Compartment,c("Bulksoil","Rhizosphere","Root")

#aggregate by soil_com
p<-6 # order
nsoilcom=6 #number of factors in soil_com
glome_order_soilcom<-matrix(NA,nsoilcom,p)

for(i in 1:p){
  a<-aggregate(t_glome_order_sum[,i], by=list(Soil_compartment=t_glome_order_sum$Soil_compartment), FUN
  glome_order_soilcom[,i]<-a$x
}

rownames(glome_order_soilcom)<-a$Soil_compartment
colnames(glome_order_soilcom)<-colnames(t_glome_order_sum[,1:p])

glome_order_soilcom
```

```
##       Archaeosporales Diversisporales Gigasporales Glomerales Paraglomerales
## Fi_BK               0               3            0         16              0
## Fi_RS               0              25            0       1671              7
## Fi_RT              16              91            0      15332            104
## Fo_BK              18               0            0          0              0
## Fo_RS            1010              14            0          0              0
## Fo_RT           11415             764           31          1              0
##       Unknown
## Fi_BK       0
## Fi_RS       0
## Fi_RT      14
## Fo_BK       0
## Fo_RS       0
## Fo_RT       0
```

Plot stack bar of orders

```
glome_order_soilcom_rc<-rownames_to_column(as.data.frame(t(glome_order_soilcom)))
glome_order_soilcom_rc_melt<-melt(glome_order_soilcom_rc,
                            rowname=c("Fi_BS", "Fi_RS","Fi_RT","Fo_BS","Fo_RS", "Fo_RT"))

glome_order_soilcom_rc_melt$Order<-factor(glome_order_soilcom_rc_melt$rowname,
                                     c("Archaeosporales","Diversisporales","Glomerales","Paraglomerales
                                       "Gigasporales","Unknown"))

cols<-c("#a6cee3", "#1f78b4", "#b2df8a", "#33a02c","#808000", "#D3D3D3") #assign colors
p<-ggplot(glome_order_soilcom_rc_melt, aes(variable, value, fill=Order)) + geom_bar(stat="identity", po
p + scale_fill_manual(values=cols) + theme_classic() +
  theme(text = element_text(size=18), axis.text.x = element_text(angle=90, hjust=1, colour = "black"))+
```



5.3. Correlation between the abundance of Glomeromycota orders and plant biomass

Subset dataset of each treatment

```
t_glome_order_sum2<-subset(t_glome_order_sum, !Soil_compartment=="Fo_BK") #remove bulksoil sample
t_glome_order_sum2<-subset(t_glome_order_sum2, !Soil_compartment=="Fi_BK") #remove bulksoil sample
t_glome_order_sum2_rc<-rownames_to_column(t_glome_order_sum2)
meta_rc<-rownames_to_column(meta)
t_glome_order_sum2_rc<-left_join(t_glome_order_sum2_rc, meta_rc[,c(1,9)], by="rowname")
FoRS_glome_order<-subset(t_glome_order_sum2_rc, Soil_compartment=="Fo_RS")
FoRT_glome_order<-subset(t_glome_order_sum2_rc, Soil_compartment=="Fo_RT")
FiRS_glome_order<-subset(t_glome_order_sum2_rc, Soil_compartment=="Fi_RS")
FiRT_glome_order<-subset(t_glome_order_sum2_rc, Soil_compartment=="Fi_RT")
```

Find correlation using negative bionomial generalized linear model

```
n = 2 # number of variables
FoRS_glome_order_res<-matrix(NA,n,4) #results contain 4 colunms
FoRT_glome_order_res<-matrix(NA,n,4)

for(i in 1:n) {
  #In forest soil, order Gigaporales, glomerales, paragomerales are almost zeros. Therefore not include
  g <-glm.nb(FoRS_glome_order[,1+i]~total_fresh_biomass_g_per_plant, data=FoRS_glome_order)
  FoRS_glome_order_res[i,]<-(as.matrix(coef(summary(g))))[2,]
  g <-glm.nb(FoRT_glome_order[,1+i]~total_fresh_biomass_g_per_plant, data=FoRT_glome_order)
  FoRT_glome_order_res[i,]<-(as.matrix(coef(summary(g))))[2,]
}
```

```
rownames(FoRS_glome_order_res)<-colnames(FoRS_glome_order[2:3])
colnames(FoRS_glome_order_res)<-colnames(coef(summary(g)))
rownames(FoRT_glome_order_res)<-colnames(FoRT_glome_order[2:3])
colnames(FoRT_glome_order_res)<-colnames(coef(summary(g)))

#In field soil, only Glomerales were dominant. Therefore, others were not tested.
g <-glm.nb(Glomerales~total_fresh_biomass_g_per_plant, data=FiRS_glome_order)
FiRS_glome_order_res<-as.matrix(coef(summary(g)))
g <-glm.nb(Glomerales~total_fresh_biomass_g_per_plant, data=FiRT_glome_order)
FiRT_glome_order_res<-as.matrix(coef(summary(g)))

FoRS_glome_order_res
```

```
##                 Estimate Std. Error  z value    Pr(>|z|)
## Archaeosporales 1.936004  0.7168595 2.700674 0.006919904
## Diversisporales 4.199837  3.2833953 1.279114 0.200856880
```

```
FoRT_glome_order_res
```

```
##                 Estimate Std. Error  z value     Pr(>|z|)
## Archaeosporales 4.886487  0.7594331 6.434389 1.239713e-10
## Diversisporales 5.419058  1.0344879 5.238397 1.619776e-07
```

```
FiRS_glome_order_res
```

```
##                                  Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)                      5.000872  0.6494890  7.699702 1.363842e-14
## total_fresh_biomass_g_per_plant -1.225460  0.4248716 -2.884308 3.922750e-03
```

```
FiRT_glome_order_res
```

```
##                                   Estimate Std. Error    z value     Pr(>|z|)
## (Intercept)                      6.0085712  0.5831987 10.3027855 6.845000e-25
## total_fresh_biomass_g_per_plant -0.1728957  0.3794320 -0.4556699 6.486274e-01
```

Test normality of residuals from significant model

```
shapiro.test(resid(glm.nb(Archaeosporales ~ total_fresh_biomass_g_per_plant, FoRT_glome_order)))[2]$p.va
```

```
## [1] 0.4948448
```

```
shapiro.test(resid(glm.nb(Diversisporales ~ total_fresh_biomass_g_per_plant, FoRT_glome_order)))[2]$p.va
```

```
## [1] 0.3572861
```

```
shapiro.test(resid(glm.nb(Archaeosporales ~ total_fresh_biomass_g_per_plant, FoRS_glome_order)))[2]$p.va
```

```
## [1] 0.3619398
```

```
shapiro.test(resid(glm.nb(Glomerales ~ total_fresh_biomass_g_per_plant, FiRS_glome_order)))[2]$p.value
```

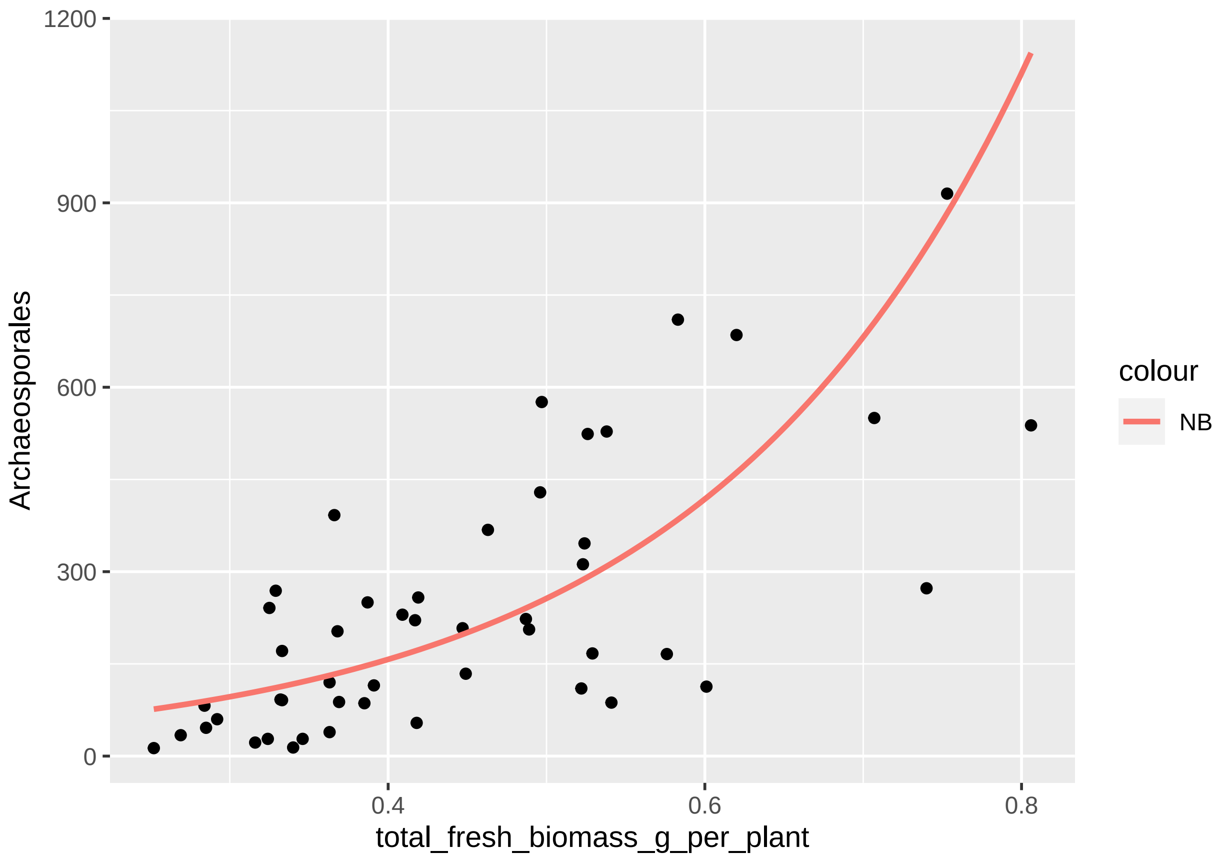
```
## [1] 0.03104962
```

Validate P value using permutation from significant models that passed Shapiro test

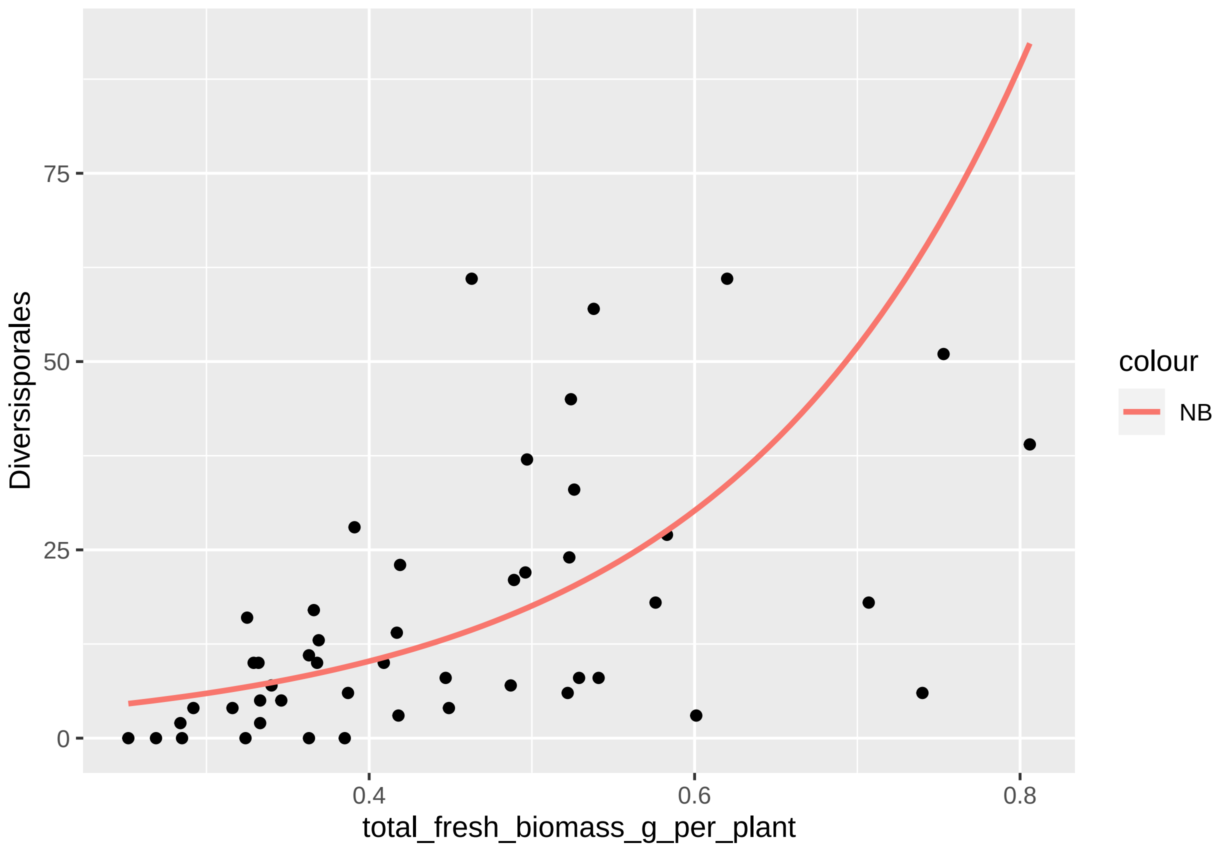
```
Glome_order_glm_res<-as.data.frame(rbind(FoRT_glome_order_res,FoRS_glome_order_res[1,]))
rownames(Glome_order_glm_res)<-c("FoRT_Archeoporales","FoRT_Diversiporales","FoRS_Archeoporales")

nPerm<-1000
Perm.list<-0*1:nPerm
Perm.Ps<-0*3

for(i in 1:nPerm){
  sp<-sample(FoRT_glome_order$total_fresh_biomass_g_per_plant)
  g <-glm.nb(Archaeosporales~sp, FoRT_glome_order)
  Perm.list[i]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g)))), 4]
}
Perm.Ps[1]<-sum(Perm.list<Glome_order_glm_res[1,4])/nPerm

for(i in 1:nPerm){
  sp<-sample(FoRT_glome_order$total_fresh_biomass_g_per_plant)
  g <-glm.nb(Diversisporales~sp, FoRT_glome_order)
  Perm.list[i]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g)))), 4]
}
Perm.Ps[2]<-sum(Perm.list<Glome_order_glm_res[2,4])/nPerm

for(i in 1:nPerm){
  sp<-sample(FoRS_glome_order$total_fresh_biomass_g_per_plant)
  g <-glm.nb(Archaeosporales~sp, FoRS_glome_order)
  Perm.list[i]<-coef(summary(g))[grepl("sp$",row.names(coef(summary(g)))), 4]
}
Perm.Ps[3]<-sum(Perm.list<Glome_order_glm_res[3,4])/nPerm

Glome_order_glm_res$Perm.P<-Perm.Ps
Glome_order_glm_res
```

```
##                       Estimate Std. Error  z value     Pr(>|z|) Perm.P
## FoRT_Archeoporales   4.886487  0.7594331 6.434389 1.239713e-10  0.000
## FoRT_Diversiporales 5.419058  1.0344879 5.238397 1.619776e-07  0.000
## FoRS_Archeoporales   1.936004  0.7168595 2.700674 6.919904e-03  0.019
```

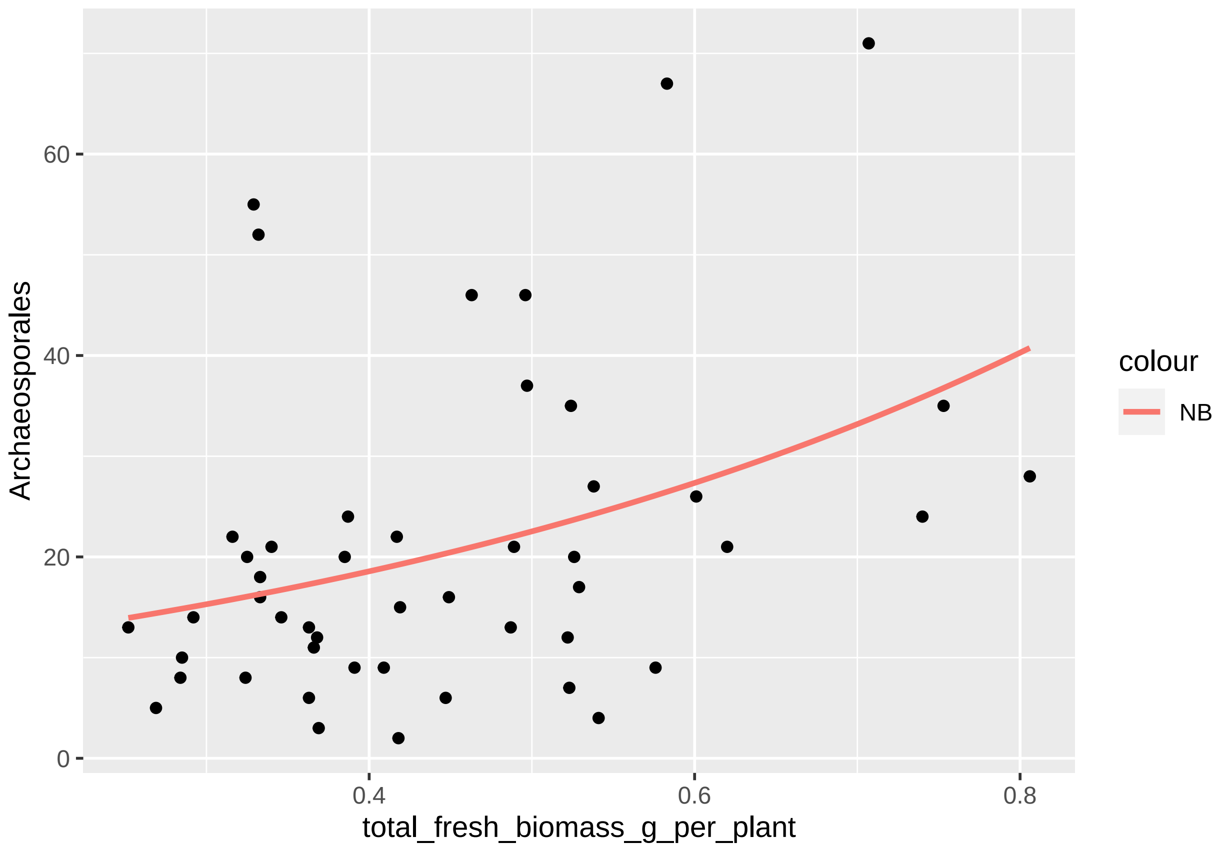
```
ggplot(FoRT_glome_order,aes(total_fresh_biomass_g_per_plant,Archaeosporales)) +
  geom_point() +
  geom_smooth(method = MASS::glm.nb, aes(color = "NB"), se = FALSE)
```



```
ggplot(FoRT_glome_order,aes(total_fresh_biomass_g_per_plant,Diversisporales)) +
  geom_point() +
  geom_smooth(method = MASS::glm.nb, aes(color = "NB"), se = FALSE)
```



```
ggplot(FoRS_glome_order,aes(total_fresh_biomass_g_per_plant,Archaeosporales)) +
  geom_point() +
  geom_smooth(method = MASS::glm.nb, aes(color = "NB"), se = FALSE)
```



6. Correlation between diversity of fungal community and SLs level (Fig 4)

As SLs were only detected in the plant roots grown on forest soil, all correlation study with SLs performed using forest soil dataset. Although we had five replicates for each experimental condition, root material was sometimes insufficient to analyze both SLs production and microbiome diversity and composition on the same sample. Therefore, three replicates were used for each analysis (total n= 48), and we used only the samples for which we had enough material to assess both SL and microbiome (n=37) for the correlation analyses between SL production and relative abundance of community.

6.1. Correlation between alpha diversity and SLs level

Subset forest soil dataset from alpha diversity measurement that I obtained earlier.

```
match_alpha<-subset(fun_alpha, Soil=="Forest"&SL_analysis=="yes")
```

The correlation between alpha diversity and SLs were examined using linear model incoperating permutation test.

```
vars=4
fourdo <-matrix(NA,vars,2)
meo5ds  <-matrix(NA,vars,2)
orb <-matrix(NA,vars,2)

for(i in 1:vars) {
  l<-lmp(match_alpha[,i]~X4DO_pmol_g, data=match_alpha)
  fourdo[i,]<-coef(summary(l))[c(2,6)] #estimate & p value
  l<-lmp(match_alpha[,i]~MeO5DS_pmol_g, data=match_alpha)
  meo5ds[i,]<-coef(summary(l))[c(2,6)] #estimate & p value
  l<-lmp(match_alpha[,i]~orobanchol_pmol_g, data=match_alpha)
  orb[i,]<-coef(summary(l))[c(2,6)] #estimate & p value
}
```

```
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
## [1] "Settings:  unique SS : numeric variables centered"
```

```
lmp_res<-cbind(fourdo, meo5ds, orb)
rownames(lmp_res)<-colnames(match_alpha[1:vars])
colnames(lmp_res)<-c("fourdo.est","fourdo.p","meo5ds.est","meo5ds.p","orb.est","orb.p")

lmp_res
```

```
##               fourdo.est  fourdo.p    meo5ds.est  meo5ds.p       orb.est
## shannon    -1.031071e-03 0.7450980  0.0008452531 0.9607843   -0.70726238
## no.species  2.403996e-01 0.9019608 -0.4372944997 0.6545455 -116.42027471
## chao1       2.403996e-01 1.0000000 -0.4372944997 0.7843137 -116.42027471
## evenness   -2.363633e-05 1.0000000  0.0003933353 0.3705882   -0.08515728
##                orb.p
## shannon    0.1937799
## no.species 0.6379310
## chao1      0.3209302
## evenness   0.2633452
```

6.2. Correlation between beta diversity and SLs level (Constrained PCoA)

Subset dataset

```
match_ps_Fo_RT<-subset_samples(rar_ps_Fo_RT,SL_analysis=="yes")
match_ps_Fo_RS<-subset_samples(rar_ps_Fo_RS,SL_analysis=="yes")

match_Fo_RT<-cbind(sample_data(match_ps_Fo_RT), otu_table(match_ps_Fo_RT))
match_Fo_RS<-cbind(sample_data(match_ps_Fo_RS), otu_table(match_ps_Fo_RS))

SLs_Fo_RT<-match_Fo_RT[,c(11:13)]
SLs_Fo_RS<-match_Fo_RS[,c(11:13)]
```

Run constrained PCoA

```
var=3
RT.p <- 0*1:var
RS.p <- 0*1:var
set.seed(111)

for(i in 1:var) {
  RT.p[i]<-(anova.cca(capscale(match_Fo_RT[14:2840]~SLs_Fo_RT[,i], match_Fo_RT, dist="bray"), step=1000
  RS.p[i]<-(anova.cca(capscale(match_Fo_RS[14:2840]~SLs_Fo_RS[,i], match_Fo_RS, dist="bray"), step=1000
}


res.p<-rbind(RT.p,RS.p)
colnames(res.p)<-colnames(SLs_Fo_RT[1:var])
res.p
```

```
##      orobanchol_pmol_g X4DO_pmol_g MeO5DS_pmol_g
## RT.p             0.045       0.279         0.632
## RS.p             0.003       0.241         0.213
```

Get species score from significant constrained model (in both roots and rhizosphere by orobanchol)

```
# in roots
FoRT_orb_scores<-(scores(capscale(match_Fo_RT[14:2840]~SLs_Fo_RT[,2], dist="bray"))$species)
FoRT_orb_scores_abs<-abs(FoRT_orb_scores)
FoRT_orb_scores2<-as.data.frame(cbind(FoRT_orb_scores,FoRT_orb_scores_abs))
FoRT_orb_scores3<-FoRT_orb_scores2[,c(1,3)]
colnames(FoRT_orb_scores3)<-c("orb_CAP1","orb_abs_CAP1")
FoRT_orb_scores3_rc<-rownames_to_column(FoRT_orb_scores3)
selected_taxa = tax_rc[which(tax_rc$rowname %in% FoRT_orb_scores3_rc$rowname),] # extract taxa
FoRT_orb_scores_tax<-full_join(FoRT_orb_scores3_rc,selected_taxa, by="rowname")
FoRT_orb_scores_tax[1:5,1:9]
```

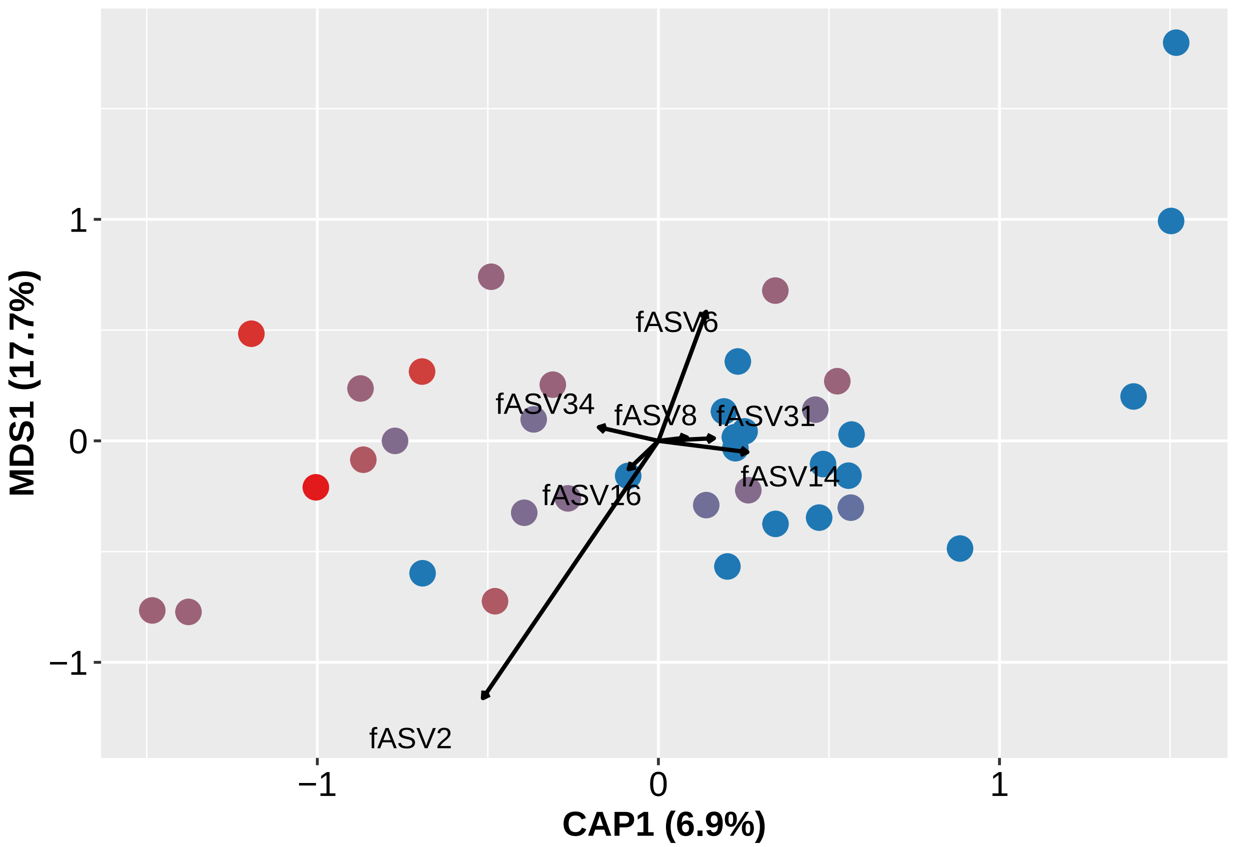
```
##   rowname      orb_CAP1 orb_abs_CAP1 Kingdom        Phylum           Class
## 1   fASV1  1.874595e-01 1.874595e-01   Fungi    Ascomycota Sordariomycetes
## 2   fASV2 -3.723999e-01 3.723999e-01   Fungi Basidiomycota         Unknown
## 3   fASV3 -1.026016e-03 1.026016e-03   Fungi    Ascomycota Sordariomycetes
## 4   fASV4 -5.951441e-05 5.951441e-05   Fungi    Ascomycota Dothideomycetes
## 5   fASV5 -6.439165e-03 6.439165e-03   Fungi Basidiomycota Tremellomycetes
##          Order              Family     Genus
## 1  Hypocreales         Nectriaceae  Fusarium
## 2      Unknown             Unknown   Unknown
## 3  Hypocreales         Nectriaceae  Fusarium
## 4 Pleosporales             Unknown   Unknown
## 5  Tremellales Trimorphomycetaceae Saitozyma
```

```
# in rhizosphere
FoRS_orb_scores<-(scores(capscale(match_Fo_RS[14:2367]~SLs_Fo_RS[,2], dist="bray"))$species)
FoRS_orb_scores_abs<-abs(FoRS_orb_scores)
FoRS_orb_scores2<-as.data.frame(cbind(FoRS_orb_scores,FoRS_orb_scores_abs))
FoRS_orb_scores3<-FoRS_orb_scores2[,c(1,3)]
colnames(FoRS_orb_scores3)<-c("orb_CAP1","orb_abs_CAP1")
FoRS_orb_scores3_rc<-rownames_to_column(FoRS_orb_scores3)
selected_taxa = tax_rc[which(tax_rc$rowname %in% FoRS_orb_scores3_rc$rowname),] # extract taxa
FoRS_orb_scores_tax<-full_join(FoRS_orb_scores3_rc,selected_taxa, by="rowname")
FoRS_orb_scores_tax[1:5,1:9]
```

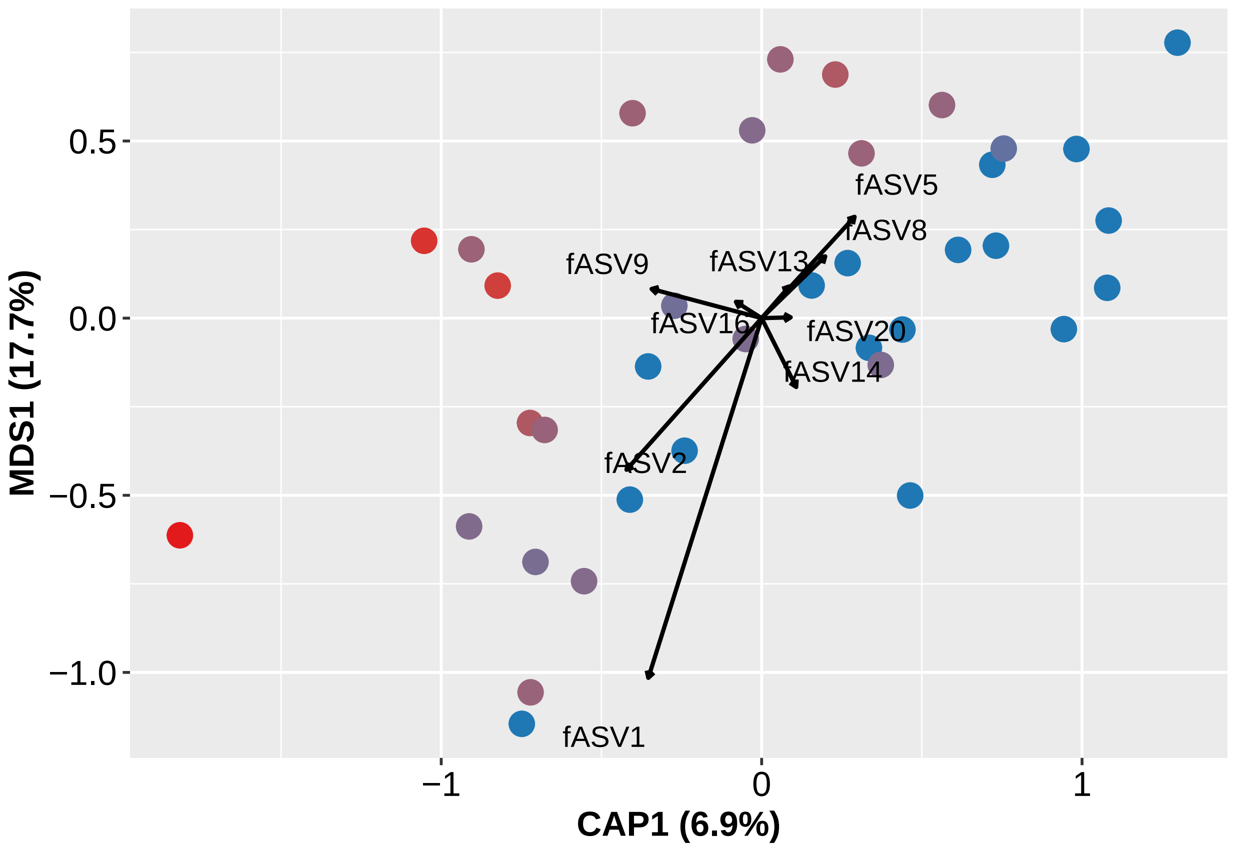
```
##   rowname      orb_CAP1 orb_abs_CAP1 Kingdom        Phylum           Class
## 1   fASV1  2.567672e-01 2.567672e-01   Fungi    Ascomycota Sordariomycetes
## 2   fASV2 -3.728837e-01 3.728837e-01   Fungi Basidiomycota         Unknown
## 3   fASV3 -1.892009e-04 1.892009e-04   Fungi    Ascomycota Sordariomycetes
## 4   fASV4  6.051322e-06 6.051322e-06   Fungi    Ascomycota Dothideomycetes
## 5   fASV5 -2.224270e-02 2.224270e-02   Fungi Basidiomycota Tremellomycetes
##          Order              Family     Genus
## 1  Hypocreales         Nectriaceae  Fusarium
## 2      Unknown             Unknown   Unknown
## 3  Hypocreales         Nectriaceae  Fusarium
## 4 Pleosporales             Unknown   Unknown
## 5  Tremellales Trimorphomycetaceae Saitozyma
```

Now plot CAP results

```
fontsize=13

# in roots
cap<-ordinate(physeq = match_ps_Fo_RT, method = "CAP", distance = "bray", formula = ~ orobanchol_pmol_g
cap_sc <- rownames_to_column(data.frame(vegan::scores(cap, display = "species")))
cap.cut <-subset(cap_sc, (abs(CAP1))>=0.08) #cutoff value
arrow_map <- aes(xend = CAP1, yend= MDS1,x = 0,y = 0,shape = NULL, color=NULL)
label_map <- aes(x =  1.1*CAP1, y =  1.1*MDS1, shape = NULL, color=NULL, label = rowname)
arrowhead = arrow(length = unit(0.01, "npc"))

plot_ordination(match_ps_Fo_RT, cap, axes = c(1,2), color = "orobanchol_pmol_g")+
  labs(x="CAP1 (6.9%)", y = "MDS1 (17.7%)") +
  scale_color_gradient(high = "#e31a1c", low = "#1f78b4")+
  geom_point(size = 4)+
  geom_segment(mapping = arrow_map, size = .75,data = cap.cut, arrow = arrowhead)+
  geom_text_repel(mapping = label_map, data = cap.cut, size=4, show.legend = F)+
  theme(axis.text.x = element_text(size = fontsize, colour = "black"),
        axis.text.y = element_text(size = fontsize, colour = "black"),
        axis.title.x = element_text(face = "bold", size = fontsize),
        axis.title.y = element_text(face = "bold", size = fontsize),
        legend.position = "none")
```



```
# in rhizosphere
cap<-ordinate(physeq = match_ps_Fo_RS, method = "CAP", distance = "bray", formula = ~ orobanchol_pmol_g
cap_sc <- rownames_to_column(data.frame(vegan::scores(cap, display = "species")))
cap.cut <-subset(cap_sc, (abs(CAP1))>=0.08) #cutoff value
arrow_map <- aes(xend = CAP1, yend= MDS1,x = 0,y = 0,shape = NULL, color=NULL)
label_map <- aes(x =  1.1*CAP1, y =  1.1*MDS1, shape = NULL, color=NULL, label = rowname)
arrowhead = arrow(length = unit(0.01, "npc"))

plot_ordination(match_ps_Fo_RS, cap, axes = c(1,2), color = "orobanchol_pmol_g")+
  labs(x="CAP1 (6.9%)", y = "MDS1 (17.7%)") +
  scale_color_gradient(high = "#e31a1c", low = "#1f78b4")+
  geom_point(size = 4)+
  geom_segment(mapping = arrow_map, size = .75,data = cap.cut, arrow = arrowhead)+
  geom_text_repel(mapping = label_map, data = cap.cut, size=4, show.legend = F)+
  theme(axis.text.x = element_text(size = fontsize, colour = "black"),
        axis.text.y = element_text(size = fontsize, colour = "black"),
        axis.title.x = element_text(face = "bold", size = fontsize),
        axis.title.y = element_text(face = "bold", size = fontsize),
        legend.position = "none")
```



7. Prepare dataset for W4 (correlation study between abundance of each genus/ASV with level of SLs)

Filter counts not seen more than 2 times in at least 40% of the sample

7.1. ASV table

```
#Roots
match_ps_Fo_RT_filt=filter_taxa(match_ps_Fo_RT, function(x) sum(x > 2) > (0.4*length(x)), TRUE)
match_Fo_RT_filt<-otu_table(match_ps_Fo_RT_filt)
match_Fo_RT_filt<-as.data.frame(match_Fo_RT_filt[,colSums(match_Fo_RT_filt[,])>2*(dim(match_Fo_RT_filt)
fun.FoRT_ASV<-cbind(sample_data(match_ps_Fo_RT_filt), match_Fo_RT_filt)

#Rhizosphere
match_ps_Fo_RS_filt=filter_taxa(match_ps_Fo_RS, function(x) sum(x > 2) > (0.4*length(x)), TRUE)
match_Fo_RS_filt<-otu_table(match_ps_Fo_RS_filt)
match_Fo_RS_filt<-as.data.frame(match_Fo_RS_filt[,colSums(match_Fo_RS_filt[,])>2*(dim(match_Fo_RS_filt)
fun.FoRS_ASV<-cbind(sample_data(match_ps_Fo_RS_filt), match_Fo_RS_filt)
```

7.2. Genus (including Glomeromycota orders)

First of all, new genus table need to be made and then filtered.

```
#in roots
t.match_RT_rc<-rownames_to_column(as.data.frame(t(otu_table(match_ps_Fo_RT))))
RT_tax<-right_join(tax_rc, t.match_RT_rc, by="rowname")
```

```
rownames(RT_tax)<-RT_tax$rowname
RT_genus<-RT_tax[,c(7,9:45)]
RT_genus$Genus <- droplevels(RT_genus)$Genus
np = length(levels(RT_genus$Genus)) #number of genus
ns = 37 #number of sample
RT_genus_sum = data.frame(matrix(ncol=ns,nrow=np))
for(i in 1:ns){
  ag<-aggregate(RT_genus[,1+i] ~ Genus, RT_genus, sum)
  RT_genus_sum[,i]<-ag[2]
}
rownames(RT_genus_sum)<-ag$Genus
colnames(RT_genus_sum)<-colnames(RT_genus[,2:38])
RT_genus_sum$percentage<-rowSums(RT_genus_sum)/sum(rowSums(RT_genus_sum))*100
RT_major_genus<-as.data.frame(t(subset(RT_genus_sum, percentage >=0.3))) #select genera which are abund
RT_major_genus = RT_major_genus[!row.names(RT_major_genus)%in% "percentage",]#remove percentage row
fun.FoRT_genus = RT_major_genus[,-21] #remove unknown genus
fun.FoRT_genus_rc<-rownames_to_column(fun.FoRT_genus)
glome_order_sum_rc<-rownames_to_column(data.frame(t(glome_order_sum)))
fun.FoRT_genus2<-left_join(fun.FoRT_genus_rc,glome_order_sum_rc[1:3])
fun.FoRT_genus2[1:5,1:23]
```

```
##   rowname Acaulospora Acidomelania Apiotrichum Cladophialophora Clonostachys
## 1  R_A7_e           6           31          17               17          130
## 2  R_A8_e          39           25           3               26          478
## 3  R_A9_e           8            6          37               28          196
## 4  R_B7_e          18           21           3               12           66
## 5  R_B8_e          51           28          15               15          378
##   Curvularia Discosia Fusarium Hyaloscypha Mariannaea Meliniomyces Mollisia
## 1          2        0      456          25          6           35       31
## 2          9       16      625          20         18           49       63
## 3          3        8      974          17         47           29      105
## 4          0       10      293          20          0           16      203
## 5         12        0      423          15         10           30       35
##   Mortierella Penicillium Pezoloma Phialocephala Saitozyma Talaromyces
## 1         325          64       18            46        13         107
## 2         224         114       36            86        54         124
## 3          62          57       47           116        53         386
## 4         152          53       22            64        45          78
## 5         224          66       49            87       100         230
##   Trichoderma Umbelopsis Archaeosporales Diversisporales
## 1          23         21             273               6
## 2          63         13             538              39
## 3         124         18             167               8
## 4          39         16             550              18
## 5         209         31             915              51
```

```
#in rhizosphere
t.match_RS_rc<-rownames_to_column(as.data.frame(t(otu_table(match_ps_Fo_RS))))
RS_tax<-right_join(tax_rc, t.match_RS_rc, by="rowname")
rownames(RS_tax)<-RS_tax$rowname
RS_genus<-RS_tax[,c(7,9:45)]
RS_genus$Genus <- droplevels(RS_genus)$Genus
np = length(levels(RS_genus$Genus)) #number of genus
ns = 37 #number of sample
```

```
RS_genus_sum = data.frame(matrix(ncol=ns,nrow=np))
for(i in 1:ns){
  ag<-aggregate(RS_genus[,1+i] ~ Genus, RS_genus, sum)
  RS_genus_sum[,i]<-ag[2]
}
rownames(RS_genus_sum)<-ag$Genus
colnames(RS_genus_sum)<-colnames(RS_genus[,2:38])
RS_genus_sum$percentage<-rowSums(RS_genus_sum)/sum(rowSums(RS_genus_sum))*100
RS_major_genus<-as.data.frame(t(subset(RS_genus_sum, percentage >=0.1))) #select genera which are abund
RS_major_genus = RS_major_genus[!row.names(RS_major_genus)%in% "percentage",]#remove percentage row
fun.FoRS_genus = RS_major_genus[,-29] #remove unknown genus
fun.FoRS_genus_rc<-rownames_to_column(fun.FoRS_genus)
fun.FoRS_genus2<-left_join(fun.FoRS_genus_rc,glome_order_sum_rc[1:2])
fun.FoRS_genus2[1:5,1:30]
```

```
##   rowname Apiotrichum Archaeorhizomyces Byssonectria Cladophialophora
## 1  R_A7_r         150                14           24               17
## 2  R_A8_r          66                13            0               29
## 3  R_A9_r         116                 0           10               37
## 4  R_B7_r          58                62           14               18
## 5  R_B8_r          72                 8            0               18
##   Clonostachys Curvularia Fusarium Hyaloscypha Ilyonectria Mariannaea
## 1          482          1      351           6           9         44
## 2          424          4      603           7           6         86
## 3          357          4     1651           5          11        129
## 4          347          0      484           4           6         16
## 5          380          0      321           7           6         32
##   Meliniomyces Mollisia Mortierella Nadsonia Neobulgaria Oidiodendron
## 1            4       12         521       48           7            0
## 2           16       20         414       54          12            5
## 3           37       21         237       34          43            5
## 4           54       15         600       55          10            9
## 5           26       19         378       52           9            7
##   Penicillium Pezoloma Phialocephala Pseudeurotium Rhizopogon Russula Saitozyma
## 1          23       10             3            12         19       4       345
## 2         155       25            15             8         11       1       329
## 3         131       34             3            11          7       7       340
## 4         117       14            20            17         21      15       459
## 5         110        8             5             9         10       2       303
##   Solicoccozyma Sugiyamaella Talaromyces Trichoderma Umbelopsis Archaeosporales
## 1            99           31         229           0         29              24
## 2           101           34          84          38         57              28
## 3            89           25         204          64         19              17
## 4            99           27          69          91         40              71
## 5            95           21         162          71         86              35
```

change object name of tax to fun_tax to use in W4

```
fun_tax<-tax_rc
```

Final outputs from 7.1 ~ 7.2 can be found in work image “W4_correlation_study_image.Rdata”.

Version

```r
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] MASS_7.3-53        lmPerm_2.1.0       ggrepel_0.9.1      tidyr_1.1.2
##  [5] reshape2_1.4.4     multcompView_0.1-8 rcompanion_2.3.27  FSA_0.8.30
##  [9] vegan_2.5-6        lattice_0.20-41    permute_0.9-5      ggplot2_3.3.2
## [13] ranacapa_0.1.0     tibble_3.0.4       dplyr_1.0.2        phyloseq_1.32.0
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-149        matrixStats_0.57.0  tools_4.0.3
##  [4] R6_2.4.1            nortest_1.0-4       BiocGenerics_0.34.0
##  [7] mgcv_1.8-33         colorspace_1.4-1    ade4_1.7-15
## [10] withr_2.3.0         tidyselect_1.1.0    Exact_2.1
## [13] compiler_4.0.3      Biobase_2.48.0      expm_0.999-6
## [16] sandwich_3.0-0      labeling_0.4.2      scales_1.1.1
## [19] lmtest_0.9-38       mvtnorm_1.1-1       stringr_1.4.0
## [22] digest_0.6.25       rmarkdown_2.7       XVector_0.28.0
## [25] pkgconfig_2.0.3     htmltools_0.5.1.1   dunn.test_1.3.5
## [28] highr_0.8           rlang_0.4.10        rstudioapi_0.11
## [31] farver_2.0.3        generics_0.0.2      zoo_1.8-8
## [34] jsonlite_1.7.1      magrittr_1.5        modeltools_0.2-23
## [37] biomformat_1.16.0   Matrix_1.2-18       Rcpp_1.0.5
## [40] DescTools_0.99.40   munsell_0.5.0       S4Vectors_0.26.1
## [43] Rhdf5lib_1.10.1     ape_5.4-1           lifecycle_0.2.0
## [46] stringi_1.5.3       multcomp_1.4-16     yaml_2.2.1
## [49] rootSolve_1.8.2.1   zlibbioc_1.34.0     rhdf5_2.32.4
## [52] plyr_1.8.6          grid_4.0.3          parallel_4.0.3
## [55] crayon_1.3.4        lmom_2.8            Biostrings_2.56.0
## [58] splines_4.0.3       multtest_2.44.0     knitr_1.31
## [61] pillar_1.4.6        igraph_1.2.6        EMT_1.1
## [64] boot_1.3-25         gld_2.6.2           codetools_0.2-16
## [67] stats4_4.0.3        glue_1.4.2          evaluate_0.14
## [70] data.table_1.13.0   vctrs_0.3.4         foreach_1.5.1
## [73] gtable_0.3.0        purrr_0.3.4         xfun_0.21
## [76] coin_1.4-0          libcoin_1.0-7       e1071_1.7-4
## [79] class_7.3-17        survival_3.2-7      iterators_1.0.13
## [82] IRanges_2.22.2       cluster_2.1.0        TH.data_1.0-10
## [85] ellipsis_0.3.1
```