230313_Reporte3PruebasdeHipotesis

Camila Silva

2023-03-10

DIVERSIDADES ALFA Y BETA FUSARIUM

```
library("phyloseq")
library("ggplot2")
library("vegan")
```

```
## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.6-4
```

```
#library("BiodiversityR")
library("RColorBrewer")
library("stringi")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("plyr")
```

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
setwd("/home/camila/GIT/Tesis_Maestria/Data/fresa_solena")
outpath = "/home/camila/GIT/Tesis_Maestria/Analisis_Comparativo/Fresa_Solena/Results_img"
### Cargado de datos originales
fresa_kraken <- import_biom("fresa_kraken.biom")
colnames(fresa_kraken@tax_table@.Data) <- c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus", "S
fresa_kraken@tax_table@.Data <- substr(fresa_kraken@tax_table@.Data,4,100)
colnames(fresa_kraken@otu_table@.Data) <- substr(colnames(fresa_kraken@otu_table@.Data),1,6)
metadata_fresa <- read.csv2("/home/camila/GIT/Tesis_Maestria/Data/fresa_solena/metadata.csv",header =  T
fresa_kraken@sam_data <- sample_data(metadata_fresa)
fresa_kraken@sam_data$Sample<-row.names(fresa_kraken@sam_data)
colnames(fresa_kraken@sam_data)<-c('Treatment','Samples')
samples_to_remove <- c("MP2079","MP2080","MP2088","MP2109","MP2137")
fresa_kraken_fil <- prune_samples(!(sample_names(fresa_kraken) %in% samples_to_remove), fresa_kraken)
percentages_fil <- transform_sample_counts(fresa_kraken_fil, function(x) x*100 / sum(x) )
percentages_df <- psmelt(percentages_fil)
```

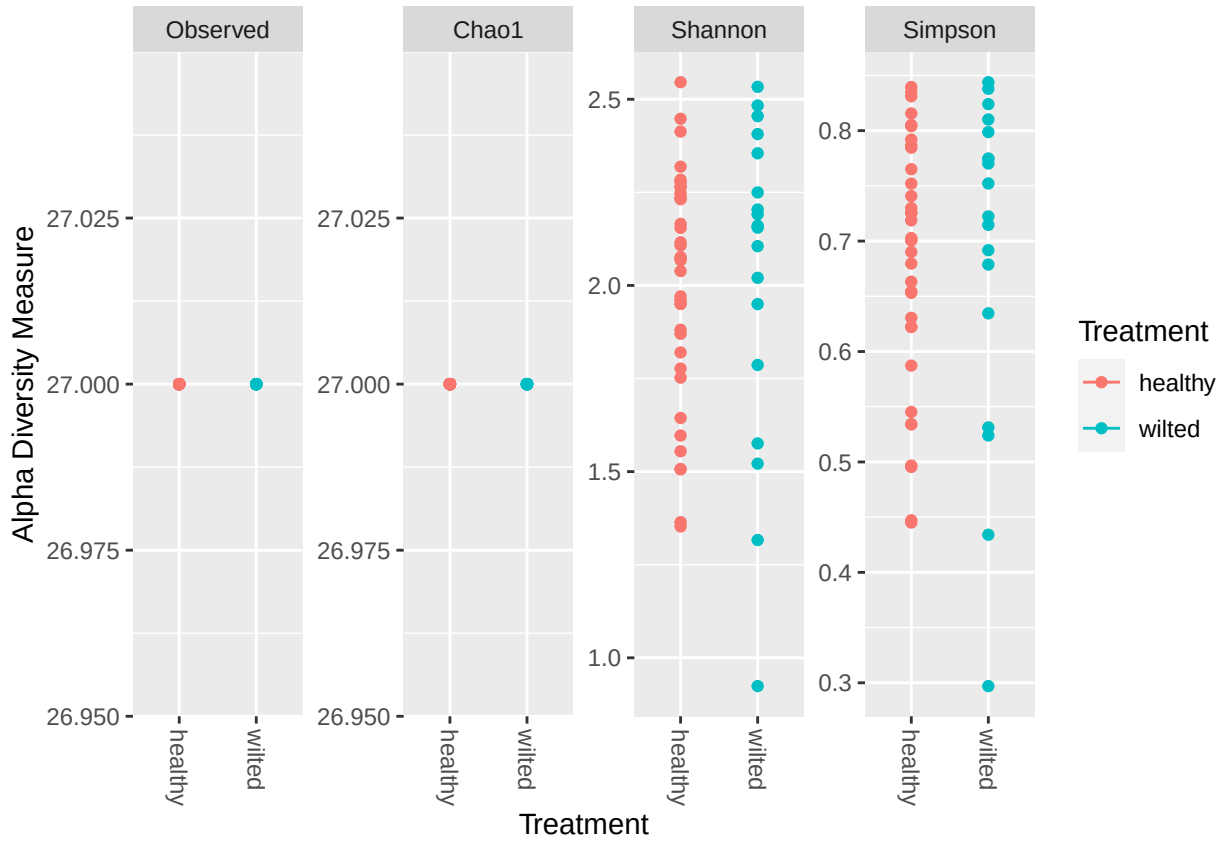
```
## Subconjunto de "Eukaryota"
merge_Eukaryota<-subset_taxa(fresa_kraken_fil,Kingdom=="Eukaryota")
## Subconjunto de "Bacteria"
merge_Bacteria<-subset_taxa(fresa_kraken_fil,Kingdom=="Bacteria")
```

```
## Aglomeramos Fusarium
#glom <- tax_glom(merge_Eukaryota, taxrank = 'Genus')
glom_Fusarium <- subset_taxa(merge_Eukaryota, Genus == 'Fusarium')

## se calcula diversidad alfa con el glom de fusarium
plot_alpha_Fusarium <- plot_richness(physeq = glom_Fusarium, measures = c("Observed","Chao1","Shannon",
```

```
## Warning in estimate_richness(physeq, split = TRUE, measures = measures): The data you have provided c
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```

```
plot_alpha_Fusarium
```

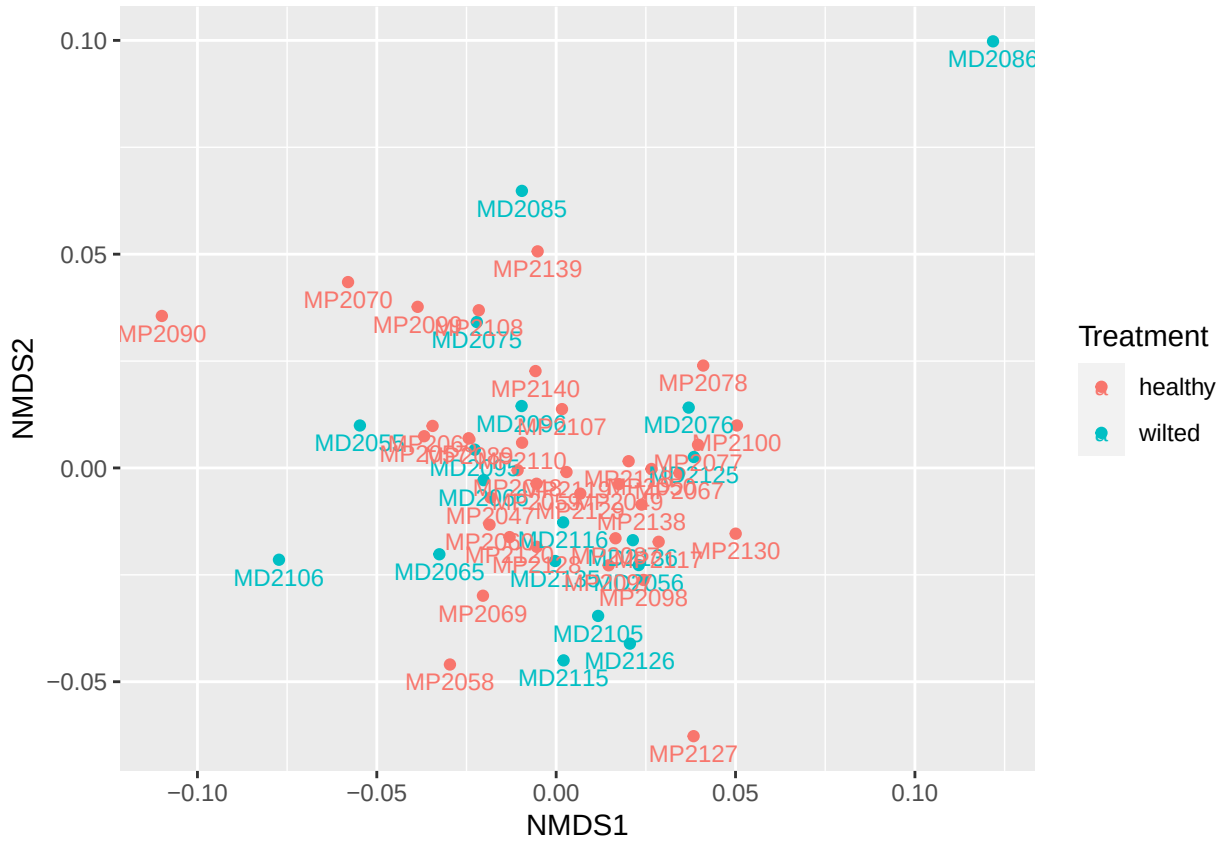



```
## se calcula la beta diversidad para Fusarium
## sacamos los porcentajes
percentages_Fusarium <- transform_sample_counts(glom_Fusarium, function(x) x*100 / sum(x) )
percentages_Fusarium_df <- psmelt(percentages_Fusarium)
meta_ord_Fusarium <- ordinate(physeq = percentages_Fusarium, method = "NMDS", distance = 'bray')
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1365369
## Run 1 stress 0.136705
## ... Procrustes: rmse 0.005485918  max resid 0.03712895
## Run 2 stress 0.1384475
## Run 3 stress 0.1502045
## Run 4 stress 0.1502046
## Run 5 stress 0.4036808
## Run 6 stress 0.1365367
## ... New best solution
## ... Procrustes: rmse 0.001399979  max resid 0.00723435
## ... Similar to previous best
## Run 7 stress 0.1367159
## ... Procrustes: rmse 0.005450815  max resid 0.03663826
## Run 8 stress 0.1365368
## ... Procrustes: rmse 0.001404876  max resid 0.007217298
```

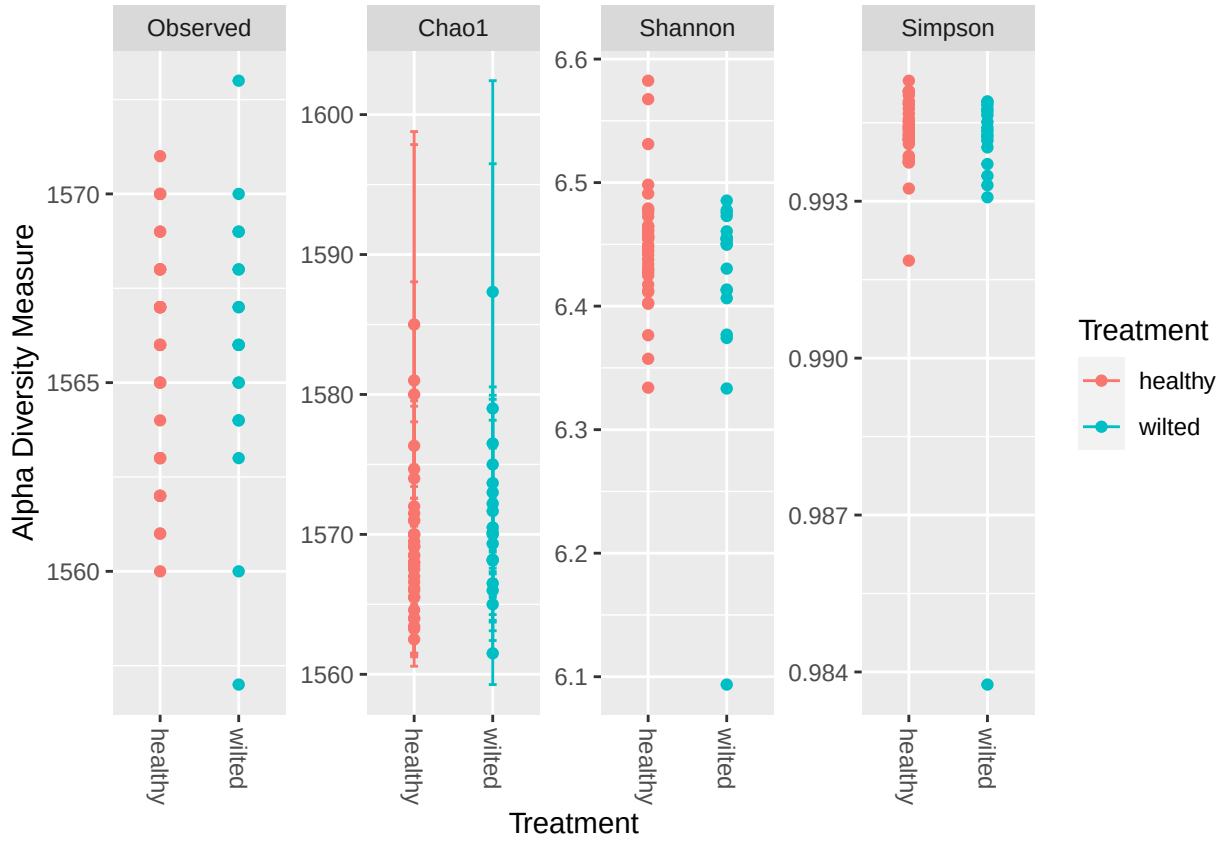
```
## ... Similar to previous best
## Run 9 stress 0.1502045
## Run 10 stress 0.1381408
## Run 11 stress 0.1382158
## Run 12 stress 0.1365365
## ... New best solution
## ... Procrustes: rmse 0.0003662819  max resid 0.002294161
## ... Similar to previous best
## Run 13 stress 0.1583245
## Run 14 stress 0.1502046
## Run 15 stress 0.1818567
## Run 16 stress 0.1502046
## Run 17 stress 0.1365366
## ... Procrustes: rmse 0.001341403  max resid 0.007605085
## ... Similar to previous best
## Run 18 stress 0.1365366
## ... Procrustes: rmse 7.065472e-05  max resid 0.0004389484
## ... Similar to previous best
## Run 19 stress 0.1365366
## ... Procrustes: rmse 0.0003142678  max resid 0.001969085
## ... Similar to previous best
## Run 20 stress 0.1365365
## ... New best solution
## ... Procrustes: rmse 0.0002669258  max resid 0.001673531
## ... Similar to previous best
## *** Best solution repeated 1 times
```

```
plot_beta_Fusarium <- plot_ordination(physeq = percentages_Fusarium, ordination = meta_ord_Fusarium, co
  geom_text(mapping = aes(label = colnames(glom_Fusarium@otu_table@.Data)), size = 3, vjust = 1.5)
plot_beta_Fusarium
```




```
## Aglomeramos Actinobacteria
#Actinobacteria
#glom <- tax_glom(merge_Bacteria, taxrank = 'Phylum')
glom_Actinobacteria <- subset_taxa(merge_Bacteria, Phylum == 'Actinobacteria')

## se calcula diversidad alfa con el glom de Actinobacteria
plot_alpha_Actinobacteria <- plot_richness(physeq = glom_Actinobacteria, measures = c("Observed","Chao1
plot_alpha_Actinobacteria
```

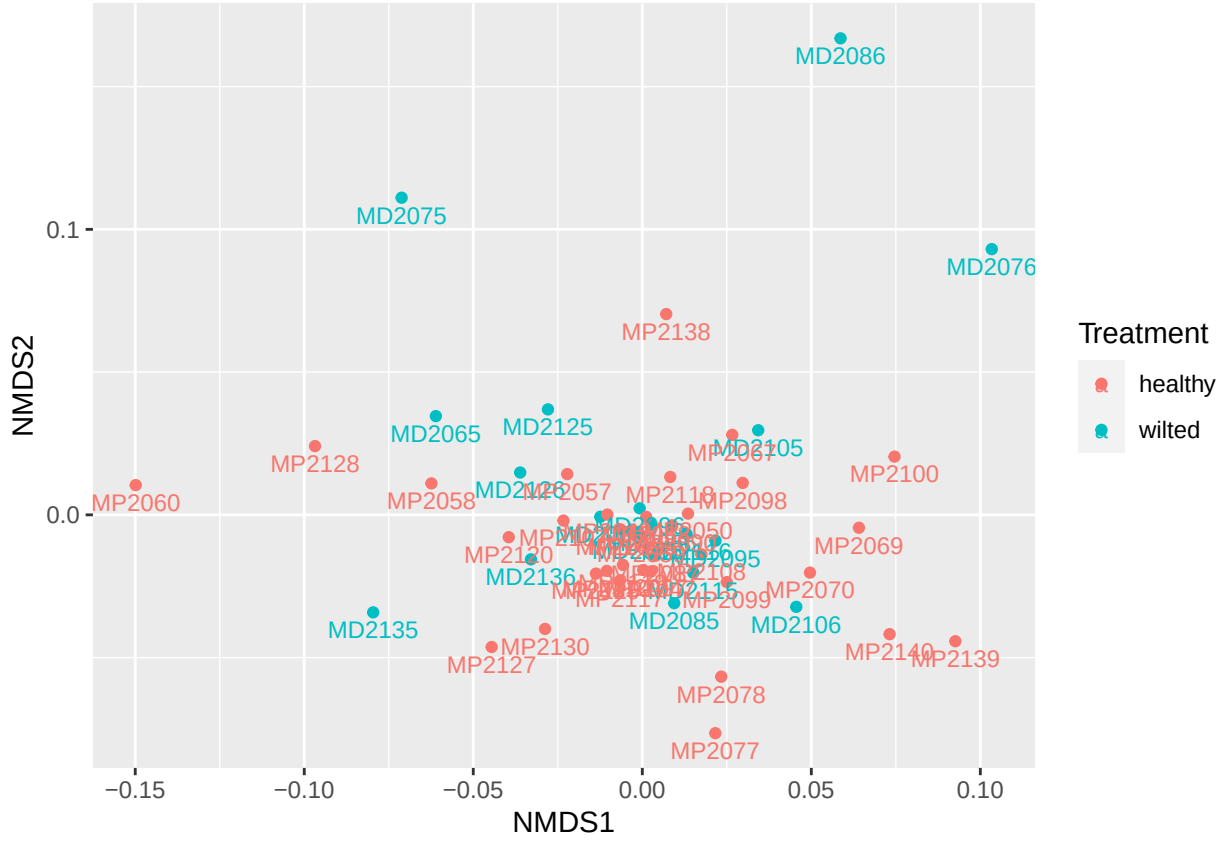



```
## se calcula la beta diversidad para Actinobacteria
## sacamos los porcentajes
percentages_Actinobacteria <- transform_sample_counts(glom_Actinobacteria, function(x) x*100 / sum(x) )
percentages_Actinobacteria_df <- psmelt(percentages_Actinobacteria)
meta_ord_Actinobacteria <- ordinate(physeq = percentages_Actinobacteria, method = "NMDS", distance = 'bi
```

```
## Wisconsin double standardization
## Run 0 stress 0.1297898
## Run 1 stress 0.1292155
## ... New best solution
## ... Procrustes: rmse 0.108649  max resid 0.6418969
## Run 2 stress 0.1345139
## Run 3 stress 0.157994
## Run 4 stress 0.1570595
## Run 5 stress 0.1270123
## ... New best solution
## ... Procrustes: rmse 0.1010982  max resid 0.6641636
## Run 6 stress 0.1292153
## Run 7 stress 0.1283594
## Run 8 stress 0.1598566
## Run 9 stress 0.127012
## ... New best solution
## ... Procrustes: rmse 0.000187527  max resid 0.0008474511
## ... Similar to previous best
## Run 10 stress 0.1292157
## Run 11 stress 0.157451
```

```
## Run 12 stress 0.1619944
## Run 13 stress 0.1270711
## ... Procrustes: rmse 0.003147405  max resid 0.01280114
## Run 14 stress 0.1297901
## Run 15 stress 0.1292159
## Run 16 stress 0.1292157
## Run 17 stress 0.1464534
## Run 18 stress 0.1292159
## Run 19 stress 0.1270121
## ... Procrustes: rmse 7.762792e-05  max resid 0.0003817968
## ... Similar to previous best
## Run 20 stress 0.1292155
## *** Best solution repeated 2 times
```

```
plot_beta_Actinobacteria <- plot_ordination(physeq = percentages_Actinobacteria, ordination = meta_ord_
  geom_text(mapping = aes(label = colnames(glom_Actinobacteria@otu_table@.Data)), size = 3, vjust = 1.5)
plot_beta_Actinobacteria
```



PRUEBA DE HIPOTESIS

```
OTU <- fresa_kraken_fil@otu_table@.Data
SAM <- fresa_kraken_fil@sam_data
```

```
## UNIR TABLA DE ABUNDANCIAS CON METADATA DE SANOS Y ENFERMOS

## Calculamos la diversidad Shannon

## Se usa la funcion diversidad del paquete vegan para calcular el indice Shannon
## Se realiza el dataframe del indice de Shannon
OTU <- t(OTU)
Shannon_OTU <- diversity(OTU, "shannon")
Shannon_OTU_df <- data.frame(sample=names(Shannon_OTU),value=Shannon_OTU,measure=rep("Shannon", length(S
total <-cbind(Shannon_OTU_df,SAM)

#mesiapor grupos
mu <- ddply(total, "Treatment", summarise, grp.mean=mean(value))

p<-ggplot(total, aes(x=value))+
  geom_histogram(color="pink",fill="black")+
  facet_grid(Treatment ~ .)
p+geom_vline(data=mu, aes(xintercept=grp.mean, color="red"),
             linetype="dashed")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```