W2. 16S analysis

Bora Kim

March 4, 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 16S amplicon sequencing from rhizosphere and roots of 16 rice genotypes grown on two natural soils for 31 days.To begin with, raw 16S amplicon sequence has been reposited in SRA database under accession number: In W2 workflow, there are 6 big main steps as: Processing sequencing data -> alpha diversity -> beta diversity -> effect of SLs on alpha diversity and beta diversity -> Picrust2 -> 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 "W2_16S_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 16S amplicon sequencing (Illumina Miseq)

The primers used in this study amplified V3-V4 16S region as below: Forward 341F- CCTACGGGNBG-CASCAG Reverse 806R- GGACTACNVGGGTWTCTAAT

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 16S V3-V4 Adapter GGCTGACTGACT Read 1 trimming AdapterRead2 CCAATTACCATA Read 2 trimming Reverse-complement R2 primer ATTAGAWACCCBNGTAGTCC for R1 read trimming Reverse-complement R1 primer CTGSTGCVNCCCGTAGG for R2 read trimming

1.1. Remove primer and adapter sequence in raw sequencing data

I used software called 'Cutadapt' employed in linux environmnet. 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 ATTAGAWACCCBNGTAGTCC -A CTGSTGCVNCCCGTAGG --no-indels -o $trimmed_for -p $trimmed_rev ${f
  done
```

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 GGCTGACTGACT -A CCAATTACCATA --no-indels -o $trimmed_for -p $trimmed_rev ${for[i]} ${rev
    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_16S_Bora/16S_primer_trim_3end/16S_adapter_trim_3end/")
path<-"~/Ricebiome/rice_sequencing_process/Rice_16S_Bora/16S_primer_trim_3end/16S_adapter_trim_3end/"
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, truncLen = c(290,170),
                     maxN=0, maxEE=c(2,2), truncQ=2, 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.n
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, "~/silva_nr_v132_train_set.fa.gz", multithread=TRUE)
taxa <- addSpecies(taxa, "~/silva_species_assignment_v132.fa.gz")
```

Afterwards, make small modification on sample names & taxa name, assign unique sequences to amplicon sequence variant (ASV), remove ASVs assigned to unwanted taxa (i.e mitocondria, chloroplast) & 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("bASV", 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)

ps_rm <- ps %>% #remove ASVs assigned to unwanted taxa
  subset_taxa(
    Kingdom == "Bacteria" &
    Phylum !="Cyanobacteria"&
    Family  != "Mitochondria"
  )

ps_rm2 <- prune_taxa(taxa_sums(ps_rm) > 1, ps_rm) #remove singletones
```

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

```
asv<-as.data.frame(otu_table(ps_rm2))
tax<-as.data.frame(tax_table(ps_rm2))
seq<-as.data.frame(refseq(ps_rm2))
```

For further use, made FASTA file.

```
test.seq<-seq
test.seq$rowname<-rownames(test.seq)

Xfasta <- character(nrow(test.seq) * 2)
Xfasta[c(TRUE, FALSE)] <- paste0(">", test.seq$rowname)
Xfasta[c(FALSE, TRUE)] <- test.seq$x # to download the table, writeLines(Xfasta, "Rice_16S_seq.fasta")
```

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_rm2, SAM)
```

As I mentioned ealier, you can find final outputs in R work image "W2_16S_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) EC: abundance table of EC from Picrust2 (please see section 6 below) PW: abundance table of pathway from Picrust2 (please see section 6 below) ec_des: description of EC path_des: description of PW

2. Getting started

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

Glimpse current datasets.

```
load("W2_16S_analysis_image.Rdata") #To load this data image, the package 'phyloseq' is required
ps2
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 2703 taxa and 198 samples ]
## sample_data() Sample Data:       [ 198 samples by 13 sample variables ]
## tax_table()   Taxonomy Table:    [ 2703 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 2703 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]
```

```
##        bASV3 bASV5 bASV6 bASV7 bASV8
## R_A2_e     0     0     0     0     0
## R_A2_r     0     0     0     0     0
## R_A3_e     0     0     0     0     0
## R_A3_r     0     0     0     0     0
## R_A4_e     0     0     0     0     0
```

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

```
##        Kingdom          Phylum               Class                  Order
## bASV3 Bacteria  Proteobacteria Gammaproteobacteria Betaproteobacteriales
## bASV5 Bacteria  Proteobacteria Gammaproteobacteria       Xanthomonadales
## bASV6 Bacteria  Proteobacteria Gammaproteobacteria       Xanthomonadales
## bASV7 Bacteria Verrucomicrobia    Verrucomicrobiae        Pedosphaerales
## bASV8 Bacteria Verrucomicrobia    Verrucomicrobiae        Pedosphaerales
##                    Family
## bASV3   Burkholderiaceae
## bASV5 Rhodanobacteraceae
## bASV6 Rhodanobacteraceae
## bASV7    Pedosphaeraceae
## bASV8    Pedosphaeraceae
```

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)
```

Have a look at the data distribution of microbiome data.

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

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

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

```
## [1] 43973
```

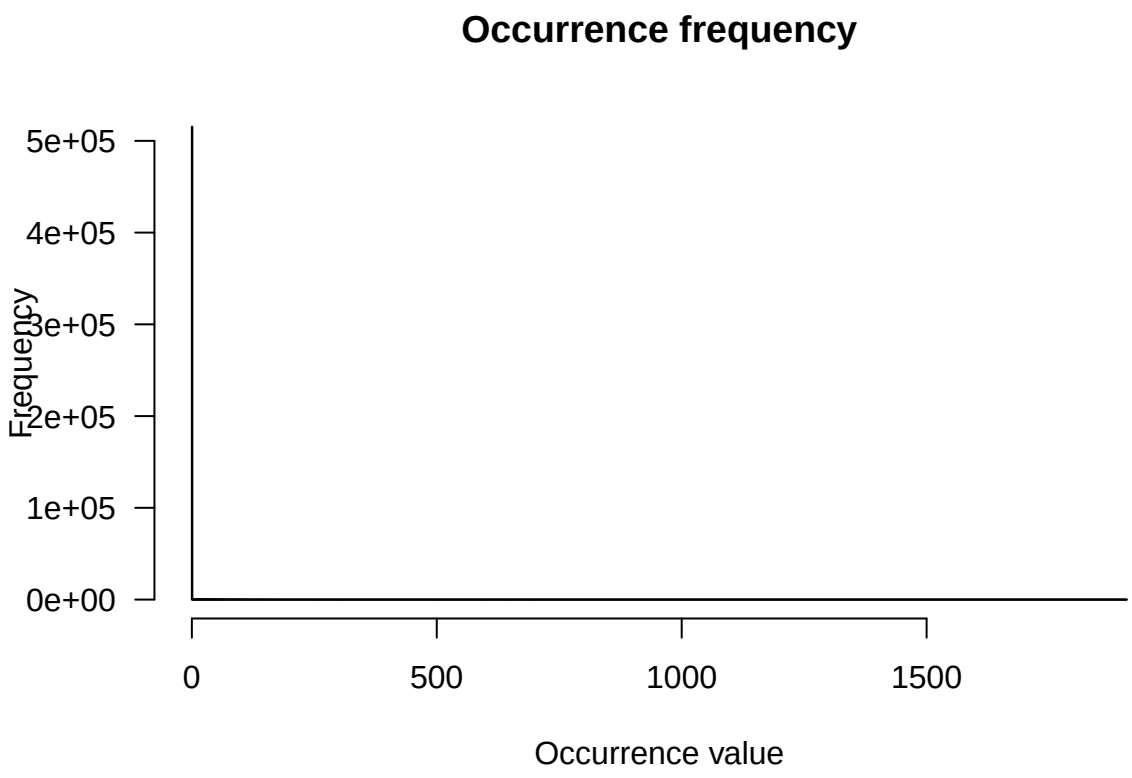
```
nsam<-dim(asv)[1] # number of samples
nvar<-dim(asv)[2] # nubmer of variables
sum(asv==0) #### Number of zeros
```

```
## [1] 515446
```

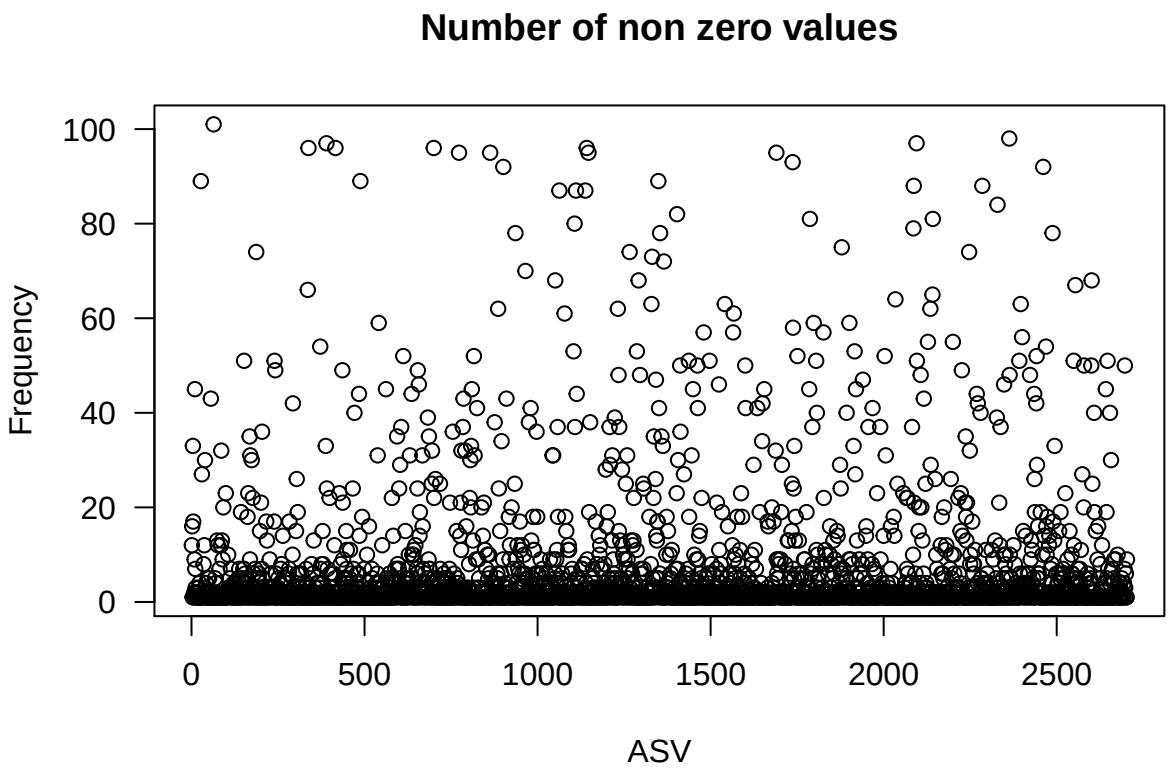
```
sum(asv==0)/(nvar*nsam)*100 #percentage of zeros
```

```
## [1] 96.31012
```

```
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
```



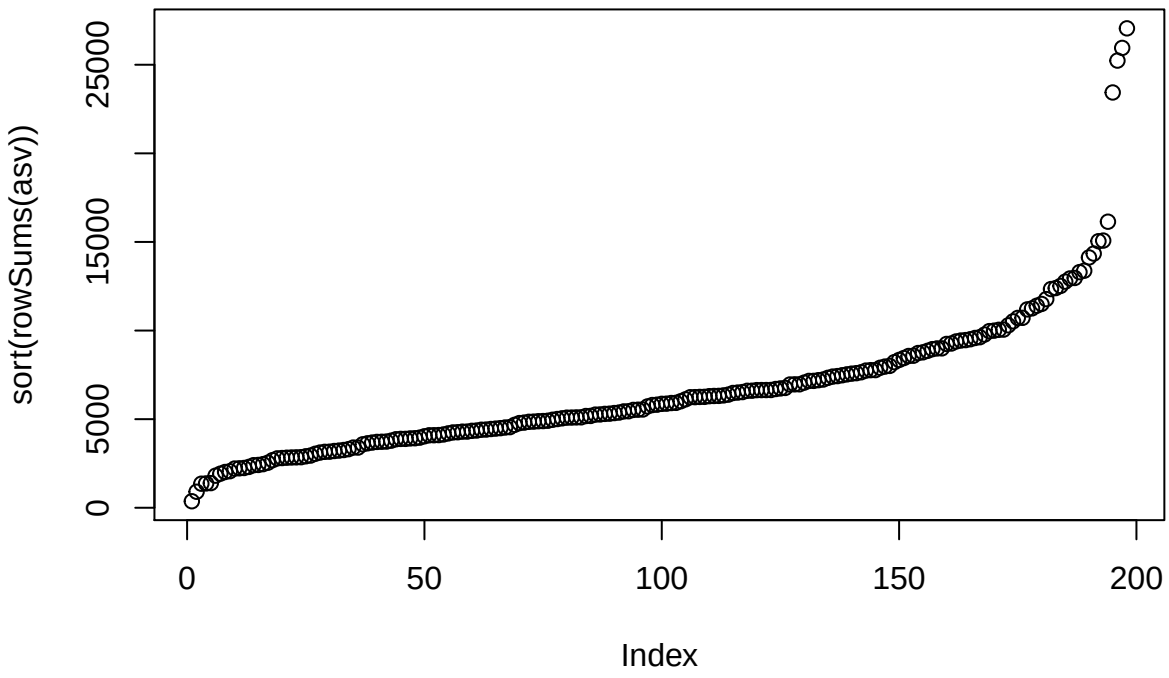

```
min(rowSums(asv)) # minimum sequencing depth in samples
```

```
## [1] 368
```

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

```
## [1] 27054
```

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



3. Alpha diversity of bacterial community (Fig 2A, 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)
```

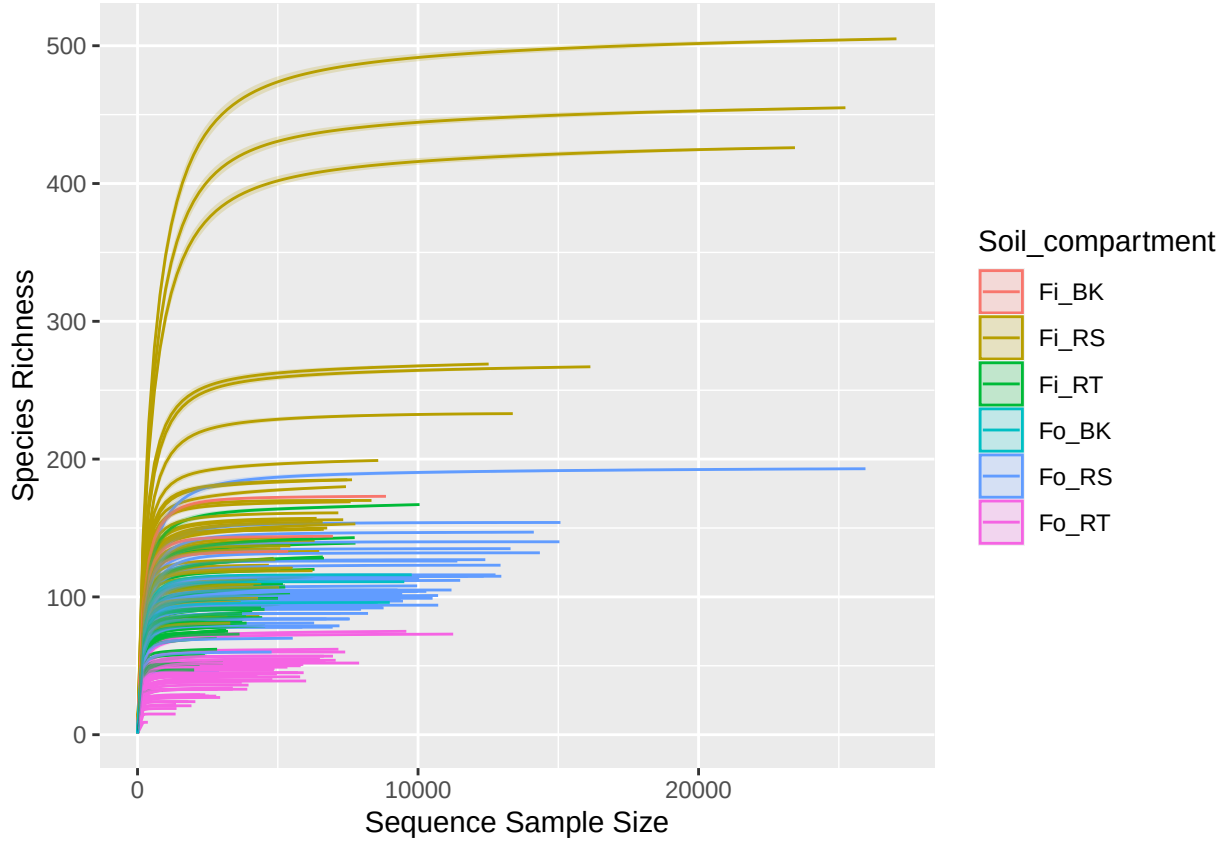
```
## rarefying sample R_A2_e
## rarefying sample R_A2_r
## rarefying sample R_A3_e
## rarefying sample R_A3_r
## rarefying sample R_A4_e
## rarefying sample R_A4_r
## rarefying sample R_A7_e
## rarefying sample R_A7_r
## rarefying sample R_A8_e
## rarefying sample R_A8_r
## rarefying sample R_A9_e
## 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
```

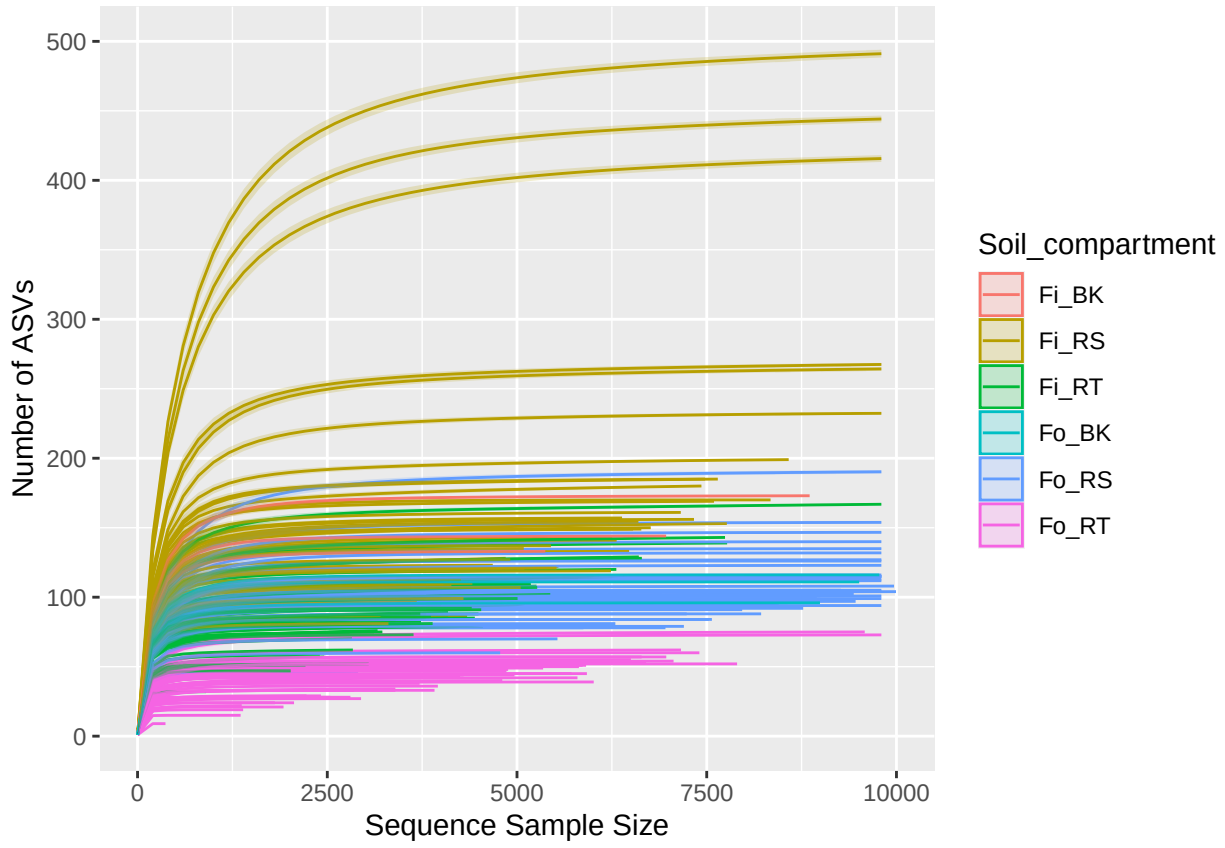
```
## rarefying sample R_G10_r
## rarefying sample R_G2_e
## rarefying sample R_G2_r
## rarefying sample R_G3_e
## rarefying sample R_G3_r
## rarefying sample R_G5_e
## rarefying sample R_G5_r
## rarefying sample R_G8_e
## rarefying sample R_G8_r
## rarefying sample R_G9_e
## rarefying sample R_G9_r
## 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
bac_alpha<-as.data.frame(cbind(shannon, no.species, chao1, evenness, sample_data(ps2)))
bac_alpha$Compartment2<-factor(bac_alpha$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(bac_alpha[,i]~bac_alpha$Soil, data=bac_alpha)
  soil.cs[i]<-k$statistic[[1]]
```

```
  soil.df[i]<-k$parameter[[1]]
  soil.p[i]<- k$p.value
  k<-kruskal.test(bac_alpha[,i]~bac_alpha$Soil_compartment, data=bac_alpha)
  soil.com.cs[i]<-k$statistic[[1]]
  soil.com.df[i]<-k$parameter[[1]]
  soil.com.p[i] <-k$p.value
  names[i]<-colnames(bac_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"    "87.3301693653452" "1"     "1.836726432684e-20"
## [2,] "no.species" "44.593270909785"  "1"     "2.42523766942052e-11"
## [3,] "chao1"      "45.6077390776652" "1"     "1.9262619820962e-11"
## [4,] "evenness"   "144.515428381577" "1"     "1.09638103688777e-32"
##      soil.com.cs        soil.com.df soil.com.p
## [1,] "152.287440992843" "5"         "8.70110225921171e-31"
## [2,] "134.321948311166" "5"         "3.83707802761924e-27"
## [3,] "133.511340196473" "5"         "4.27758266787645e-27"
## [4,] "160.765709862443" "5"         "2.71905083311669e-32"
```

Make summary table of results

```
data=bac_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]
bac_alpha_summary_soil<-st2
bac_alpha_summary_soil
```

```
##                Field.n.99   Forest.n.99             KW_adj.p
## shannon       4.403±0.418    3.603±0.56 8.70110225921171e-31
```

```
## no.species 126.313±73.444 73.162±37.532 3.83707802761924e-27
## chao1      127.289±74.551 73.165±37.543 4.27758266787645e-27
## evenness      0.931±0.011    0.87±0.025 2.71905083311669e-32
```

3.4. Dunn test on alpha diversity indices among soil_compartment group

```
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(bac_alpha[,i]~Soil_compartment, data=bac_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(bac_alpha[1:indices])
rownames(P.unadj) <- PT$res$Comparison
colnames(P.unadj) <- colnames(bac_alpha[1:indices])
rownames(P.adj) <- PT$res$Comparison
colnames(P.adj) <- colnames(bac_alpha[1:indices])
rownames(Let) <- cl$Group
colnames(Let) <- colnames(bac_alpha[1:indices])

Let
```

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

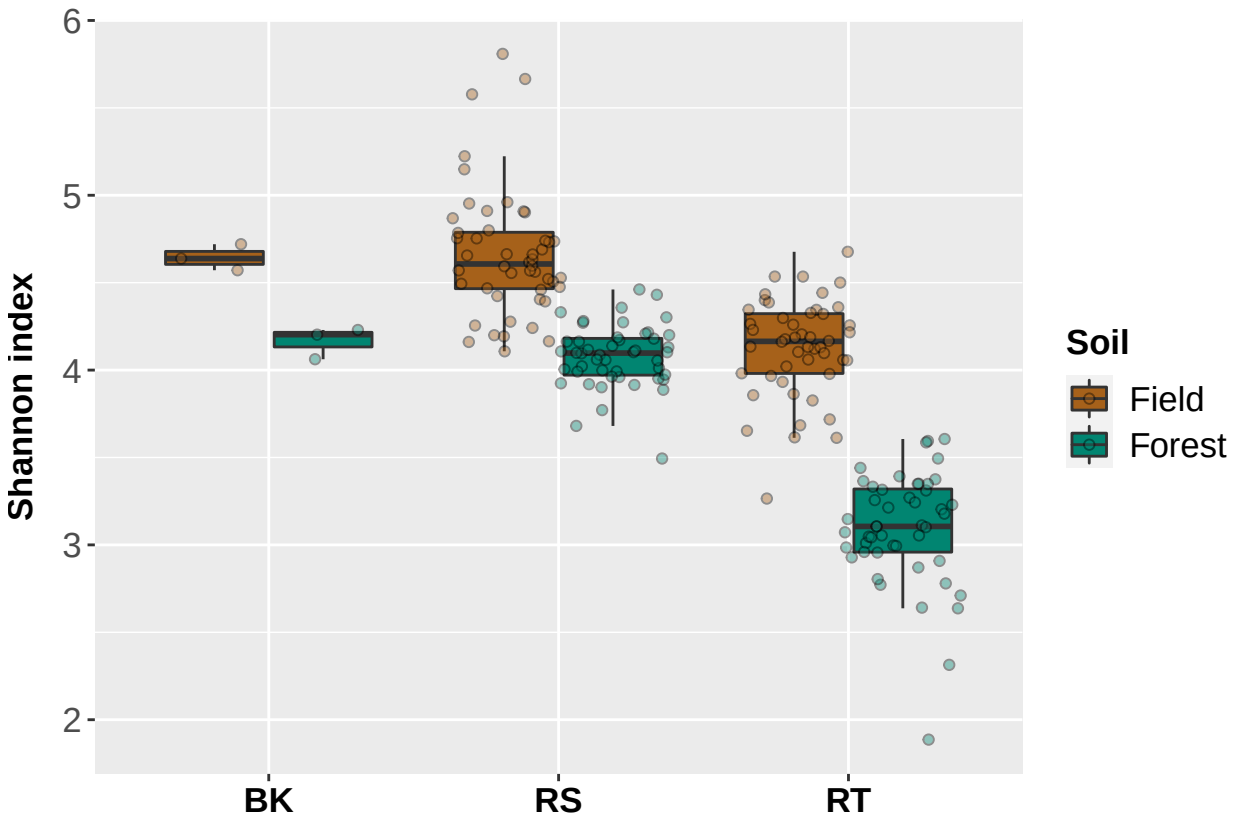
```
data=bac_alpha
by=data$Soil_compartment


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
  }
```

```
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
bac_alpha_summary_soil_com<-as.data.frame(t(st2[,-(5:6)]))

bac_alpha_summary_soil_com
```

```
##                 Fi_BK,n=3       Fi_RS,n=48      Fi_RT,n=48          Fo_BK,n=3
## shannon    4.643±0.075,ab    4.664±0.368,a   4.127±0.28,bc     4.165±0.09,abc
## no.species  150±20.664,ab 160.917±89.121,a 90.229±26.419,c 107.667±10.408,abc
## chao1       150±20.664,ab 161.942±90.764,a  91.216±27.28,c 107.667±10.408,abc
## evenness   0.928±0.01,abc     0.936±0.01,a    0.926±0.01,b    0.891±0.001,bcd
##                   Fo_RS,n=48      Fo_RT,n=48
## shannon         4.078±0.18,c   3.093±0.322,d
## no.species  103.604±24.88,bc 40.562±14.347,d
## chao1      103.611±24.905,bc 40.562±14.347,d
## evenness       0.885±0.012,c   0.853±0.026,d
```

3.5. plot shannon index

```
ggplot(bac_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 2B, Fig 2C)

4.1. Rarefying abundance table

Prior to rarefying, we removed the sample "R_J9_e" due to its low sequencing depth.

```
rar_ps = subset_samples(ps2, sample_names(ps2) != "R_J9_e")
set.seed(1234) # to have reproducible result
rar_ps =rarefy_even_depth(rar_ps, 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(3:4,9:205)]
phyla[1:5,1:5]
```

```
##                Phylum               Class R_A2_e R_A2_r R_A3_e
## bASV3  Proteobacteria Gammaproteobacteria      0      0      0
## bASV5  Proteobacteria Gammaproteobacteria      0      0      0
## bASV6  Proteobacteria Gammaproteobacteria      0      0      0
## bASV7 Verrucomicrobia    Verrucomicrobiae      0      0      0
## bASV8 Verrucomicrobia    Verrucomicrobiae      0      0      0
```

As Proteobacteria occupy huge porpotion of phylum in bacterial community. Therefore, Proteobacteria was subsitute with class level.

```
ptb<-subset(phyla, Phylum == "Proteobacteria")
ptb$Phylum<-ptb$Class
non_ptb<-subset(phyla, !Phylum == "Proteobacteria")
new_phyla<-(rbind(ptb,non_ptb))[,-2]
new_phyla$Phylum <- droplevels(new_phyla)$Phylum
new_phyla[1:5,1:5]
```

```
##                      Phylum R_A2_e R_A2_r R_A3_e R_A3_r
## bASV3  Gammaproteobacteria      0      0      0      0
## bASV5  Gammaproteobacteria      0      0      0      0
## bASV6  Gammaproteobacteria      0      0      0      0
## bASV10 Gammaproteobacteria      0      0      0      0
## bASV12 Gammaproteobacteria      0      0      0      0
```

Create sum of phylum

```
np = length(levels(new_phyla$Phylum)) #number of phylum
ns = 197 #number of sample
phyla_sum = data.frame(matrix(ncol=ns,nrow=np))

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

rownames(phyla_sum)<-ag$Phylum
colnames(phyla_sum)<-colnames(new_phyla[,2:198])
phyla_sum[1:5,1:5]
```

```
##                     R_A2_e R_A2_r R_A3_e R_A3_r R_A4_e
## Actinobacteria          18     10     35     11     20
## Alphaproteobacteria    104    145    148    102     99
## Chlamydiae               0      0      0      0      0
## Deltaproteobacteria     58     36     51     45     44
## Elusimicrobia            0      0      0      3      0
```

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 >=1)
phyla_minor<-subset(phyla_sum, percentage <1)
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]
```

```
##        Actinobacteria Alphaproteobacteria Deltaproteobacteria
## R_A2_e             18                 104                  58
## R_A2_r             10                 145                  36
## R_A3_e             35                 148                  51
## R_A3_r             11                 102                  45
## R_A4_e             20                  99                  44
```

```
##        Gammaproteobacteria Acidobacteria Bacteroidetes Chloroflexi
## R_A2_e                 258             1           317          31
## R_A2_r                 195           118           189          26
## R_A3_e                 244            21           263          74
## R_A3_r                 191           130           216          14
## R_A4_e                 294             6           309          35
##        Patescibacteria Verrucomicrobia Others
## R_A2_e               2              89     25
## R_A2_r              21             151     12
## R_A3_e               2              40     25
## R_A3_r              25             133     36
## R_A4_e               7              53     36
```

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 =197, 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]
```

```
##        Actinobacteria Alphaproteobacteria Deltaproteobacteria
## R_A2_e       1.993355            11.51717            6.423034
## R_A2_r       1.107420            16.05759            3.986711
## R_A3_e       3.875969            16.38981            5.647841
## R_A3_r       1.218162            11.29568            4.983389
## R_A4_e       2.214839            10.96346            4.872647
##        Gammaproteobacteria Acidobacteria Bacteroidetes Chloroflexi
## R_A2_e            28.57143     0.1107420      35.10520    3.433001
## R_A2_r            21.59468    13.0675526      20.93023    2.879291
## R_A3_e            27.02104     2.3255814      29.12514    8.194906
## R_A3_r            21.15172    14.3964563      23.92027    1.550388
## R_A4_e            32.55814     0.6644518      34.21927    3.875969
##        Patescibacteria Verrucomicrobia   Others
## R_A2_e       0.2214839        9.856035 2.768549
## R_A2_r       2.3255814       16.722038 1.328904
## R_A3_e       0.2214839        4.429679 2.768549
## R_A3_r       2.7685493       14.728682 3.986711
## R_A4_e       0.7751938        5.869324 3.986711
```

Run dunn test to compare composition among soil_compartment groups. In this test, phylum 'others' was not included as its comparison is meaningless.

```
indices=9 #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
```

```
##       Actinobacteria Alphaproteobacteria Deltaproteobacteria
## Fi_BK             ab                  ab                 abc
## Fi_RS             ac                   a                  ad
## Fi_RT              b                   a                   d
## Fo_BK              b                   c                 abd
## Fo_RS              a                  bc                   b
## Fo_RT              c                   a                   c
##       Gammaproteobacteria Acidobacteria Bacteroidetes Chloroflexi
## Fi_BK                   a             a          abcd          ab
## Fi_RS                   a             a             a           a
## Fi_RT                   b             b             b           a
## Fo_BK                   a            ac           acd          ab
## Fo_RS                   b             c             c           b
## Fo_RT                   c             c             d           c
##       Patescibacteria Verrucomicrobia
## Fi_BK              ab             abc
## Fi_RS               a               a
## Fi_RT               c               b
## Fo_BK               b             acd
## Fo_RS               b               d
## Fo_RT               d               c
```

4.4. Phylum stack bar plot

Prepare dataset for stack bar plot

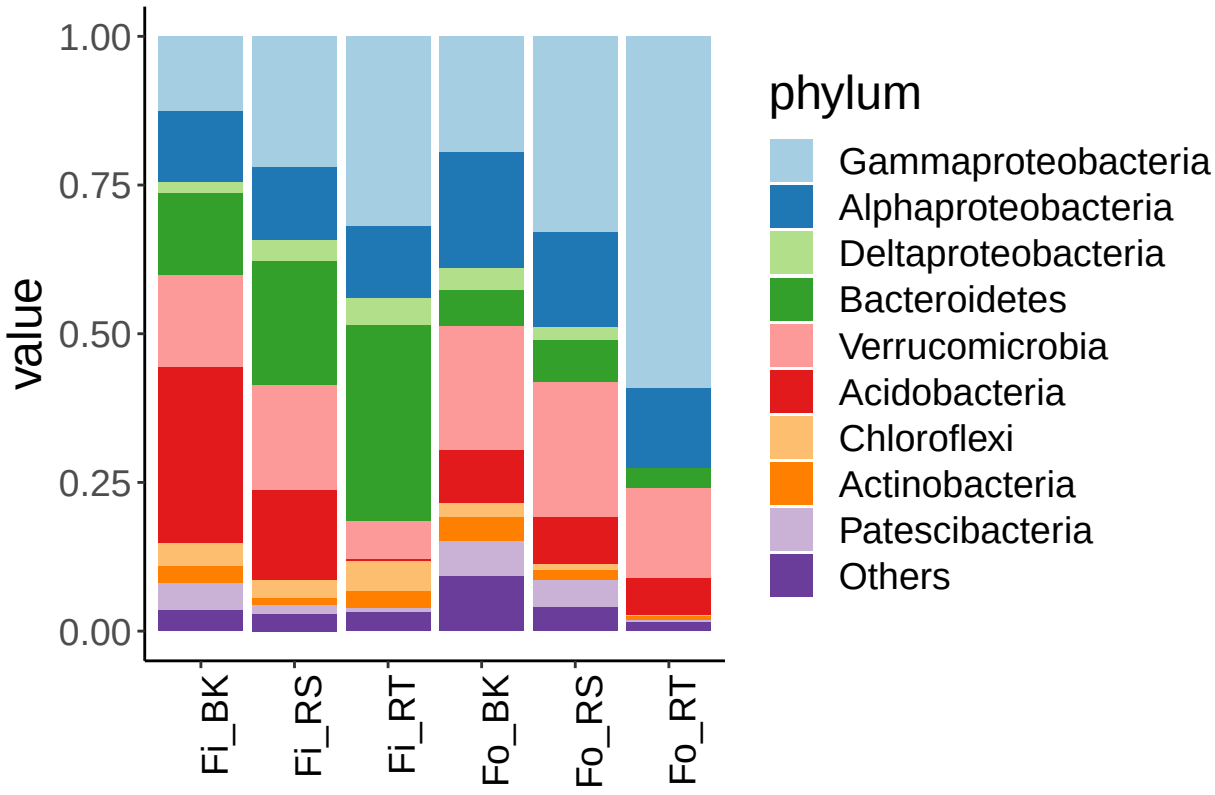
```
np2=10  #number of phylums
nsoilcom=6 #number of factors in soil_com
phyla_soilcom<-matrix(NA,nsoilcom,np2)

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_BS", "Fi_RS","Fi_RT","Fo_BS","Fo_RS", "Fo_RT"))

phyla_soilcom_rc_melt$phylum<-factor(phyla_soilcom_rc_melt$rowname,
                                     c("Gammaproteobacteria","Alphaproteobacteria","Deltaproteobacteria
                                       "Verrucomicrobia","Acidobacteria","Chloroflexi","Actinobacteria"

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

```
##               rowname variable     value              phylum
## 1      Actinobacteria    Fi_BK  8.748616      Actinobacteria
## 2 Alphaproteobacteria    Fi_BK 36.212625 Alphaproteobacteria
## 3 Deltaproteobacteria    Fi_BK  5.426357 Deltaproteobacteria
## 4 Gammaproteobacteria    Fi_BK 37.430786 Gammaproteobacteria
## 5       Acidobacteria    Fi_BK 88.925803       Acidobacteria
```

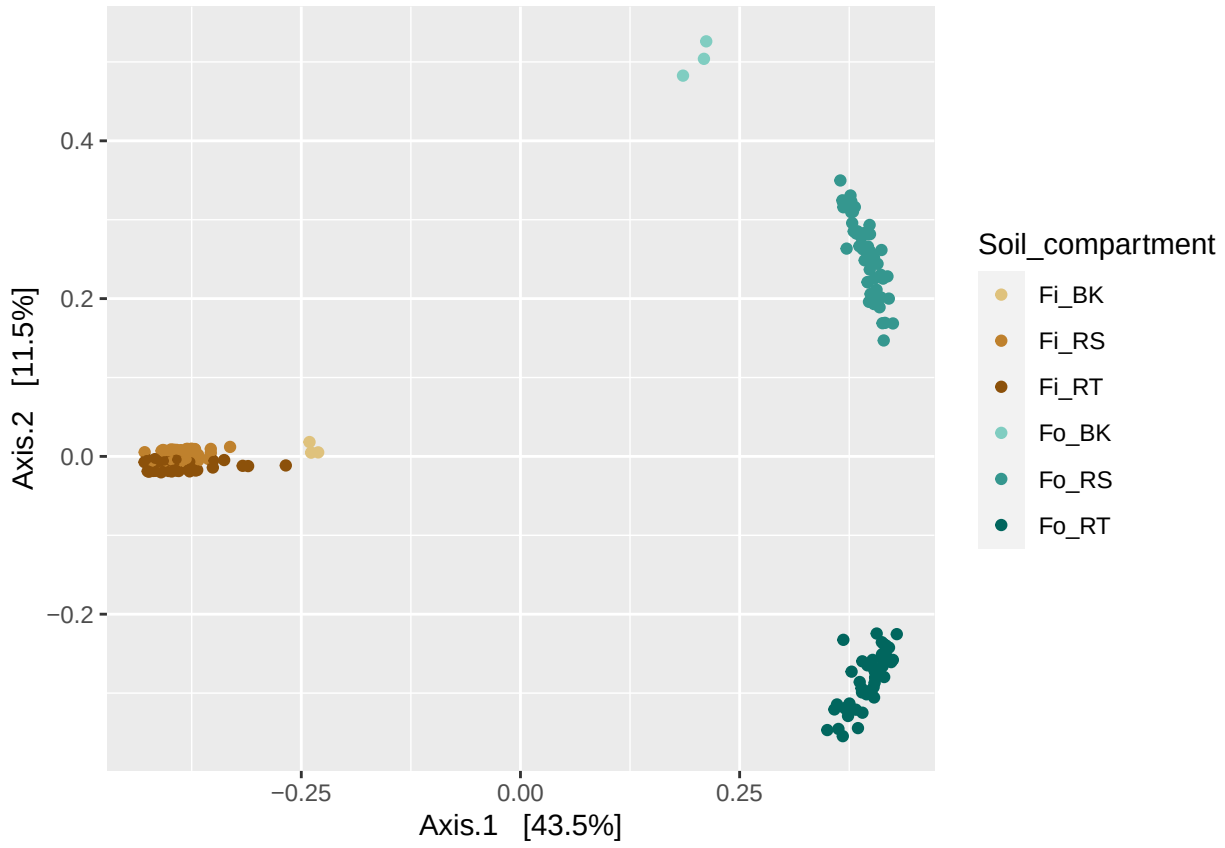
creat plot

```
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 29.542150 29.5421500 260.2893473 0.43131448
## Compartment                   2  7.686742  3.8433712  33.8630935 0.11222620
## Genotype                     15  2.753435  0.1835623   1.6173268 0.04020006
## Soil:Compartment              2  7.660194  3.8300971  33.7461384 0.11183860
## Soil:Genotype                15  2.726229  0.1817486   1.6013467 0.03980286
## Compartment:Genotype         15  1.635535  0.1090356   0.9606890 0.02387876
## Soil:Compartment:Genotype    15  1.620858  0.1080572   0.9520678 0.02366447
## Residuals                   131 14.868152  0.1134973          NA 0.21707457
## Total                       196 68.493295         NA          NA 1.00000000
##                             Pr(>F)
## Soil                        0.0001
## Compartment                 0.0001
## Genotype                    0.0032
## Soil:Compartment            0.0001
## Soil:Genotype               0.0029
## Compartment:Genotype        0.5508
```

```
## Soil:Compartment:Genotype 0.5684
## Residuals                      NA
## Total                          NA
```

subsetting dataset for further use

```
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. Correlation between diversity of bacterial 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.

5.1. Correlation between alpha diversity and SLs level

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

```
match_alpha<-subset(bac_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]~Me05DS_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     orb.p
## shannon    -0.0043394425 0.7450980 -0.0004140046 0.9215686 0.22888511 0.7450980
## no.species -0.2907305039 0.5104167  0.0624749857 0.8627451 1.25357027 1.0000000
## chao1      -0.2907305039 1.0000000  0.0624749857 0.9803922 1.25357027 1.0000000
## evenness   -0.0003106182 0.4000000 -0.0002077470 0.8431373 0.05388696 0.1507092
```

5.2. 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(123)

for(i in 1:var) {
  RT.p[i]<-(anova.cca(capscale(match_Fo_RT[14:2367]~SLs_Fo_RT[,i], match_Fo_RT, dist="bray"), step=1000)
  RS.p[i]<-(anova.cca(capscale(match_Fo_RS[14:2367]~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 Me05DS_pmol_g
## RT.p             0.032       0.186         0.274
## RS.p             0.007       0.863         0.663
```

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

```
# in roots
FoRT_orb_scores<-(scores(capscale(match_Fo_RT[14:2367]~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]
```

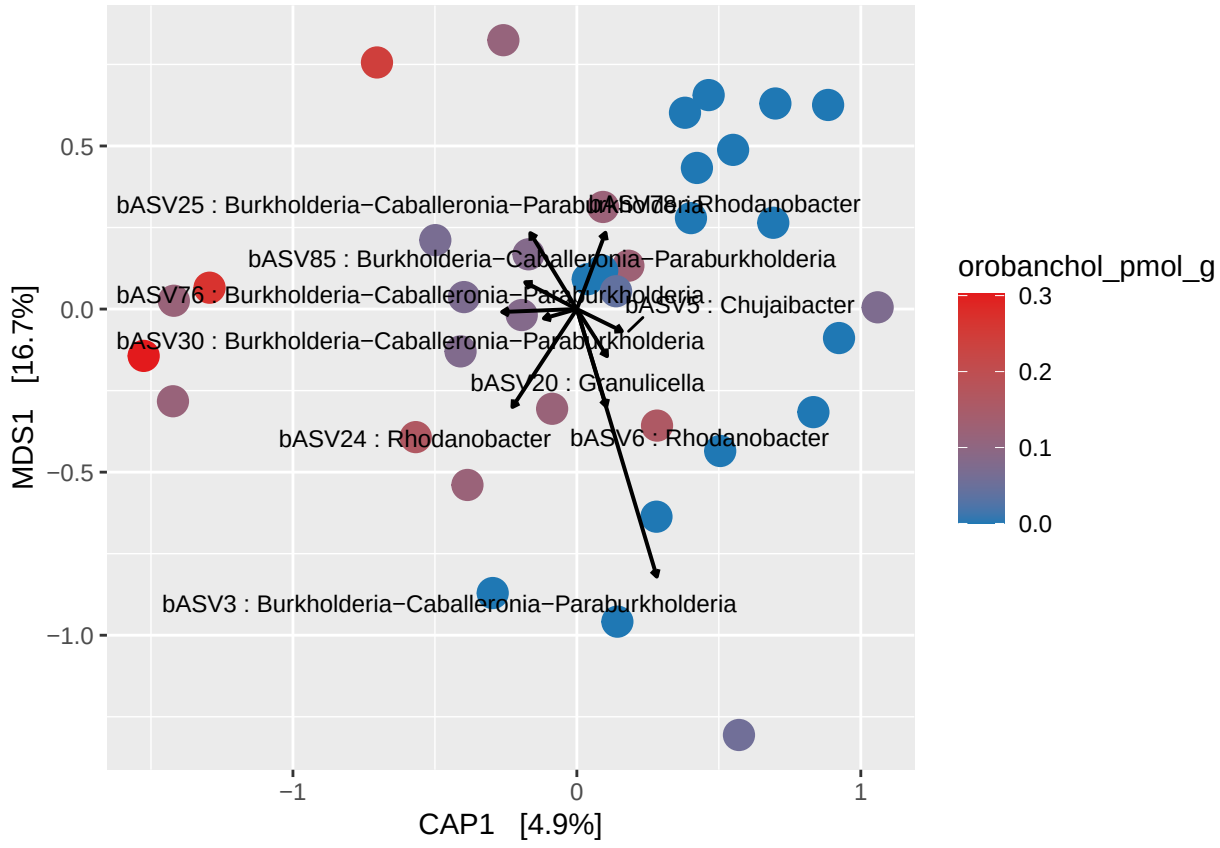
```
##   rowname     orb_CAP1 orb_abs_CAP1  Kingdom          Phylum
## 1   bASV3 -0.300692950  0.300692950 Bacteria  Proteobacteria
## 2   bASV5 -0.013322140  0.013322140 Bacteria  Proteobacteria
## 3   bASV6 -0.018410145  0.018410145 Bacteria  Proteobacteria
## 4   bASV7  0.012140125  0.012140125 Bacteria Verrucomicrobia
## 5   bASV8  0.006097812  0.006097812 Bacteria Verrucomicrobia
##                Class                 Order             Family
## 1 Gammaproteobacteria Betaproteobacteriales   Burkholderiaceae
## 2 Gammaproteobacteria       Xanthomonadales Rhodanobacteraceae
## 3 Gammaproteobacteria       Xanthomonadales Rhodanobacteraceae
## 4    Verrucomicrobiae        Pedosphaerales    Pedosphaeraceae
## 5    Verrucomicrobiae        Pedosphaerales    Pedosphaeraceae
##                                    Genus
## 1 Burkholderia-Caballeronia-Paraburkholderia
## 2                             Chujaibacter
## 3                            Rhodanobacter
## 4                                  Unknown
## 5                                  Unknown
```

```
# 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   bASV3  0.05647725   0.05647725 Bacteria  Proteobacteria Gammaproteobacteria
## 2   bASV5 -0.07164689   0.07164689 Bacteria  Proteobacteria Gammaproteobacteria
## 3   bASV6 -0.08350647   0.08350647 Bacteria  Proteobacteria Gammaproteobacteria
## 4   bASV7 -0.08838933   0.08838933 Bacteria Verrucomicrobia    Verrucomicrobiae
## 5   bASV8  0.02542523   0.02542523 Bacteria Verrucomicrobia    Verrucomicrobiae
##                   Order             Family
## 1 Betaproteobacteriales   Burkholderiaceae
## 2       Xanthomonadales Rhodanobacteraceae
## 3       Xanthomonadales Rhodanobacteraceae
## 4        Pedosphaerales    Pedosphaeraceae
## 5        Pedosphaerales    Pedosphaeraceae
##                                    Genus
## 1 Burkholderia-Caballeronia-Paraburkholderia
## 2                             Chujaibacter
## 3                            Rhodanobacter
## 4                                  Unknown
## 5                                  Unknown
```

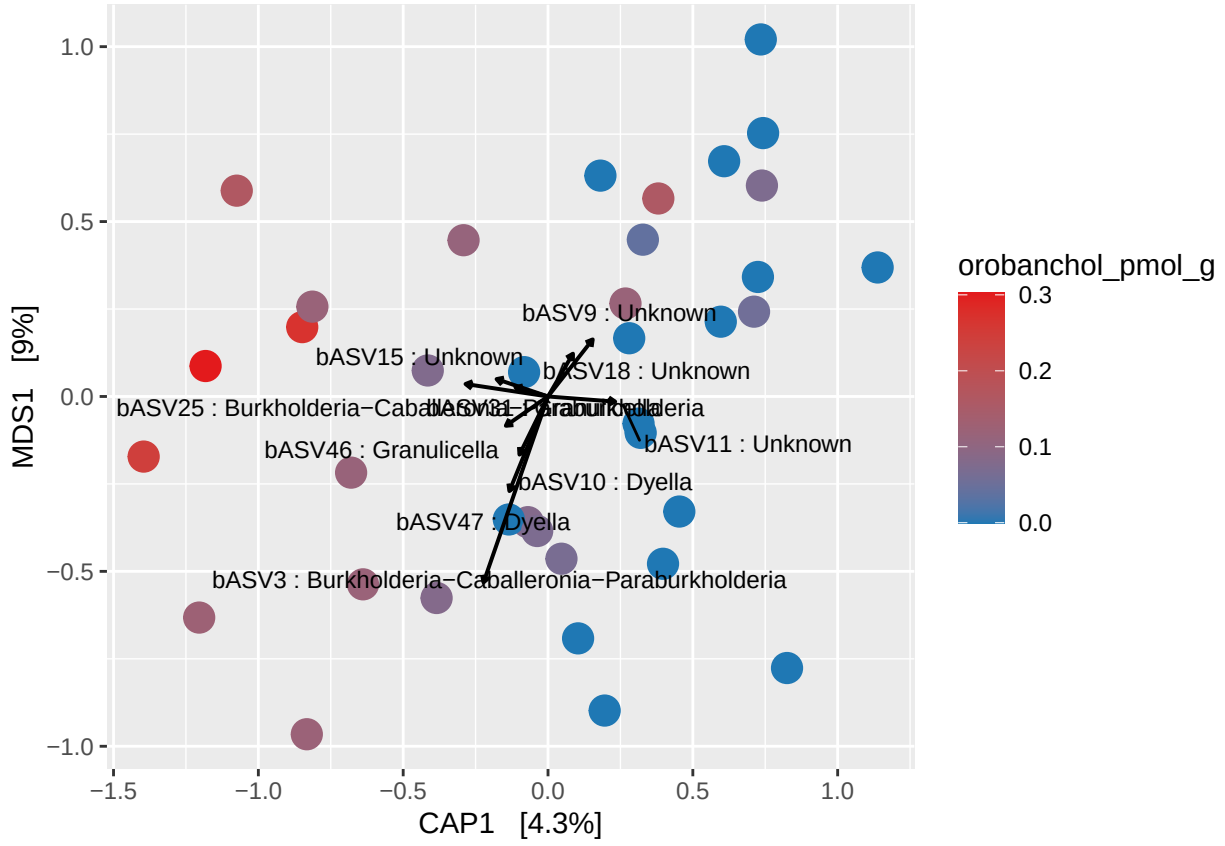
Now plot CAP results

```
# in roots
cap<-ordinate(physeq = match_ps_Fo_RT, method = "CAP", distance = "bray", formula = ~ orobanchol_pmol_g)
cap.p<-plot_ordination(physeq = match_ps_Fo_RT, ordination = cap, axes = c(1,2),
                       color = "orobanchol_pmol_g")+ geom_point(size = 5)+
  scale_color_gradient(high = "#e31a1c", low = "#1f78b4")

cap.sc <- data.frame(vegan::scores(cap, display = "species"))
cap.sc <- rownames_to_column(cap.sc)

cap.cut <-subset(cap.sc, (abs(CAP1))>=0.1) #cutoff value
cap.tax<-left_join(cap.cut, tax_rc, by="rowname")
cap.tax<-unite(cap.tax, col=newname, c(rowname, Genus), sep=" : ", remove=F)

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 = newname)
arrowhead = arrow(length = unit(0.01, "npc"))

cap.p + geom_segment(mapping = arrow_map, size = .7,data = cap.tax, arrow = arrowhead) +
  geom_text_repel(mapping = label_map,data = cap.tax, size=3, show.legend = F)
```




```
# in rhizosphere
cap<-ordinate(physeq = match_ps_Fo_RS, method = "CAP", distance = "bray", formula = ~ orobanchol_pmol_g)
cap.p<-plot_ordination(physeq = match_ps_Fo_RS, ordination = cap, axes = c(1,2),
                       color = "orobanchol_pmol_g")+ geom_point(size = 5)+
  scale_color_gradient(high = "#e31a1c", low = "#1f78b4")
```

```
cap.sc <- data.frame(vegan::scores(cap, display = "species"))
cap.sc <- rownames_to_column(cap.sc)

cap.cut <-subset(cap.sc, (abs(CAP1))>=0.08) #cutoff value
cap.tax<-left_join(cap.cut, tax_rc, by="rowname")
cap.tax<-unite(cap.tax, col=newname, c(rowname, Genus), sep=" : ", remove=F)

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 = newname)
arrowhead = arrow(length = unit(0.01, "npc"))

cap.p + geom_segment(mapping = arrow_map, size = .7,data = cap.tax, arrow = arrowhead) +
  geom_text_repel(mapping = label_map,data = cap.tax, size=3, show.legend = F)
```



6. PiCRUST2

PiCRUST2 was employed to predict functionality of bacterial community. As it required linux envirionment, I made virtual machine ubuntu (version 18.04, 64.bit) in VMware workstaion 15 player. Due to differnt operating environment and language (python),here I only share code without intermediate data or results. However, final output will be shared in work image.

```
conda activate picurst2
cd rice_picrust # directory where my files are

#-s: my sequencing file (fasta format), -i:my ASV table, -o:output dir name
picrust2_pipeline.py -s Rice_16S_seq.fasta -i Rice_16S_asv_picrust2.txt -o picrust2_out_pipeline
```

```
#Add descriptions on output

cd picrust2_out_pipeline

add_descriptions.py -i EC_metagenome_out/pred_metagenome_unstrat.tsv.gz -m EC \
-o EC_metagenome_out/pred_metagenome_unstrat_EC_descrip.tsv.gz

add_descriptions.py -i KO_metagenome_out/pred_metagenome_unstrat.tsv.gz -m KO \
-o KO_metagenome_out/pred_metagenome_unstrat_KO_descrip.tsv.gz

add_descriptions.py -i pathways_out/path_abun_unstrat.tsv.gz -m METACYC \
-o pathways_out/path_abun_unstrat_descrip.tsv.gz
```

Final outputs from PiCRUST2 were imported to current R studio in local computer and the outputs named EC (abundance table of EC), PW (abundance table of pathway), and their description (ec_des, path_des) can be found in provided work image "W2_16S_analaysis_image".

For further analyses, abundance table of both EC and pathway were rarefied

```
# EC table
tEC<-as.data.frame(t(EC[,-1]))
tEC_round <- round(tEC)
tEC.ps<-phyloseq(otu_table(as.matrix(tEC_round), taxa_are_rows = FALSE),sample_data(ps2))
rar_tEC.ps = subset_samples(tEC.ps, sample_names(tEC.ps) != "R_J9_e") #remove low depth sample
set.seed(1234)
rar_tEC.ps =rarefy_even_depth(rar_tEC.ps, rngseed=T, replace = F)

# pathway table
tPW<-as.data.frame(t(PW[,-1]))
tPW_round <- round(tPW)
tPW.ps<-phyloseq(otu_table(as.matrix(tPW_round), taxa_are_rows = FALSE),sample_data(ps2))
rar_tPW.ps = subset_samples(tPW.ps, sample_names(tPW.ps) != "R_J9_e") #remove low depth sample
set.seed(1234)
rar_tPW.ps =rarefy_even_depth(rar_tPW.ps, rngseed=T, replace = F)
```

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

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

7.1. Picrust dataset

```
#EC dataset, Root
match_Fo_RT_rar_tEC.ps<-subset_samples(rar_tEC.ps,Soil_compartment=="Fo_RT"&SL_analysis=="yes")
match_Fo_RT_rar_tEC.ps_filt=filter_taxa(match_Fo_RT_rar_tEC.ps, function(x) sum(x > 2) > (0.4*length(x)
match_Fo_RT_rar_tEC_filt<-otu_table(match_Fo_RT_rar_tEC.ps_filt)
FoRT_EC <-as.data.frame(match_Fo_RT_rar_tEC_filt[,colSums(match_Fo_RT_rar_tEC_filt[,])>2*(dim(match_Fo_


#EC dataset, Rhizosphere
match_Fo_RS_rar_tEC.ps<-subset_samples(rar_tEC.ps,Soil_compartment=="Fo_RS"&SL_analysis=="yes")
match_Fo_RS_rar_tEC.ps_filt=filter_taxa(match_Fo_RS_rar_tEC.ps, function(x) sum(x > 2) > (0.4*length(x)
match_Fo_RS_rar_tEC_filt<-otu_table(match_Fo_RS_rar_tEC.ps_filt)
FoRS_EC<-as.data.frame(match_Fo_RS_rar_tEC_filt[,colSums(match_Fo_RS_rar_tEC_filt[,])>2*(dim(match_Fo_R
```

```
#Pathway dataset, Root
match_Fo_RT_rar_tPW.ps<-subset_samples(rar_tPW.ps,Soil_compartment=="Fo_RT"&SL_analysis=="yes")
match_Fo_RT_rar_tPW.ps_filt=filter_taxa(match_Fo_RT_rar_tPW.ps, function(x) sum(x > 2) > (0.4*length(x))
match_Fo_RT_rar_tPW_filt<-otu_table(match_Fo_RT_rar_tPW.ps_filt)
FoRT_PW<-as.data.frame(match_Fo_RT_rar_tPW_filt[,colSums(match_Fo_RT_rar_tPW_filt[,])>2*(dim(match_Fo_RT

#Pathway dataset, rhizosphere
match_Fo_RS_rar_tPW.ps<-subset_samples(rar_tPW.ps,Soil_compartment=="Fo_RS"&SL_analysis=="yes")
match_Fo_RS_rar_tPW.ps_filt=filter_taxa(match_Fo_RS_rar_tPW.ps, function(x) sum(x > 2) > (0.4*length(x))
match_Fo_RS_rar_tPW_filt<-otu_table(match_Fo_RS_rar_tPW.ps_filt)
FoRS_PW<-as.data.frame(match_Fo_RS_rar_tPW_filt[,colSums(match_Fo_RS_rar_tPW_filt[,])>2*(dim(match_Fo_RS
```

7.2. ASVs

```
#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)
bac.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)
bac.FoRS_ASV<-cbind(sample_data(match_ps_Fo_RS_filt), match_Fo_RS_filt)
```

7.3. Genus level

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$rowname
RT_genus<-RT_tax[,c(7,9:44)]
RT_genus$Genus <- droplevels(RT_genus)$Genus
np = length(levels(RT_genus$Genus)) #number of genus
ns = 36 #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$Genus
colnames(RT_genus_sum)<-colnames(RT_genus[,2:37])
RT_genus_sum$percentage<-rowSums(RT_genus_sum)/sum(rowSums(RT_genus_sum))*100
RT_major_genus<-as.data.frame(t(subset(RT_genus_sum, percentage >=1))) #select genera which are abundan
RT_major_genus = RT_major_genus[!row.names(RT_major_genus)%in% "percentage",]#remove percentage row
bac.FoRT_genus = RT_major_genus[,-10] #remove unknown genus
bac.FoRT_genus[1:5,1:9]
```

```
##        Acidocella Asticcacaulis Bordetella
## R_A7_e         13            38         65
## R_A8_e         15            44         46
```

```
## R_A9_e         15            62         34
## R_B7_e          8            32         75
## R_B8_e          4            44         28
##        Burkholderia-Caballeronia-Paraburkholderia Chujaibacter Dyella
## R_A7_e                                        175           45    129
## R_A8_e                                        165           37     78
## R_A9_e                                        206           37     83
## R_B7_e                                        168           56    139
## R_B8_e                                        252           25    136
##        Granulicella Mucilaginibacter Rhodanobacter
## R_A7_e           33               18           142
## R_A8_e           68               25           136
## R_A9_e           27               26           161
## R_B7_e           51               13           140
## R_B8_e           48               22           118
```

```
#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$Genus
colnames(RS_genus_sum)<-colnames(RS_genus[,2:38])
#Select genera
RS_genus_sum$percentage<-rowSums(RS_genus_sum)/sum(rowSums(RS_genus_sum))*100
RS_major_genus<-as.data.frame(t(subset(RS_genus_sum, percentage >=0.5))) #select genera which are abund
RS_major_genus = RS_major_genus[!row.names(RS_major_genus)%in% "percentage",]#remove percentage row
bac.FoRS_genus = RS_major_genus[,-18] #remove unknown genus
bac.FoRS_genus[1:5,1:9]
```

```
##        Acidibacter Acidicaldus Acidipila Acidocella Acidothermus Asticcacaulis
## R_A7_r          20           0         3          9           22            15
## R_A8_r          14           0         9         12            5             7
## R_A9_r           9           7        11         16            0            11
## R_B7_r          12           6         6          6           23             8
## R_B8_r           5           9        14         11            7            14
##        Bordetella Bradyrhizobium Burkholderia-Caballeronia-Paraburkholderia
## R_A7_r         15             10                                        107
## R_A8_r         22              8                                         86
## R_A9_r         27              7                                        165
## R_B7_r         13              8                                        114
## R_B8_r         17              9                                         97
```

change object name of tax to bac_tax to use in W4

```
bac_tax<-tax_rc
```

Final outputs from 7.1 ~ 7.3 can be found in work image "W4_correlation_study_image.Rdata".

Version

```
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] lmPerm_2.1.0       ggrepel_0.9.1      tidyr_1.1.2        reshape2_1.4.4
##  [5] multcompView_0.1-8 rcompanion_2.3.27  FSA_0.8.30         vegan_2.5-6
##  [9] lattice_0.20-41    permute_0.9-5      ggplot2_3.3.2      ranacapa_0.1.0
## [13] 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] MASS_7.3-53         rootSolve_1.8.2.1   zlibbioc_1.34.0
## [52] rhdf5_2.32.4        plyr_1.8.6          grid_4.0.3
## [55] parallel_4.0.3      crayon_1.3.4        lmom_2.8
## [58] Biostrings_2.56.0   splines_4.0.3       multtest_2.44.0
## [61] knitr_1.31          pillar_1.4.6        igraph_1.2.6
## [64] EMT_1.1             boot_1.3-25         gld_2.6.2
## [67] codetools_0.2-16    stats4_4.0.3        glue_1.4.2
## [70] evaluate_0.14       data.table_1.13.0   vctrs_0.3.4
## [73] foreach_1.5.1       gtable_0.3.0        purrr_0.3.4
## [76] xfun_0.21           coin_1.4-0          libcoin_1.0-7
## [79] e1071_1.7-4         class_7.3-17        survival_3.2-7
```

```
## [82] iterators_1.0.13    IRanges_2.22.2      cluster_2.1.0
## [85] TH.data_1.0-10      ellipsis_0.3.1
```