

## ARTICLE TEMPLATE

# Scripts: Bacterial communities in the rhizosphere at different growth stages of maize cultivated in soil under conventional and conservation agricultural practices

Yendi E. Navarro-Noya<sup>a</sup>, Stephanie Hereira-Pacheco<sup>b</sup>

<sup>a</sup>Centro Tlaxcala de Biología de la Conducta, Universidad Autónoma de Tlaxcala, Tlaxcala, Mexico; <sup>b</sup>Laboratory of Soil Ecology, CINVESTAV-IPN, Ciudad de México, Mexico

## ARTICLE HISTORY

Compiled July 30, 2021

## 1. I. QIIME2 AND PICRUST2 SCRIPTS

Raw sequences were import to QIIME2 (Bolyen et al. 2019) workflow and then PICRUST2 (Douglas et al. 2020) was done to predict functionality.

### 1.0.1. IMPORT TO QIIME AND DEMULTIPLEX SEQUENCES

```
qiime tools import --type EMPPairedEndSequences \
--input-path barcode_extracted/ \
--output-path yen.qza
```

--type : type of file , in this case paired end sequences. Check other import types<sup>1</sup>.

--input-path: directory with the files to import

--output-path: artifact name output

And then, we perform the demultiplexing:

```
qiime demux emp-paired \
--i-seqs yen.qza \
--m-barcodes-file Map_rhizos.txt \
--m-barcodes-column BarcodeSequence \
--o-per-sample-sequences demux.qza \
--o-error-correction-details errordetails.qza \
--p-no-golay-error-correction
```

--i-seqs : artifact with the import paired end sequences

--m-barcodes-file : mapping file containing information of the sequences

--m-barcodes-column: column name of the Barcode sequences

--o-per-sample-sequences : output of the sequences demultiplexed

CONTACT Yendi E. Navarro-Noya. Email: nyendi@hotmail.com;, Stephanie Hereira-Pacheco. Email: shereirap@gmail.com

<sup>1</sup><https://docs.qiime2.org/2021.4/tutorials/importing/>

```
-o-error-correction-details: file with correction details  
-p-no-golay-error-correction: by default perform a correction with a barcode of 12  
nt if not use this option (in our case is 16 nt)
```

To visualice the demux file:

```
qiime demux summarize  
--i-data demux.qza \  
--o-visualization demux.qzv
```

```
-i-data : demultiplexed and/or trimmed sequences  
-o-visualization : output
```

In this case, due to de the low quality of reverse reads we will continue with the forward sequences and let's set the truncation length of 120 bp for forward reads.

#### 1.0.2. RUN DADA2

```
qiime dada2 denoise-single \  
--i-demultiplexed-seqs ../demultiplex/demux_yen.qza \  
--p-trim-left 0 --p-trunc-len 120 \  
--o-representative-sequences rep-seq-dada-forward.qza \  
--o-table table-dada-forward.qza \  
--o-denoising-stats stats-dada-forward.qza
```

```
-i-demultiplexed-seqs : demultiplexed and trimmed sequences  
-p-trunc-len-f : length to trunc in forward sequences sequences to obtain good quality  
(usually when sequencing drops)  
-p-trunc-len-r : length to trunc in resverse sequences sequences to obtain good quality  
(usually when sequencing drops)  
-output-dir : output directory that will contain feature-table and representative  
sequences
```

#### 1.0.3. FILTERING FORM ALIGNMENT (REMOVE UNASSIGNED BASED ON GREEN GENES DATABASE)

First, we do the alignment against the green genes database:

```
qiime quality-control exclude-seqs \  
--i-query-sequences rep-seq-dada-forward.qza \  
--i-reference-sequences ../references/99_ottus.qza \  
--p-method vsearch \  
--p-perc-identity 0.97 \  
--p-perc-query-aligned 0.95 \  
--p-threads 4 \  
--o-sequence-hits hits.qza \  
--o-sequence-misses misses.qza
```

```
-i-query-sequences : representative sequences obtained from dada2  
-i-reference-sequences : reference sequences imported to qiime2  
-p-method : alignment method
```

- p-perc-identity : identity percent
- p-perc-query-aligned : query aligned percent
- p-threads : number of threads
- o-sequence-hits : output with hits sequences
- o-sequence-misses : output with misses sequences (not aligned)

Now, filter the feature table to remove this misses sequences:

```
qiime feature-table filter-features \
--i-table table-dada-forward.qza \
--m-metadata-file misses.qza \
--o-filtered-table no-misses-table.qza \
--p-exclude-ids
```

- i-table : feature table from dada2
- m-metadata-file : metadata mapping file
- o-filtered-table : filtered table
- p-exclude-ids : argument to exclude the ids from ‘misses’ file

#### 1.0.4. ASSIGN TAXONOMY

```
qiime feature-classifier classify-sklearn \
--i-reads rep-seq-dada_forward.qza \
--i-classifier /home/steph/Descargas/gg-13-8-99-nb-classifier.qza \
--o-classification taxonomy.qza
```

- cclassify-sklearn : using sklearn (other options are vsearch and blast)
- i-reads : seqs merged
- i-classifier: artifact classifier full-length (<https://docs.qiime2.org/2021.4/data-resources/>)
- o-classification output artifact with taxonomy

#### 1.0.5. FILTERING TABLE

- Removing taxa of chloroplast and mitochondria

```
qiime taxa filter-table
--i-table no-misses-table.qza
--i-taxonomy taxonomy.qza
--p-exclude mitochondria,chloroplast
--o-filtered-table table_filtered.qza
```

- i-table : merge table
- i-taxonomy : taxonomy (from assign taxonomy)
- p-exclude : taxa to exclude
- o-filtered-table : output/artifact

- Visualizing the taxonomy in a barplot

```
qiime taxa barplot --i-table table_filtered.qza \
--i-taxonomy taxonomy.qza \
```

```
--m-metadata-file Map_rhizos.txt \
--o-visualization taxa_barplot.qzv
```

```
qiime tools view taxa-barplot.qzv
```

```
-i-table : input table
-m-metadata-file : mapping file
-i-taxonomy : taxonomy
-o-visualization: .qzv of barplot
```

#### 1.0.6. FILTERING SEQUENCES

For this step we will filter the representative sequences base on the table filtered.

```
qiime feature-table filter-seqs \
--i-data rep-seq-dada-forward.qza \
--i-table table_filtered.qza \
--o-filtered-data rep-seqs-filter-exclude.qza
```

```
-i-data : input sequences
-i-table : input table use to filter
-o-filtered-data : output/filtered sequences
```

#### 1.0.7. BUILDING THE TREE

For this step we will build the phylogenetic tree *denovo*.

```
qiime phylogeny align-to-tree-mafft-fasttree \
--i-sequences rep-seqs-filter-exclude.qza \
{output-dir tree/}
```

```
-i-sequences : sequences filtered
--output-dir : output director that will contain the alignment, masked alignment,
the tree and the rooted treed.
```

#### 1.0.8. EXPORTING SEQUENCES, TABLE AND TAXONOMY

```
#export sequences
qiime tools export \
--input-path rep-seqs-filter-exclude.qza \
--output-path exported

#expor the feature table
qiime tools export \
--input-path .table_filtered.qza \
--output-path exported/

#export the taxonomy
```

```

qiime tools export \
--input-path taxonomy.qza \
--output-path exported/

#join the feature table and taxonomy
biom add-metadata \
-i exported/feature-table_grouped.biom \
--observation-metadata-fp exported/taxonomy.tsv \
-o otutable_with_taxonomy.biom

#convert biom to tsu to check the otutable (feature-table)
biom convert -i otutable_with_taxonomy.biom
-o otutable.txt --to-tsv --header-key taxonomy

```

-input-path: artifact to export (table or taxonomy)  
 -output-path: directory output  
 -i : feature-table in biom format  
 -observation-metadata-fp : taxonomy file (already changed)  
 -o: output  
 --to-tsv --header-key taxonomy : options to convert and add taxonomy to  
 otutable/feature-table

#### 1.0.9. PICRUST2

```

picrust2_pipeline.py \
-s exported/dna-sequences.fasta \
-i exported/feature-table.biom \
-o picrust2

add_descriptions.py \
-i picrust2/EC_metagenome_out/pred_metagenome_unstrat.tsv.gz \
-m EC -o picrust2/EC_metagenome_out/pred_metagenome_unstrat_descrip.tsv.gz

add_descriptions.py \
-i picrust2/pathways_out/path_abun_unstrat.tsv.gz \
-m METACYC -o picrust2/pathways_out/path_abun_unstrat_descrip.tsv.gz

```

-s : exported sequences from qiime2 in fasta format  
 -i : exported table from qiime2 in biom format  
 -o: directory that contains the results (EC, KO, pathways)  
 In the add\_descriptions.py (script to add the descriptions to EC and pathways file):  
 -i : file output from PICRUST2 pipeline (EC or pathways)  
 -m METACYC/EC : map type  
 -o : output file with descriptions

*\*The files obtained from these scripts were imported into R for downstream analyses.*

## 2. II. ALPHA AND FUNCTIONALITY PLOT

### 2.0.1. Loading libraries

```
library(hillR)
library(gplots)
library(lme4)
library(nlme)
library(ggplot2)
library(cowplot)
library(pgirmess) # includes PermTest()
library(dplyr)
```

### 2.0.2. Loading files and formatting

```
#Species as rows, traits as columns
EC_predicted <- read.delim(gzfile("../Data/EC_predicted.tsv.gz"), row.names=1)
K0_predicted <- read.delim(gzfile("../Data/K0_predicted.tsv.gz"), row.names=1)

#Sites as rows, species as columns
otutable <- read.delim("../Data/otutable_final_picrust2.txt", row.names=1)
otu_table<- otutable[,1:72]
totutable <- t(otu_table)
totutable <- totutable[ , match(rownames(K0_predicted), colnames(totutable))]
```

### 2.0.3. Functional diversity with Hill numbers

```
#Calculate the functional diversity (Not running due to long time)

#func_parti_q0<-hill_func_parti(totutable, traits = EC_predicted, q = 0)
#func_parti_q1<-hill_func_parti(totutable, traits = EC_predicted, q = 1)
#func_parti_q2<-hill_func_parti(totutable, traits = EC_predicted, q = 2)

#func_q0<- hill_func(totutable, traits = EC_predicted, q = 0)
#func_q1<- hill_func(totutable, traits = EC_predicted, q = 1)
#func_q2<- hill_func(totutable, traits = EC_predicted, q = 2)

#func_q0_K0<- hill_func(totutable, traits = K0_predicted, q = 0)
#func_q1_K0<- hill_func(totutable, traits = K0_predicted, q = 1)
#func_q2_K0<- hill_func(totutable, traits = K0_predicted, q = 2)

#write.table(t(func_q0), file="../Data/func_q0.txt", sep="\t")
#write.table(t(func_q2_K0), file="../Data/func_q2_K0.txt", sep="\t")
```

#### 2.0.4. Plotting functional diversity

```

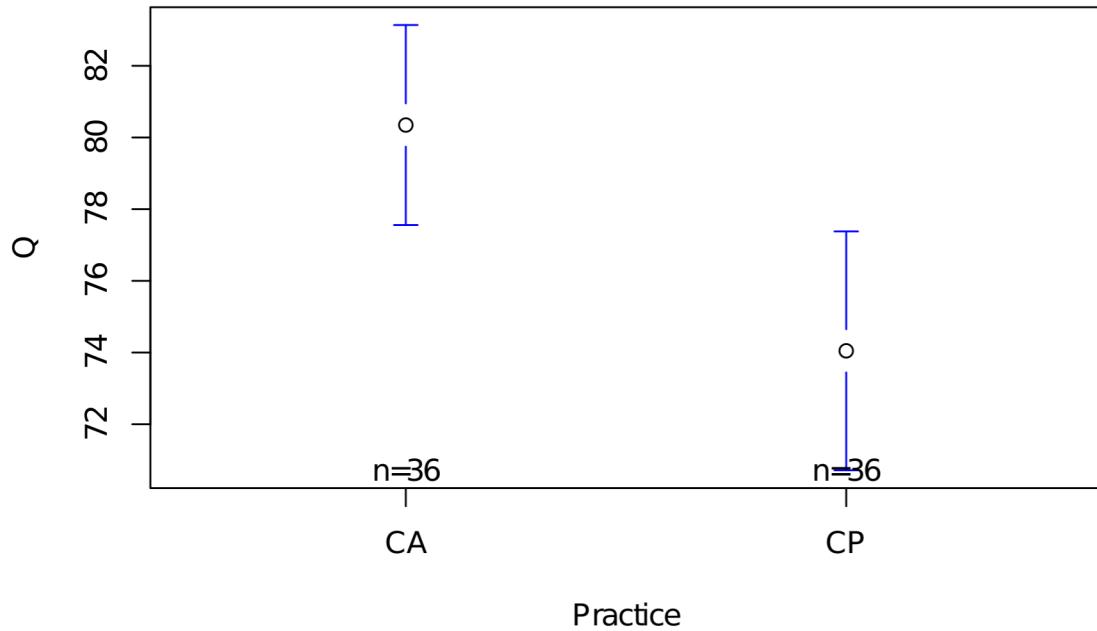
func_q0<- t(read.delim("../Data/func_q0.txt"))
func_q1<- t(read.delim("../Data/func_q1.txt"))
func_q2<- t(read.delim("../Data/func_q2.txt"))
Alpha.t_asv_table<- read.csv("../Data/Alpha-t_otu_table.txt.csv", check.names = F, row.n

funct_q0<-t(func_q0)
funct_q1<-t(func_q1)
funct_q2<-t(func_q2)

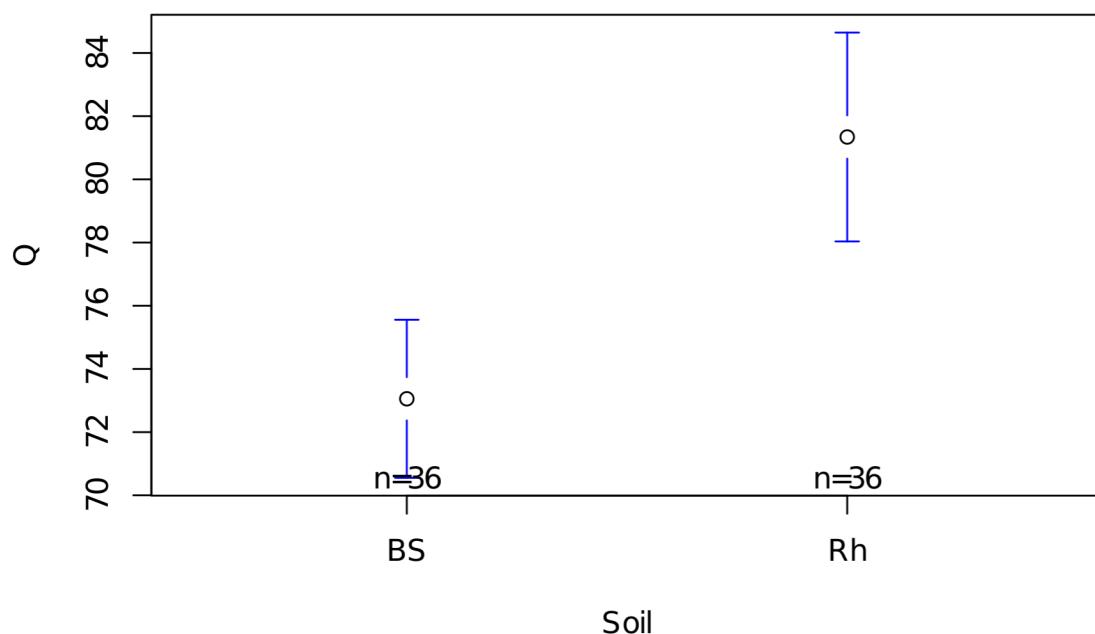
FD_q0<- merge(Alpha.t_asv_table, funct_q0, by=0)
FD_q1<- merge(Alpha.t_asv_table, funct_q1, by=0)
FD_q2<- merge(Alpha.t_asv_table, funct_q2, by=0)

plotmeans(Q~Practice, FD_q0, connect=F)

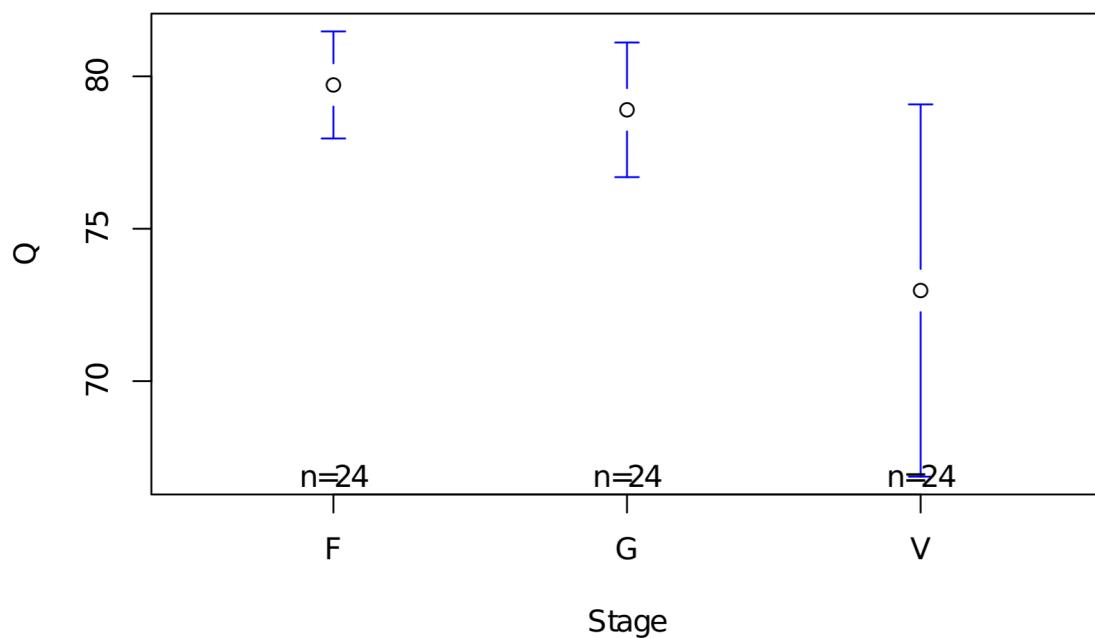
```



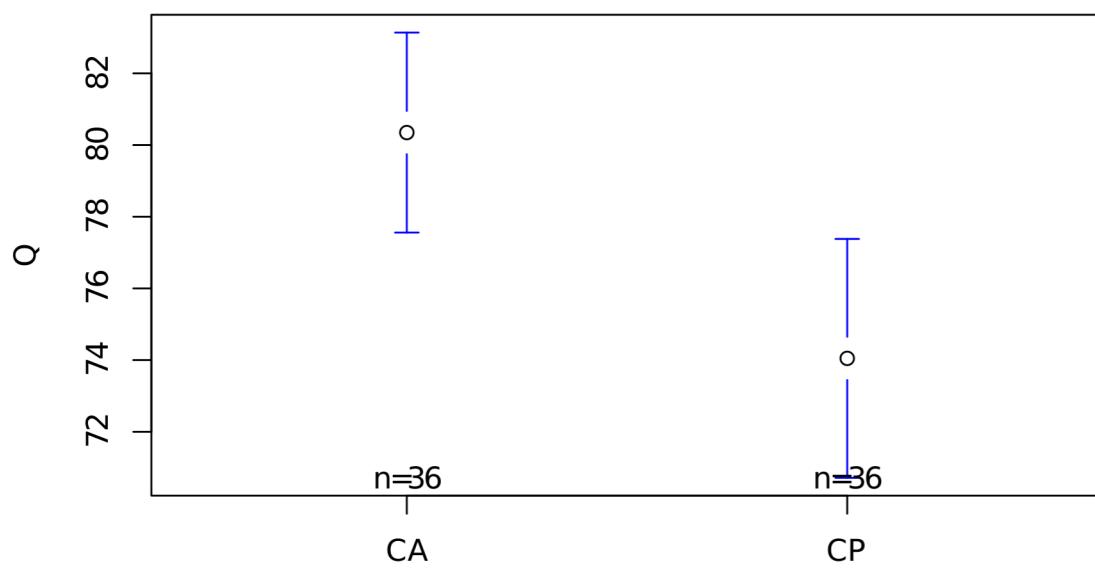
```
plotmeans(Q~Soil, FD_q0, connect=F)
```



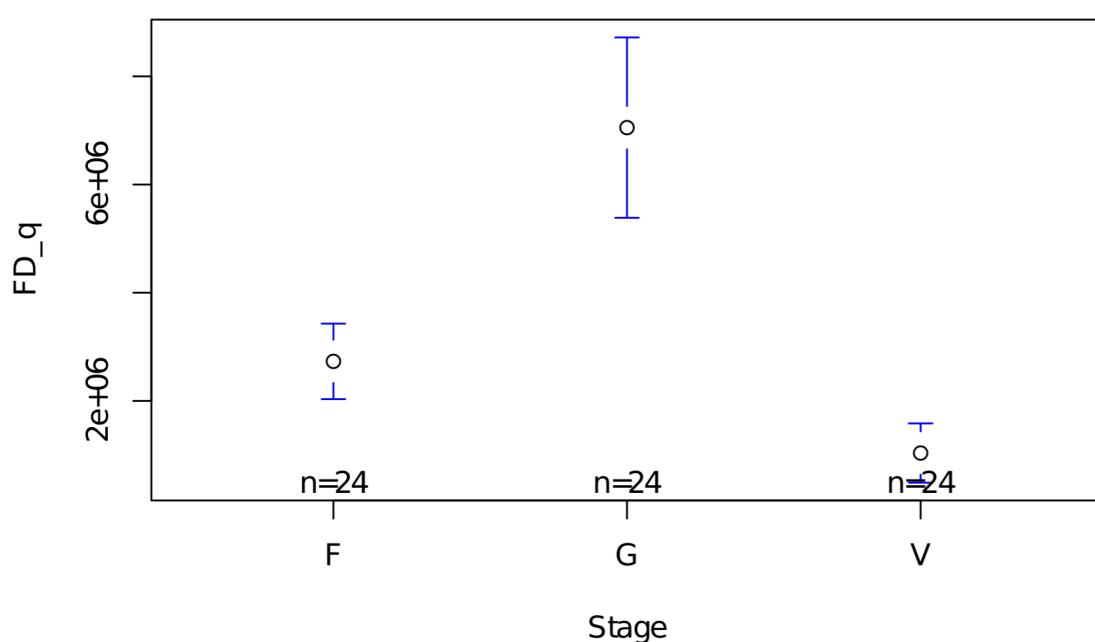
```
plotmeans(Q~Stage, FD_q1, connect=F)
```



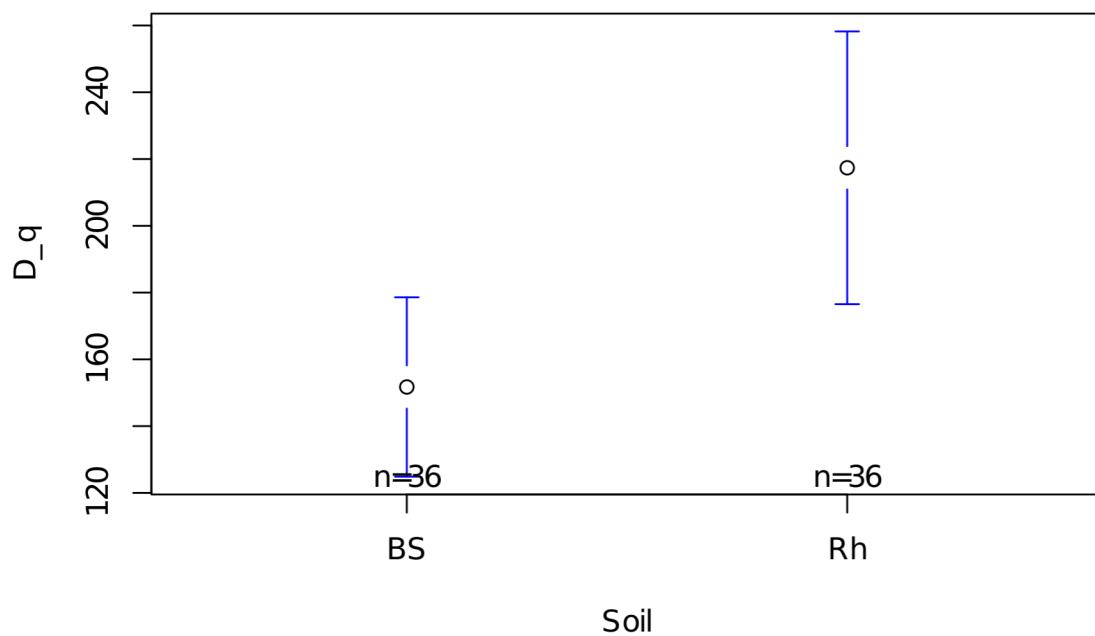
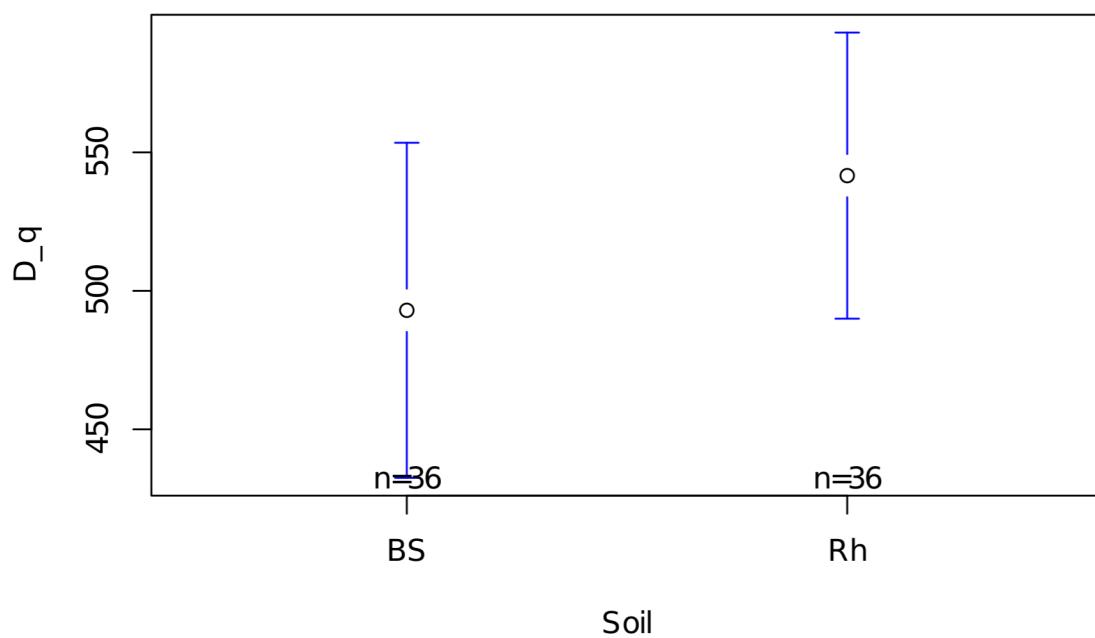
```
plotmeans(Q~Practice, FD_q0, connect=F)
```

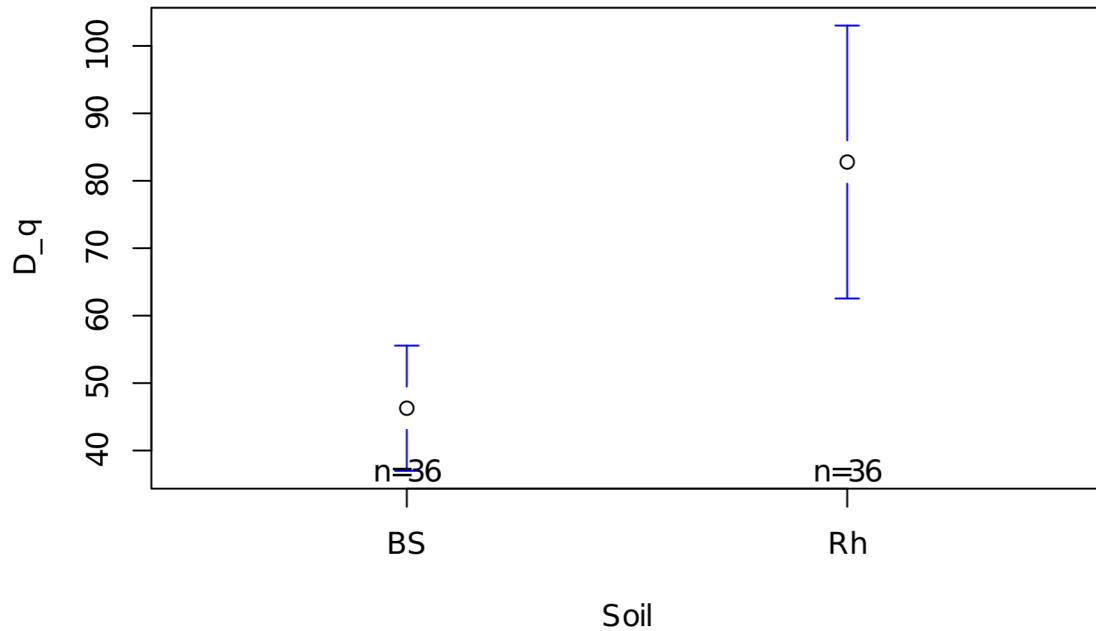


```
plotmeans(FD_q~Stage, FD_q1, connect=F)
```



```
plotmeans(D_q~Soil, FD_q0, connect=F)
```





#### 2.0.5. General linear model of functional diversity

```

func_MDq<- read.delim("../Data/func_MDq.txt", check.names = F, row.names = 1)

a<-lme(FD_q~Practice.Location*Stage, random=~1 |Plot, FD_q2)%>%PermTest
summary(a)

##          Length Class      Mode
## resultats  1     data.frame list
## B          1     -none-    numeric
## call       2     -none-    call

b<-lme(FD_q~Soil, random=~1 |Plot, FD_q2)
summary(b)

## Linear mixed-effects model fit by REML
##   Data: FD_q2
##   AIC      BIC      logLik
## 2100.708 2109.702 -1046.354
##
## Random effects:
##   Formula: ~1 | Plot
##             (Intercept) Residual
##   StdDev:    145905.9 704699.1
##
## Fixed effects: FD_q ~ Soil
##                  Value Std.Error DF t-value p-value
## (Intercept) 216353.2 138262.8 67 1.564797 0.1223
## SoilRh      618085.1 166099.2 67 3.721181 0.0004

```

```

## Correlation:
##          (Intr)
## SoilRh -0.601
##
## Standardized Within-Group Residuals:
##      Min       Q1       Med       Q3       Max
## -1.23073653 -0.63084391 -0.15614166  0.08395973  3.69128696
##
## Number of Observations: 72
## Number of Groups: 4

c<- lme(FD_q~Stage, random=~1 |Plot, FD_q2)%>%
PermTest

0<-ggplot(func_MDq, aes(x=Practice, y=MD_q0, fill=Soil))+
geom_boxplot()

I<-ggplot(FD_q1, aes(x=Practice, y=MD_q, fill=Soil))+
geom_boxplot()

II<- ggplot(FD_q2, aes(x=Practice, y=MD_q, fill=Soil))+
geom_boxplot()

0s<-ggplot(FD_q0, aes(x=Soil, y=MD_q, fill=Stage))+
geom_boxplot()

Is<-ggplot(FD_q1, aes(x=Soil, y=MD_q, fill=Stage))+
geom_boxplot()

IIs<- ggplot(FD_q2, aes(x=Soil, y=MD_q, fill=Stage))+
geom_boxplot()

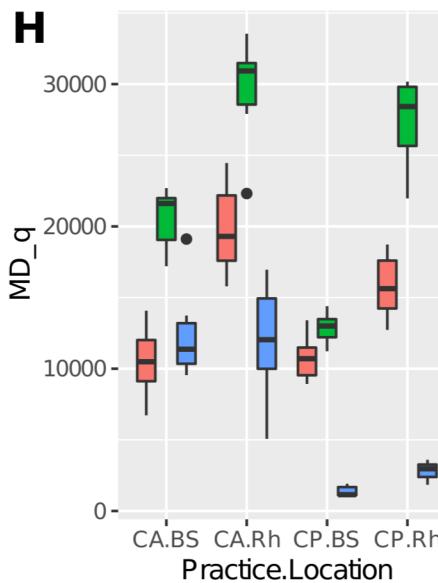
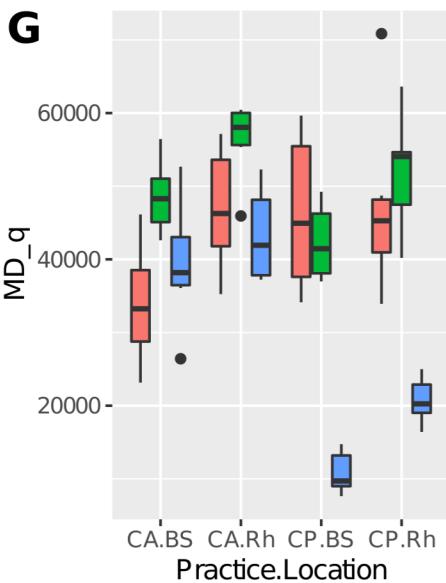
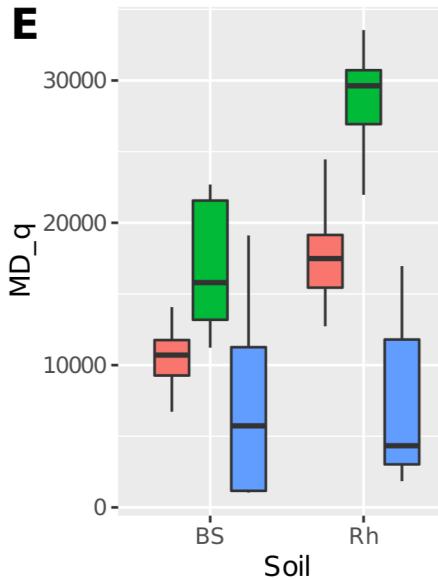
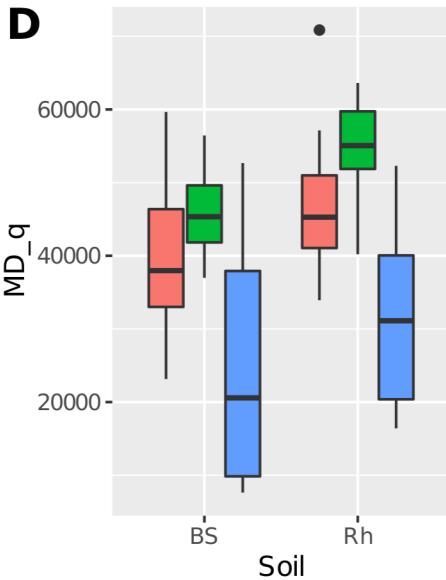
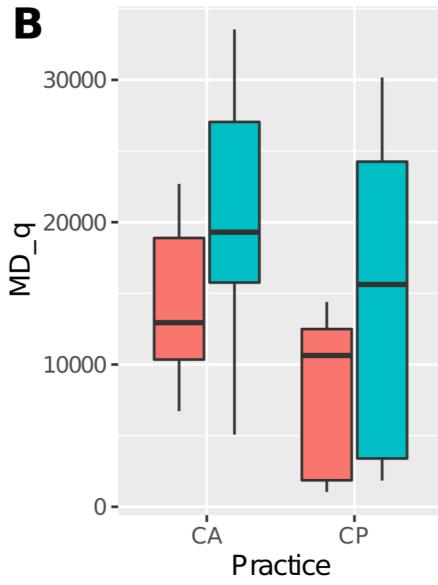
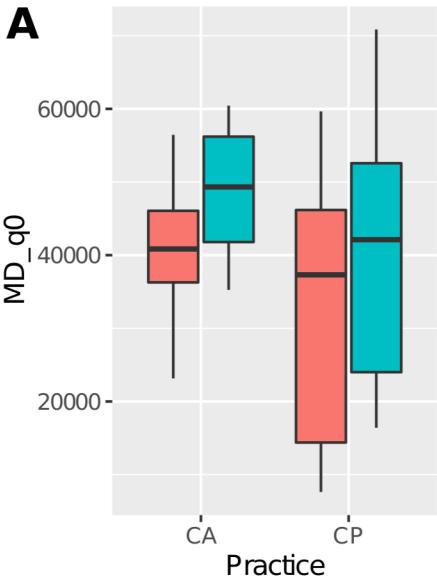
0ss<-ggplot(FD_q0, aes(x=Practice.Location, y=MD_q, fill=Stage))+
geom_boxplot()

Iss<-ggplot(FD_q1, aes(x=Practice.Location, y=MD_q, fill=Stage))+
geom_boxplot()

IIss<- ggplot(FD_q2, aes(x=Practice.Location, y=MD_q, fill=Stage))+
geom_boxplot()

r<-plot_grid(0, I, II, 0s, Is, IIs, 0ss, Iss, IIss,
             labels = "AUTO",
             label_size = 17, nrow=3, ncol = 3)
r

```



```
#pdf("FigX_FUNDIV-interactions.pdf", width=10, height=8)
#print(r)
#dev.off()
```

Plot S3

```
library(ggpubr)
library(cowplot)
func_MDq <- read.delim("../Data/func_MDq.txt", row.names=1)

F0.p <- ggboxplot(data = func_MDq, x = "Practice", y= "MD_q0",
                    fill= "Practice", palette = c("#212F3D", "#839192"),
                    width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Mean functional diversity")+
  theme_gray() +
  theme(text = element_text (size = 12))+
  theme(legend.position = "none")+
  theme(plot.title = element_text("q=0"))+
  theme(legend.position = "none",
        axis.ticks.x = element_blank())+
  stat_compare_means(method = "t.test")

F1.p <- ggboxplot(data = func_MDq, x = "Practice", y= "MD_q1",
                    fill = "Practice", palette = c("#212F3D", "#839192"),
                    width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Mean functional diversity")+
  theme_gray() +
  theme(text = element_text (size = 12))+
  theme(legend.position = "none")+
  theme(plot.title = element_text("q=0"))+
  theme(legend.position = "none",
        axis.ticks.x = element_blank())+
  stat_compare_means(method = "t.test")

F2.p <- ggboxplot(data = func_MDq, x = "Practice", y= "MD_q2",
                    fill = "Practice", palette = c("#212F3D", "#839192"),
                    width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Mean functional diversity")+
  theme_gray() +
  theme(text = element_text (size = 12))+
  theme(legend.position = "none")+
  theme(plot.title = element_text("q=0"))+
  theme(legend.position = "none",
        axis.ticks.x = element_blank())+
  stat_compare_means(method = "t.test")

F0.s <- ggboxplot(data = func_MDq, x = "Soil", y= "MD_q0",
                    fill = "Soil", palette = c("darkgoldenrod4", "#365238"),
                    width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Mean functional diversity")+
  theme_gray() +
```

```

theme(text = element_text (size = 12))+  

theme(legend.position = "none")+
theme(plot.title = element_text("q=0"))+
theme(legend.position = "none",
      axis.ticks.x = element_blank())+
stat_compare_means(method = "t.test")
F1.s <- ggboxplot(data = func_MDq, x = "Soil", y= "MD_q1",
                   fill = "Soil", palette = c("darkgoldenrod4", "#365238"),
                   width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Mean functional diversity")+
  theme_gray() +
  theme(text = element_text (size = 12))+  

theme(legend.position = "none")+
theme(plot.title = element_text("q=0"))+
theme(legend.position = "none",
      axis.ticks.x = element_blank())+
stat_compare_means(method = "t.test")
F2.s <- ggboxplot(data = func_MDq, x = "Soil", y= "MD_q2",
                   fill = "Soil", palette = c("darkgoldenrod4", "#365238"),
                   width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Mean functional diversity")+
  theme_gray() +
  theme(text = element_text (size = 12))+  

theme(legend.position = "none")+
theme(plot.title = element_text("q=0"))+
theme(legend.position = "none",
      axis.ticks.x = element_blank())+
stat_compare_means(method = "t.test")

div <- read.delim("../Data/Alpha-t_asv_table.txt", row.names=1)

D0.p <- ggboxplot(data = div, x = "Practice", y= "q0",
                   fill = "Practice", palette = c("#212F3D", "#839192"),
                   width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Effective number of OTUs")+
  theme_gray() +
  theme(text = element_text (size = 12))+  

theme(legend.position = "none")+
theme(plot.title = element_text("q=0"))+
theme(legend.position = "none",
      axis.ticks.x = element_blank())+
stat_compare_means(method = "t.test")
D1.p <- ggboxplot(data = div, x = "Practice", y= "q1",
                   fill = "Practice", palette = c("#212F3D", "#839192"),
                   width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Effective number of OTUs")+
  theme_gray() +
  theme(text = element_text (size = 12))+  

theme(legend.position = "none")+

```

```

theme(plot.title = element_text("q=0"))+
theme(legend.position = "none",
      axis.ticks.x = element_blank())+
stat_compare_means(method = "t.test")

D2.p <- ggboxplot(data = div, x = "Practice", y= "q2",
                    fill = "Practice", palette = c("#212F3D", "#839192"),
                    width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Effective number of OTUs")+
  theme_gray() +
  theme(text = element_text (size = 12))+
  theme(legend.position = "none")+
  theme(plot.title = element_text("q=0"))+
  theme(legend.position = "none",
        axis.ticks.x = element_blank())+
  stat_compare_means(method = "t.test")

D0.s <- ggboxplot(data = div, x = "Soil", y= "q0",
                    fill = "Soil", palette = c("darkgoldenrod4", "#365238"),
                    width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Effective number of OTUs")+
  theme_gray() +
  theme(text = element_text (size = 12))+
  theme(legend.position = "none")+
  theme(plot.title = element_text("q=0"))+
  theme(legend.position = "none",
        axis.ticks.x = element_blank())+
  stat_compare_means(method = "t.test")

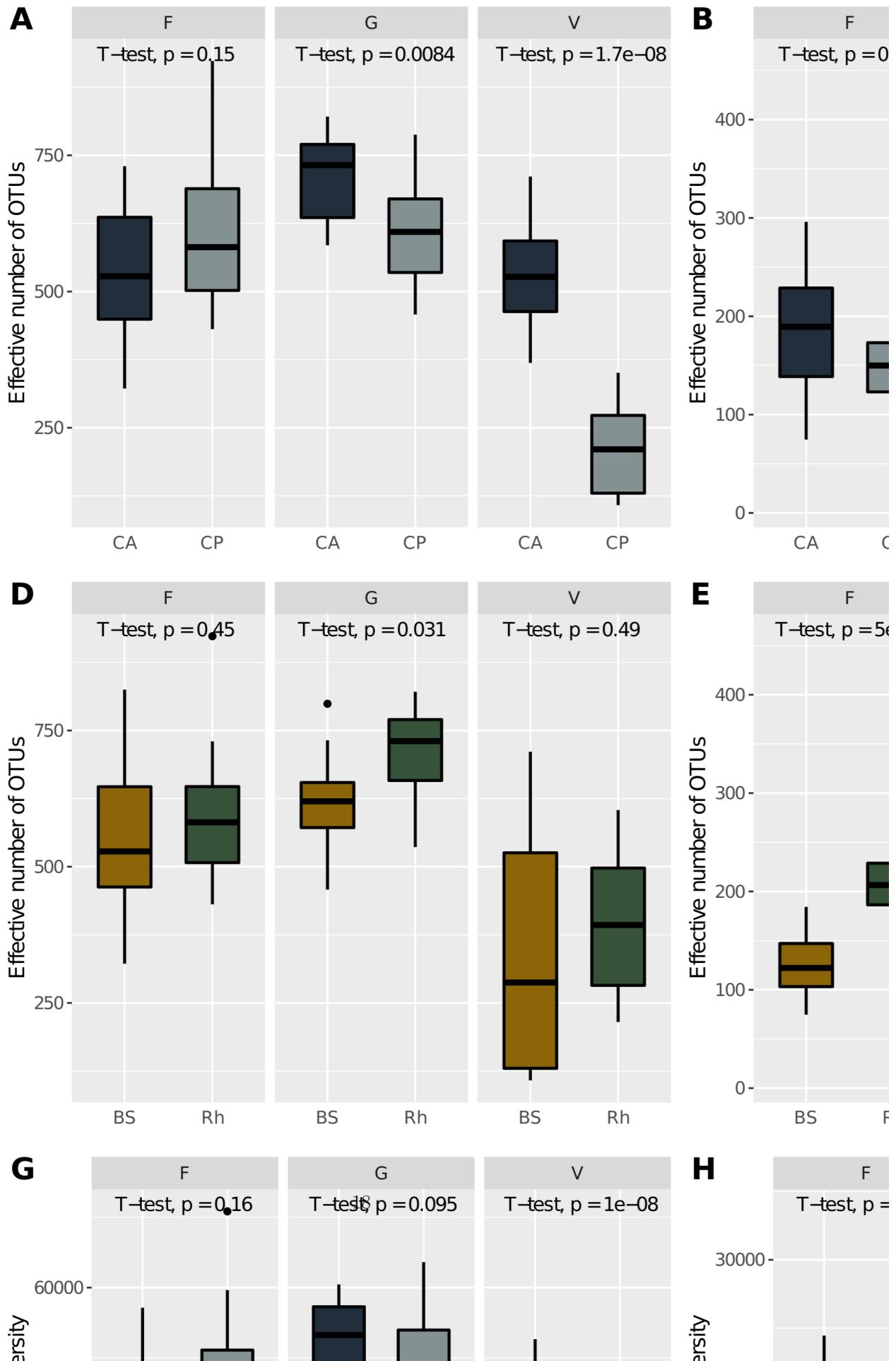
D1.s <- ggboxplot(data = div, x = "Soil", y= "q1",
                    fill = "Soil", palette = c("darkgoldenrod4", "#365238"),
                    width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Effective number of OTUs")+
  theme_gray() +
  theme(text = element_text (size = 12))+
  theme(legend.position = "none")+
  theme(plot.title = element_text("q=0"))+
  theme(legend.position = "none",
        axis.ticks.x = element_blank())+
  stat_compare_means(method = "t.test")

D2.s <- ggboxplot(data = div, x = "Soil", y= "q2",
                    fill = "Soil", palette = c("darkgoldenrod4", "#365238"),
                    width = 0.6, lwd=0.8, facet.by = "Stage") +
  labs(x = element_blank(), y = "Effective number of OTUs")+
  theme_gray() +
  theme(text = element_text (size = 12))+
  theme(legend.position = "none")+
  theme(plot.title = element_text("q=0"))+
  theme(legend.position = "none",
        axis.ticks.x = element_blank())

```

```
axis.ticks.x = element_blank())+
stat_compare_means(method = "t.test")

r<-plot_grid(D0.p, D1.p,D2.p,D0.s,D1.s,D2.s, F0.p, F1.p,F2.p,F0.s,F1.s,F2.s,
              labels = "AUTO",
              label_size = 17, nrow=4, ncol = 3)
```



```

#pdf("FigS3_Div_block_by_stage.pdf", width=16, height=18)
#print(r)
#dev.off()

```

---



---

### 3. III. BETA-DIVERSITY PLOT

#### 3.0.1. Loading libraries

```

library(cowplot)
library(tidyverse)
library(ggpubr)
library(circlize)
library(viridis)
library(RColorBrewer)
library(grid)
library(ggplot2)

```

#### 3.0.2. Loading and formatting files

```

beta<- read_tsv("../Data/beta_diversity.txt") %>% mutate(qs = case_when(
  q == 0 ~ "q=0 (species richness)",
  q == 1 ~ "q=1 (frequent species)",
  q == 2 ~ "q=2 (dominant species)") %>% rename("ASVs_turnover" = TurnoverComp)
head(beta)

## # A tibble: 6 x 12
##   q ID1  ID2    beta LocalOverlap RegionalOverlap Homogeneity ASVs_turnover
##   <dbl> <chr> <chr>    <dbl>        <dbl>        <dbl>        <dbl>
## 1     0 CAFB~ CAFR~    1.66      0.343      0.207      0.207
## 2     1 CAFB~ CAFR~    1.35      0.572      0.572      0.486
## 3     2 CAFB~ CAFR~    1.04      0.923      0.960      0.923
## 4     0 CAFB~ CAFR~    1.63      0.370      0.227      0.227
## 5     1 CAFB~ CAFR~    1.32      0.598      0.598      0.513
## 6     2 CAFB~ CAFR~    1.09      0.828      0.906      0.828
## # ... with 4 more variables: Comparison <chr>, PlotCompare <chr>, Type <chr>,
## #   qs <chr>

```

#### 3.0.3. Treatment plot

```

ann_text_treatment<-data.frame(
  Comparison=c("CA_BSvsRh", "CA_BSvsRh", "CA_BSvsRh"),
  "ASVs_turnover"=c(1,1,1),

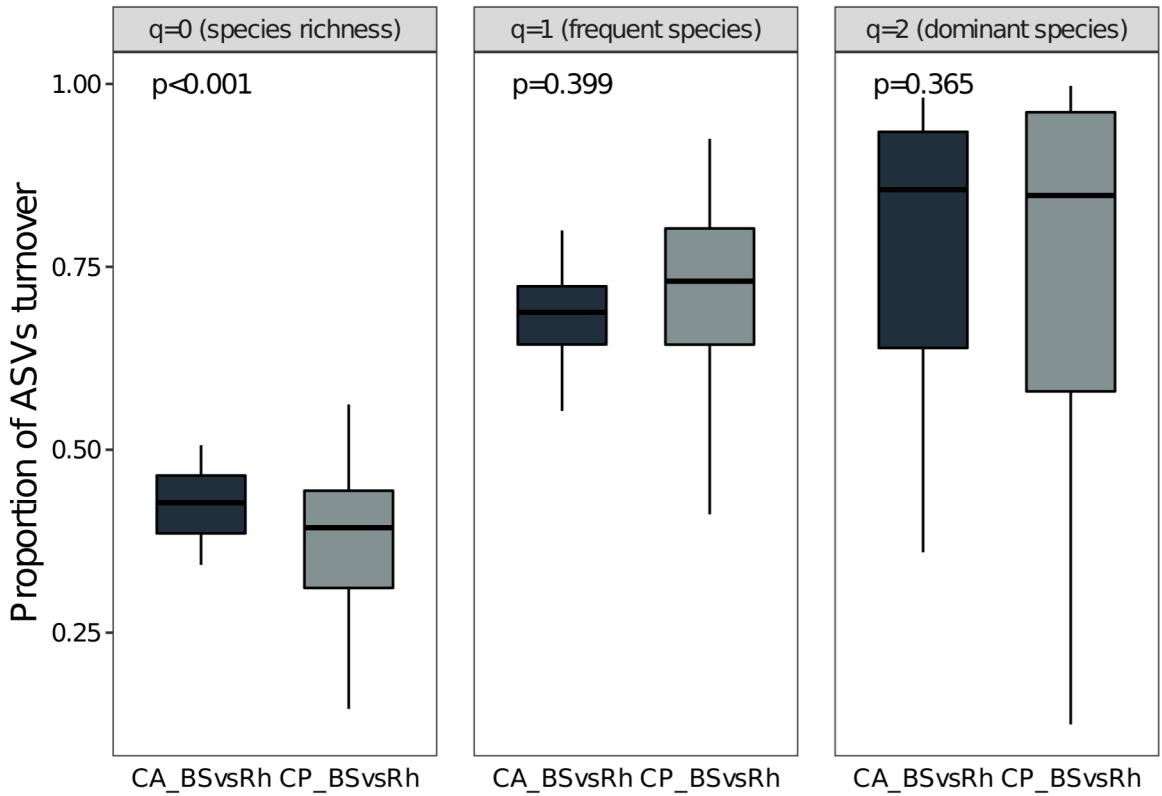
```

```

qs=c("q=0 (species richness)","q=1 (frequent species)","q=2 (dominant species)") ,
label=c("p<0.001","p=0.399", "p=0.365")) #tittles and positiong in y axis

beta_treatment<- beta %>%
  filter(str_detect(Comparison, '^CA|^CP'))%>% ggplot(
    aes(y=~ASVs_turnover~,x=Comparison, fill=Comparison)) +
  geom_boxplot(position=position_dodge(1), outlier.shape = NA, color="black",
               width=0.6)+theme_bw()+
  labs(y = "Proportion of ASVs turnover")+
  facet_grid(~qs, scales = "free")+
  theme(panel.spacing=unit(1,"lines"),
        # strip.background=element_rect(color="grey30", fill="gray90"),
        # panel.border=element_rect(color="black"),
        #strip.text.x = element_text(
        #  size = 12, color = "black", face = "bold"),
        strip.text.x = element_text(size = 10),
        axis.text = element_text(colour = "black", size = 10),
        axis.ticks.x=element_blank(),
        axis.title.x = element_blank(),
        legend.title = element_blank(),
        axis.title.y = element_text(size = 14),
        # legend.text = element_text(size=16),
        # axis.text.x = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        # legend.position = c(0.6,0.8),
        legend.direction = "vertical" ,
        legend.position = "none")+scale_fill_manual(values = c("#212F3D","#839192"))+
  geom_text(data = ann_text_treatment,label=ann_text_treatment$label)
beta_treatment

```



```
#pdf("fig_beta_treatment.pdf", width=6, height=3)
#print(beta_treatment)
#dev.off()
```

### 3.0.4. Soil Plot

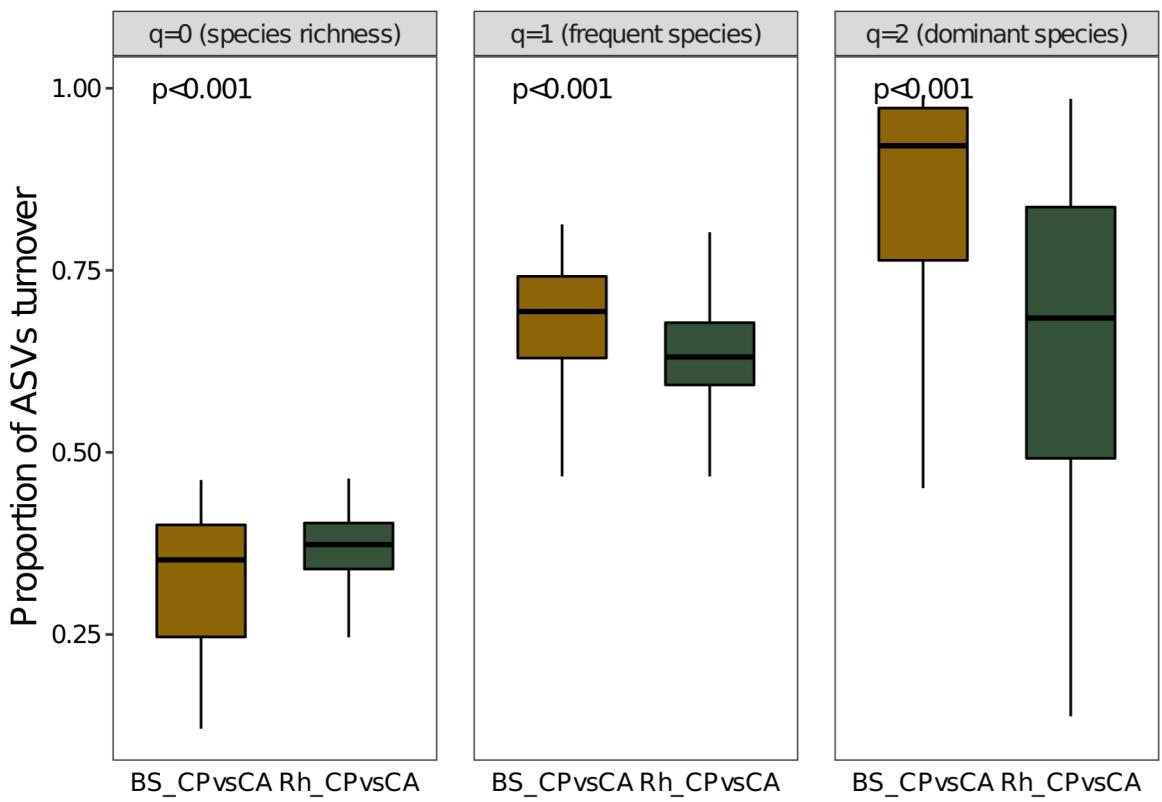
```
ann_text_soil<-data.frame(
  Comparison=c("BS_CPVsCA", "BS_CPVsCA", "BS_CPVsCA"),
  "ASVs_turnover"=c(1,1,1),
  qs=c("q=0 (species richness)", "q=1 (frequent species)", "q=2 (dominant species)"),
  label=c("p<0.001", "p<0.001", "p<0.001")) #titles and positiong in y axis

beta_soil<- beta %>%
  filter(!str_detect(Comparison, '^CA|^CP'))%>% ggplot(
    aes(y=`ASVs_turnover`, x=Comparison, fill=Comparison)) +
  geom_boxplot(position=position_dodge(1), outlier.shape = NA, color="black",
               width=0.6)+theme_bw()+
  labs(y = "Proportion of ASVs turnover")+
  facet_grid(~qs, scales = "free")+
  theme(panel.spacing=unit(1,"lines"),
        # strip.background=element_rect(color="grey30", fill="gray90"),
        # panel.border=element_rect(color="black"),
        #strip.text.x = element_text(
```

```

# size = 12, color = "black", face = "bold"),
strip.text.x = element_text(size = 10),
axis.text = element_text(colour = "black", size = 10),
axis.ticks.x=element_blank(),
axis.title.x = element_blank(),
legend.title = element_blank(),
axis.title.y = element_text(size = 14),
# legend.text = element_text(size=16),
# axis.text.x = element_blank(),
panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
# legend.position = c(0.6,0.8),
legend.direction = "vertical" ,
legend.position = "none")+scale_fill_manual(values = c("darkgoldenrod4", "#36523
geom_text(data = ann_text_soil,label=ann_text_soil$label)
beta_soil

```



```

#pdf("fig_beta_soil.pdf", width=6, height=3)
#print(beta_soil)
#dev.off()

```

## 4. IV. SOIL FIGURE

### 4.0.1. Loading libraries

```
library(cowplot)
library(tidyverse)
library(ggpubr)
library(ComplexHeatmap)
library(circlize)
library(viridis)
library(RColorBrewer)
library(grid)
library(CoDaSeq)
library(ggplot2)
require(compositions) # exploratory data analysis of compositional data
require(zCompositions) # used for 0 substitution
require(ALDEx2) # used for per-OTU comparisons
library(CoDaSeq)
library(ggrepel)
```

### 4.0.2. Loadings files and Barplot Text annotations

```
alpha<- read.table("../Data/alpha_diversity") %>% gather(
  q0:q4, key = "q", value = "value") %>% filter(
  q %in% c("q0", "q1", "q2")) %>% mutate(qs= case_when(
  str_detect(q, "q0") ~ "q=0 (species richness)",
  str_detect(q, "q1") ~ "q=1 (frequent species)",
  str_detect(q, "q2") ~ "q=2 (dominant species)"))
head(alpha)

##   Practice Soil Practice.Location Stage Age Plant Plot ExpUnit  q value
## 1       CA   BS             CA.BS     F   1     1   18 CA.BS.F q0    426
## 2       CA   BS             CA.BS     F   1     2   18 CA.BS.F q0    646
## 3       CA   BS             CA.BS     F   1     3   18 CA.BS.F q0    510
## 4       CA   BS             CA.BS     F   1     1   59 CA.BS.F q0    546
## 5       CA   BS             CA.BS     F   1     2   59 CA.BS.F q0    391
## 6       CA   BS             CA.BS     F   1     3   59 CA.BS.F q0    322
##                               qs
## 1 q=0 (species richness)
## 2 q=0 (species richness)
## 3 q=0 (species richness)
## 4 q=0 (species richness)
## 5 q=0 (species richness)
## 6 q=0 (species richness)

func<- read.table("../Data/func_MDq.txt") %>% gather(
  MD_q0:MD_q2, key = "q", value = "value") %>% mutate(fs= case_when(
  str_detect(q, "q0") ~ "q=0 (species richness)",
```

```

str_detect(q, "q1") ~ "q=1 (frequent species)",
str_detect(q, "q2") ~ "q=2 (dominant species"))
head(func)

##   Practice Soil Practice.Location Stage Age Plant Plot ExpUnit     q    value
## 1       CA  BS             CA.BS      F  1     1  18 CA.BS.F MD_q0 29629.84
## 2       CA  BS             CA.BS      F  1     2  18 CA.BS.F MD_q0 46138.20
## 3       CA  BS             CA.BS      F  1     3  18 CA.BS.F MD_q0 36859.81
## 4       CA  BS             CA.BS      F  1     1  59 CA.BS.F MD_q0 39086.65
## 5       CA  BS             CA.BS      F  1     2  59 CA.BS.F MD_q0 28482.32
## 6       CA  BS             CA.BS      F  1     3  59 CA.BS.F MD_q0 23152.35
##                               fs
## 1 q=0 (species richness)
## 2 q=0 (species richness)
## 3 q=0 (species richness)
## 4 q=0 (species richness)
## 5 q=0 (species richness)
## 6 q=0 (species richness)

#df with the p values to show in the figures
ann_text<-data.frame(Soil=c("BS", "BS", "BS"), value=c(800,350,150),
qs=c("q=0 (species richness)", "q=1 (frequent species)",
"q=2 (dominant species)", label=c("p=0.157", "p=0.001", "p<0.0001"))
#tittles and position in y axis

ann_text_f<-data.frame(Soil=c("BS", "BS", "BS"), value=c(60000,30000,10000),
fs=c("q=0 (species richness)", "q=1 (frequent species)",
"q=2 (dominant species)", label=c("p=0.075", "p<0.0001", "p<0.0001"))
#tittles and position in y axis

```

#### 4.0.3. Barplots alpha and functional diversity

```

#Alpha diversity barplot soil
boxplot_soil<-alpha %>%
  ggbarplot(x="qs", y="value", fill = "Soil", add = "mean_se",
            position = position_dodge())+
  theme_bw()+
  labs(y = "Effective number of ASVs")+
  facet_wrap(~qs, scales = "free", dir = "v")+
  theme(panel.spacing=unit(1,"lines"),
        strip.text.x = element_text(size = 10),
        axis.text = element_text(colour = "black", size = 10),
        axis.ticks.x=element_blank(),
        legend.title = element_text(size = 14),
        legend.text = element_text(size=14),
        axis.text.x = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),

```

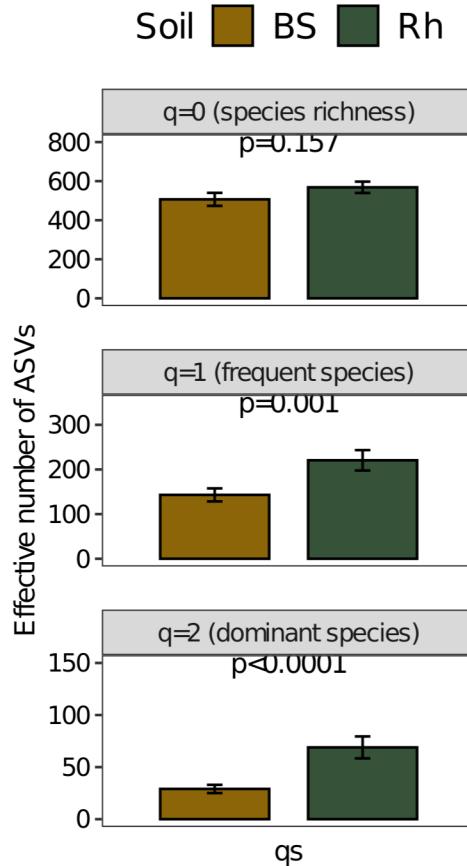
```

    legend.direction = "horizontal" ,
    legend.position = "top")+scale_fill_manual(
      values = c("darkgoldenrod4", "#365238"))+ labs(fill = "Soil")

boxplot_soil<-boxplot_soil +  geom_text(data = ann_text,label=ann_text$label)

boxplot_soil

```



```

#Functional diversity barplot soil
boxplot_soil_f<-func %>%
  ggbarplot(x="fs", y="value", fill = "Soil", add = "mean_se",
             position = position_dodge())+
  theme_bw()+
  labs(y = "Mean functional diversity")+
  facet_wrap(~fs, scales = "free", dir = "v")+
  theme(panel.spacing=unit(1,"lines"),
        strip.text.x = element_text(size = 10),
        axis.text = element_text(colour = "black", size = 10),
        axis.ticks.x=element_blank(),
        legend.title = element_text(size = 14),
        legend.text = element_text(size=14),

```

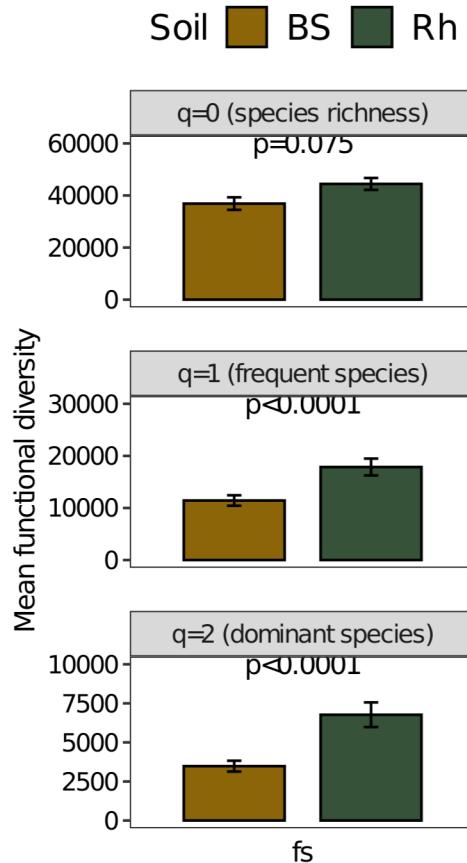
```

axis.text.x = element_blank(),
panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
legend.direction = "horizontal" ,
legend.position = "top")+scale_fill_manual(values = c(
  "darkgoldenrod4", "#365238"))+ labs(fill = "Soil")

boxplot_soil_f<-boxplot_soil_f + geom_text(data = ann_text_f,label=ann_text_f$label)

boxplot_soil_f

```



```

#ggsave('./fig_alpha_soil.png',
#       width = 2.5, height = 5, dpi = 300, plot = boxplot_soil)

#ggsave('./fig_func_soil.png',
#       width = 2.5, height = 5, dpi = 300, plot = boxplot_soil_f)

#pdf("fig_alpha_soil.pdf", width=2.5, height=5)
#print(boxplot_soil)
#dev.off()
#pdf("fig_func_soil.pdf", width=2.5, height=5)
#print(boxplot_soil_f)

```

```
#dev.off()
```

#### 4.0.4. Aldex results heatmap from Soil

```
#file to heatmap
aldex_all_dif<- read_tsv("../Data/aldex_soil.tsv")

my_fun <- function(x) {
  x %>% separate(
    "Taxon", c("k", "phylum", "c", "o", "f", "g"),
    sep = "\\|;", remove = F) %>% dplyr::select(
      Taxon, p.value, effect, diff.btw, rab.win.0, rab.win.1, phylum,
      "FeatureID"="Feature.ID")%>%
  drop_na(.)%>%
  rownames_to_column(var="rows")%>%
  mutate_all(funs(str_replace(., "k_Bacteria;", "")))%>%
  mutate_all(funs(str_replace(., "; c__; o__; f__; g__; s__", "")))%>%
  mutate_all(funs(str_replace(., "; o__; f__; g__; s__", "")))%>%
  mutate_all(funs(str_replace(., "; f__; g__; s__", "")))%>%
  mutate_all(funs(str_replace(., "; g__; s__", "")))%>%
  mutate_all(funs(str_replace(., "; s__", "")))%>%mutate(
    tax= str_extract(Taxon, "[^_]+$")) %>%mutate(
      taxo = paste(rows,"_",tax))%>% mutate_at(
        c(3:7), as.numeric) %>%
  mutate_at(c(3), funs(p.Value = case_when(
    . <= 0.001 ~ "<0.001",
    . > 0.001 & . <= 0.01 ~ "<0.01",
    . > 0.01 & . <= 0.05 ~ "<0.05")))%>%
  arrange(diff.btw)%>%column_to_rownames(
    var = "taxo")%>% mutate_at(c(8),funns(str_replace(., "p__", "")))
}

#We are going to multiplicate for -1 in order to change
#the direction of the figure (e.g, bulk soil first and then rhizosphere)

annotation_heatmap <- my_fun(aldex_all_dif) %>% mutate(
  diff.btw2 = diff.btw*-1, effect2 = effect*-1 ) %>% arrange(diff.btw2) %>% mutate(
  taxo= paste(rows,tax, sep = "_"))
data_heatmap<- annotation_heatmap%>%dplyr::select(rab.win.1, rab.win.0) %>% rename(
  rab.win.Rh=rab.win.0 , rab.win.Bs=rab.win.1)

color_heatmap= colorRamp2(seq(min(data_heatmap), max(data_heatmap), length = 5), c(
  "#0000FF", "#5499C7", "#DAE7E4", "red", "#FF0000"))

#Annotation Phylum
cols_ann <- list('phylum' = c(
  " Acidobacteria" = 'red2',
  " Actinobacteria" = 'royalblue',
```

```

    " Bacteroidetes"="yellow",
    " Chloroflexi" ="pink",
    " Firmicutes"= "green",
    " Gemmatimonadetes" = "black",
    " Nitrospirae" ="purple",
    " Planctomycetes" ="dark green",
    " Proteobacteria" ="gray",
    " Verrucomicrobia" ="brown"))
colAnn <- HeatmapAnnotation(phylum = annotation_heatmap$phylum,
                             which = 'row',
                             col = cols_ann,
                             show_legend = T)

#Annotation pvalue

cols_pvalue <- list('p-value' = c("<0.001" = '#AB0000',
                                    "<0.01" = '#FF0000',
                                    "<0.05"="#FFB6B6"))

annP2 = HeatmapAnnotation("p-value" = annotation_heatmap$p.Value,
                         which = "row", col = cols_pvalue,
                         show_legend = T)##, gp = gpar(col = "white"))

#Annotation effect size
effect_col_fun =colorRamp2(c(-1.5, 0, 1.5), c("lightsalmon4", "white", "lightseagreen"))

annEffect = HeatmapAnnotation("effect-size" = annotation_heatmap$effect2,
                             which = "row", col = list("effect-size" = effect_col_fun),
                             show_legend = T,
                             gp = gpar(col = "white"))

# gap = unit(10, "cm")

#Annotation barplot
bardif= rowAnnotation("difference \n between groups" = anno_barplot(
  annotation_heatmap$diff.btw2, width = unit(4, "cm")))

#Annotation taxonomy

labels = c("RB41", "iiii1-15", "Bacillus", "Halomonas", rep("", 7),"Burkholderiaceae",
          "Comamonadaceae","Comamonadaceae", "Xanthomonadales", "Oxalobacteraceae",
          "Rhodospirillaceae", "Solibacterales","Comamonadaceae", "Rhizobiales","Rhizot
          "Comamonadaceae" , "Oxalobacteraceae")

#Heat map
heatmap_aldex_soil<-ComplexHeatmap:: Heatmap(data_heatmap, col = color_heatmap,
row_dend_reorder = F, width = ncol(data_heatmap)*unit(1, "cm"),

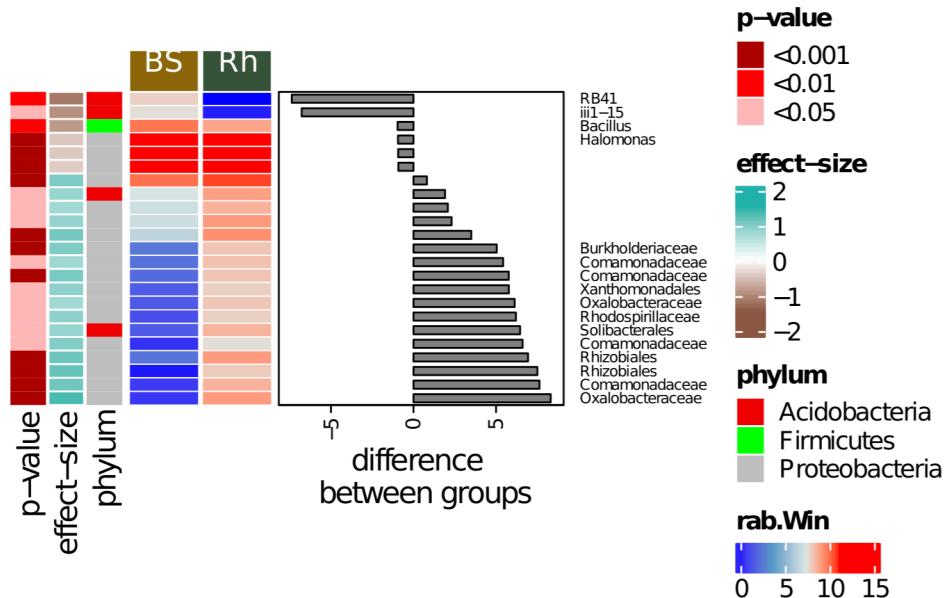
```

```

height = ncol(data_heatmap)*unit(2.2, "cm"),
left_annotation = c(annP2, annEffect, colAnn),
cluster_column_slices = F,
heatmap_legend_param = list(direction = "horizontal" ),
right_annotation = c(bardif),
column_split = c("BS", "Rh"),
cluster_rows = F,
cluster_columns = F,
column_km = 1,
column_title_gp = gpar(fill = c("darkgoldenrod4", "#365238" ), col="white"),
border = F, column_gap = unit(0.5, "mm"), row_dend_side = "left",
row_names_side = "right", show_row_names = F,
rect_gp = gpar(col = "white", lwd = 0.2),
row_names_gp = gpar(fontface ="italic", fontsize=10),
show_column_names = F, name = "rab.Win")+
rowAnnotation(labels = anno_text(labels, which = "row",
gpar(col = "black", fontsize = 6)),
width = unit(2, "cm"))

```

heatmap\_aldex\_soil



```

#pdf("fig_aldex_soil.pdf", width=6, height=5)
#print(heatmap_aldex_soil)
#dev.off()

```

#### 4.0.5. PCA plot

```
#loading files and formatting
```

```

d.pro.0<- read_tsv("../Data/otutable.tsv")%>% column_to_rownames(var = "#OTU ID")
meta<-read_tsv("../Data/metadata.tsv")

meta$Soil<- factor(meta$Soil_sample,
                     levels = c( "bulksoil", "Rhizosphere"),
                     labels = c("BS", "Rh"))

tax<-read_tsv("../Data/taxonomy.tsv") %>% dplyr::select(-Confidence)%>%
  mutate_all(funs(str_replace(., "k_Bacteria;", "")))%>%
  mutate_all(funs(str_replace(., "p_ ", "")))%>%
  mutate_all(funs(str_replace(., "c_ ", "")))%>%
  mutate_all(funs(str_replace(., "o_ ", "")))%>%
  mutate_all(funs(str_replace(., "f_ ", "")))%>%
  mutate_all(funs(str_replace(., "g_ ", "")))%>%
  mutate_all(funs(str_replace(., "s_ ", "")))%>%
  mutate_all(funs(str_replace(., "; ; ;", " ")))%>%
  mutate_all(funs(str_replace(., "; ; ", " "))) %>% rename(
  "FeatureID"=~#OTU ID` , Taxon= taxonomy)

tax2<- read_tsv("../Data/taxonomy.tsv") %>% dplyr::select(
-Confidence) %>% rename(
"FeatureID"=~#OTU ID` , Taxon= taxonomy)

#transforming data
d.pro <- cmultRepl(t(d.pro.0), method="CZM", output="p-counts")
d.clr.abund.codaseq<-codaSeq.clr(x = d.pro,samples.by.row = F)

#run pca
pcx.abund <- prcomp(d.clr.abund.codaseq)

#labels to pca axis

PC1 <- paste("PC1", round(sum(pcx.abund$sdev[1] ^ 2) / mvar(d.clr.abund.codaseq) * 100,
PC2 <- paste("P21", round(sum(pcx.abund$sdev[2] ^ 2) / mvar(d.clr.abund.codaseq) * 100,

#let's choose som of the significant groups from aldex analysis

vars_chosen<- c("14_RB41",
                "3_iii1-15",
                "16_Oxalobacteraceae" ,
                "11_Comamonadaceae",
                "13_Rhizobiales",
                "21_Solibacterales",
                "20_Rhodospirillaceae")

#these ones were chosen from before (some aldex significant groups)

vars_to_choose<- annotation_heatmap %>% filter(taxo %in% vars_chosen)

```

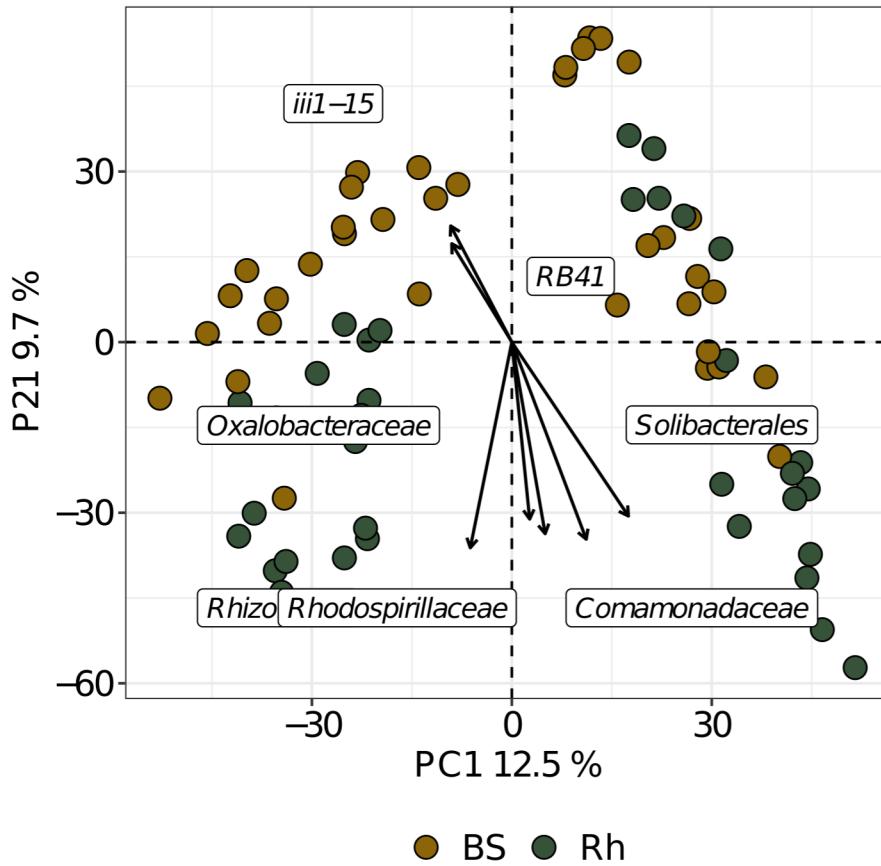
```

vars_choosing<- data.frame(pcx.abund$rotation) %>% rownames_to_column(var = "FeatureID")
  mutate(a=sqrt(PC1^2+PC2^2)) %>%
  mutate(PC1=PC1*500, PC2=PC2*500) %>% left_join(tax2)%>% dplyr::select(
    Taxon, PC1, PC2, FeatureID)%>%right_join(vars_to_choose, by = "FeatureID")

#create the base plot with only the arrows
pca_soil_arrows<- ggplot() +
  theme_bw() +
  xlab(PC1) +
  ylab(PC2) +
  theme(axis.text = element_text(colour = "black", size = 14), #setting theme
        axis.title = element_text(colour = "black", size = 14),
        legend.text = element_text(size = 14),
        legend.title = element_blank(),
        legend.position = "bottom") +
  geom_point(                                     #individuals
             data=data.frame(pcx.abund$x) %>%   rownames_to_column(var = "SampleID")%>%
               left_join(meta, by = "SampleID"),
             aes(x=PC1, y=PC2, fill=Soil),
             shape=21, size=4) +
  geom_vline(xintercept = 0, linetype = 2) +    #lines-cross
  geom_hline(yintercept = 0, linetype = 2) +
  scale_fill_manual(values = c("darkgoldenrod4", "#365238"))+
  ggrepel::geom_label_repel(data = vars_choosing, aes(x=PC1, y=PC2, label= tax),
                            segment.colour = NA, box.padding = 2, fontface="italic")+
  geom_segment(data = vars_choosing, #arrows and names
               aes(x=0, y=0, xend=PC1, yend=PC2),
               arrow=arrow(length=unit(0.15,"cm")),
               size= 0.6)

pca_soil_arrows

```



```
#pdf("fig_pca_soil.pdf", width=5, height=5)
#print(pca_soil_arrows)
#dev.off()
```

## 5. V. TREATMENT FIGURE

### 5.0.1. Loading libraries

```
library(cowplot)
library(tidyverse)
library(ggpubr)
library(ComplexHeatmap)
library(circlize)
library(viridis)
library(RColorBrewer)
library(grid)
library(CoDaSeq)
library(ggplot2)
require(compositions) # exploratory data analysis of compositional data
require(zCompositions) # used for 0 substitution
```

```

require(ALDEx2) # used for per-OTU comparisons
library(CoDaSeq)
library(ggrepel)

```

### 5.0.2. Loadings files and Barplot Text annotations

```

alpha<- read.table("../Data/alpha_diversity") %>% gather(
  q0:q4, key = "q", value = "value") %>% filter(
  q %in% c("q0", "q1", "q2"))%>%mutate(qs= case_when(
  str_detect(q, "q0") ~ "q=0 (species richness)",
  str_detect(q, "q1") ~ "q=1 (frequent species)",
  str_detect(q, "q2") ~ "q=2 (dominant species)"))
head(alpha)

##   Practice Soil Practice.Location Stage Age Plant Plot ExpUnit   q value
## 1       CA   BS             CA.BS     F   1    1  18 CA.BS.F q0   426
## 2       CA   BS             CA.BS     F   1    2  18 CA.BS.F q0   646
## 3       CA   BS             CA.BS     F   1    3  18 CA.BS.F q0   510
## 4       CA   BS             CA.BS     F   1    1  59 CA.BS.F q0   546
## 5       CA   BS             CA.BS     F   1    2  59 CA.BS.F q0   391
## 6       CA   BS             CA.BS     F   1    3  59 CA.BS.F q0   322
##                               qs
## 1 q=0 (species richness)
## 2 q=0 (species richness)
## 3 q=0 (species richness)
## 4 q=0 (species richness)
## 5 q=0 (species richness)
## 6 q=0 (species richness)

func<- read.table("../Data/func_MDq.txt") %>% gather(
  MD_q0:MD_q2, key = "q", value = "value")%>%mutate(fs= case_when(
  str_detect(q, "q0") ~ "q=0 (species richness)",
  str_detect(q, "q1") ~ "q=1 (frequent species)",
  str_detect(q, "q2") ~ "q=2 (dominant species)"))
head(func)

##   Practice Soil Practice.Location Stage Age Plant Plot ExpUnit   q value
## 1       CA   BS             CA.BS     F   1    1  18 CA.BS.F MD_q0 29629.84
## 2       CA   BS             CA.BS     F   1    2  18 CA.BS.F MD_q0 46138.20
## 3       CA   BS             CA.BS     F   1    3  18 CA.BS.F MD_q0 36859.81
## 4       CA   BS             CA.BS     F   1    1  59 CA.BS.F MD_q0 39086.65
## 5       CA   BS             CA.BS     F   1    2  59 CA.BS.F MD_q0 28482.32
## 6       CA   BS             CA.BS     F   1    3  59 CA.BS.F MD_q0 23152.35
##                               fs
## 1 q=0 (species richness)
## 2 q=0 (species richness)
## 3 q=0 (species richness)
## 4 q=0 (species richness)

```

```

## 5 q=0 (species richness)
## 6 q=0 (species richness)

#df with the p values to show in the figures
ann_text<-data.frame(Practice=c("CA", "CA", "CA"),value=c(800,350,150),
qs=c("q=0 (species richness)","q=1 (frequent species)",
"q=2 (dominant species)"),label=c("p=0.009","p=0.002", "p=0.011"))
#tittles and positiong in y axis

ann_text_f<-data.frame(Practice=c("BS", "BS", "BS"),value=c(60000,30000,10000),
fs=c("q=0 (species richness)","q=1 (frequent species)",
"q=2 (dominant species)"),label=c("p=0.059","p=0.015", "p=0.026"))
#tittles and positiong in y axis

```

### 5.0.3. Barplots alpha and functional diversity

```

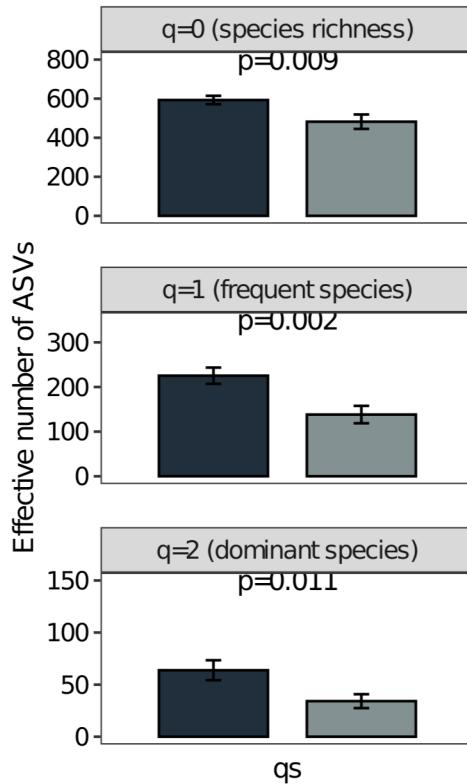
#Alpha diversity barplot soil
boxplot_practice<-alpha %>%
  ggbarplot(x="qs", y="value", fill = "Practice", add = "mean_se",
             position = position_dodge())+
  theme_bw()+
  labs(y = "Effective number of ASVs")+
  facet_wrap(~qs, scales = "free", dir = "v")+
  theme(panel.spacing=unit(1,"lines"),
        strip.text.x = element_text(size = 10),
        axis.text = element_text(colour = "black", size = 10),
        axis.ticks.x=element_blank(),
        legend.title = element_text(size = 14),
        legend.text = element_text(size=14),
        axis.text.x = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        legend.direction = "horizontal" ,
        legend.position = "top")+scale_fill_manual(values = c("#212F3D", "#839192"))+ la

boxplot_practice<-boxplot_practice +  geom_text(data = ann_text,label=ann_text$label)

boxplot_practice

```

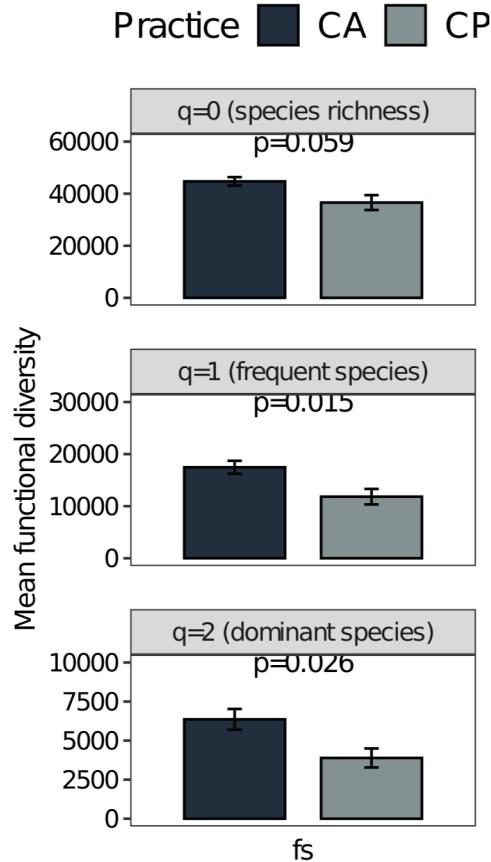
## Practice CA CP



```
#boxplot
boxplot_practice_f<-func %>%
  ggbarplot(x="fs", y="value", fill = "Practice", add = "mean_se",
             position = position_dodge())+
  theme_bw()+
  labs(y = "Mean functional diversity")+
  facet_wrap(~fs, scales = "free", dir = "v")+
  theme(panel.spacing=unit(1,"lines"),
        strip.text.x = element_text(size = 10),
        axis.text = element_text(colour = "black", size = 10),
        axis.ticks.x=element_blank(),
        legend.title = element_text(size = 14),
        legend.text = element_text(size=14),
        axis.text.x = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        legend.direction = "horizontal" ,
        legend.position = "top")+scale_fill_manual(values = c("#212F3D", "#839192"))+ la

boxplot_practice_f<-boxplot_practice_f + geom_text(data = ann_text_f,label=ann_text_f$la

boxplot_practice_f
```



```
#pdf("fig_alpha_practice.pdf", width=2.7, height=5)
#print(boxplot_practice)
#dev.off()
#pdf("fig_func_practice.pdf", width=2.7, height=5)
#print(boxplot_practice_f)
#dev.off()
```

#### 5.0.4. Aldex results heatmap from Soil

```
#file to heatmap
aldex_all_dif<- read_tsv("../Data/aldex_treatment.tsv")

my_fun <- function(x) {
  x %>% separate(
    "Taxon", c("k", "phylum", "c", "o", "f", "g"),
    sep = "\\", remove = F) %>% dplyr::select(
      Taxon, p.value, effect, diff.btw, rab.win.0, rab.win.1, phylum,
      "FeatureID"=="Feature.ID")%>%
  drop_na(.)%>%
  rownames_to_column(var="rows")%>%
  mutate_all(funs(str_replace(., "k_Bacteria;", "")))%>%
```

```

    mutate_all(funs(str_replace(., " ; c__; o__; f__; g__; s__", "")))%>%
    mutate_all(funs(str_replace(., " ; o__; f__; g__; s__", "")))%>%
    mutate_all(funs(str_replace(., " ; f__; g__; s__", "")))%>%
    mutate_all(funs(str_replace(., " ; g__; s__", "")))%>%
    mutate_all(funs(str_replace(., " ; s__", "")))%>%mutate(
      tax= str_extract(Taxon, "[^_]+$")) %>%mutate(
        taxo = paste(rows,"_",tax))%>% mutate_at(
          c(3:7), as.numeric) %>%
    mutate_at(c(3), funs(p.Value = case_when(
      . <= 0.001 ~ "<0.001",
      . > 0.001 & . <= 0.01 ~ "<0.01",
      . > 0.01 & . <= 0.05 ~ "<0.05")))%>%
    arrange(diff.btw)%>%column_to_rownames(
      var = "taxo")%>% mutate_at(c(8),funns(str_replace(., "p__", "")))
}

#We are going to multiplicate for -1 in order to change
#the direction of the figure (e.g, bulk soil first and then rhizosphere)

annotation_heatmap <- my_fun(aldex_all_dif) %>%
  rename(rab.win.CA = rab.win.0, rab.win.CP = rab.win.1) %>%
  mutate(taxo= paste(rows,tax, sep = "_"))
data_heatmap<- annotation_heatmap%>%dplyr::select(rab.win.CA, rab.win.CP)

color_heatmap= colorRamp2(seq(min(data_heatmap), max(data_heatmap), length = 5),
                         c("#0000FF", "#5499C7", "#DAE7E4", "red", "#FF0000"))

#Annotation Phylum
cols_ann <- list('phylum' = c(
  " Acidobacteria" = 'red2',
  " Actinobacteria" = 'royalblue',
  " Bacteroidetes"="yellow",
  " Chloroflexi" ="pink",
  " Firmicutes"= "green",
  " Gemmatimonadetes" = "black",
  " Nitrospirae" ="purple",
  " Planctomycetes" ="dark green",
  " Proteobacteria" = "gray",
  " Verrucomicrobia" = "brown"))
colAnn <- HeatmapAnnotation(phylum = annotation_heatmap$phylum,
                            which = 'row',
                            col = cols_ann,
                            show_legend = T)

cols_pvalue <- list('p-value' = c("<0.001" = '#AB0000',
                                    "<0.01" = '#FF0000',
                                    "<0.05"="#FFB6B6"))

annP2 = HeatmapAnnotation("p-value" = annotation_heatmap$p.Value,

```

```

    which = "row", col = cols_pvalue,
    show_legend = T)#, gp = gpar(col = "white"))

#Annotation effect size
effect_col_fun = colorRamp2(c(-1.5, 0, 1.5), c("lightsalmon4", "white", "lightseagreen"))

annEffect = HeatmapAnnotation("effect-size" = annotation_heatmap$effect,
                               which = "row", col = list("effect-size" = effect_col_fun),
                               show_legend = T,
                               gp = gpar(col = "white"))

#Annotation barplot
bardif= rowAnnotation("difference \n between groups" = anno_barplot(
  annotation_heatmap$diff.btw, width = unit(4, "cm")))

#Annotation taxonomy

labels = c("Ellin6075", "SC-I-84", "", "iiii1-15" , "Gemm-1","", "Rhizobiales" , "Ellin5301", "Xanthomonadaceae" , "Cytophagaceae" , "iiii1-15" , "" , "Flavisolibacter" , "iiii1-15" , "Chitinophagaceae" , "Acidobacteria-6" , "Chitinophagaceae" , "iiii1-15" , "iiii1-15" , "OR-59" , "Ellin6075" , "Microbacteriaceae" , "Nitrospira" , "Chitinophagaceae" , "Ellin5290" , "" , "Ellin6529" , "Flavobacterium" , "Cytophagaceae" , "Chitinophagaceae" , "Xanthomonadaceae" , "Chitinophagaceae" , "OR-59" , "Chitinophagaceae" , "Chitinophagaceae" , "mb2424" , "Xanthomonadaceae" , "Chitinophagaceae" , "iiii1-15" , "Chitinophagaceae" , "" , "Xanthomonadaceae" , "[Chthoniobacteraceae]" , "C111" , "iiii1-15" , "Flavisolibacter" , "" , "[Chthoniobacteraceae]" , "" , "Micromonosporaceae" , "Rhizobiales" , "WD2101" , "iiii1-15" , "Flavobacterium" , rep("", 11) , "Roseococcus" , "Comamonadaceae" , "Chitinophagaceae" , "Chitinophagaceae" , "DA-101" , "" , "WD2101" , "" , "WD2101" , "Oxalobacteraceae" , "Ellin5301" , "iiii1-15" , "iiii1-15" , "Ellin5290" , "" , "" , "" , "Flavisolibacter" , "Oxalobacteraceae" , "RB41" , "iiii1-15")

heatmap_aldex_treatment<-ComplexHeatmap:: Heatmap(data_heatmap, col = color_heatmap, rows = NULL,
height = ncol(data_heatmap)*unit(8, "cm"),
left_annotation = c(annP2, annEffect, colAnn),
cluster_column_slices = F,
heatmap_legend_param = list(direction = "horizontal" ),
right_annotation = c(bardif),
column_split = rep(c("CA", "CP")),
cluster_rows = F,
cluster_columns = F,
column_km = 1, column_title_gp = gpar(
fill = c("#212F3D", "#839192" ), col="white"),
border = F, column_gap = unit(0.5, "mm"),
row_dend_side = "left", row_names_side = "right",
show_row_names = F ,rect_gp = gpar(col = "white", lwd = 0.2),
row_names_gp = gpar(fontface ="italic", fontsize=10),

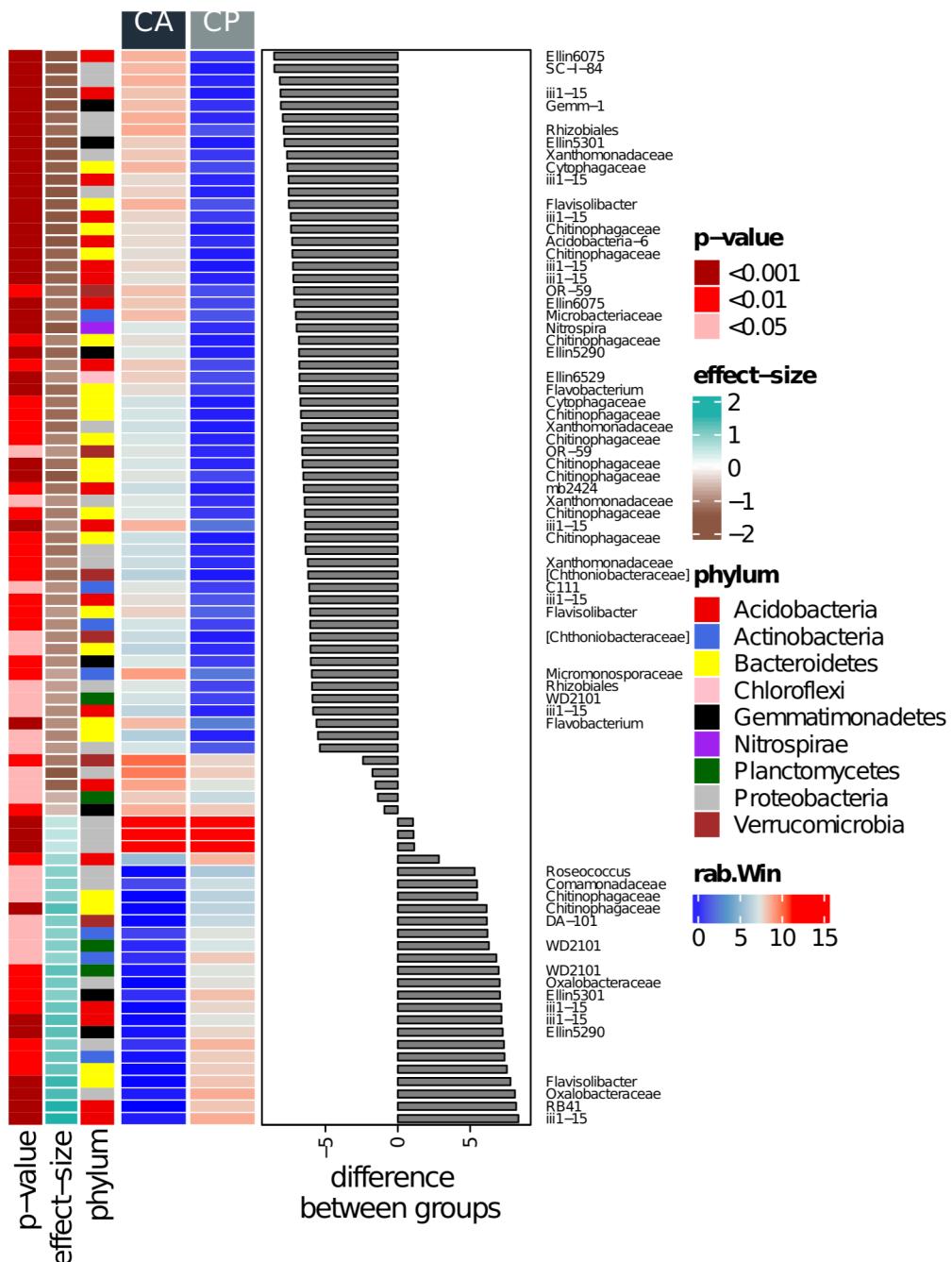
```

```

show_column_names = F, name = "rab.Win") +
rowAnnotation(labels = anno_text(labels, which = "row", gpar(
  col = "black", fontsize = 6)), width = unit(2, "cm"))

```

heatmap\_aldex\_treatment



```

#pdf("fig_aldex_TREATMENT.pdf", width=6, height=8)
#print(heatmap_aldex_treatment)
#dev.off()

```

### 5.0.5. PCA plot

```

#loading files and formatting

d.pro.0<- read_tsv("../Data/otutable.tsv")%>% column_to_rownames(var = "#OTU ID")
meta<-read_tsv("../Data/metadata.tsv")

meta$Treatment<- factor(meta$Treatment,
                         levels = c( "AC", "AT"),
                         labels = c("CA", "CP"))

tax<-read_tsv("../Data/taxonomy.tsv") %>% dplyr::select(-Confidence)%>%
  mutate_all(funns(str_replace, "k_Bacteria;", ""))
  mutate_all(funns(str_replace, "p_", ""))
  mutate_all(funns(str_replace, "c_", ""))
  mutate_all(funns(str_replace, "o_", ""))
  mutate_all(funns(str_replace, "f_", ""))
  mutate_all(funns(str_replace, "g_", ""))
  mutate_all(funns(str_replace, "s_", ""))
  mutate_all(funns(str_replace, "; ;", ""))
  mutate_all(funns(str_replace, "; ;", "")) %>% rename(
    "FeatureID"="#OTU ID", Taxon= taxonomy)

tax2<- read_tsv("../Data/taxonomy.tsv") %>% dplyr::select(
-Confidence) %>% rename(
  "FeatureID"="#OTU ID", Taxon= taxonomy)

#transforming data
d.pro <- cmultRepl(t(d.pro.0), method="CZM", output="p-counts")
d.clr.abund.codaseq<-codaSeq.clr(x = d.pro,samples.by.row = F)

#run pca
pcx.abund <- prcomp(d.clr.abund.codaseq)

#labels to pca axis

PC1 <- paste("PC1", round(sum(pcx.abund$sdev[1] ^ 2) / mvar(d.clr.abund.codaseq) * 100,
PC2 <- paste("P21", round(sum(pcx.abund$sdev[2] ^ 2) / mvar(d.clr.abund.codaseq) * 100,

#let's choose som of the significant groups from aldex analysis

vars_chosen<- c("52_Flavisolibacter",

```

```

    "37_OR-59",
    "47_Nitrospira" ,
    "16_Halomonas",
    "29_Flavobacterium",
    " 27_Steroidobacter" ,
    "36_Roseococcus")
#these ones were chosen from before (some aldex significant groups)
#these ones were chosen from before (some aldex significant groups)

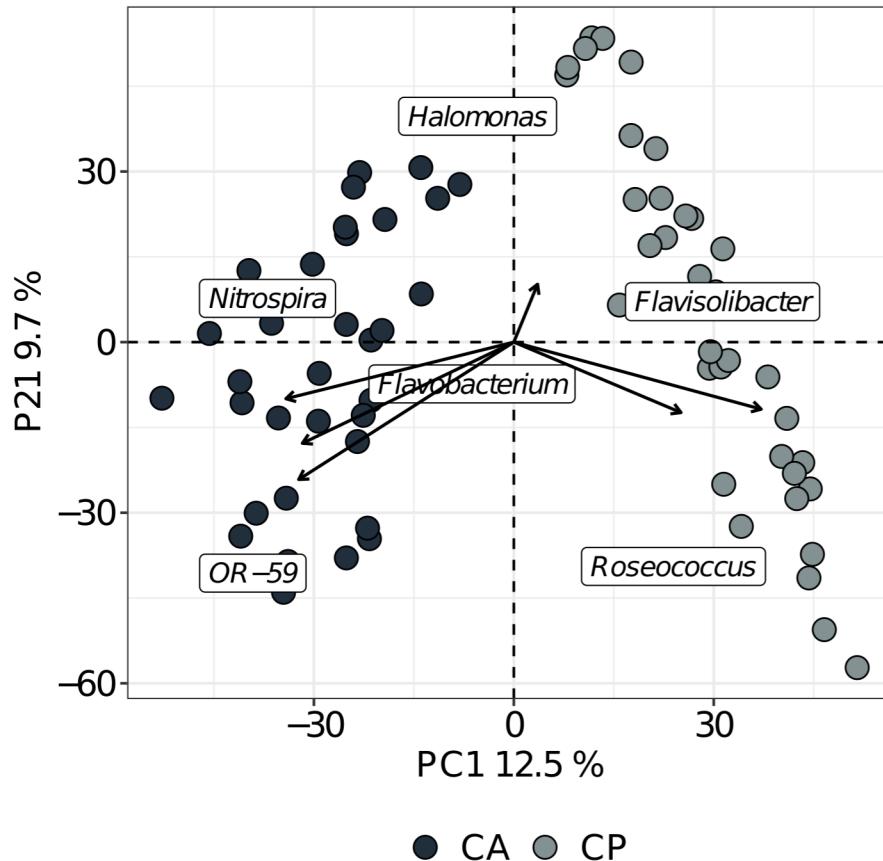
vars_to_choose<- annotation_heatmap %>% filter(taxo %in% vars_chosen)

vars_choosing<- data.frame(pcx.abund$rotation) %>% rownames_to_column(var = "FeatureID"
  mutate(a=sqrt(PC1^2+PC2^2)) %>%
  mutate(PC1=PC1*500, PC2=PC2*500) %>% left_join(tax2)%>% dplyr::select(
    Taxon, PC1, PC2, FeatureID)%>%right_join(vars_to_choose, by = "FeatureID")

#pca plot
pca_treatment_arrows<- ggplot() +
  theme_bw() +
  xlab(PC1) +
  ylab(PC2) +
  theme(axis.text = element_text(colour = "black", size = 14),#setting theme
        axis.title = element_text(colour = "black", size = 14),
        legend.text = element_text(size = 14),
        legend.title = element_blank(),
        legend.position = "bottom") +
  geom_point(#individuals
            data=data.frame(pcx.abund$x) %>% rownames_to_column(var = "SampleID")%>%
            left_join(meta, by = "SampleID"),
            aes(x=PC1, y=PC2, fill=Treatment),
            shape=21, size=4) +
  geom_vline(xintercept = 0, linetype = 2) + #lines-cross
  geom_hline(yintercept = 0, linetype = 2) +
  scale_fill_manual(values = c("#212F3D", "#839192"))+
  ggrepel::geom_label_repel(data = vars_choosing, aes(x=PC1, y=PC2, label= tax),
                            segment.colour = NA, box.padding = 2, fontface="italic")+
  geom_segment(data = vars_choosing, aes(x=0, y=0, xend=PC1, yend=PC2),
               arrow=arrow(length=unit(0.15,"cm")),
               size= 0.6)

pca_treatment_arrows

```



```
#pdf("fig_pca_treatment.pdf", width=5, height=5)
#print(pca_treatment_arrows)
#dev.off()
```

## 6. VI. STAGE FIGURE

### 6.0.1. Loading libraries

```
library(cowplot)
library(tidyverse)
library(ggpubr)
library(ComplexHeatmap)
library(circlize)
library(viridis)
library(RColorBrewer)
library(grid)
library(CoDaSeq)
library(ggplot2)
require(compositions) # exploratory data analysis of compositional data
require(zCompositions) # used for 0 substitution
```

```

require(ALDEx2) # used for per-OTU comparisons
library(CoDaSeq)
library(ggrepel)

```

### 6.0.2. Loadings files and Barplot Text annotations

```

alpha<- read.delim("../Data/Alpha-t_asv_table.txt") %>% gather(
  q0:q4, key = "q", value = "value") %>% filter(
  q %in% c("q0", "q1", "q2"))%>%mutate(qs= case_when(
    str_detect(q, "q0") ~ "q0 (species richness)",
    str_detect(q, "q1") ~ "q1 (frequent species)",
    str_detect(q, "q2") ~ "q2 (dominant species)"))

alpha$Stage <- factor(alpha$Stage,
                      levels = c('V', 'F', 'G'), ordered = TRUE)

alpha<-alpha%>%arrange(Stage)

head(alpha)

##          X Practice Soil Practice.Location Stage Age Plant Plot ExpUnit   q
## 1 CAVBS.18.1      CA   BS           CA.BS     V  3    1   18 CA.BS.V q0
## 2 CAVBS.18.2      CA   BS           CA.BS     V  3    2   18 CA.BS.V q0
## 3 CAVBS.18.3      CA   BS           CA.BS     V  3    3   18 CA.BS.V q0
## 4 CAVBS.59.1      CA   BS           CA.BS     V  3    1   59 CA.BS.V q0
## 5 CAVBS.59.2      CA   BS           CA.BS     V  3    2   59 CA.BS.V q0
## 6 CAVBS.59.3      CA   BS           CA.BS     V  3    3   59 CA.BS.V q0
##   value             qs
## 1 524 q0 (species richness)
## 2 711 q0 (species richness)
## 3 516 q0 (species richness)
## 4 625 q0 (species richness)
## 5 369 q0 (species richness)
## 6 530 q0 (species richness)

func<- read.table("../Data/func_MDq.txt") %>% gather(
  MD_q0:MD_q2, key = "q", value = "value")%>%mutate(fs= case_when(
    str_detect(q, "q0") ~ "q=0 (species richness)",
    str_detect(q, "q1") ~ "q=1 (frequent species)",
    str_detect(q, "q2") ~ "q=2 (dominant species)"))

func$Stage <- factor(func$Stage,
                      levels = c('V', 'F', 'G'), ordered = TRUE)

func<-func%>%arrange(Stage)

head(func)

```

```

## Practice Soil Practice.Location Stage Age Plant Plot ExpUnit      q     value
## 1       CA   BS          CA.BS     V  3    1   18 CA.BS.V MD_q0 37655.61
## 2       CA   BS          CA.BS     V  3    2   18 CA.BS.V MD_q0 52669.45
## 3       CA   BS          CA.BS     V  3    3   18 CA.BS.V MD_q0 36080.17
## 4       CA   BS          CA.BS     V  3    1   59 CA.BS.V MD_q0 44486.28
## 5       CA   BS          CA.BS     V  3    2   59 CA.BS.V MD_q0 26415.95
## 6       CA   BS          CA.BS     V  3    3   59 CA.BS.V MD_q0 38715.64
##                               fs
## 1 q=0 (species richness)
## 2 q=0 (species richness)
## 3 q=0 (species richness)
## 4 q=0 (species richness)
## 5 q=0 (species richness)
## 6 q=0 (species richness)

#df with the p values to show in the figures
ann_text<-data.frame(Stage=c("G", "G", "G"),value=c(890,400,200),
                      qs=c("q0 (species richness)","q1 (frequent species)","q2 (dominant species)",label=
                          "p<0.0001","p<0.0001", "p<0.0001")) #tittles and positiong in y axis
#tittles and positiong in y axis

ann_text_f<-data.frame(Practice=c("G", "G", "G"),value=c(60000,30000,15000),
                       fs=c("q=0 (species richness)","q=1 (frequent species)",
                            "q=2 (dominant species)",label=c(
                                "p<0.0001","p<0.0001", "p<0.0001")))
#tittles and positiong in y axis

```

### 6.0.3. Barplots alpha and functional diversity

```

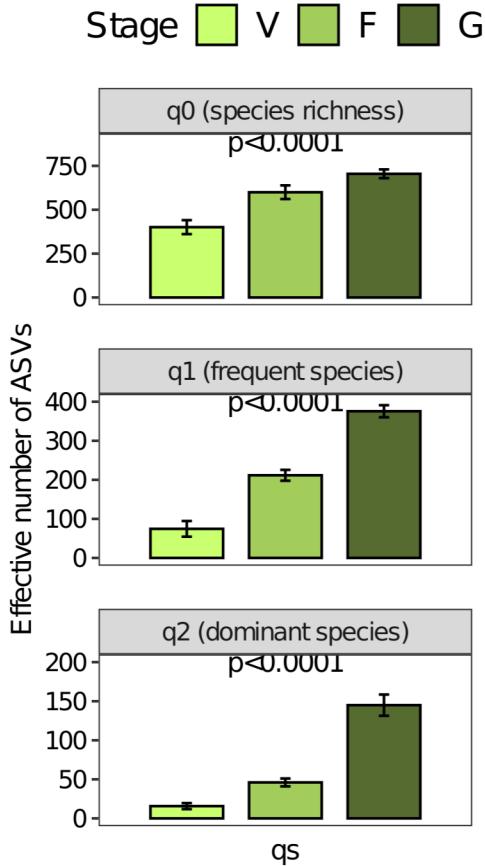
#Alpha diversity barplot
boxplot_rhizo_stage<-subset(alpha, Soil=="Rh") %>%
  ggbarplot(x="qs", y="value", fill = "Stage", add = "mean_se",
             position = position_dodge())+
  theme_bw()+
  labs(y = "Effective number of ASVs")+
  facet_wrap(~qs, scales = "free", dir = "v")+
  theme(panel.spacing=unit(1,"lines"),
        strip.text.x = element_text(size = 10),
        axis.text = element_text(colour = "black", size = 10),
        axis.ticks.x=element_blank(),
        legend.title = element_text(size = 14),
        legend.text = element_text(size=14),
        axis.text.x = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        legend.direction = "horizontal" ,
        legend.position = "top")+scale_fill_manual(values = c(
          "darkolivegreen1","darkolivegreen3","darkolivegreen"))+ labs(fill = "Stage")

```

```

boxplot_rhizo_stage<-boxplot_rhizo_stage + geom_text(data = ann_text,label=ann_text$label)
boxplot_rhizo_stage

```



```

boxplot_bulk_stage<-subset(alpha, Soil=="BS") %>%
  ggbarplot(x="qs", y="value", fill = "Stage", add = "mean_se",
             position = position_dodge())+
  theme_bw()+
  labs(y = "Effective number of ASVs")+
  facet_wrap(~qs, scales = "free", dir = "v")+
  theme(panel.spacing=unit(1,"lines"),
        strip.text.x = element_text(size = 10),
        axis.text = element_text(colour = "black", size = 10),
        axis.ticks.x=element_blank(),
        legend.title = element_text(size = 14),
        legend.text = element_text(size=14),
        axis.text.x = element_blank(),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        legend.direction = "horizontal",
        legend.position = "top")+scale_fill_manual(values = c(
          "darkolivegreen1","darkolivegreen3","darkolivegreen"))+ labs(fill = "Stage")

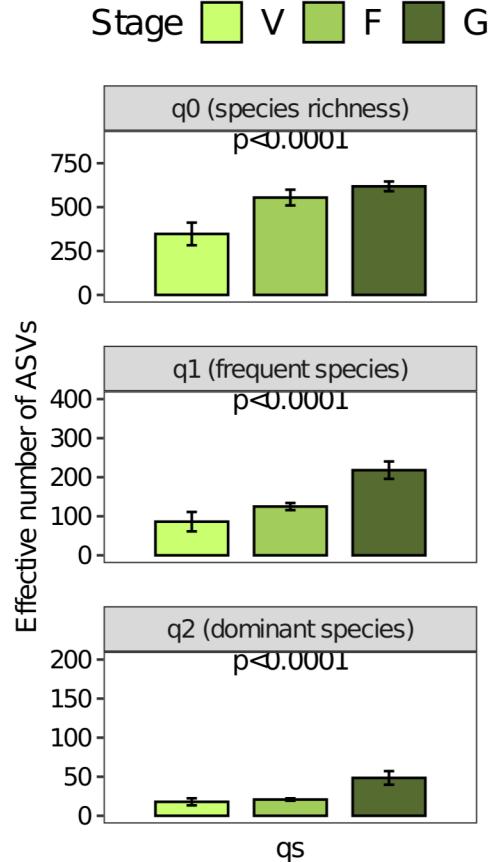
```

```

boxplot_bulk_stage<-boxplot_bulk_stage + geom_text(data = ann_text,label=ann_text$label)

boxplot_bulk_stage

```



```

#pdf("fig_bulk_stage.pdf", width=2.7, height=5)
#print(boxplot_bulk_stage)
#dev.off()
#pdf("fig_rhizo_stage.pdf", width=2.7, height=5)
#print(boxplot_rhizo_stage)
#dev.off()

```

```

#Functional diversity barplot

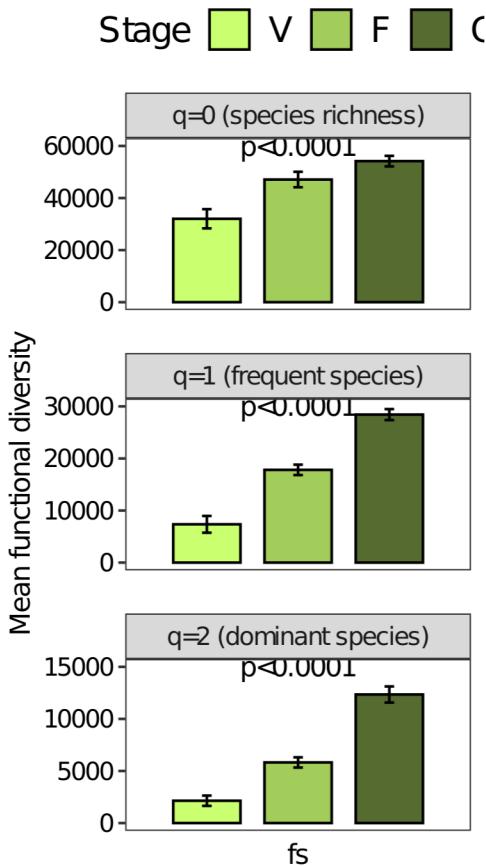
boxplot_rhizo_stage_f<-subset(func, Soil=="Rh") %>%
  ggbarplot(x="fs", y="value", fill = "Stage", add = "mean_se",
             position = position_dodge())+
  theme_bw()+
  labs(y = "Mean functional diversity")+
  facet_wrap(~fs, scales = "free", dir = "v")+
  theme(panel.spacing=unit(1,"lines"),
        strip.text.x = element_text(size = 10),

```

```

axis.text = element_text(colour = "black", size = 10),
axis.ticks.x=element_blank(),
legend.title = element_text(size = 14),
legend.text = element_text(size=14),
axis.text.x = element_blank(),
panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
legend.direction = "horizontal" ,
legend.position = "top")+scale_fill_manual(values = c("darkolivegreen1","darkoli
boxplot_rhizo_stage_f<-boxplot_rhizo_stage_f + geom_text(data = ann_text_f,label=ann_text
boxplot_rhizo_stage_f

```



```

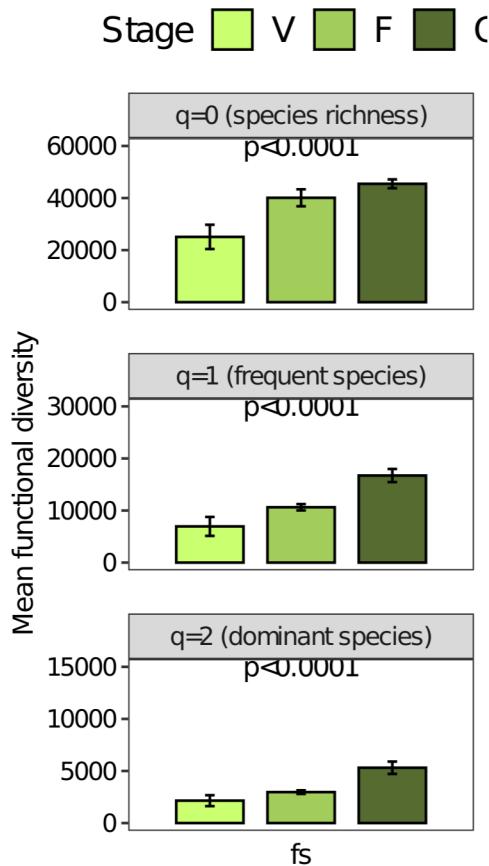
boxplot_bulk_stage_f<-subset(func, Soil=="BS") %>%
  ggbarplot(x="fs", y="value", fill = "Stage", add = "mean_se",
            position = position_dodge())+
  theme_bw()+
  labs(y = "Mean functional diversity")+
  facet_wrap(~fs, scales = "free", dir = "v")+
  theme(panel.spacing=unit(1,"lines"),
        strip.text.x = element_text(size = 10),

```

```

axis.text = element_text(colour = "black", size = 10),
axis.ticks.x=element_blank(),
legend.title = element_text(size = 14),
legend.text = element_text(size=14),
axis.text.x = element_blank(),
panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
legend.direction = "horizontal" ,
legend.position = "top")+scale_fill_manual(values = c("darkolivegreen1","darkoli
boxplot_bulk_stage_f<-boxplot_bulk_stage_f + geom_text(data = ann_text_f,label=ann_text_f
boxplot_bulk_stage_f

```



```

#pdf("fig_bulk_stage_f.pdf", width=2.7, height=5)
#print(boxplot_bulk_stage_f)
#dev.off()
#pdf("fig_rhizo_stage_f.pdf", width=2.7, height=5)
#print(boxplot_rhizo_stage_f)
#dev.off()

```

#### 6.0.4. Aldex results heatmap from Soil

```

#function to heatmap
my_fun <- function(x) {
  x %>% separate(
    "Taxon", c("k", "phylum", "c", "o", "f", "g"),
    sep = "\\\; ", remove = F) %>% dplyr::select(
      Taxon, p.value, effect, diff.btw, rab.win.0, rab.win.1, phylum,
      "FeatureID"=="Feature.ID")%>%
  drop_na(.)%>%
  rownames_to_column(var="rows")%>%
  mutate_all(funs(str_replace(., "k_Bacteria;", "")))%>%
  mutate_all(funs(str_replace(., "; c_ ; o_ ; f_ ; g_ ; s_ ", "")))%>%
  mutate_all(funs(str_replace(., "; o_ ; f_ ; g_ ; s_ ", "")))%>%
  mutate_all(funs(str_replace(., "; f_ ; g_ ; s_ ", "")))%>%
  mutate_all(funs(str_replace(., "; g_ ; s_ ", "")))%>%
  mutate_all(funs(str_replace(., "; s_ ", "")))%>%mutate(
    tax= str_extract(Taxon, "[^_]+$")) %>%mutate(
      taxo = paste(rows,"_",tax))%>% mutate_at(
        c(3:7), as.numeric) %>%
  mutate_at(c(3), funs(p.Value = case_when(
    . <= 0.001 ~ "<0.001",
    . > 0.001 & . <= 0.01 ~ "<0.01",
    . > 0.01 & . <= 0.05 ~ "<0.05")))%>%
  arrange(diff.btw)%>%column_to_rownames(
    var = "taxo")%>% mutate_at(c(8),funns(str_replace(., "p_ ", "")))}

#VvsF
#file to heatmap
aldex_all_dif_VvsF<- read_tsv("../Data/aldex_all_dif_VvsF.tsv")

annotation_heatmap1 <- my_fun(aldex_all_dif_VvsF)
data_heatmap<- annotation_heatmap1%>%dplyr::select(rab.win.0, rab.win.1)

#Setting colors to heatmap
colo_heatmap= colorRamp2(seq(min(data_heatmap), max(
  data_heatmap), length = 5), c(
  "#0000FF", "#5499C7", "#DAE7E4", "red", "#FF0000"))

#annotation phylum
cols_ann <- list('phylum' = c(" Acidobacteria" = 'red2',
                                " Actinobacteria" = 'royalblue',
                                " Bacteroidetes"="yellow",
                                " Chloroflexi" ="pink",
                                " Firmicutes"= "green",
                                " Gemmatimonadetes" = "black",
                                " Proteobacteria" = "gray",
                                " Verrucomicrobia" = "brown",
                                " Nitrospirae" ="DarkGreen",

```

```

    " TM7"= "blue",
    " Planctomycetes" ="purple"))
colAnn <- HeatmapAnnotation(phylum = annotation_heatmap1$phylum,
                           which = 'row',
                           col = cols_ann,
                           show_legend = T)

#pvalue annotation

cols_pvalue <- list('p-value' = c("<0.001" = '#AB0000',
                                    "<0.01" = '#FF0000',
                                    "<0.05"="#FFB6B6"))

annP2 = HeatmapAnnotation("p-value" = annotation_heatmap1$p.Value,
                         which = "row", col = cols_pvalue,
                         show_legend = T)##, gp = gpar(col = "white"))

#effect annotation
effect_col_fun =colorRamp2(c(-1.5, 0, 1.5), c(
  "lightsalmon4", "white", "lightseagreen"))

annEffect = HeatmapAnnotation("effect-size" = annotation_heatmap1$effect,
                             which = "row", col = list("effect-size" = effect_col_fun),
                             show_legend = T,
                             gp = gpar(col = "white"))

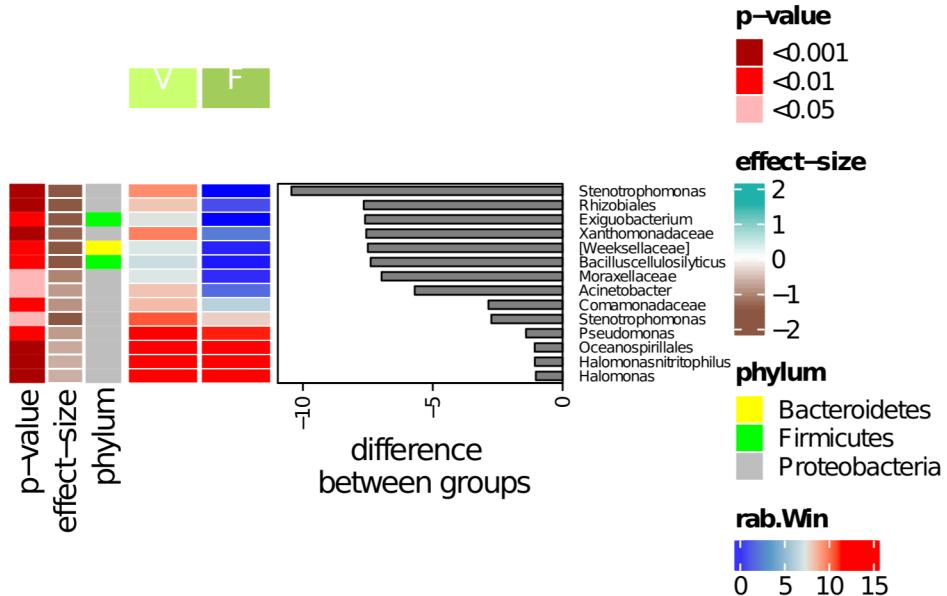
#barplot annotation
bardif= rowAnnotation(
  "difference \n between groups" = anno_barplot(
    annotation_heatmap1$diff.btw, width = unit(4, "cm")))

labels1 = (annotation_heatmap1$tax)

htVvsF<- ComplexHeatmap::Heatmap(
  as.matrix(data_heatmap), col = colo_heatmap, row_dend_reorder = F,
  height = ncol(data_heatmap)*unit(1.4, "cm"),
  left_annotation = c(annP2,annEffect, colAnn),
  heatmap_legend_param = list(direction = "horizontal" ),
  right_annotation = c(bardif),
  column_split = factor(rep(c("V", "F")), levels = c("V", "F")),
  cluster_rows = F, column_km = 1,
  column_title_gp = gpar(fill = c("darkolivegreen1","darkolivegreen3"), col="white"),
  border = F, column_gap = unit(0.5, "mm"), row_dend_side = "left",
  row_names_side = "right",show_row_names = F ,
  rect_gp = gpar(col = "white", lwd = 0.2), row_names_gp = gpar(
  fontface ="italic", fontsize=10),show_column_names = F, name = "rab.Win",
  cluster_column_slices = F) +rowAnnotation(labels = anno_text(
  labels1, which = "row", gpar(col = "black", fontsize = 6)),width = unit(2, "cm"))

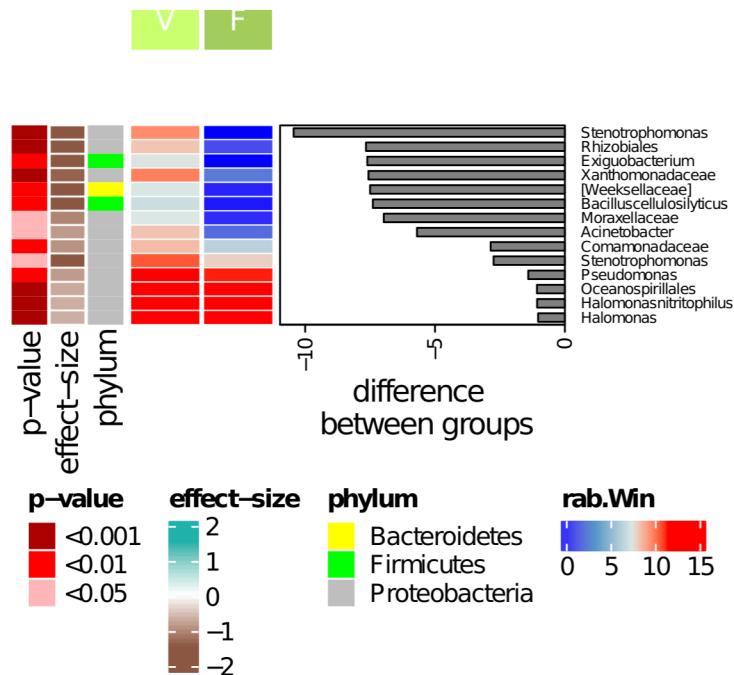
```

htVvsF



```
#pdf("fig_aldex_VvsF.pdf", width=6, height=5)
#print(htVvsF)
#dev.off()

htVvsF.2<-draw(htVvsF, heatmap_legend_side = "bottom",
annotation_legend_side = "bottom")
```



```

#pdf("fig_aldex_VvsF2.pdf", width=6, height=6)
#print(htVvsF.2)
#dev.off()

#FVSG
#loading file
aldex_all_dif_FvsG<-read_tsv("../Data/aldex_all_dif_FvsG.tsv")

annotation_heatmap2 <- my_fun(aldex_all_dif_FvsG)
data_heatmap<- annotation_heatmap2%>%dplyr::select(rab.win.0, rab.win.1)

#Setting colors to heatmap
colo_heatmap= colorRamp2(seq(min(data_heatmap), max(data_heatmap),
length = 5), c("#0000FF", "#5499C7", "#DAE7E4", "red", "#FF0000"))

#annotation phylum
cols_ann <- list('phylum' = c(
  " Acidobacteria" = 'red2',
  " Actinobacteria" = 'royalblue',
  " Bacteroidetes"="yellow",
  " Chloroflexi" ="pink",
  " Firmicutes"= "green",
  " Gemmatimonadetes" = "black",
  " Proteobacteria"  ="gray",
  " Verrucomicrobia" = "brown",
  " TM7"= "blue",
  " Planctomycetes" = "purple"))

colAnn <- HeatmapAnnotation(phylum = annotation_heatmap2$phylum,
                           which = 'row',
                           col = cols_ann,
                           show_legend = F)

#pvalue annotation
cols_pvalue <- list('p-value' = c("<0.001" = '#AB0000',
                                    "<0.01" = '#FF0000',
                                    "<0.05"="#FFB6B6"))

annP2 = HeatmapAnnotation("p-value" = annotation_heatmap2$p.Value,
                         which = "row", col = cols_pvalue,
                         show_legend = F)

#effect annotation
effect_col_fun =colorRamp2(c(-1.5, 0, 1.5), c(
  "lightsalmon4", "white", "lightseagreen"))

```

```

annEffect = HeatmapAnnotation("effect-size" = annotation_heatmap2$effect,
                                which = "row", col = list(
                                    "effect-size" = effect_col_fun),
                                show_legend = F,
                                gp = gpar(col = "white"))

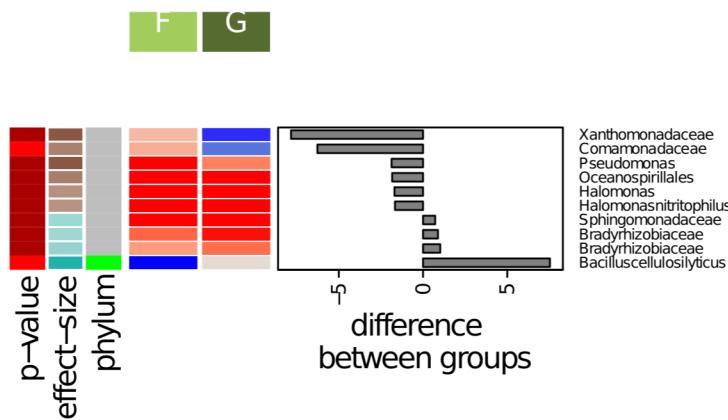
#barplot annotation
bardif= rowAnnotation(
  "difference \n between groups" = anno_barplot(
    annotation_heatmap2$diff.btw, width = unit(4, "cm")))

labels2 = (annotation_heatmap2$tax)

htFvsG<-ComplexHeatmap::Heatmap(
  data_heatmap, col = colo_heatmap, row_dend_reorder = F,
  width = ncol(data_heatmap)*unit(1, "cm"),
  height = ncol(data_heatmap)*unit(1, "cm"),
  left_annotation = c(annP2, annEffect, colAnn),
  heatmap_legend_param = list(direction = "horizontal" ),
  right_annotation = c(bardif),
  column_split = rep(c("F", "G")),
  cluster_rows = F, show_heatmap_legend = F,
  cluster_column_slices = F,
  column_km = 1, column_title_gp = gpar(
    fill = c("darkolivegreen3","darkolivegreen"), col="white"),
  border = F, column_gap = unit(0.5, "mm"),
  row_dend_side = "left",row_names_side = "right",show_row_names = F ,
  rect_gp = gpar(col = "white", lwd = 0.2), row_names_gp = gpar(
    fontface ="italic", fontsize=10),show_column_names = F,
  name = "rab.Win") + rowAnnotation(
    labels = anno_text(labels2, which = "row", gpar(
      col = "black", fontsize = 6)), width = unit(2, "cm"))

htFvsG

```



```

#pdf("fig_aldex_FvsG.pdf", width=6, height=5)
#print(htFvsG)
#dev.off()

# VvsG

aldex_all_dif_VvsG<-read_tsv("../Data/aldex_all_dif_VvsG.tsv")
annotation_heatmap3 <- my_fun(aldex_all_dif_VvsG)
data_heatmap<- annotation_heatmap3%>%dplyr::select(rab.win.0, rab.win.1)

#Setting colors to heatmap
colo_heatmap= colorRamp2(seq(min(data_heatmap), max(data_heatmap),
length = 5), c("#0000FF", "#5499C7", "#DAE7E4", "red", "#FF0000"))

#annotation phylum
cols_ann <- list('phylum' = c(
  " Acidobacteria" = 'red2',
  " Actinobacteria" = 'royalblue',
  " Bacteroidetes"="yellow",
  " Chloroflexi" ="pink",
  " Firmicutes"= "green",
  " Gemmatimonadetes" = "black",
  " Proteobacteria"  ="gray",
  " Verrucomicrobia" = "brown",
  " TM7"= "blue",
  " Planctomycetes" ="purple"))

colAnn <- HeatmapAnnotation(phylum = annotation_heatmap3$phylum,
                           which = 'row',
                           col = cols_ann,
                           show_legend = F)

#pvalue annotation

cols_pvalue <- list('p-value' = c("<0.001" = '#AB0000',
                                    "<0.01" = '#FF0000',
                                    "<0.05"="#FFB6B6"))

annP2 = HeatmapAnnotation("p-value" = annotation_heatmap3$p.Value,
                          which = "row", col = cols_pvalue,
                          show_legend = F)##, gp = gpar(col = "white"))

#effect annotation
effect_col_fun =colorRamp2(c(-1.5, 0, 1.5), c(
  "lightsalmon4", "white", "lightseagreen"))

annEffect = HeatmapAnnotation("effect-size" = annotation_heatmap3$effect,
                             which = "row",

```

```

        col = list("effect-size" = effect_col_fun),
        show_legend = F,
        gp = gpar(col = "white"))

#barplot annotation
bardif= rowAnnotation(
  "difference \n between groups" = anno_barplot(
    annotation_heatmap3$diff.btw, width = unit(4, "cm")))

labels3 = (annotation_heatmap3$tax)

htVvsG<-ComplexHeatmap::Heatmap(
  data_heatmap, col = colo_heatmap, row_dend_reorder = F,
  width = ncol(data_heatmap)*unit(1, "cm"),
  height = ncol(data_heatmap)*unit(1.4, "cm"),
  left_annotation = c(annP2, annEffect, colAnn),
  heatmap_legend_param = list(direction = "horizontal" ),
  right_annotation = c(bardif),
  column_split = factor(rep(c("V", "G")), levels = c("V", "G")),
  cluster_rows = F,show_heatmap_legend = F,
  column_km = 1, column_title_gp = gpar(fill = c(
  "darkolivegreen1", "darkolivegreen"), col="white"),
  border = F, column_gap = unit(0.5, "mm"),
  row_dend_side = "left",row_names_side = "right",show_row_names = F ,
  rect_gp = gpar(col = "white", lwd = 0.2), row_names_gp = gpar(
  fontface ="italic", fontsize=10),show_column_names = F, name = "rab.Win")+
  rowAnnotation(labels = anno_text(labels3, which = "row",
  gpar(col = "black", fontsize = 6)),width = unit(2, "cm"))

#pdf("fig_aldex_VvsG.pdf", width=6, height=5)
#print(htVvsG)
#dev.off()

```

#### 6.0.5. PCA plot

```

#loading files and formatting
d.pro.0<- read_tsv("../Data/otutable.tsv") %>% column_to_rownames(var = "#OTU ID")

meta<-read_tsv("../Data/metadata.tsv")

meta$Stage<- factor(meta$Maize_development_stage,
  levels = c( "Vegetative", "Flowering", "Grainfilling"),
  labels = c("V", "F", "G"))

tax2<- read_tsv("../Data/taxonomy.tsv")%>% rename(
  "FeatureID"=~#OTU ID`, Taxon= taxonomy)

tax3<-tax2%>% separate(
  "Taxon", c("k", "phylum","c", "o","f","g"),

```

```

sep = "\\; ", remove = F) %>%
rownames_to_column(var="rows")%>%
mutate_all(funs(str_replace(., "k_Bacteria;", "")))%>%
mutate_all(funs(str_replace(., "; c_; o_; f_; g_; s_"; "", "")))%>%
mutate_all(funs(str_replace(., "; o_; f_; g_; s_"; "", "")))%>%
mutate_all(funs(str_replace(., "; f_; g_; s_"; "", "")))%>%
mutate_all(funs(str_replace(., "; g_; s_"; "", "")))%>%
mutate_all(funs(str_replace(., "; s_"; "", "")))%>%mutate(
  tax= str_extract(Taxon, "[^_]+$"))

sample_to_choose<- meta %>% filter(Soil_sample=="Rhizosphere")

#transforming data
d.pro.0.rhizo<- d.pro.0  %>% dplyr::select(0, sample_to_choose$SampleID)
d.pro.rhizo <- t(cmultRepl(t(d.pro.0.rhizo), method="CZM", output="p-counts"))
d.clr.abund.codaseq.rhizo<-codaSeq.clr(x = d.pro.rhizo,samples.by.row = F)

#run pca
pcx.abund.rhizo <- prcomp(t(d.clr.abund.codaseq.rhizo))

#labels to pca axis

PC1 <- paste(
  "PC1", round(sum(pcx.abund.rhizo$sdev[1] ^ 2) / mvar(d.clr.abund.codaseq.rhizo) , 1),
PC2 <- paste(
  "PC2", round(sum(pcx.abund.rhizo$sdev[2] ^ 2) / mvar(d.clr.abund.codaseq.rhizo) , 1),

#let's choose som of the significant groups from aldex analysis

annot_heat<- merge(annotation_heatmap1,
                     annotation_heatmap2, by = "FeatureID") %>%full_join(
  annotation_heatmap3, by = "FeatureID")

vars_chosen<- c("d0dbf2a66c655edf1f45eb0fe9415866",
               "2553e8df6ec901e443d9f4ed5f7ea2fe",
               "008e9d51155f32838e58a5a6eb48f335",
               "#"61d320df173b3b20ac4bb8a0b9adcb3c",
               "f35cd29ecc2c92909b596ad30084ea48",
               "f75c3dab2258512ada2c3af6f86e5865",
               "cf75802eef23e2082bcb012af233a01b")
               # "3882df43374c4d647c02bb95fc46c3ed",
               # "2553e8df6ec901e443d9f4ed5f7ea2fe",
               # "087cf9bebbcc26a354bc475125443455")
#these ones were chosen from before (some aldex significant groups)

vars_to_choose<- annotation_heatmap3 %>% rownames_to_column(
  var = "ids")%>%filter(FeatureID %in% vars_chosen)

vars_choosing<- data.frame(

```

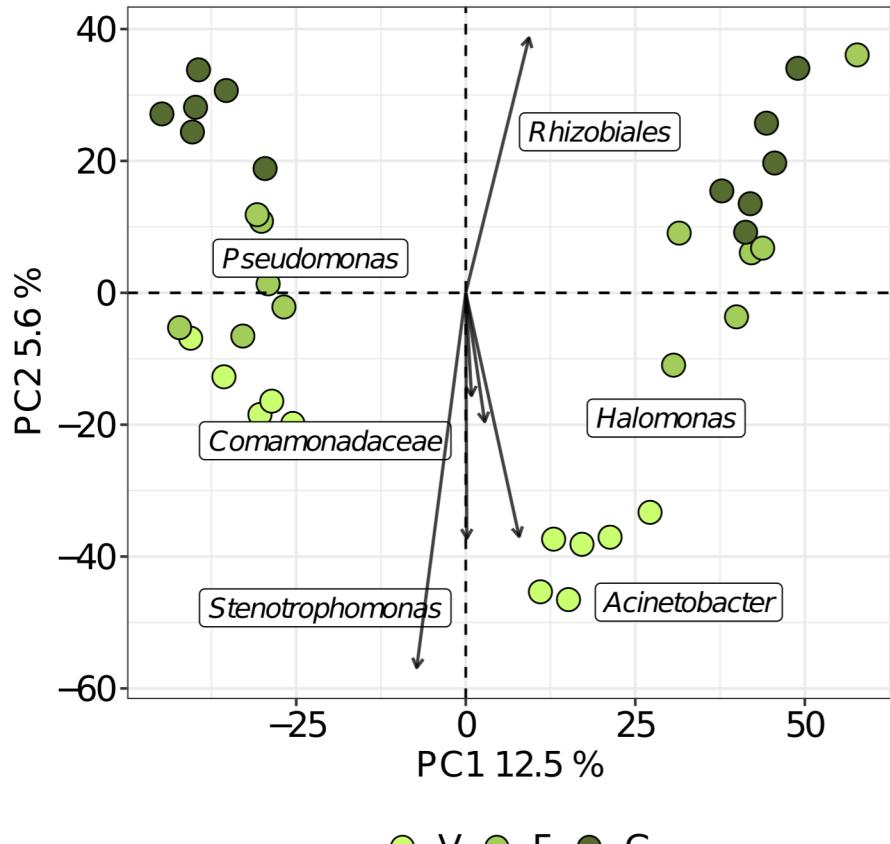
```

pcx.abund.rhizo$rotation) %>% rownames_to_column(
  var = "FeatureID")%>%
  mutate(a=sqrt(PC1^2+PC2^2)) %>%
  mutate(PC1=PC1*500, PC2=PC2*500) %>% dplyr::select(
    PC1, PC2, FeatureID)%>%right_join(vars_to_choose, by = "FeatureID")

#pca-plot
pca_stage_arrows<- ggplot() +
  theme_bw() +
  xlab(PC1) +
  ylab(PC2) +
  theme(axis.text = element_text(colour = "black", size = 14), #setting themes
        axis.title = element_text(colour = "black", size = 14),
        legend.text = element_text(size = 14),
        legend.title = element_blank(),
        legend.position = "bottom",
        legend.box = "horizontal",
        legend.direction = "horizontal") +
  geom_point(                                     #individuals
    data=data.frame(pcx.abund.rhizo$x) %>%   rownames_to_column(var = "SampleID")%>%
    left_join(meta, by = "SampleID"),
    aes(x=PC1, y=PC2, fill=Stage),
    shape=21, size=4) +
  geom_vline(xintercept = 0, linetype = 2) +    #lines-cross
  geom_hline(yintercept = 0, linetype = 2) +
  scale_fill_manual(values = c( "darkolivegreen1","darkolivegreen3","darkolivegreen"))+
  ggrepel::geom_label_repel(data = vars_choosing, aes(x=PC1, y=PC2, label= tax),
                            segment.colour = NA, box.padding = 2, fontface="italic")+
  geom_segment(data = vars_choosing, aes(x = 0, y = 0, xend = PC1, yend = PC2),
               arrow=arrow(length=unit(0.15,"cm")), #arros and names
               alpha = 0.75, color = 'black', size= 0.6)

pca_stage_arrows

```



● V ● F ● G

```
#pdf("fig_PCA_rhizo_stage.pdf", width=5, height=5)
#print(pca_stage_arrows)
#dev.off()

# PCA VEGETATIVE STAGE

sample_to_choose_v<- meta %>% filter(Stage=="V")
d.pro.O.V<- d.pro.O %>% dplyr::select(0, sample_to_choose_v$SampleID)
d.pro.V <- t(cmultRepl(t(d.pro.O.V), method="CZM", output="p-counts")) #tratamiento de

d.clr.abund.codaseq.V<-codaSeq.clr(x = d.pro.V, samples.by.row = F) #transformacion clr

pcx.abund.V <- prcomp(t(d.clr.abund.codaseq.V))

PC1 <- paste(
  "PC1", round(sum(pcx.abund.V$sdev[1] ^ 2) / mvar(d.clr.abund.codaseq.V), 1), "%")
PC2 <- paste(
  "PC2", round(sum(pcx.abund.V$sdev[2] ^ 2) / mvar(d.clr.abund.codaseq.V) , 1), "%")

vars_choosing<- data.frame(pcx.abund.V$rotation) %>% rownames_to_column(var = "Feature")
```

```

    mutate(a=sqrt(PC1^2+PC2^2)) %>%
    mutate(PC1=PC1*500, PC2=PC2*500) %>% top_n(8, a) %>% dplyr::select(
      PC1, PC2, FeatureID) %>% right_join(tax3, by = "FeatureID")

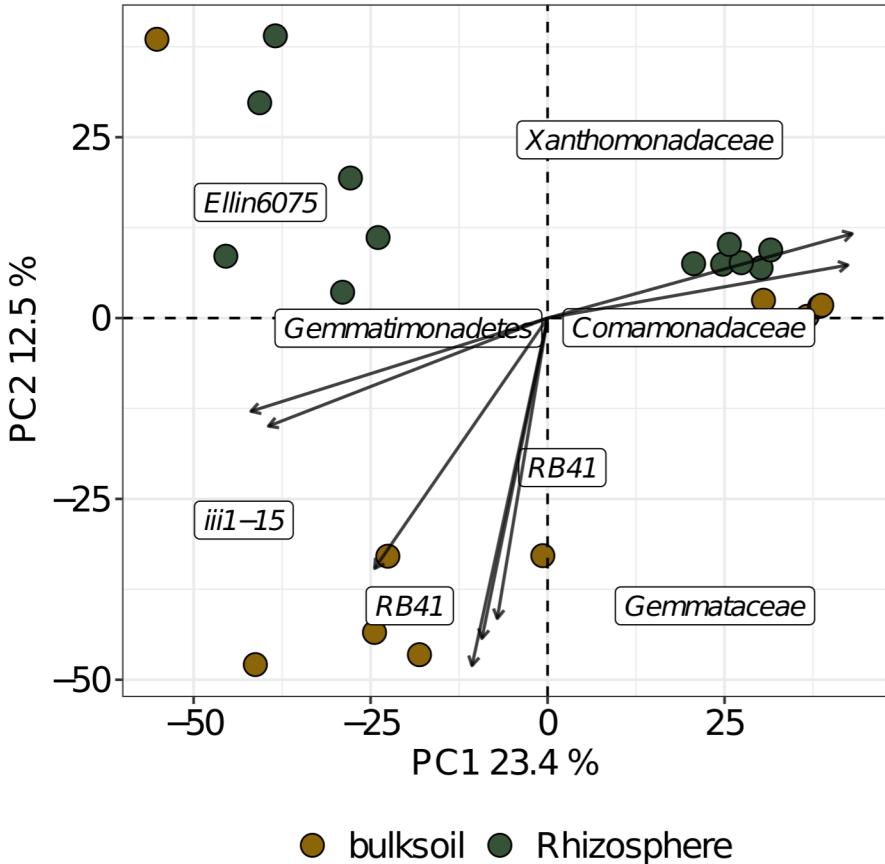


#pca-plot


pca_stage_arrows_V<- ggplot() +
  theme_bw() +
  xlab(PC1) +
  ylab(PC2) +
  theme(axis.text = element_text(colour = "black", size = 14), #setting theme
        axis.title = element_text(colour = "black", size = 14),
        legend.text = element_text(size = 14),
        legend.title = element_blank(),
        legend.position = "bottom",
        legend.box = "horizontal",
        legend.direction = "horizontal") +
  geom_point(                                     #individuals
             data=data.frame(pcx.abund.V$x) %>%   rownames_to_column(var = "SampleID")%>%
               left_join(meta, by = "SampleID"),
             aes(x=PC1, y=PC2, fill=Soil.sample),
             shape=21, size=4) +
  geom_vline(xintercept = 0, linetype = 2) +    #lines-cross
  geom_hline(yintercept = 0, linetype = 2) +
  scale_fill_manual(values = c("darkgoldenrod4", "#365238"))+
  ggrepel::geom_label_repel(data = vars_choosing, aes(x=PC1, y=PC2, label= tax),
                            segment.colour = NA, box.padding = 2, fontface="italic")+
  geom_segment(data = vars_choosing, aes(x = 0, y = 0, xend = PC1, yend = PC2),
               arrow=arrow(length=unit(0.15,"cm")), #arrows and names
               alpha = 0.75, color = 'black', size= 0.6)

pca_stage_arrows_V

```



```

#pdf("fig_PCA_vegetative.pdf", width=5, height=5)
#print(pca_stage_arrows_V)
#dev.off()

# PCA FLOWERING STAGE

sample_to_choose_f<- meta %>% filter(Stage=="F")
d.pro.O.F<- d.pro.O %>% dplyr::select(0, sample_to_choose_f$SampleID)
d.pro.F <- t(cmultRepl(t(d.pro.O.F), method="CZM", output="p-counts")) #tratamiento de

d.clr.abund.codaseq.F<-codaSeq.clr(x = d.pro.F, samples.by.row = F) #transformacion clr

pcx.abund.F <- prcomp(t(d.clr.abund.codaseq.F))

PC1 <- paste(
  "PC1", round(sum(pcx.abund.F$sdev[1] ^ 2) / mvar(d.clr.abund.codaseq.F) * 100, 1), "%"
PC2 <- paste(
  "PC2", round(sum(pcx.abund.F$sdev[2] ^ 2) / mvar(d.clr.abund.codaseq.F) * 100, 1), "%"

vars_choosing<- data.frame(pcx.abund.F$rotation) %>% rownames_to_column(var = "Feature"
  mutate(a=sqrt(PC1^2+PC2^2)) %>%

```

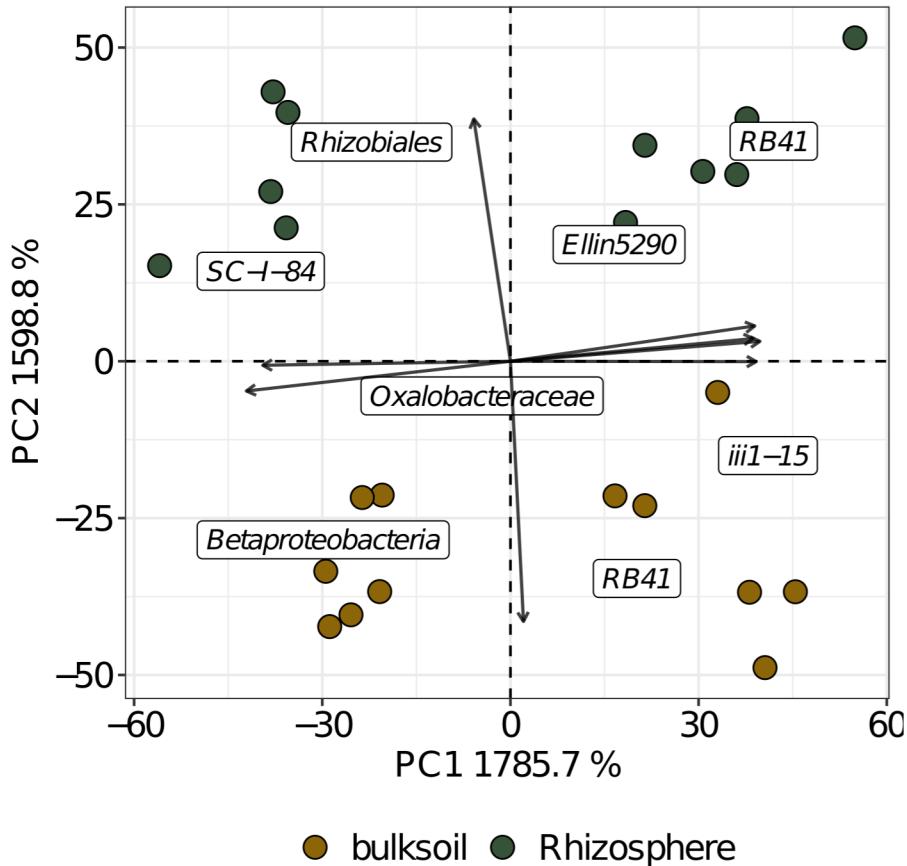
```

    mutate(PC1=PC1*500, PC2=PC2*500) %>% top_n(8, a) %>% dplyr::select(
      PC1, PC2, FeatureID) %>% right_join(tax3, by = "FeatureID")

#create the base plot with only the arrows
pca_stage_arrows_F<- ggplot() +
  theme_bw() +
  xlab(PC1) +
  ylab(PC2) +
  theme(axis.text = element_text(colour = "black", size = 14), #setting themes
        axis.title = element_text(colour = "black", size = 14),
        legend.text = element_text(size = 14),
        legend.title = element_blank(),
        legend.position = "bottom",
        legend.box = "horizontal",
        legend.direction = "horizontal") +
  geom_point(                                     #individuals
    data=data.frame(pcx.abund.F$x) %>% rownames_to_column(var = "SampleID")%>%
    left_join(meta, by = "SampleID"),
    aes(x=PC1, y=PC2, fill=Soil_sample),
    shape=21, size=4) +
  geom_vline(xintercept = 0, linetype = 2) +    #lines-cross
  geom_hline(yintercept = 0, linetype = 2) +
  scale_fill_manual(values = c("darkgoldenrod4", "#365238"))+
  ggrepel::geom_label_repel(data = vars_choosing, aes(x=PC1, y=PC2, label= tax),
                            segment.colour = NA, box.padding = 2, fontface="italic")+
  geom_segment(data = vars_choosing, aes(x = 0, y = 0, xend = PC1, yend = PC2),
               arrow=arrow(length=unit(0.15,"cm")), #arrows and names
               alpha = 0.75, color = 'black', size= 0.6)

pca_stage_arrows_F

```



```

#pdf("fig_PCA_flowering.pdf", width=5, height=5)
#print(pca_stage_arrows_F)
#dev.off()

# PCA GRAIN FILLING STAGE

sample_to_choose_g<- meta %>% filter(Stage=="G")
d.pro.0.G<- d.pro.0 %>% dplyr::select(0, sample_to_choose_g$SampleID)
d.pro.G <- t(cmultRepl(t(d.pro.0.G), method="CZM", output="p-counts")) #tratamiento de

d.clr.abund.codaseq.G<-codaSeq.clr(x = d.pro.G, samples.by.row = F) #transformacion clr

pcx.abund.G <- prcomp(t(d.clr.abund.codaseq.G))

PC1 <- paste("PC1", round(sum(pcx.abund.G$sdev[1] ^ 2) / mvar(d.clr.abund.codaseq.G) ,
PC2 <- paste("PC2", round(sum(pcx.abund.G$sdev[2] ^ 2) / mvar(d.clr.abund.codaseq.G) ,

vars_choosing<- data.frame(pcx.abund.G$rotation) %>% rownames_to_column(var = "Feature"
  mutate(a=sqrt(PC1^2+PC2^2)) %>%
  mutate(PC1=PC1*500, PC2=PC2*500) %>% top_n(8, a) %>% dplyr::select(

```

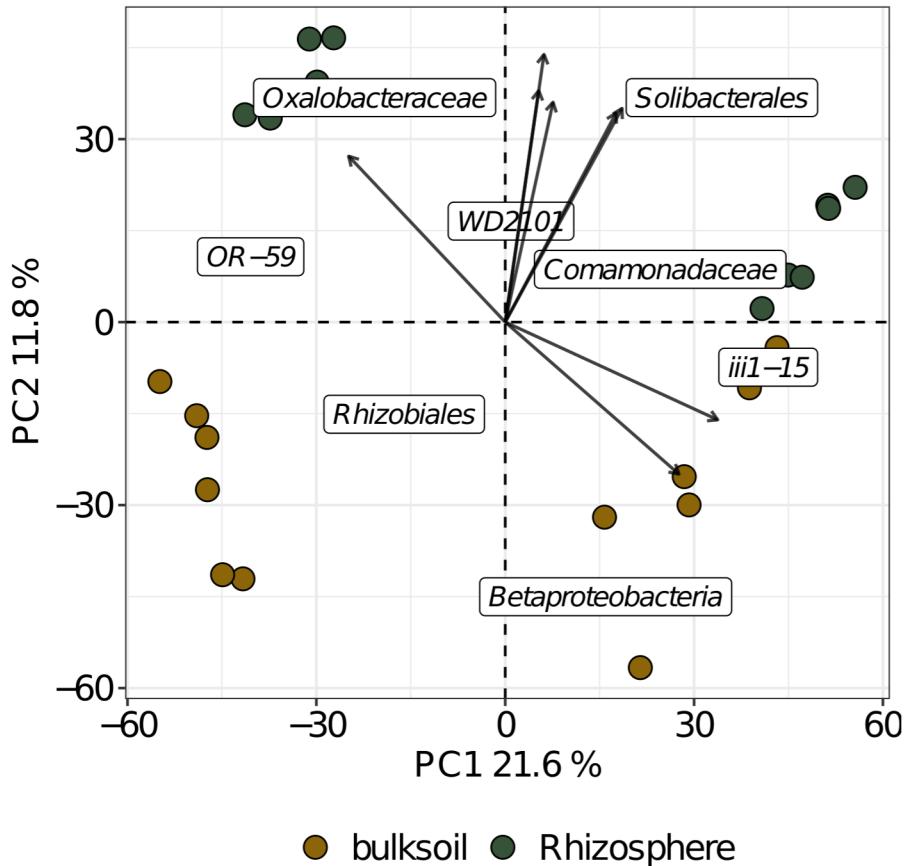
```

PC1, PC2, FeatureID) %>% right_join(tax3, by = "FeatureID")

#create the base plot with only the arrows
pca_stage_arrows_G<- ggplot() +
  theme_bw() +
  xlab(PC1) +
  ylab(PC2) +
  theme(axis.text = element_text(colour = "black", size = 14), #setrrting theme
        axis.title = element_text(colour = "black", size = 14),
        legend.text = element_text(size = 14),
        legend.title = element_blank(),
        legend.position = "bottom",
        legend.box = "horizontal",
        legend.direction = "horizontal") +
  geom_point(                                         #individuals
    data=data.frame(pcx.abund.G$x) %>%  rownames_to_column(var = "SampleID")%>%
      left_join(meta, by = "SampleID"),
    aes(x=PC1, y=PC2, fill=Soil_sample),
    shape=21, size=4) +
  geom_vline(xintercept = 0, linetype = 2) +   #lines-cross
  geom_hline(yintercept = 0, linetype = 2) +
  scale_fill_manual(values = c("darkgoldenrod4", "#365238"))+
  ggrepel::geom_label_repel(data = vars_choosing, aes(x=PC1, y=PC2, label= tax),
                            segment.colour = NA, box.padding = 2, fontface="italic")+
  geom_segment(data = vars_choosing, aes(x = 0, y = 0, xend = PC1, yend = PC2),
               arrow=arrow(length=unit(0.15,"cm"))), #arrows and names
               alpha = 0.75, color = 'black', size= 0.6)

pca_stage_arrows_G

```



```
#pdf("fig_PCA_grainfilling.pdf", width=5, height=5)
#print(pca_stage_arrows_G)
#dev.off()
```

## 7. VII. PICRUST PLOT

### 7.0.1. Loading libraries

```
library(ComplexHeatmap)
library(tidyverse)
library(circlize)
library(viridis)
library(RColorBrewer)
library(cowplot)
```

### 7.0.2. Setting common annotations to heatmap

```
levels<- read_tsv( "../Data/levels.tsv")
```

```

## 
## -- Column specification -----
## cols(
##   pathway = col_character(),
##   description = col_character(),
##   level1 = col_character(),
##   level2 = col_character(),
##   level3 = col_character()
## )

cols_ann <- list('Superclass' = c(
  "Alcohol Degradation"="#A6CEE3",
  "Aldehyde Degradation"="#00FFFF",
  "Amine and Polyamine Biosynthesis"="#B2DF8A",
  "Amine and Polyamine Degradation"="#3300CC",
  "Amino Acid Biosynthesis"="#33A02C",
  "Amino Acid Degradation"="#99FFFF",
  "Aminoacyl-tRNA Charging"="#99CC66",
  "Aromatic Compound Degradation"="#006699",
  "C1 Compound Utilization and Assimilation"="#6699CC",
  "Carbohydrate Biosynthesis"="#B3DE69",
  "Carbohydrate Degradation"="#6699FF",
  "Carboxylate Degradation"="#0033CC",
  "Cell Structure Biosynthesis"="#CCEBC5",
  "Cofactor, Carrier, and Vitamin Biosynthesis"="#66FF00",
  "Cofactor, Prosthetic Group, Electron Carrier Degradation"="#00CCFF",
  "Degradation/Utilization/Assimilation"="#666699",
  "Fatty Acid and Lipid Biosynthesis"="#66CC33",
  "Fatty Acid and Lipid Degradation"="#000666",
  "Fermentation"="#CC0000",
  "Glycolysis"="#993333",
  "Inorganic Nutrient Metabolism"="#6666FF",
  "Metabolic Regulator Biosynthesis"="#669933",
  "Nucleic Acid Processing"="#FFFF00",
  "Nucleoside and Nucleotide Biosynthesis"="#339933",
  "Nucleoside and Nucleotide Degradation"="#99CCFF",
  "Other"="#000000",
  "Other Biosynthesis"="#069966",
  "Pentose Phosphate Pathways"="#FF6666",
  "Polyprenyl Biosynthesis"="#00FF33",
  "Respiration"="#CC6666",
  "Secondary Metabolite Biosynthesis"="#99CC00",
  "Secondary Metabolite Degradation"="#66CCCC",
  "TCA cycle"="#990033",
  "Tetrapyrrole Biosynthesis"="#CCFF99"))

cols_pvalue <- list('p-value' = c("<0.001" = '#AB0000',
                                 "<0.01" = '#FF0000',
                                 "<0.05" = '#FFB6B6"))

```

```
effect_col_fun =colorRamp2(c(-1.5, 0, 1.5), c(
  "lightsalmon4", "white", "lightseagreen"))
```

### 7.0.3. Treatment Picrust

```
aldex_all_dif<- read_tsv( "../Data/aldex_Treatment_picrust.tsv")

annotation_heatmap<- aldex_all_dif%>% left_join(
  levels, by = c("Feature.ID"="pathway"))%>% dplyr::select(
  level2, Feature.ID, p.Value, effect, diff.btw) %>% mutate_at(c(
    3), funs(p.value = case_when(
      . <= 0.001 ~ "<0.001",
      . > 0.001 & . <= 0.01 ~ "<0.01",
      . > 0.01 & . <= 0.05 ~ "<0.05")))%>%arrange(
    diff.btw)%>%column_to_rownames(var = "Feature.ID")

data_heatmap<- aldex_all_dif %>% arrange(diff.btw)%>%column_to_rownames(
  var = "Feature.ID")%>%dplyr::select(
  rab.win.CA, rab.win.CP, diff.btw) %>% arrange(diff.btw)

color_heatmap= colorRamp2(seq(min(data_heatmap), max(data_heatmap), length = 5), c(
  "#0000FF", "#5499C7", "#DAE7E4", "red", "#FF0000"))

colAnn <- HeatmapAnnotation(Superclass = annotation_heatmap$level2,
  which = 'row',
  col = cols_ann,
  show_legend = F)

annP2 = HeatmapAnnotation("p-value" = annotation_heatmap$p.value,
  which = "row", col = cols_pvalue,
  show_legend = F)

annEffect = HeatmapAnnotation("effect-size" = annotation_heatmap$effect,
  which = "row", col = list(
  "effect-size" = effect_col_fun),
  show_legend = F,
  gp = gpar(col = "white"))

bardif= rowAnnotation(
  "difference between groups" = anno_barplot(
  annotation_heatmap$diff.btw, width = unit(4, "cm")))

ht5<-ComplexHeatmap::Heatmap(
  data_heatmap[-3],
  row_dend_reorder = F, col = color_heatmap,
  width = ncol(data_heatmap)*unit(0.6, "cm"),
```

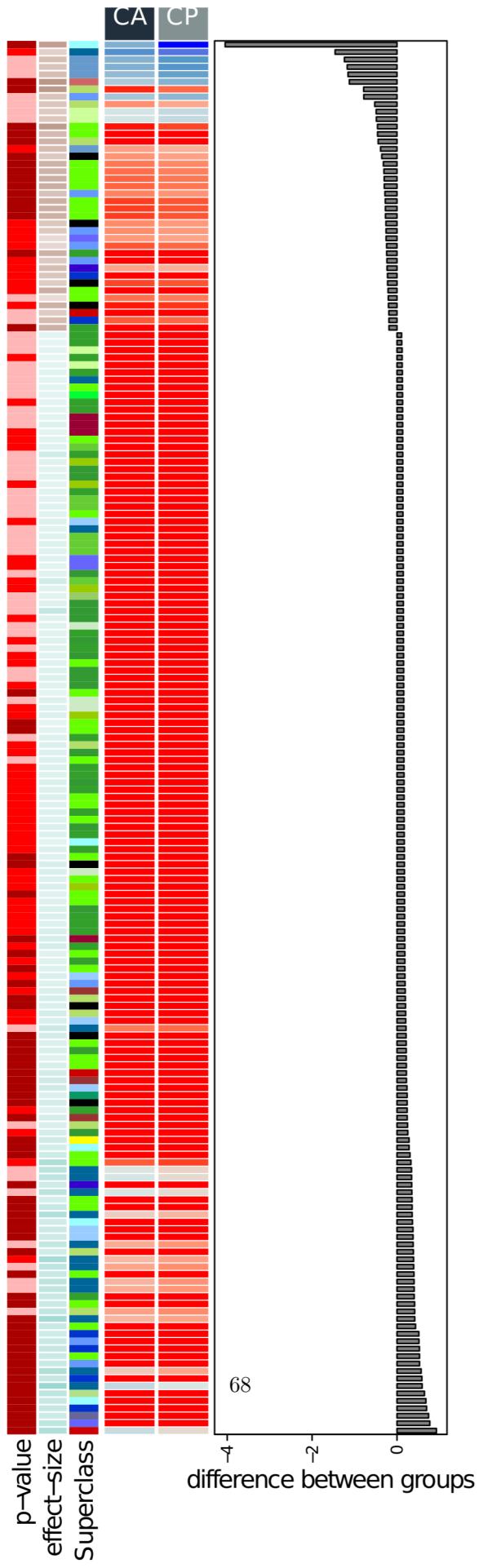
```

height = ncol(data_heatmap)*unit(8, "cm"),
left_annotation = c(annP2, annEffect, colAnn),
heatmap_legend_param = list(direction = "vertical" ),
right_annotation = c(bardif),
cluster_column_slices = FALSE,
column_split = rep(c("CA", "CP")),
cluster_rows = F,
column_km = 1, column_title_gp = gpar(
fill = c("#212F3D", "#839192" ), col="white"),
border = F, column_gap = unit(0.5, "mm"),
row_dend_side = "left", row_names_side = "right", show_row_names = F ,
rect_gp = gpar(col = "white", lwd = 0.2), row_names_gp = gpar(
fontface ="italic", fontsize=10),
cluster_columns = F,
show_column_names = F, name = "rab.Win")

#ht5

ht5.2<-draw(ht5, heatmap_legend_side = "bottom")

```



```

#pdf("fig-picrust_TREATMENT2.pdf", width=7, height=20)
#print(ht5.2)
#dev.off()

#pdf("fig-picrust_TREATMENT.pdf", width=10, height=10)
#print(ht5)
#dev.off()

```

#### 7.0.4. Soil Picrust

```

aldex_all_dif<- read_tsv( "../Data/aldex_Soil_picrust.tsv")

annotation_heatmap<- aldex_all_dif%>% left_join(levels, by = c(
  "Feature.ID"="pathway"))%>% dplyr::select(
  level2, Feature.ID, p.Value, effect, diff.btw, rab.win.Rh, rab.win.Bs )%>% mutate_at(
  c(3), funs(p.value = case_when(
    . <= 0.001 ~ "<0.001",
    . > 0.001 & . <= 0.01 ~ "<0.01",
    . > 0.01 & . <= 0.05 ~ "<0.05")))%>% mutate(
    diff.btw2 = diff.btw*-1, effect2 = effect*-1 ) %>% arrange(
    diff.btw2)%>%column_to_rownames(var = "Feature.ID")

data_heatmap<- annotation_heatmap%>%dplyr::select(
  rab.win.Bs, rab.win.Rh, diff.btw2 ) %>% arrange(
  diff.btw2)

color_heatmap= colorRamp2(
  seq(min(data_heatmap), max(data_heatmap),
  length = 5), c("#0000FF", "#5499C7", "#DAE7E4", "red", "#FF0000"))

colAnn <- HeatmapAnnotation(Superclass = annotation_heatmap$level2,
  which = 'row',
  col = cols_ann,
  show_legend = F)

annP2 = HeatmapAnnotation("p-value" = annotation_heatmap$p.value,
  which = "row", col = cols_pvalue,
  show_legend = F)

annEffect = HeatmapAnnotation("effect-size" = annotation_heatmap$effect,
  which = "row", col = list(
  "effect-size" = effect_col_fun),
  show_legend = F,
  gp = gpar(col = "white"))

bardif= rowAnnotation(
  "difference between groups" = anno_barplot(

```

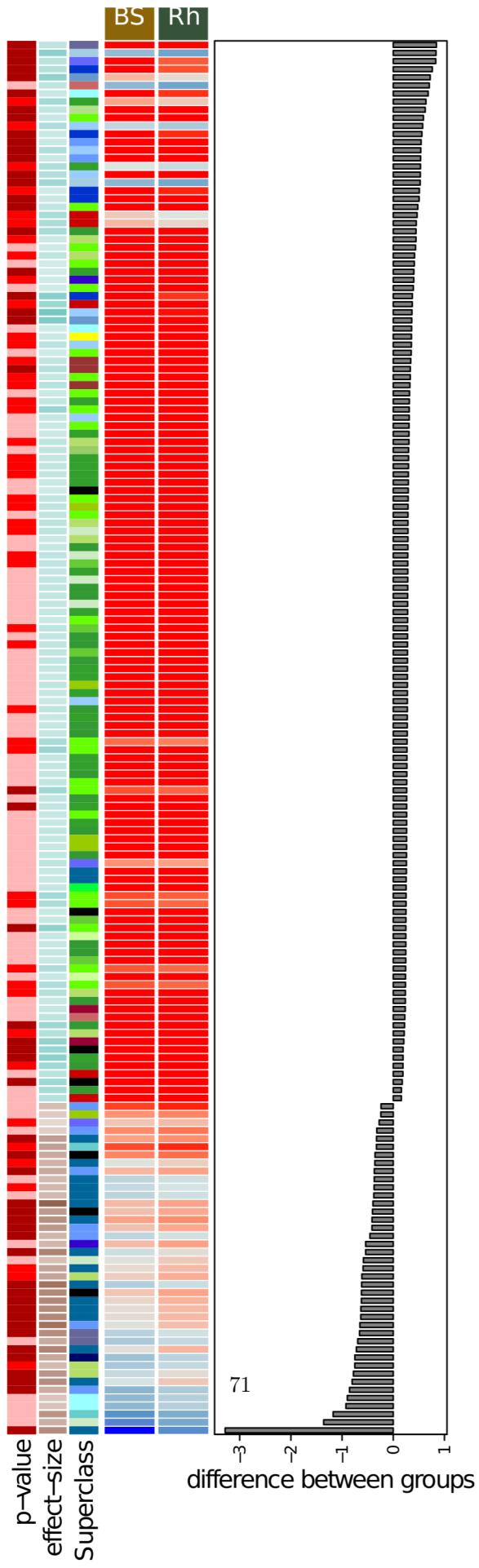
```

annotation_heatmap$diff.btw, width = unit(4, "cm")))

ht4<-ComplexHeatmap::Heatmap(
  data_heatmap[-3], row_dend_reorder = F, col = color_heatmap,
  width = ncol(data_heatmap)*unit(0.6, "cm"),
  height = ncol(data_heatmap)*unit(8, "cm"),
  left_annotation = c(annP2, annEffect, colAnn),
  heatmap_legend_param = list(direction = "vertical" ),
  right_annotation = c(bardif),
  cluster_column_slices = FALSE,
  column_split = rep(c("BS", "Rh")),
  show_heatmap_legend = T,
  cluster_rows = F,
  column_km = 1, column_title_gp = gpar(
    fill = c("darkgoldenrod4", "#365238" ), col="white"),
  border = F, column_gap = unit(0.5, "mm"),
  row_dend_side = "left",row_names_side = "right",show_row_names = F ,
  rect_gp = gpar(col = "white", lwd = 0.2), row_names_gp = gpar(
    fontface ="italic", fontsize=10),
  cluster_columns = F,
  show_column_names = F, name = "rab.Win")

ht4.2<-draw(ht4, heatmap_legend_side = "bottom")

```



```

#pdf("fig_picrust_soil2.pdf", width=7, height=20)
#print(ht4.2)
#dev.off()

#pdf("fig_picrust_soil.pdf", width=10, height=10)
#print(ht4)
#dev.off()

```

#### 7.0.5. Stage Picrust

```

# VvsF
aldex_all_dif<- read_tsv( "../Data/aldex_Stage_vvsf_picrust.tsv")

#construct heatmap
annotation_heatmap<- aldex_all_dif%>% left_join(
  levels, by = c("Feature.ID"="pathway"))%>% dplyr::select(
  level2, Feature.ID, p.Value, effect, diff.btw) %>% mutate_at(c(3), funs(
    p.value = case_when(
      . <= 0.001 ~ "<0.001",
      . > 0.001 & . <= 0.01 ~ "<0.01",
      . > 0.01 & . <= 0.05 ~ "<0.05")))%>%arrange(
    diff.btw)%>%column_to_rownames(var = "Feature.ID")

data_heatmap<- aldex_all_dif %>% arrange(
  diff.btw)%>%column_to_rownames(
  var = "Feature.ID")%>%dplyr::select(
  rab.win.0, rab.win.1, diff.btw) %>% rename(
  Ve=rab.win.0 , Fl=rab.win.1 )%>% arrange(diff.btw)

colAnn <- HeatmapAnnotation(
  Superclass = annotation_heatmap$level2,
  which = 'row',
  col = cols_ann,
  show_legend = F)

annP2 = HeatmapAnnotation("p-value" = annotation_heatmap$p.value,
  which = "row", col = cols_pvalue,
  show_legend = F)

#effect annotation

annEffect = HeatmapAnnotation(
  "effect-size" = annotation_heatmap$effect,
  which = "row", col = list("effect-size" = effect_col_fun),

```

```

show_legend  =F,
gp = gpar(col = "white"))

#barplot annotation
bardif= rowAnnotation(
  "difference between groups" = anno_barplot(
    annotation_heatmap$diff.btw, width = unit(4, "cm")))

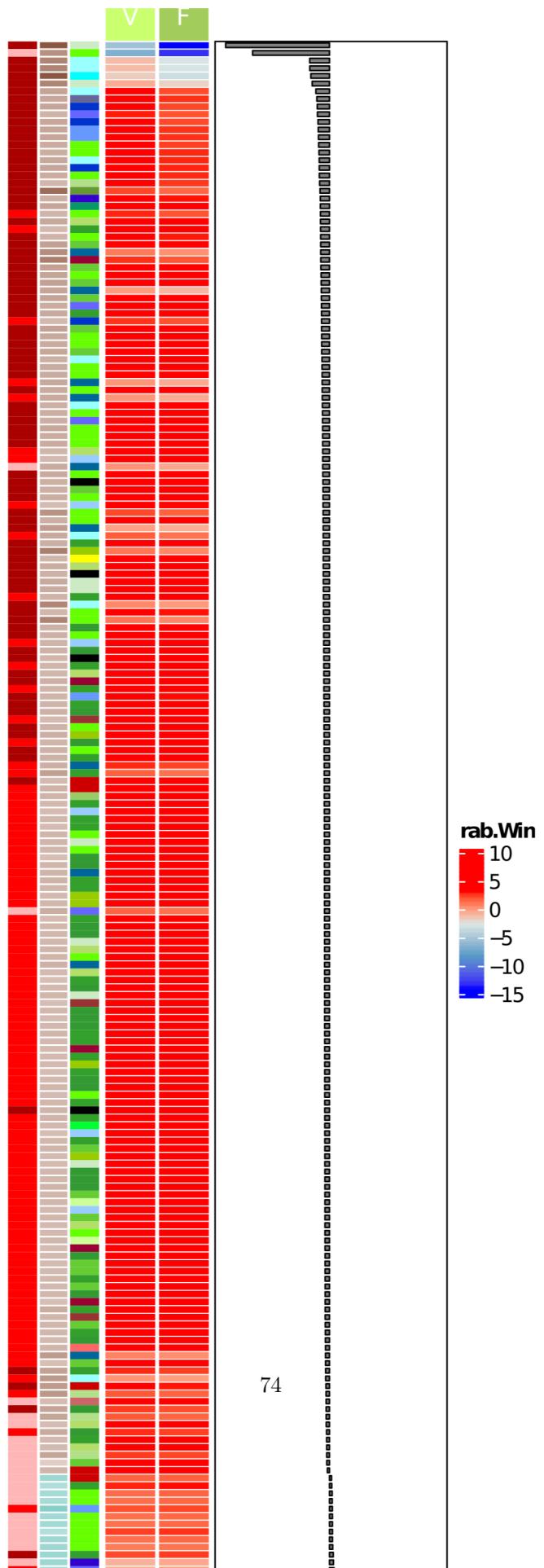
color_heatmap= colorRamp2(
  seq(min(data_heatmap), max(
    data_heatmap), length = 5), c(
      "#0000FF", "#5499C7", "#DAE7E4", "red", "#FF0000"))

htVvsF<- ComplexHeatmap::Heatmap(
  as.matrix(data_heatmap[-3]), col = color_heatmap, row_dend_reorder = F,
  width = ncol(data_heatmap)*unit(0.6, "cm"),
  height = ncol(data_heatmap)*unit(10, "cm"),
  left_annotation = c(annP2,annEffect, colAnn),
  heatmap_legend_param = list(direction = "vertical" ),
  right_annotation = c(bardif),
  column_split = factor(rep(c("V", "F")), levels = c("V", "F")),
  cluster_rows = F,
  column_km = 1,
  column_title_gp = gpar(fill = c(
    "darkolivegreen1", "darkolivegreen3"), col="white"),
  border = F, column_gap = unit(0.5, "mm"), row_dend_side = "left",
  row_names_side = "right", show_row_names = F ,
  rect_gp = gpar(col = "white", lwd = 0.2), row_names_gp = gpar(
    fontface ="italic", fontsize=10),
  show_column_names = F, name = "rab.Win",
  cluster_columns = F,
  cluster_column_slices = F)

htVvsF

```

#### 7.0.5.1. *Vegetative vs Flowering.*



74