

Diversidades alfa y beta en datos metagenomicos de shotgun de cultivo de fresa, a diferentes niveles taxonomicos

Camila Silva

2023-05-31

Observacion en diferentes niveles taxonomicos

```
library("phyloseq")
library("ggplot2")
library("igraph")

##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
## 
##     decompose, spectrum

## The following object is masked from 'package:base':
## 
##     union

library("readr")
library("patchwork")
library("vegan")

## Loading required package: permute

##
## Attaching package: 'permute'

## The following object is masked from 'package:igraph':
## 
##     permute

## Loading required package: lattice

## This is vegan 2.6-4

##
## Attaching package: 'vegan'
```

```

## The following object is masked from 'package:igraph':
##
##     diversity

library("GUniFrac")
library("kableExtra")

## Registered S3 method overwritten by 'httr':
##   method      from
##   print.response rmutil

library("RColorBrewer")

```

Cargado de datos originales

```

setwd("/home/camila/GIT/Tesis_Maestria/Data/fresa_solena/Data1")
outpath = "/home/camila/GIT/Tesis_Maestria/Analisis_Comparativo/Fresa_Solena/Results_img"

fresa_kraken <- import_biom("fresa_kraken.biom")
colnames(fresa_kraken@tax_table@.Data) <- c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus", "Species")
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/Data1/metadata.csv", header=TRUE)
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)

```

De aqui en adelante solo trabajaremos con los datos filtrados por calidad, ya que no vale la pena continuar con los datos que contiene muestras en ceros, ya que esto altera las visualizaciones.

Queremos explorar nuestras muestras a diferentes niveles taxonómicos específicos, empezando por una agrupación de los datos a nivel de reino y filo, Para agrupar todas las OTU que tienen la misma taxonomía en un determinado rango taxonómico, utilizaremos la función *tax_grom()*.

A nivel de Kingdom

```

percentages_glom_kingdom <- tax_grom(percentages_fil, taxrank = 'Kingdom')
percentages_glom_kingdom@tax_table@.Data

```

	Kingdom	Phylum	Class	Order	Family	Genus	Species
## 1883	"Bacteria"	NA	NA	NA	NA	NA	NA
## 5515	"Eukaryota"	NA	NA	NA	NA	NA	NA

En reino podemos ver que tenemos solamente dos grupos que son: Bacteria y Eucariota.

A nivel de Phylum

```
percentages_glom_phylum <- tax_glom(percentages_fil, taxrank = 'Phylum')
percentages_glom_phylum@tax_table@.Data
```

	Kingdom	Phylum	Class	Order	Family	Genus
## 1883	"Bacteria"	"Actinobacteria"	NA	NA	NA	NA
## 1406	"Bacteria"	"Firmicutes"	NA	NA	NA	NA
## 1298	"Bacteria"	"Deinococcus-Thermus"	NA	NA	NA	NA
## 1173025	"Bacteria"	"Cyanobacteria"	NA	NA	NA	NA
## 2057	"Bacteria"	"Chloroflexi"	NA	NA	NA	NA
## 1005039	"Bacteria"	"Armatimonadetes"	NA	NA	NA	NA
## 2132	"Bacteria"	"Tenericutes"	NA	NA	NA	NA
## 374	"Bacteria"	"Proteobacteria"	NA	NA	NA	NA
## 2528023	"Bacteria"	"Planctomyces"	NA	NA	NA	NA
## 1882749	"Bacteria"	"Verrucomicrobia"	NA	NA	NA	NA
## 1307763	"Bacteria"	"Kiritimatiellaeota"	NA	NA	NA	NA
## 389348	"Bacteria"	"Chlamydiae"	NA	NA	NA	NA
## 1930593	"Bacteria"	"Candidatus Omnitrophica"	NA	NA	NA	NA
## 354356	"Bacteria"	"Bacteroidetes"	NA	NA	NA	NA
## 274537	"Bacteria"	"Chlorobi"	NA	NA	NA	NA
## 1457365	"Bacteria"	"Balneolaeota"	NA	NA	NA	NA
## 1134405	"Bacteria"	"Ignavibacteriae"	NA	NA	NA	NA
## 861299	"Bacteria"	"Gemmatimonadetes"	NA	NA	NA	NA
## 833	"Bacteria"	"Fibrobacteres"	NA	NA	NA	NA
## 456827	"Bacteria"	"Candidatus Cloacimonetes"	NA	NA	NA	NA
## 2802971	"Bacteria"	"Acidobacteria"	NA	NA	NA	NA
## 42253	"Bacteria"	"Nitrospirae"	NA	NA	NA	NA
## 154	"Bacteria"	"Spirochaetes"	NA	NA	NA	NA
## 638849	"Bacteria"	"Synergistetes"	NA	NA	NA	NA
## 2026885	"Bacteria"	"Candidatus Bipolaricaulota"	NA	NA	NA	NA
## 1332188	"Bacteria"	"Candidatus Saccharibacteria"	NA	NA	NA	NA
## 2735562	"Bacteria"	"Candidatus Absconditabacteria"	NA	NA	NA	NA
## 2335	"Bacteria"	"Thermotogae"	NA	NA	NA	NA
## 412593	"Bacteria"	"Aquificae"	NA	NA	NA	NA
## 2047767	"Bacteria"	"Thermodesulfobacteria"	NA	NA	NA	NA
## 139438	"Bacteria"	"Deferribacteres"	NA	NA	NA	NA
## 848	"Bacteria"	"Fusobacteria"	NA	NA	NA	NA
## 936456	"Bacteria"	"Chrysioigenetes"	NA	NA	NA	NA
## 187145	"Bacteria"	"Calditrichaeota"	NA	NA	NA	NA
## 423605	"Bacteria"	"Elusimicrobia"	NA	NA	NA	NA
## 693075	"Bacteria"	"Caldiserica"	NA	NA	NA	NA
## 35786	"Bacteria"	"Coprothermobacterota"	NA	NA	NA	NA
## 2847778	"Bacteria"	"Atribacterota"	NA	NA	NA	NA
## 13	"Bacteria"	"Dictyoglomi"	NA	NA	NA	NA
## 5515	"Eukaryota"	"Ascomycota"	NA	NA	NA	NA
## 84751	"Eukaryota"	"Basidiomycota"	NA	NA	NA	NA
## 6035	"Eukaryota"	"Microsporidia"	NA	NA	NA	NA
## 4785	"Eukaryota"	"Oomycota"	NA	NA	NA	NA
##	Species					
## 1883	NA					
## 1406	NA					
## 1298	NA					

```

## 1173025 NA
## 2057 NA
## 1005039 NA
## 2132 NA
## 374 NA
## 2528023 NA
## 1882749 NA
## 1307763 NA
## 389348 NA
## 1930593 NA
## 354356 NA
## 274537 NA
## 1457365 NA
## 1134405 NA
## 861299 NA
## 833 NA
## 456827 NA
## 2802971 NA
## 42253 NA
## 154 NA
## 638849 NA
## 2026885 NA
## 1332188 NA
## 2735562 NA
## 2335 NA
## 412593 NA
## 2047767 NA
## 139438 NA
## 848 NA
## 936456 NA
## 187145 NA
## 423605 NA
## 693075 NA
## 35786 NA
## 2847778 NA
## 13 NA
## 5515 NA
## 84751 NA
## 6035 NA
## 4785 NA

```

A nivel de filo, ya podemos ver mas variedad en nuestras muestras, teniendo como mayoria los filo pertenecientes a bacteria.

Usaremos la función de phyloseq ***psmelt()***, que fusiona objetos de phyloseq en un data.frame para manipularlos con paquetes como ggplot2 y vegan. Con esta creamos un dataframe con los porcentajes de “phylum”, este nos dará los porcentajes de cada OTU, perteneciente a cada muestra, a nivel taxonomico de filo.

```

percentages_df_phylum <- psmelt(percentages_glom_phylum)
str(percentages_df_phylum)

```

```

## 'data.frame': 2279 obs. of 7 variables:
## $ OTU      : chr "1883" "1883" "1883" "1883" ...
## $ Sample   : chr "MP2060" "MP2058" "MP2059" "MP2049" ...

```

```

## $ Abundance: num 56.3 52.7 52.7 52.7 52.4 ...
## $ Treatment: chr "healthy" "healthy" "healthy" "healthy" ...
## $ Samples : chr "MP2060" "MP2058" "MP2059" "MP2049" ...
## $ Kingdom : chr "Bacteria" "Bacteria" "Bacteria" "Bacteria" ...
## $ Phylum : chr "Actinobacteria" "Actinobacteria" "Actinobacteria" "Actinobacteria" ...

head(percentages_df_phylum)

##      OTU Sample Abundance Treatment Samples Kingdom          Phylum
## 738 1883 56.28736   healthy MP2060 Bacteria Actinobacteria
## 709 1883 52.71057   healthy MP2058 Bacteria Actinobacteria
## 726 1883 52.70278   healthy MP2059 Bacteria Actinobacteria
## 701 1883 52.70204   healthy MP2049 Bacteria Actinobacteria
## 690 1883 52.35707    wilted MD2055 Bacteria Actinobacteria
## 725 1883 51.64839   healthy MP2120 Bacteria Actinobacteria

```

Ahora, creamos un data frame con los datos originales. Esta estructura nos ayudará a comparar la abundancia absoluta con la relativa

```

absolute_glm_phylum <- tax_glm(physeq = fresa_kraken_fil, taxrank = "Phylum")
absolute_df_phylum <- psmelt(absolute_glm_phylum)
str(absolute_df_phylum)

```

```

## 'data.frame': 2279 obs. of 7 variables:
## $ OTU      : chr "1883" "1883" "1883" "374" ...
## $ Sample    : chr "MP2058" "MP2087" "MP2068" "MP2068" ...
## $ Abundance: num 11028897 10342054 9967507 9190530 8805431 ...
## $ Treatment: chr "healthy" "healthy" "healthy" "healthy" ...
## $ Samples   : chr "MP2058" "MP2087" "MP2068" "MP2068" ...
## $ Kingdom   : chr "Bacteria" "Bacteria" "Bacteria" "Bacteria" ...
## $ Phylum    : chr "Actinobacteria" "Actinobacteria" "Actinobacteria" "Proteobacteria" ...

```

```

absolute_df_phylum$Phylum <- as.factor(absolute_df_phylum$Phylum)
phylum_colors_abs <- colorRampPalette(brewer.pal(8,"Dark2")) (length(levels(absolute_df_phylum$Phylum)))

```

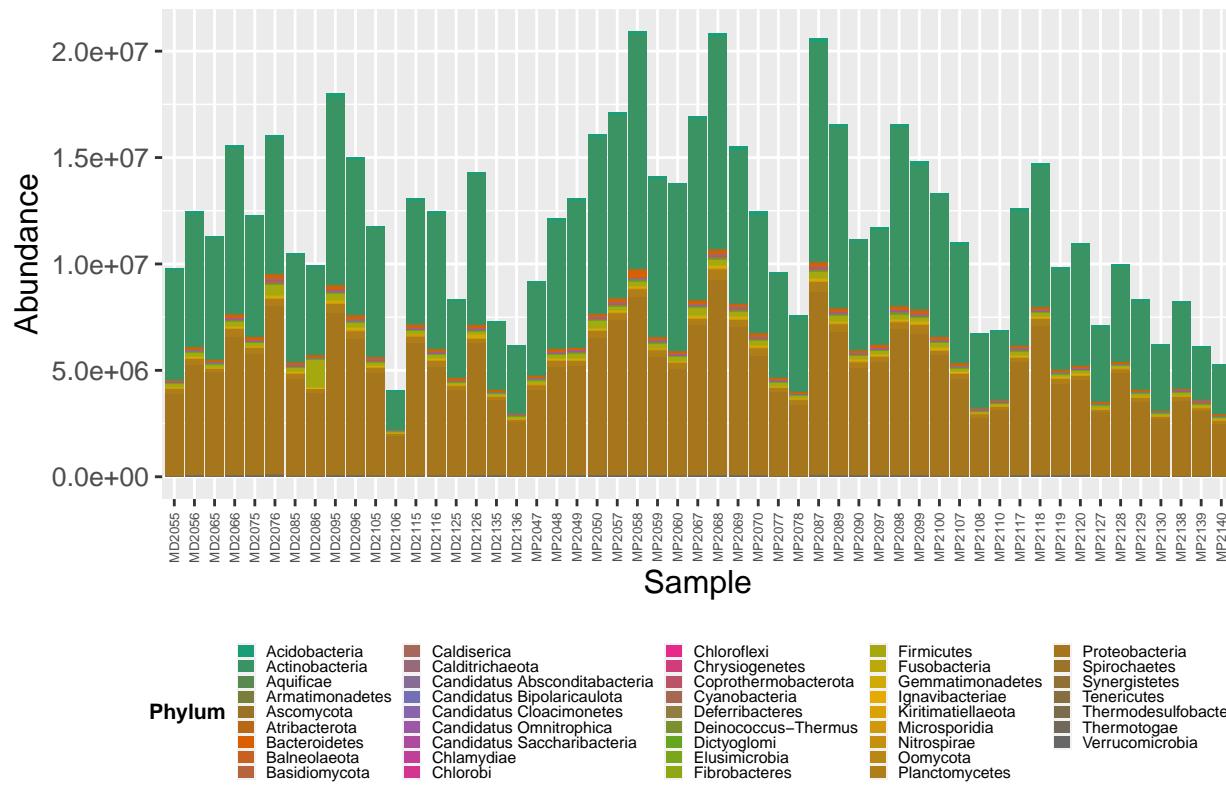
```

absolute_plot <- ggplot(data= absolute_df_phylum, aes(x=Sample, y=Abundance, fill=Phylum))+
  geom_bar(aes(), stat="identity", position="stack")+
  scale_fill_manual(values = phylum_colors_abs) +
  labs(title = "Abundance", x='Sample', y='Abundance') +
  theme(legend.key.size = unit(0.2, "cm"),
        legend.key.width = unit(0.25,"cm"),
        legend.position = "bottom",
        legend.direction = "horizontal",
        legend.title=element_text(size=8, face = "bold"),
        legend.text=element_text(size=6),
        text = element_text(size=12),
        axis.text.x = element_text(angle=90, size=5, hjust=1, vjust=0.5))

```

```
absolute_plot
```

Abundance

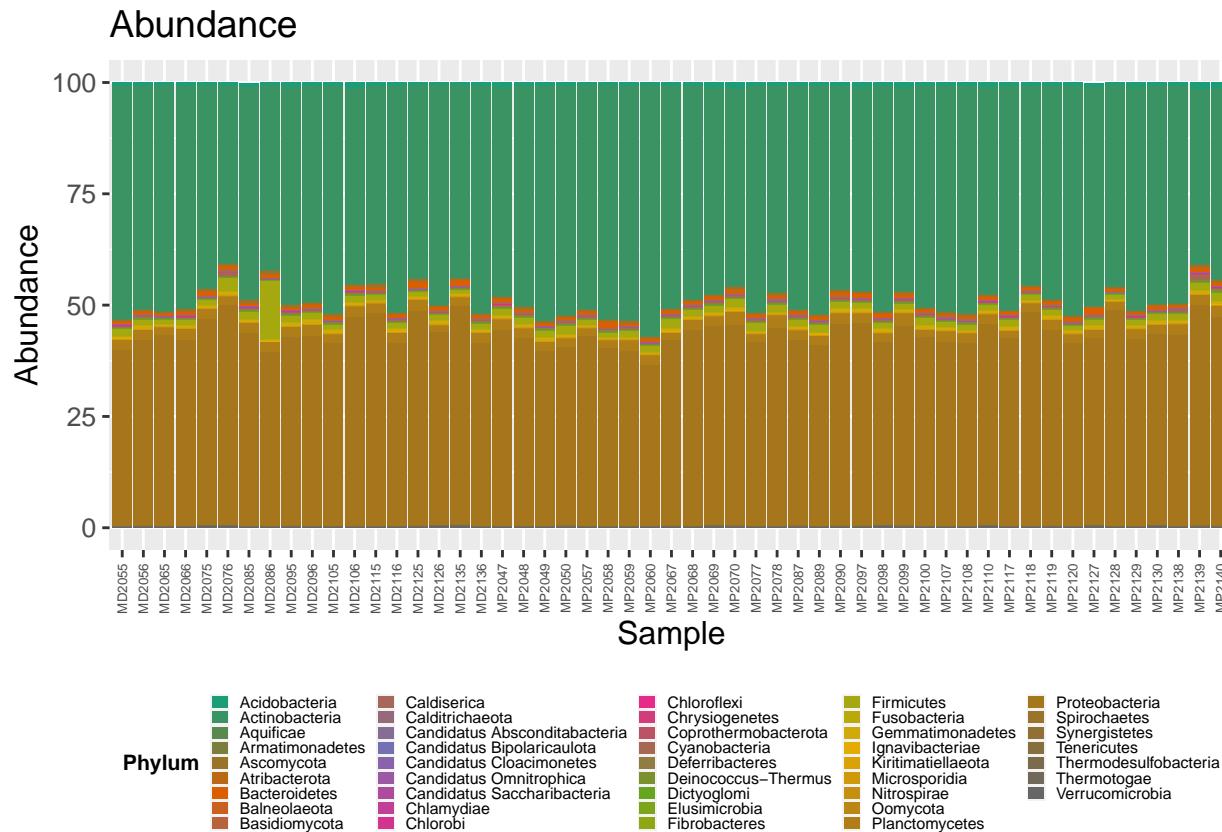


```

percentages_df_phylum$Phylum <- as.factor(percentages_df_phylum$Phylum)
phylum_colors_rel <- colorRampPalette(brewer.pal(8, "Dark2")) (length(levels(percentages_df_phylum$Phylum))
relative_plot <- ggplot(data=percentages_df_phylum, aes(x=Sample, y=Abundance, fill=Phylum)) +
  geom_bar(aes(), stat="identity", position="stack")+
  scale_fill_manual(values = phylum_colors_rel) +
  labs(title = "Abundance", x='Sample', y='Abundance') +
  theme(legend.key.size = unit(0.2, "cm"),
        legend.key.width = unit(0.25,"cm"),
        legend.position = "bottom",
        legend.direction = "horizontal",
        legend.title=element_text(size=8, face = "bold"),
        legend.text=element_text(size=6),
        text = element_text(size=12),
        axis.text.x = element_text(angle=90, size=5, hjust=1, vjust=0.5))

relative_plot

```

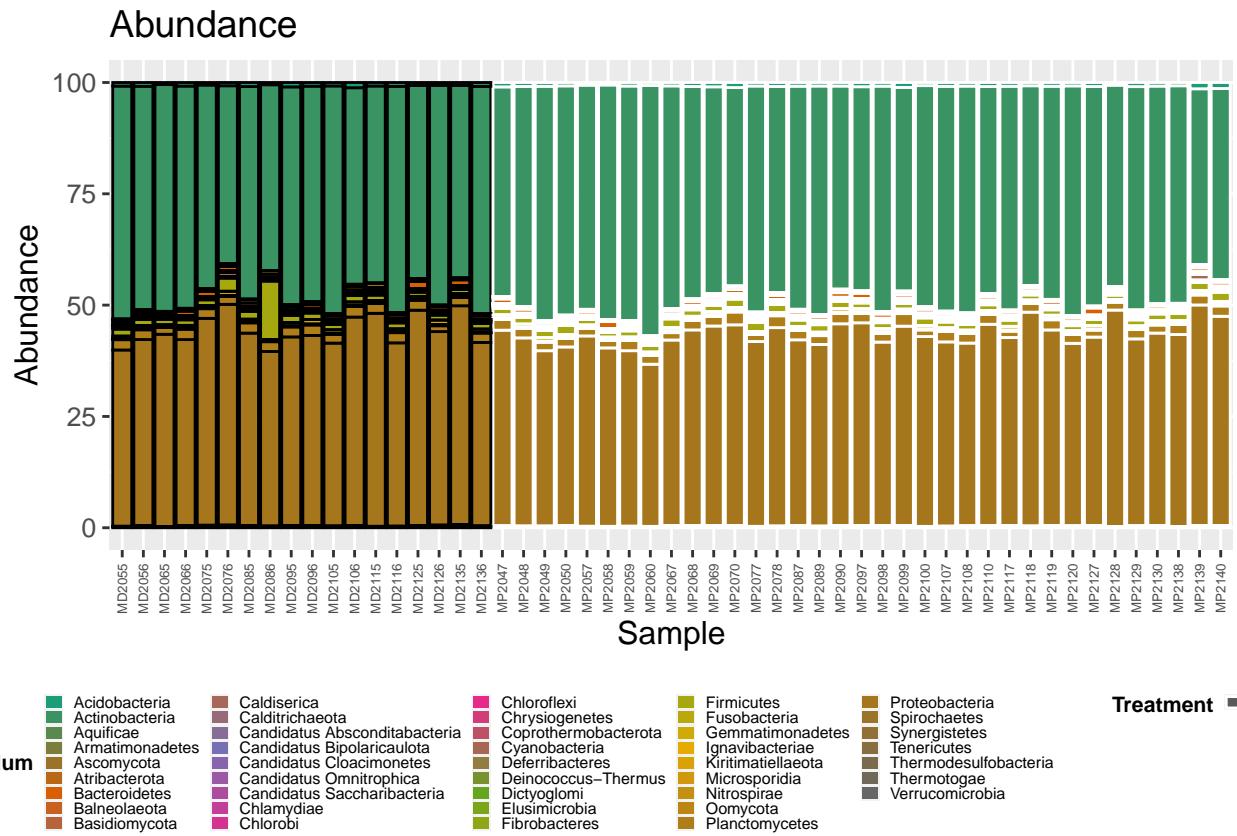


Podemos ver algunas diferencias entre los diferentes filo, sin embargo, se nota que tenemos demasiados taxones para distinguir adecuadamente el color de cada uno, a menos que se tenga una abundancia muy grande.

Como nuestro objetivo es diferenciar entre muestras sanas y enfermas, le agregaremos una diferenciación entre la variable ‘Treatment’, con color blanco las muestras sanas y negras las muestras enfermas.

```
percentages_df_phylum$Phylum <- as.factor(percentages_df_phylum$Phylum)
phylum_colors_rel <- colorRampPalette(brewer.pal(8, "Dark2")) (length(levels(percentages_df_phylum$Phylum))
relative_plot <- ggplot(data=percentages_df_phylum, aes(x=Sample, y=Abundance, fill=Phylum, color=Treatment))
  scale_colour_manual(values=c('white','black')) +
  geom_bar(aes(), stat="identity", position="stack")+
  scale_fill_manual(values = phylum_colors_rel) +
  labs(title = "Abundance", x='Sample', y='Abundance') +
  theme(legend.key.size = unit(0.2, "cm"),
        legend.key.width = unit(0.25,"cm"),
        legend.position = "bottom",
        legend.direction = "horizontal",
        legend.title=element_text(size=8, face = "bold"),
        legend.text=element_text(size=6),
        text = element_text(size=12),
        axis.text.x = element_text(angle=90, size=5, hjust=1, vjust=0.5))

relative_plot
```



Igualmente tenemos demasiados taxones para ver una distinción entre las muestras, más adelante crearemos subconjuntos mas pequeños para tener una mejor observación de nuestros datos.

Por otro lado, podemos usar un comando llamado “unique()” para explorar cuántos filos y reinos tenemos.

```
unique(fresa_kraken_fil@tax_table@Data[, "Kingdom"])
```

```
## [1] "Bacteria" "Eukaryota"
```

```
unique(fresa_kraken_fil@tax_table@Data[, "Phylum"])
```

```
## [1] "Actinobacteria" "Firmicutes"
## [3] "Deinococcus-Thermus" "Cyanobacteria"
## [5] "Chloroflexi" "Armatimonadetes"
## [7] "Tenericutes" "Proteobacteria"
## [9] "Planctomycetes" "Verrucomicrobia"
## [11] "Kiritimatiellaeota" "Chlamydiae"
## [13] "Candidatus Omnitrophica" "Bacteroidetes"
## [15] "Chlorobi" "Balneolaeota"
## [17] "Ignavibacteriae" "Gemmatimonadetes"
## [19] "Fibrobacteres" "Candidatus Cloacimonetes"
## [21] "Acidobacteria" "Nitrospirae"
## [23] "Spirochaetes" "Synergistetes"
## [25] "Candidatus Bipolaricaulota" "Candidatus Saccharibacteria"
## [27] "" "Candidatus Absconditabacteria"
```

```

## [29] "Thermotogae"           "Aequificae"
## [31] "Thermodesulfobacteria"  "Deferrribacteres"
## [33] "Fusobacteria"          "Chrysogenetes"
## [35] "Calditrichaeota"        "Elusimicrobia"
## [37] "Caldisericia"          "Coprothermobacterota"
## [39] "Atribacterota"          "Dictyoglomi"
## [41] "Ascomycota"             "Basidiomycota"
## [43] "Microsporidia"          "Oomycota"

```

Ahora, podemos ver cuantos “Eukaryota” tenemos en “Kingdom”.

```
sum(fresa_kraken_fil@tax_table@Data[, "Kingdom"] == "Eukaryota")
```

```
## [1] 181
```

y cuantos “Bacteria”

```
sum(fresa_kraken_fil@tax_table@Data[, "Kingdom"] == "Bacteria")
```

```
## [1] 8822
```

Verificando así lo que notamos anteriormente con la observación de los subconjuntos aglomerados, tenemos mayor cantidad de muestras clasificadas como bacterias, que las clasificadas como eucariota.

Diversidad Beta

Como se menciono anteriormente el objetivo de este documento es explorar las visualizaciones de: barras, alfa diversidad y beta diversidad, para subconjuntos seleccionados de nuestros datos.

Veremos aquí solo el reino Eucariota:

```
merge_Eukaryota <- subset_taxa(fresa_kraken_fil, Kingdom == "Eukaryota")
```

sacamos las abundancias relativas,

```
percentages_Eukaryota <- transform_sample_counts(merge_Eukaryota, function(x) x*100 / sum(x) )
percentages_Eukaryota_df <- psmelt(percentages_Eukaryota)
```

Beta diversidad de Eukaryota

```

meta_ord_Eukaryota <- ordinate(physeq = percentages_Eukaryota, method = "NMDS", distance = "bray")

## Wisconsin double standardization
## Run 0 stress 0.1341389
## Run 1 stress 0.1540356
## Run 2 stress 0.1742741
## Run 3 stress 0.1394567
## Run 4 stress 0.1536393

```

```

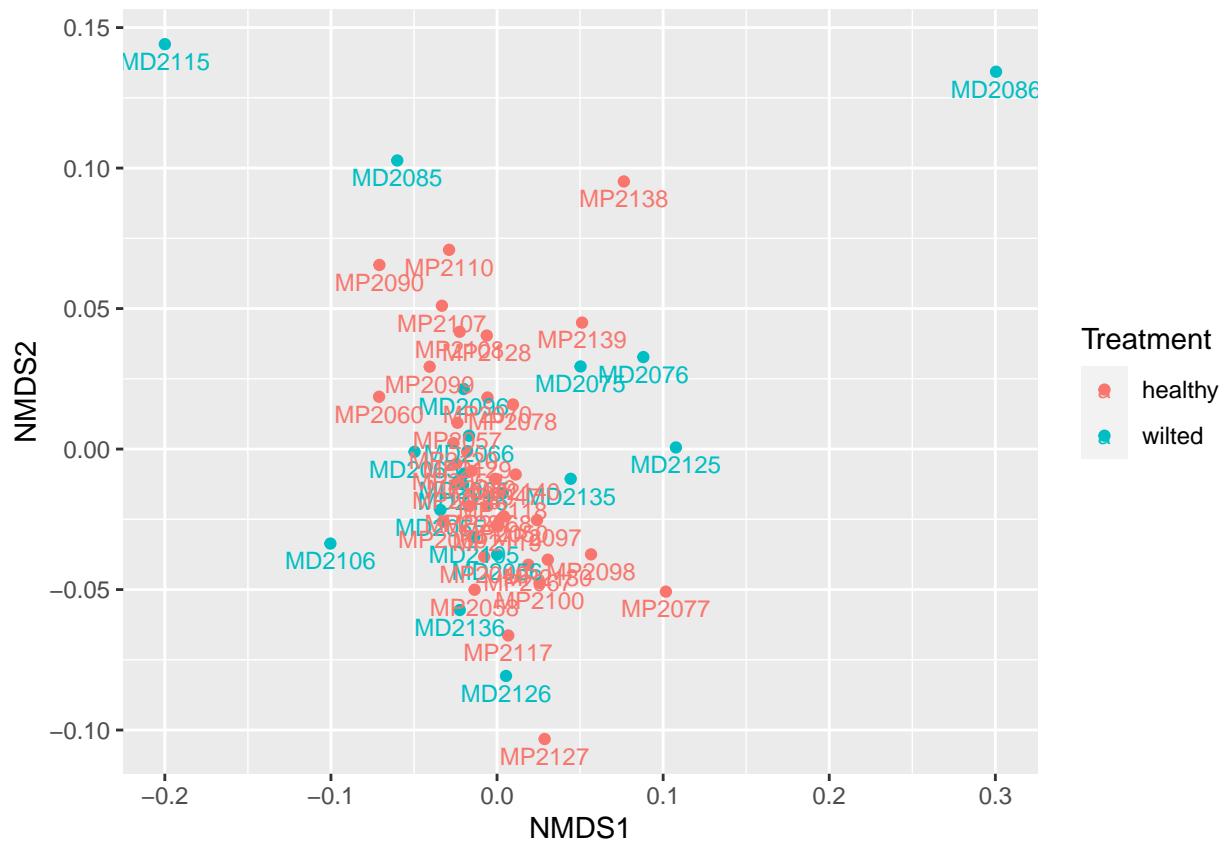
## Run 5 stress 0.1519372
## Run 6 stress 0.1667878
## Run 7 stress 0.1359151
## Run 8 stress 0.1507837
## Run 9 stress 0.1341966
## ... Procrustes: rmse 0.01189101 max resid 0.08040136
## Run 10 stress 0.1397088
## Run 11 stress 0.1537812
## Run 12 stress 0.1516085
## Run 13 stress 0.1611188
## Run 14 stress 0.1689251
## Run 15 stress 0.1466938
## Run 16 stress 0.1341406
## ... Procrustes: rmse 0.001429061 max resid 0.008314108
## ... Similar to previous best
## Run 17 stress 0.1649096
## Run 18 stress 0.1536124
## Run 19 stress 0.1628183
## Run 20 stress 0.139464
## *** Best solution repeated 1 times

```

```

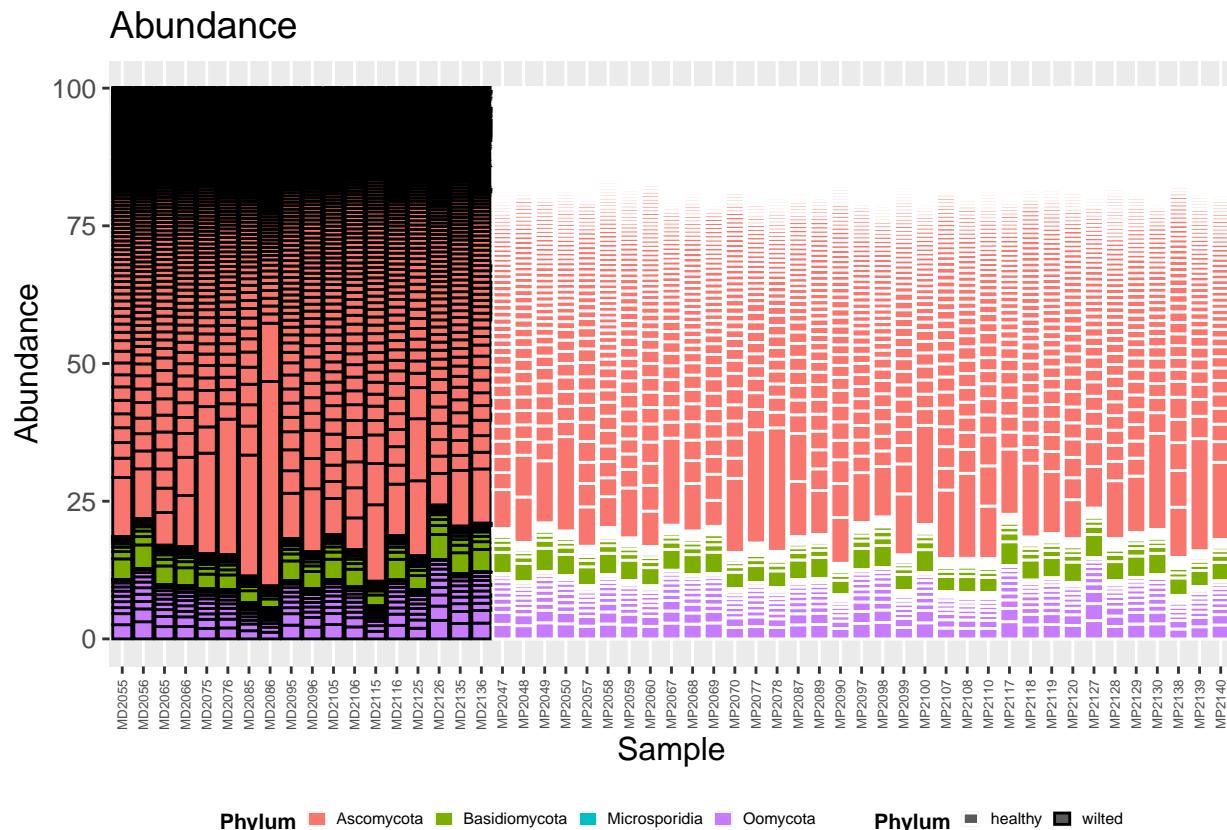
plot_ordination(physeq = percentages_Eukaryota, ordination = meta_ord_Eukaryota, color = "Treatment") +
  geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@Data)), size = 3, vjust = 1.5)

```



Eukaryota por Phylum

```
ggplot(data= percentages_Eukaryota_df, aes(x=Sample, y=Abundance, fill=Phylum, color=Treatment)) +
  scale_colour_manual(values=c('white','black')) +
  geom_bar(aes(), stat="identity", position="stack") +
  labs(title = "Abundance", x='Sample', y='Abundance', color = 'Phylum') +
  theme(legend.key.size = unit(0.2, "cm"),
        legend.key.width = unit(0.25,"cm"),
        legend.position = "bottom",
        legend.direction = "horizontal",
        legend.title=element_text(size=8, face = "bold"),
        legend.text=element_text(size=6),
        text = element_text(size=12),
        axis.text.x = element_text(angle=90, size=5, hjust=1, vjust=0.5))
```



```
merge_Eukaryota_Phylum<-tax_glom(merge_Eukaryota,taxrank="Phylum")
```

sacamos las abundancias relativas

```
percentages_Eukaryota_Phylum <- transform_sample_counts(merge_Eukaryota_Phylum, function(x) x*100 / sum(percentages_Eukaryota_Phylum))
percentages_Eukaryota_Phylum_df <- psmelt(percentages_Eukaryota_Phylum)
meta_ord_Eukaryota_Phylum <- ordinate(physeq = percentages_Eukaryota_Phylum, method = "NMDS", distance =
```

```
## Square root transformation
```

```

## Wisconsin double standardization
## Run 0 stress 0.01148681
## Run 1 stress 9.646159e-05
## ... New best solution
## ... Procrustes: rmse 0.03465767 max resid 0.0908727
## Run 2 stress 0.002640858
## Run 3 stress 9.756458e-05
## ... Procrustes: rmse 0.0001053051 max resid 0.0002353465
## ... Similar to previous best
## Run 4 stress 5.897372e-05
## ... New best solution
## ... Procrustes: rmse 3.531445e-05 max resid 9.339526e-05
## ... Similar to previous best
## Run 5 stress 9.785774e-05
## ... Procrustes: rmse 0.0001671586 max resid 0.0004562006
## ... Similar to previous best
## Run 6 stress 0.0006662634
## Run 7 stress 6.965347e-05
## ... Procrustes: rmse 3.53283e-05 max resid 7.52576e-05
## ... Similar to previous best
## Run 8 stress 9.48467e-05
## ... Procrustes: rmse 9.185617e-05 max resid 0.0002659007
## ... Similar to previous best
## Run 9 stress 8.877074e-05
## ... Procrustes: rmse 3.089355e-05 max resid 8.285961e-05
## ... Similar to previous best
## Run 10 stress 0.0002199881
## ... Procrustes: rmse 0.0006517098 max resid 0.001723586
## ... Similar to previous best
## Run 11 stress 7.339722e-05
## ... Procrustes: rmse 3.191856e-05 max resid 7.162121e-05
## ... Similar to previous best
## Run 12 stress 9.670656e-05
## ... Procrustes: rmse 0.0001671632 max resid 0.0004528372
## ... Similar to previous best
## Run 13 stress 0.0001083394
## ... Procrustes: rmse 0.0003196863 max resid 0.00084982
## ... Similar to previous best
## Run 14 stress 0.0001002617
## ... Procrustes: rmse 0.0002926547 max resid 0.0007819431
## ... Similar to previous best
## Run 15 stress 7.072036e-05
## ... Procrustes: rmse 3.22457e-05 max resid 8.654915e-05
## ... Similar to previous best
## Run 16 stress 8.680872e-05
## ... Procrustes: rmse 0.0001073028 max resid 0.0002930709
## ... Similar to previous best
## Run 17 stress 9.712712e-05
## ... Procrustes: rmse 0.0001205651 max resid 0.0003290698
## ... Similar to previous best
## Run 18 stress 8.070774e-05
## ... Procrustes: rmse 3.517541e-05 max resid 9.123802e-05
## ... Similar to previous best
## Run 19 stress 9.084046e-05

```

```

## ... Procrustes: rmse 5.21782e-05 max resid 0.0001038878
## ... Similar to previous best
## Run 20 stress 9.943326e-05
## ... Procrustes: rmse 0.0001062638 max resid 0.0003068705
## ... Similar to previous best
## *** Best solution repeated 16 times

```

```

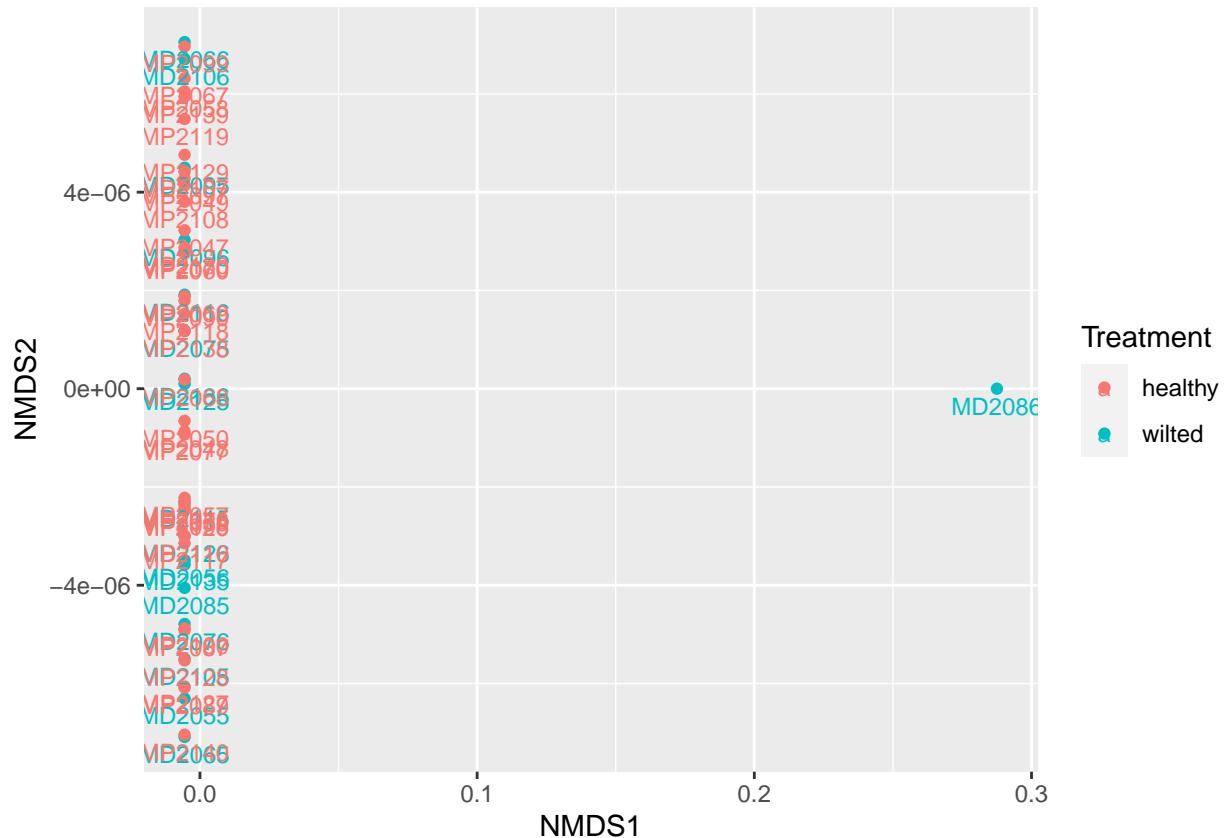
## Warning in metaMDS(veganifyOTU(physeq), distance, ...): stress is (nearly)
## zero: you may have insufficient data

```

```

plot_ordination(physeq = percentages_Eukaryota_Phylum, ordination = meta_ord_Eukaryota_Phylum, color =
  geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@Data))), size = 3, vjust = 1.5)

```



```

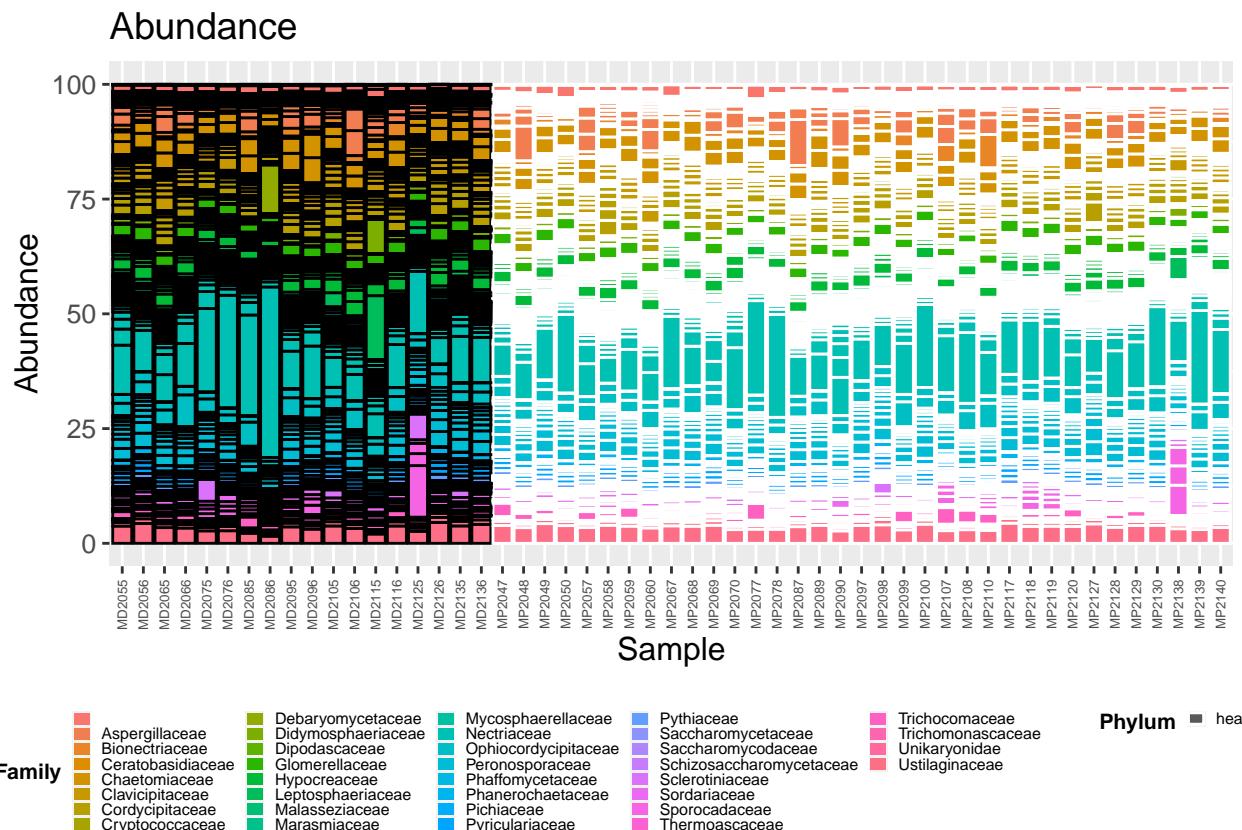
## Eukaryota por Family

```

```

ggplot(data= percentages_Eukaryota_df, aes(x=Sample, y=Abundance, fill=Family, color=Treatment)) +
  scale_colour_manual(values=c('white','black')) +
  geom_bar(aes(), stat="identity", position="stack") +
  labs(title = "Abundance", x='Sample', y='Abundance', color = 'Phylum') +
  theme(legend.key.size = unit(0.2, "cm"),
        legend.key.width = unit(0.25,"cm"),
        legend.position = "bottom",
        legend.direction = "horizontal",
        legend.title=element_text(size=8, face = "bold"),
        legend.text=element_text(size=6),
        text = element_text(size=12),
        axis.text.x = element_text(angle=90, size=5, hjust=1, vjust=0.5))

```



```
merge_Eukaryota_Family <- tax_glom(merge_Eukaryota, taxrank = "Family")
```

sacamos las abundancias relativas

```
percentages_Eukaryota_Family <- transform_sample_counts(merge_Eukaryota_Family, function(x) x*100 / sum(x))
percentages_Eukaryota_Family_df <- psmelt(percentages_Eukaryota_Family)
meta_ord_Eukaryota_Family <- ordinate(physeq = percentages_Eukaryota_Family, method = "NMDS", distance = "euclidean")
```

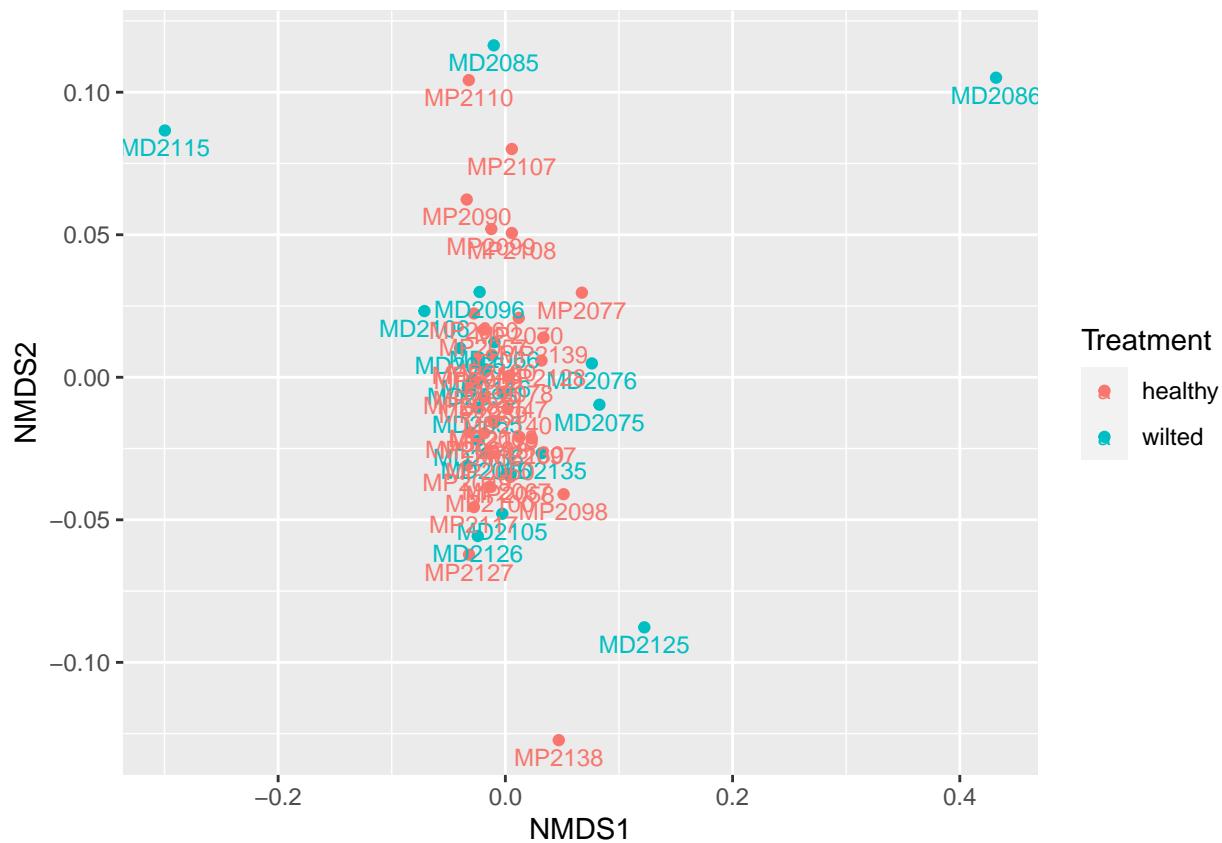
```
## Wisconsin double standardization
## Run 0 stress 0.1089081
## Run 1 stress 0.1088971
## ... New best solution
## ... Procrustes: rmse 0.002665604 max resid 0.009156696
## ... Similar to previous best
## Run 2 stress 0.1089302
## ... Procrustes: rmse 0.00859201 max resid 0.05460502
## Run 3 stress 0.1089298
## ... Procrustes: rmse 0.008636386 max resid 0.05489538
## Run 4 stress 0.1263362
## Run 5 stress 0.1089185
## ... Procrustes: rmse 0.002203581 max resid 0.008824426
## ... Similar to previous best
## Run 6 stress 0.1088894
```

```

## ... New best solution
## ... Procrustes: rmse 0.001754418 max resid 0.009032942
## ... Similar to previous best
## Run 7 stress 0.126382
## Run 8 stress 0.1089291
## ... Procrustes: rmse 0.008417777 max resid 0.054603
## Run 9 stress 0.1089488
## ... Procrustes: rmse 0.008117131 max resid 0.05416036
## Run 10 stress 0.1089295
## ... Procrustes: rmse 0.00843432 max resid 0.05463096
## Run 11 stress 0.1091134
## ... Procrustes: rmse 0.009817098 max resid 0.05407622
## Run 12 stress 0.108929
## ... Procrustes: rmse 0.008411416 max resid 0.0545915
## Run 13 stress 0.1089304
## ... Procrustes: rmse 0.008470212 max resid 0.05468782
## Run 14 stress 0.108888
## ... New best solution
## ... Procrustes: rmse 0.000274177 max resid 0.001569151
## ... Similar to previous best
## Run 15 stress 0.1091131
## ... Procrustes: rmse 0.009837527 max resid 0.05393734
## Run 16 stress 0.1089485
## ... Procrustes: rmse 0.008087662 max resid 0.05416526
## Run 17 stress 0.1089297
## ... Procrustes: rmse 0.003901092 max resid 0.02006705
## Run 18 stress 0.1089442
## ... Procrustes: rmse 0.008797539 max resid 0.05506747
## Run 19 stress 0.1091133
## ... Procrustes: rmse 0.00984114 max resid 0.05391384
## Run 20 stress 0.1091139
## ... Procrustes: rmse 0.00983247 max resid 0.05385692
## *** Best solution repeated 1 times

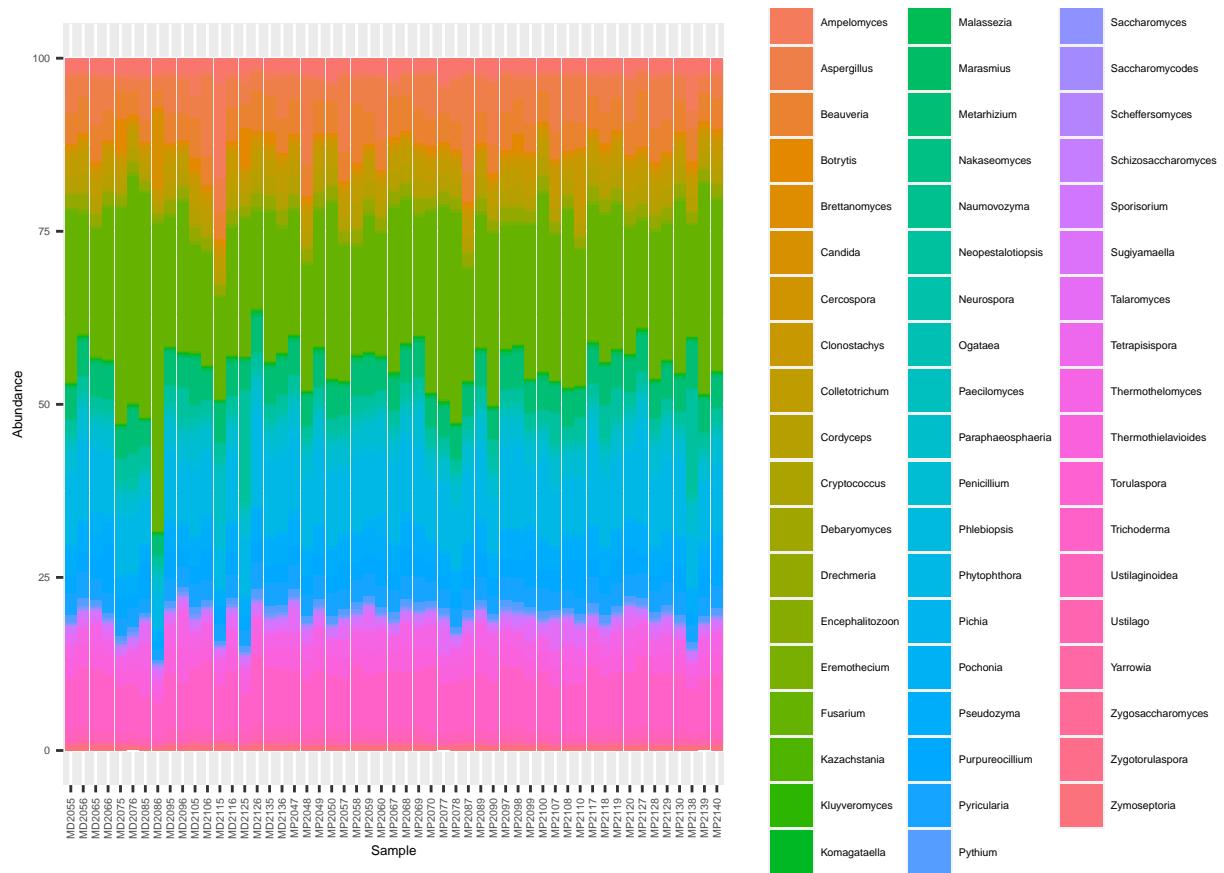
plot_ordination(physeq = percentages_Eukaryota_Family, ordination = meta_ord_Eukaryota_Family, color =
  geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@Data))), size = 3, vjust = 1.5)

```



Eukaryota por Genus

```
ggplot(data= percentages_Eukaryota_df, aes(x=Sample, y=Abundance, fill=Genus))+  
  geom_bar(aes(), stat="identity", position="stack") +  
  theme(text = element_text(size = 5),  
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



```
merge_Eukaryota_Genus <- tax_glom(merge_Eukaryota, taxrank = "Genus")
```

sacamos las abundancias relativas

```
percentages_Eukaryota_Genus <- transform_sample_counts(merge_Eukaryota_Genus, function(x) x*100 / sum(x))
percentages_Eukaryota_Genus_df <- psmelt(percentages_Eukaryota_Genus)
meta_ord_Eukaryota_Genus <- ordinate(physeq = percentages_Eukaryota_Genus, method = "NMDS", distance =
```

```
## Wisconsin double standardization
## Run 0 stress 0.110709
## Run 1 stress 0.1119814
## Run 2 stress 0.1119823
## Run 3 stress 0.11185
## Run 4 stress 0.1107117
## ... Procrustes: rmse 0.001624258 max resid 0.009286264
## ... Similar to previous best
## Run 5 stress 0.1235033
## Run 6 stress 0.11071
## ... Procrustes: rmse 0.001283405 max resid 0.007352181
## ... Similar to previous best
## Run 7 stress 0.1235026
## Run 8 stress 0.1235114
## Run 9 stress 0.1236301
## Run 10 stress 0.1223832
## Run 11 stress 0.1112554
```

```

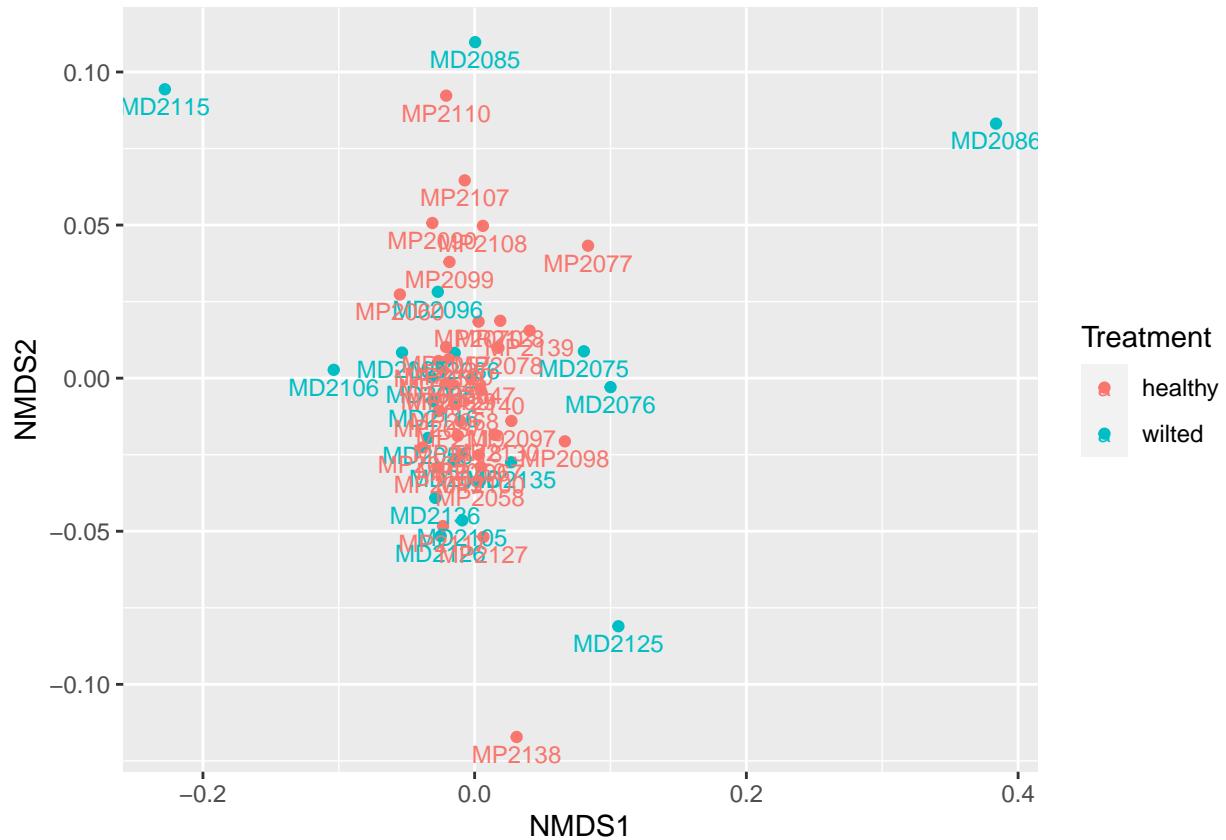
## Run 12 stress 0.1108472
## ... Procrustes: rmse 0.007457018 max resid 0.04412116
## Run 13 stress 0.1107095
## ... Procrustes: rmse 0.001138336 max resid 0.006530864
## ... Similar to previous best
## Run 14 stress 0.1107094
## ... Procrustes: rmse 0.001069403 max resid 0.005655563
## ... Similar to previous best
## Run 15 stress 0.1119819
## Run 16 stress 0.1235023
## Run 17 stress 0.1107087
## ... New best solution
## ... Procrustes: rmse 0.0001401562 max resid 0.0008062254
## ... Similar to previous best
## Run 18 stress 0.1107088
## ... Procrustes: rmse 5.850425e-05 max resid 0.0003253864
## ... Similar to previous best
## Run 19 stress 0.1119814
## Run 20 stress 0.1223836
## *** Best solution repeated 2 times

```

```

plot_ordination(physeq = percentages_Eukaryota_Genus, ordination = meta_ord_Eukaryota_Genus, color = "Treatment",
geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@Data)), size = 3, vjust = 1.5)

```



tomando diferentes porcentajes de abundancia

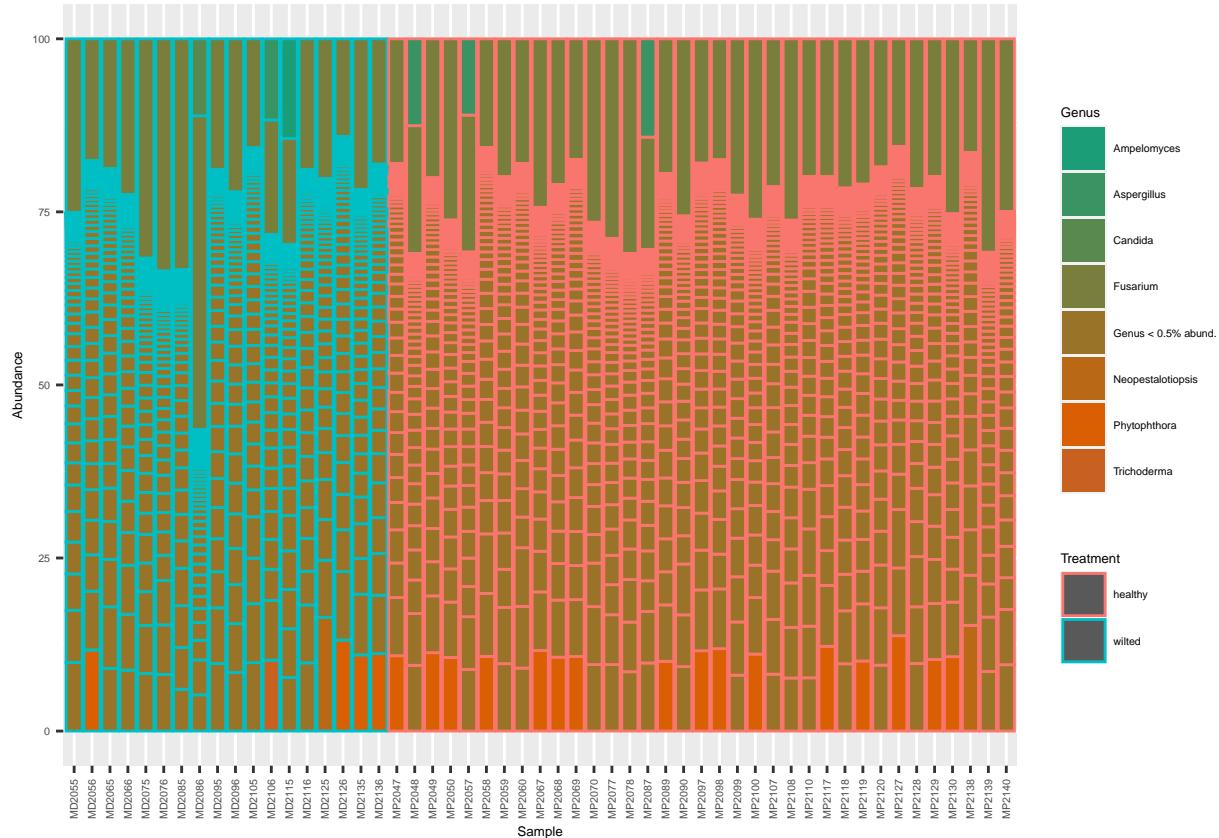
```

percentages_Eukaryota_Genus_df$Genus[percentages_Eukaryota_Genus_df$Abundance < 10.0] <- "Genus < 0.5% ab."
percentages_Eukaryota_Genus_df$Genus <- as.factor(percentages_Eukaryota_Genus_df$Genus)

genus_colors_rel <- colorRampPalette(brewer.pal(8, "Dark2")) (length(levels(percentages_Eukaryota_Genus_df$Genus))
relative_plot <- ggplot(data=percentages_Eukaryota_Genus_df, aes(x=Sample, y=Abundance, fill=Genus ,color=Genus))
  geom_bar(aes(), stat="identity", position="stack") +
  scale_fill_manual(values = phylum_colors_rel) +
  theme(text = element_text(size = 5),
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))

relative_plot

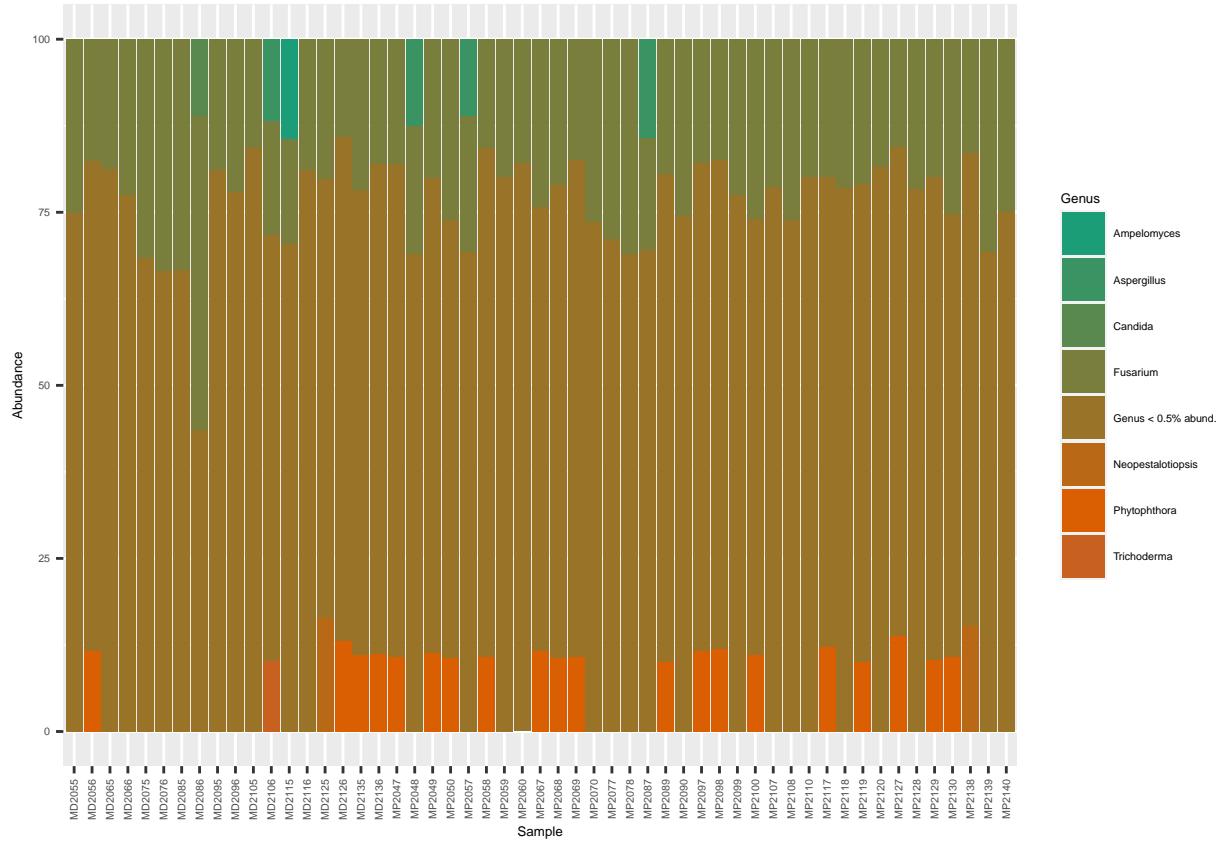
```



```

relative_plot <- ggplot(data=percentages_Eukaryota_Genus_df, aes(x=Sample, y=Abundance, fill=Genus))+
  geom_bar(aes(), stat="identity", position="stack") +
  scale_fill_manual(values = phylum_colors_rel) +
  theme(text = element_text(size = 5),
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
relative_plot

```



sacando la beta diversidad con aglomerado de 10%

```
meta_ord_Eukaryota_Genus <- ordinate(physeq = percentages_Eukaryota_Genus, method = "NMDS", distance = 

## Wisconsin double standardization
## Run 0 stress 0.110709
## Run 1 stress 0.1112576
## Run 2 stress 0.1223838
## Run 3 stress 0.1223842
## Run 4 stress 0.1107094
## ... Procrustes: rmse 0.001116441 max resid 0.006386784
## ... Similar to previous best
## Run 5 stress 0.1107096
## ... Procrustes: rmse 0.0001850301 max resid 0.001026053
## ... Similar to previous best
## Run 6 stress 0.1236311
## Run 7 stress 0.1223842
## Run 8 stress 0.1236485
## Run 9 stress 0.1107101
## ... Procrustes: rmse 0.0002907165 max resid 0.001586952
## ... Similar to previous best
## Run 10 stress 0.1119783
## Run 11 stress 0.1222478
## Run 12 stress 0.1119789
## Run 13 stress 0.1107094
## ... Procrustes: rmse 0.0001044938 max resid 0.0005657647
```

```

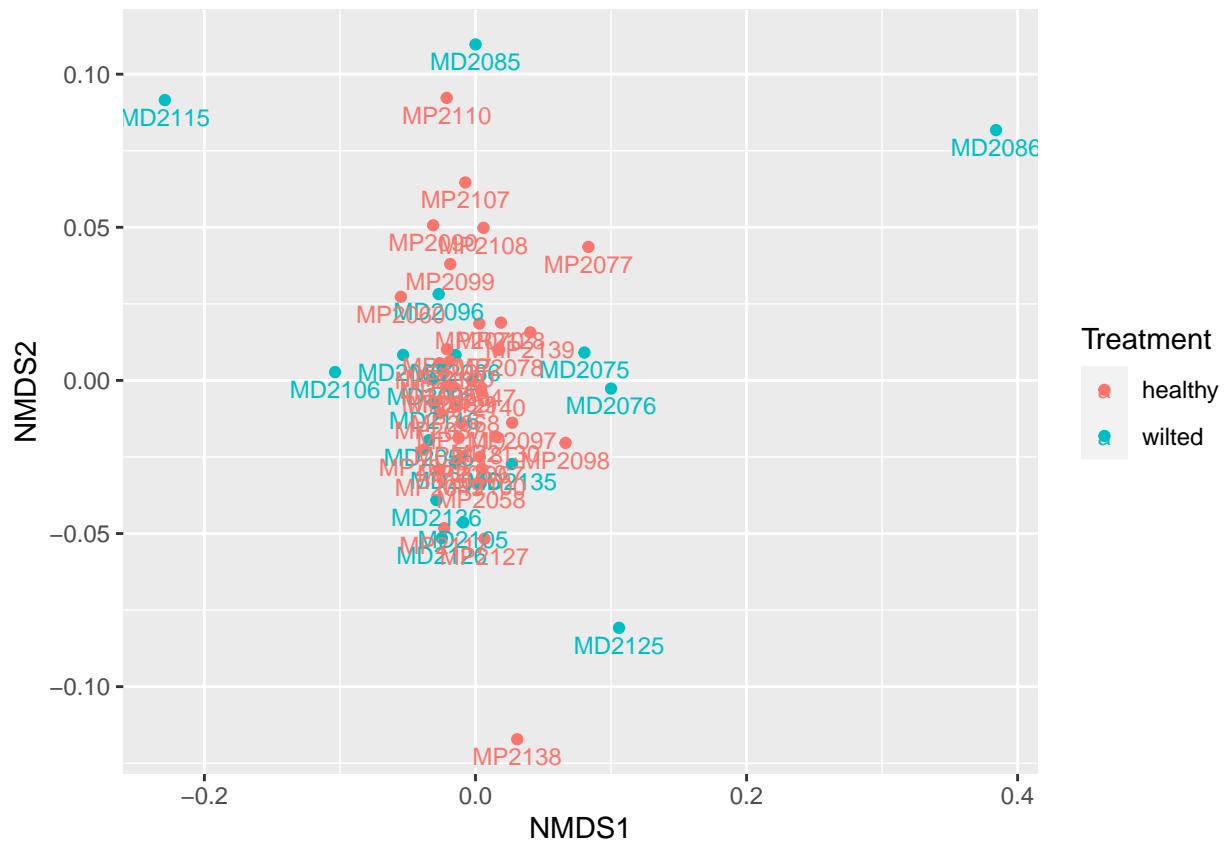
## ... Similar to previous best
## Run 14 stress 0.1107102
## ... Procrustes: rmse 0.001279573 max resid 0.007310778
## ... Similar to previous best
## Run 15 stress 0.1119821
## Run 16 stress 0.110709
## ... New best solution
## ... Procrustes: rmse 0.0009253661 max resid 0.005296653
## ... Similar to previous best
## Run 17 stress 0.1119811
## Run 18 stress 0.1107101
## ... Procrustes: rmse 0.001247762 max resid 0.007008158
## ... Similar to previous best
## Run 19 stress 0.111979
## Run 20 stress 0.1107096
## ... Procrustes: rmse 0.001129286 max resid 0.006412118
## ... Similar to previous best
## *** Best solution repeated 3 times

```

```

plot_ordination(physeq = percentages_Eukaryota_Genus, ordination = meta_ord_Eukaryota_Genus, color = "Treatment"
geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@Data)), size = 3, vjust = 1.5)

```



Veremos aqui solo el reino Bacteriano

```
merge_Bacteria<-subset_taxa(fresa_kraken_fil,Kingdom=="Bacteria")
```

sacamos las abundancias relativas

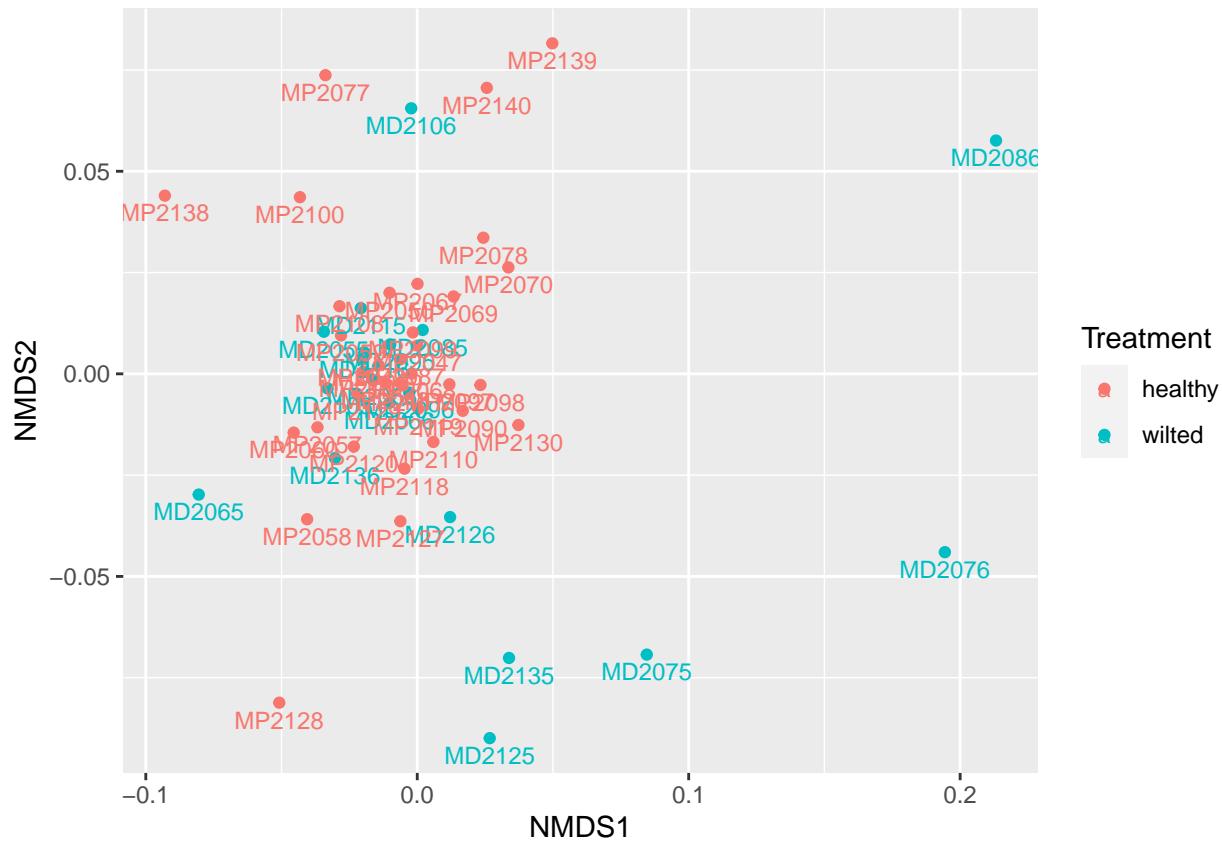
```
percentages_Bacteria <- transform_sample_counts(merge_Bacteria, function(x) x*100 / sum(x) )
percentages_Bacteria_df <- psmelt(percentages_Bacteria)
```

beta diversidad de Bacteria

```
meta_ord_Bacteria <- ordinate(physeq = percentages_Bacteria, method = "NMDS", distance = "bray")
```

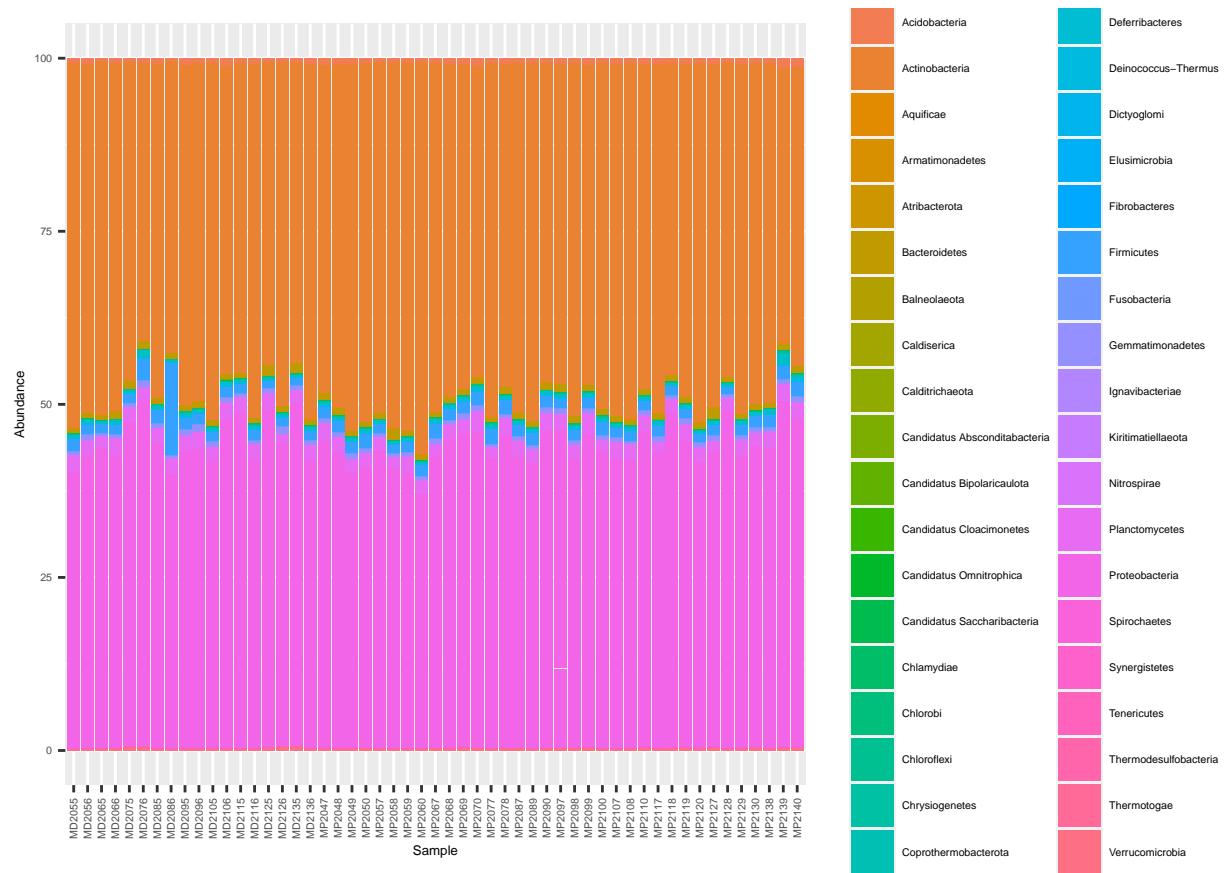
```
## Wisconsin double standardization
## Run 0 stress 0.1680232
## Run 1 stress 0.1696258
## Run 2 stress 0.1976482
## Run 3 stress 0.1639932
## ... New best solution
## ... Procrustes: rmse 0.05672611 max resid 0.2254394
## Run 4 stress 0.1801078
## Run 5 stress 0.1631266
## ... New best solution
## ... Procrustes: rmse 0.04086273 max resid 0.1602159
## Run 6 stress 0.1666047
## Run 7 stress 0.1820413
## Run 8 stress 0.1639946
## Run 9 stress 0.1668663
## Run 10 stress 0.1933575
## Run 11 stress 0.1643944
## Run 12 stress 0.1824765
## Run 13 stress 0.1647139
## Run 14 stress 0.1724096
## Run 15 stress 0.1793737
## Run 16 stress 0.1638
## Run 17 stress 0.1641467
## Run 18 stress 0.1667916
## Run 19 stress 0.1657375
## Run 20 stress 0.1727847
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      1: no. of iterations >= maxit
##      19: stress ratio > sratmax
```

```
plot_ordination(physeq = percentages_Bacteria, ordination = meta_ord_Bacteria, color = "Treatment") +
  geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@Data)), size = 3, vjust = 1.5)
```



Bacterias por Phylum

```
ggplot(data= percentages_Bacteria_df, aes(x=Sample, y=Abundance, fill=Phylum))+  
  geom_bar(aes(), stat="identity", position="stack") +  
  theme(text = element_text(size = 5),  
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



```
merge_Bacteria_Phylum<-tax_glm(merge_Bacteria, taxrank="Phylum")
```

sacamos las abundancias relativas

```
percentages_Bacteria_Phylum <- transform_sample_counts(merge_Bacteria_Phylum, function(x) x*100 / sum(x))
percentages_Bacteria_Phylum_df <- psmelt(percentages_Bacteria_Phylum)
meta_ord_Bacteria_Phylum <- ordinate(physeq = percentages_Bacteria_Phylum, method = "NMDS", distance =
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1520004
## Run 1 stress 0.1663748
## Run 2 stress 0.1783249
## Run 3 stress 0.1723688
## Run 4 stress 0.1606012
## Run 5 stress 0.170769
## Run 6 stress 0.157849
## Run 7 stress 0.160791
## Run 8 stress 0.1739657
## Run 9 stress 0.160738
## Run 10 stress 0.1609939
## Run 11 stress 0.1746317
## Run 12 stress 0.1698149
## Run 13 stress 0.1593176
## Run 14 stress 0.1658336
```

```

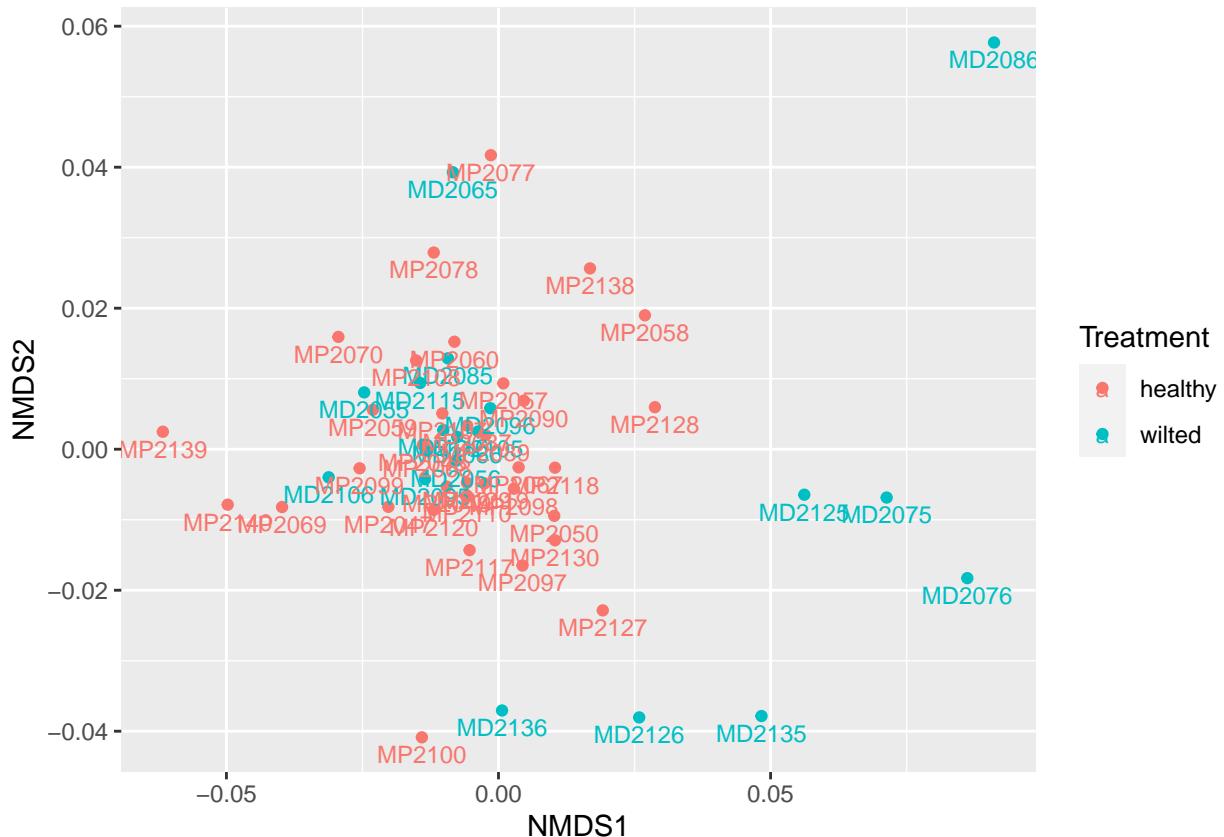
## Run 15 stress 0.1552182
## Run 16 stress 0.175528
## Run 17 stress 0.1731612
## Run 18 stress 0.1520603
## ... Procrustes: rmse 0.04718055 max resid 0.3156681
## Run 19 stress 0.1595159
## Run 20 stress 0.1730442
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      20: stress ratio > sratmax

```

```

plot_ordination(physeq = percentages_Bacteria_Phylum, ordination = meta_ord_Bacteria_Phylum, color = "Treatment"
geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@Data)), size = 3, vjust = 1.5)

```



```

## Bacterias por Family

```

```

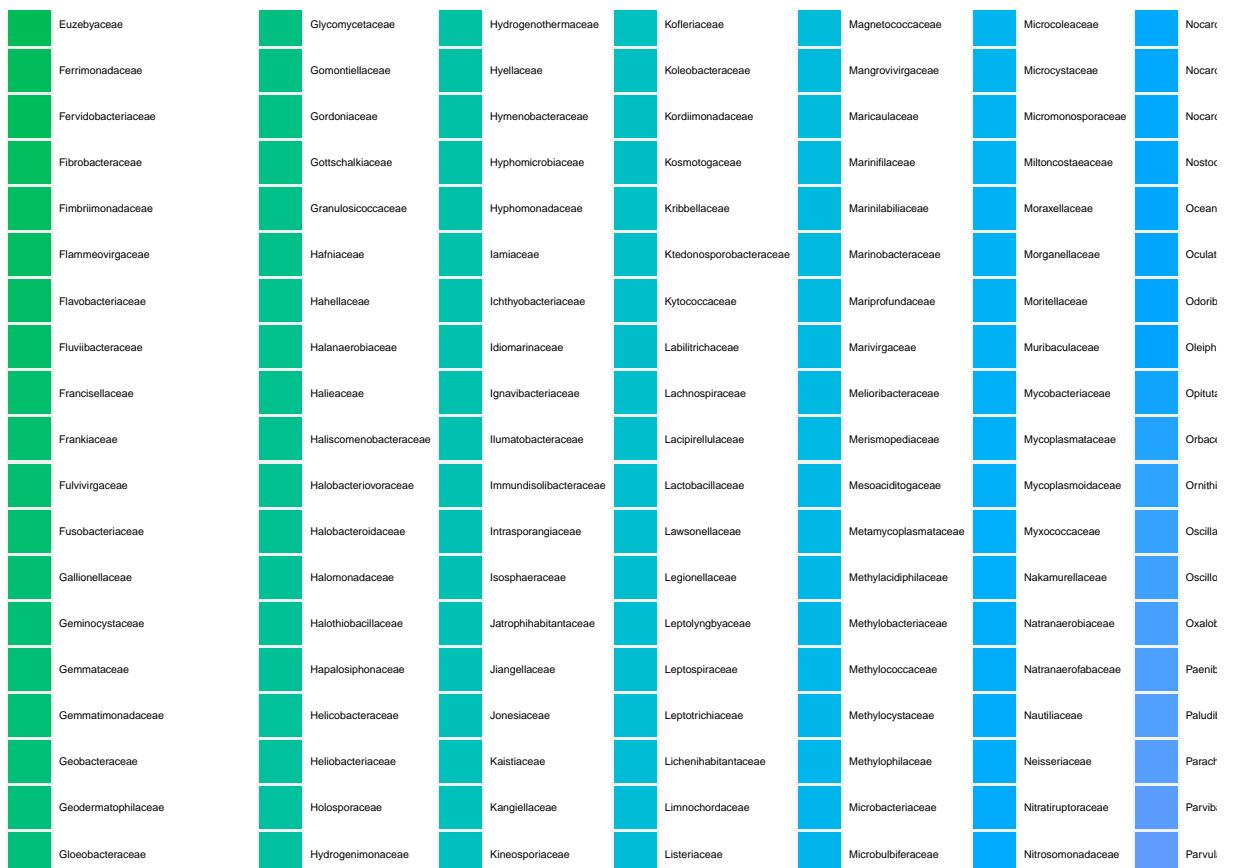
ggplot(data= percentages_Bacteria_df, aes(x=Sample, y=Abundance, fill=Family))+  

  geom_bar(aes(), stat="identity", position="stack") +  

  theme(text = element_text(size = 5),  

        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))

```



```
merge_Bacteria_Family <- tax_glom(merge_Bacteria, taxrank = "Family")
```

sacamos las abundancias relativas

```
percentages_Bacteria_Family <- transform_sample_counts(merge_Bacteria_Family, function(x) x*100 / sum(x))
percentages_Bacteria_Family_df <- psmelt(percentages_Bacteria_Family)
meta_ord_Bacteria_Family <- ordinate(physeq = percentages_Bacteria_Family, method = "NMDS", distance =
```

```
## Wisconsin double standardization
## Run 0 stress 0.1412073
## Run 1 stress 0.1522417
## Run 2 stress 0.1372659
## ... New best solution
## ... Procrustes: rmse 0.06574469 max resid 0.3653103
## Run 3 stress 0.1544647
## Run 4 stress 0.1369863
## ... New best solution
## ... Procrustes: rmse 0.01215007 max resid 0.05736917
## Run 5 stress 0.180529
## Run 6 stress 0.1395937
## Run 7 stress 0.1799178
## Run 8 stress 0.1441744
## Run 9 stress 0.1789251
```

```

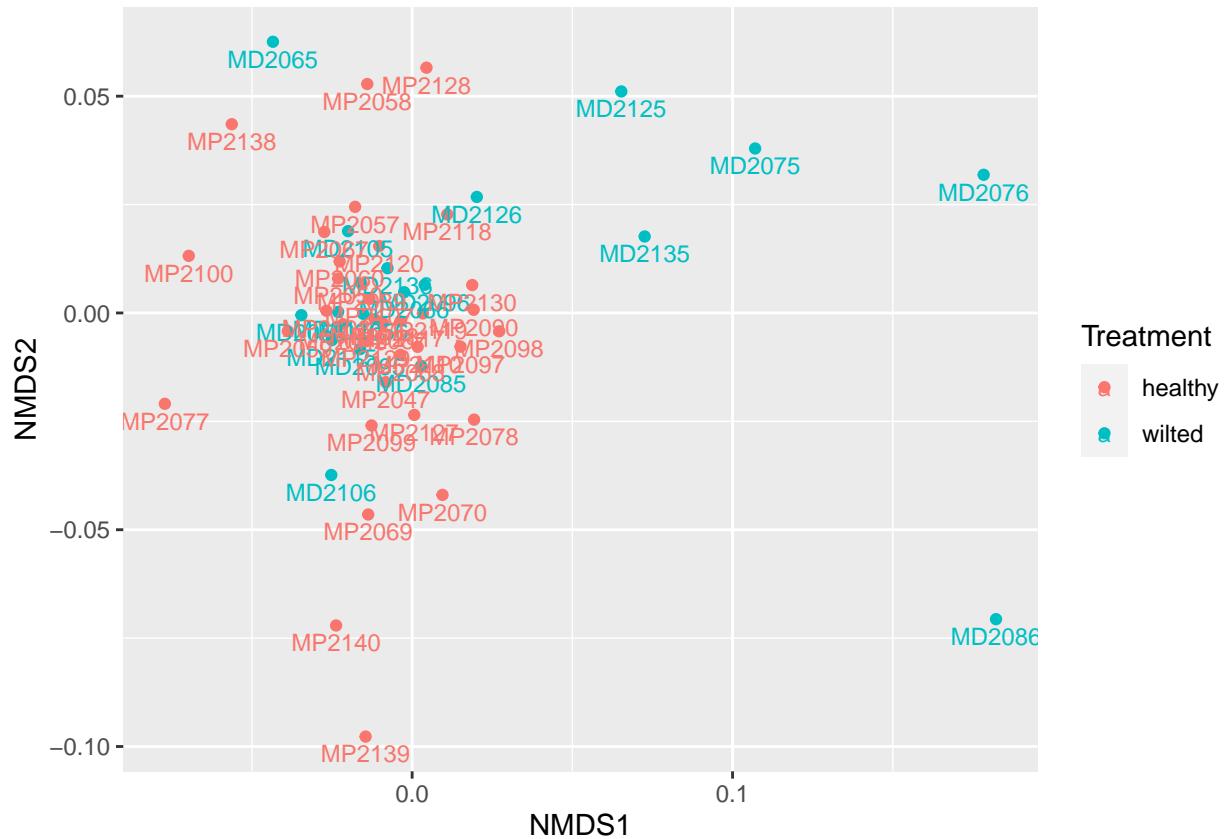
## Run 10 stress 0.1845183
## Run 11 stress 0.1675176
## Run 12 stress 0.1412071
## Run 13 stress 0.1391521
## Run 14 stress 0.156404
## Run 15 stress 0.1412081
## Run 16 stress 0.1373767
## ... Procrustes: rmse 0.01554212 max resid 0.0592079
## Run 17 stress 0.1423478
## Run 18 stress 0.1517987
## Run 19 stress 0.1400611
## Run 20 stress 0.1721753
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      5: no. of iterations >= maxit
##      15: stress ratio > sratmax

```

```

plot_ordination(physeq = percentages_Bacteria_Family, ordination = meta_ord_Bacteria_Family, color = "Treatment",
geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@Data))), size = 3, vjust = 1.5)

```



```

## Bacterias por Genero

```

```

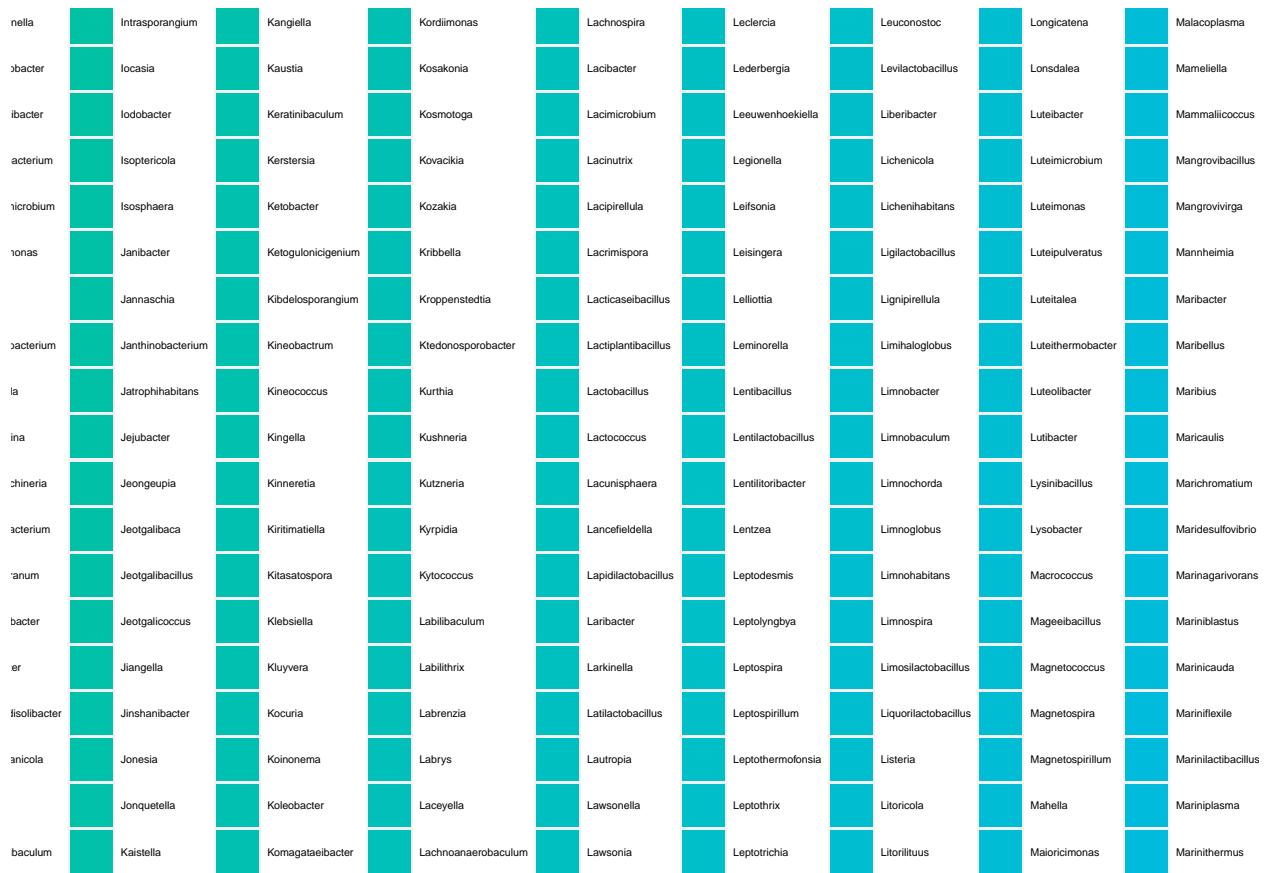
ggplot(data= percentages_Bacteria_df, aes(x=Sample, y=Abundance, fill=Genus))+  

  geom_bar(aes(), stat="identity", position="stack") +  

  theme(text = element_text(size = 5),  

        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))

```



```
merge_Bacteria_Genus <- tax_glom(merge_Bacteria, taxrank = "Genus")
```

sacamos las abundancias relativas

```
percentages_Bacteria_Genus <- transform_sample_counts(merge_Bacteria_Genus, function(x) x*100 / sum(x))
percentages_Bacteria_Genus_df <- ps melt(percentages_Bacteria_Genus)
meta_ord_Bacteria_Genus <- ordinate(physeq = percentages_Bacteria_Genus, method = "NMDS", distance = "bray")
```

```
## Wisconsin double standardization
## Run 0 stress 0.1467945
## Run 1 stress 0.1495483
## Run 2 stress 0.150053
## Run 3 stress 0.1483211
## Run 4 stress 0.1437859
## ... New best solution
## ... Procrustes: rmse 0.09209116 max resid 0.3390236
## Run 5 stress 0.1523147
## Run 6 stress 0.1476747
## Run 7 stress 0.1458784
## Run 8 stress 0.1504334
## Run 9 stress 0.1477492
## Run 10 stress 0.1438667
## ... Procrustes: rmse 0.05236895 max resid 0.2964459
## Run 11 stress 0.1478
## Run 12 stress 0.1457498
```

```

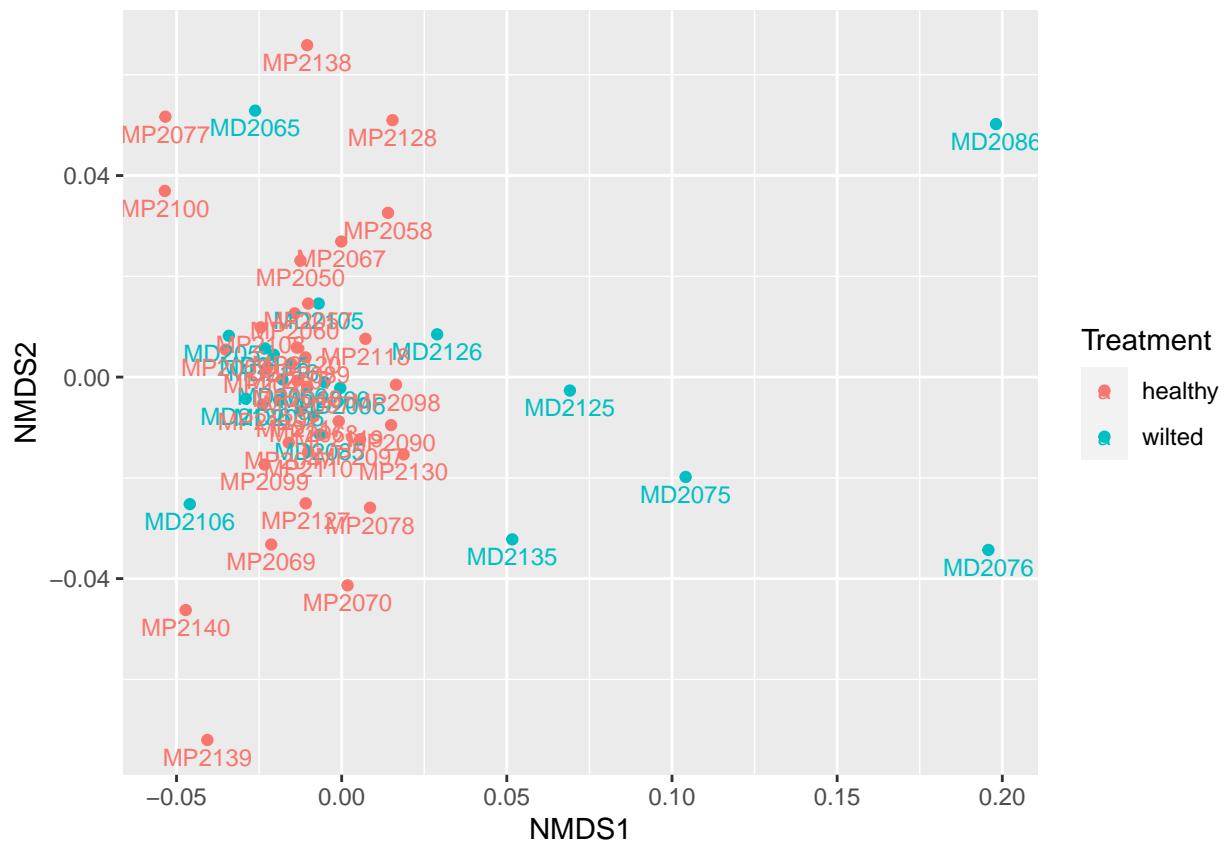
## Run 13 stress 0.1543387
## Run 14 stress 0.1467427
## Run 15 stress 0.1499607
## Run 16 stress 0.1504896
## Run 17 stress 0.1432832
## ... New best solution
## ... Procrustes: rmse 0.01411852 max resid 0.08644691
## Run 18 stress 0.1514187
## Run 19 stress 0.1438132
## Run 20 stress 0.1438149
## *** Best solution was not repeated -- monoMDS stopping criteria:
##     4: no. of iterations >= maxit
##     16: stress ratio > sratmax

```

```

plot_ordination(physeq = percentages_Bacteria_Genus, ordination = meta_ord_Bacteria_Genus, color = "Treatment"
geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@Data)), size = 3, vjust = 1.5)

```



tomando diferentes porcentajes de abundancia

```

percentages_Bacteria_Genus_df$Genus[percentages_Bacteria_Genus_df$Abundance < 10.0] <- "Genus < 0.5% abundance"
percentages_Bacteria_Genus_df$Genus <- as.factor(percentages_Bacteria_Genus_df$Genus)

genus_colors_rel <- colorRampPalette(brewer.pal(8, "Dark2")) (length(levels(percentages_Bacteria_Genus_df$Genus))
relative_plot <- ggplot(data=percentages_Bacteria_Genus_df, aes(x=Sample, y=Abundance, fill=Genus ,color=Genus))
  geom_bar(aes(), stat="identity", position="stack") +
  scale_fill_manual(values = genus_colors_rel) +

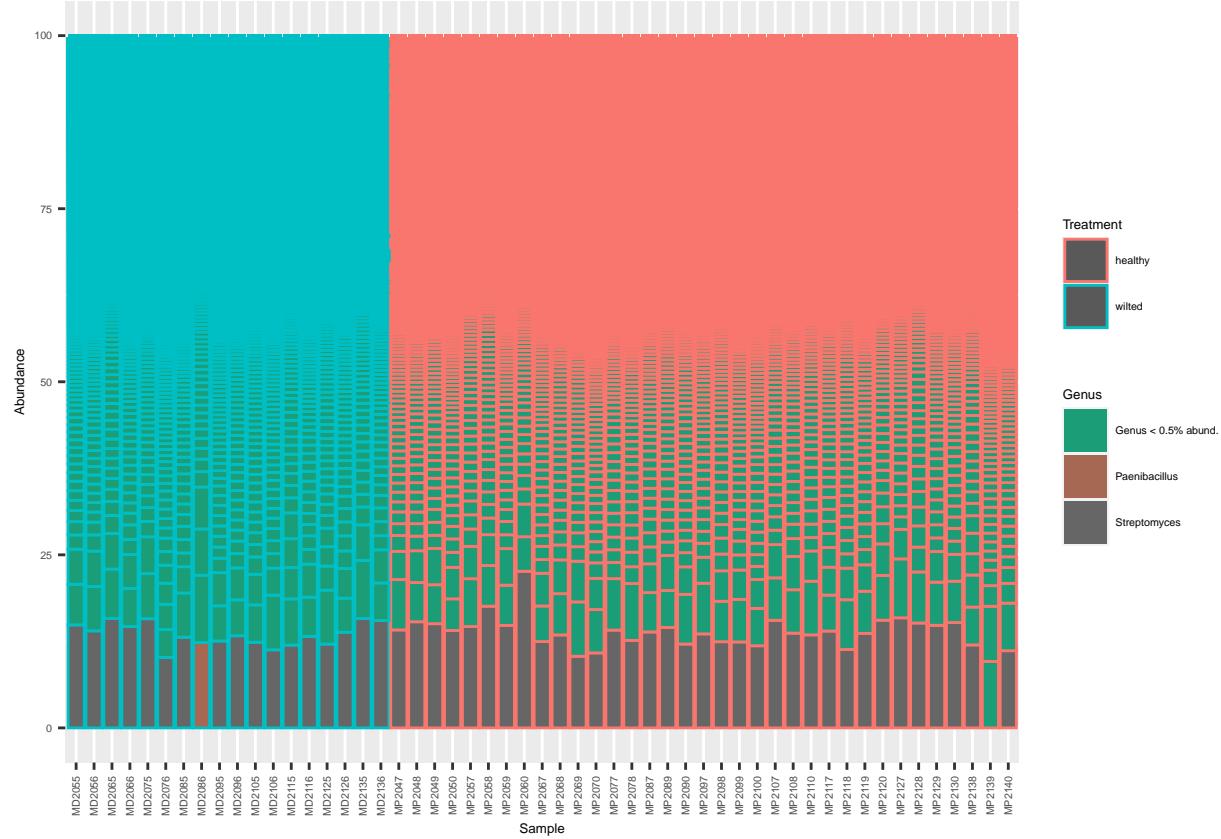
```

```

theme(text = element_text(size = 5),
      axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))

relative_plot

```

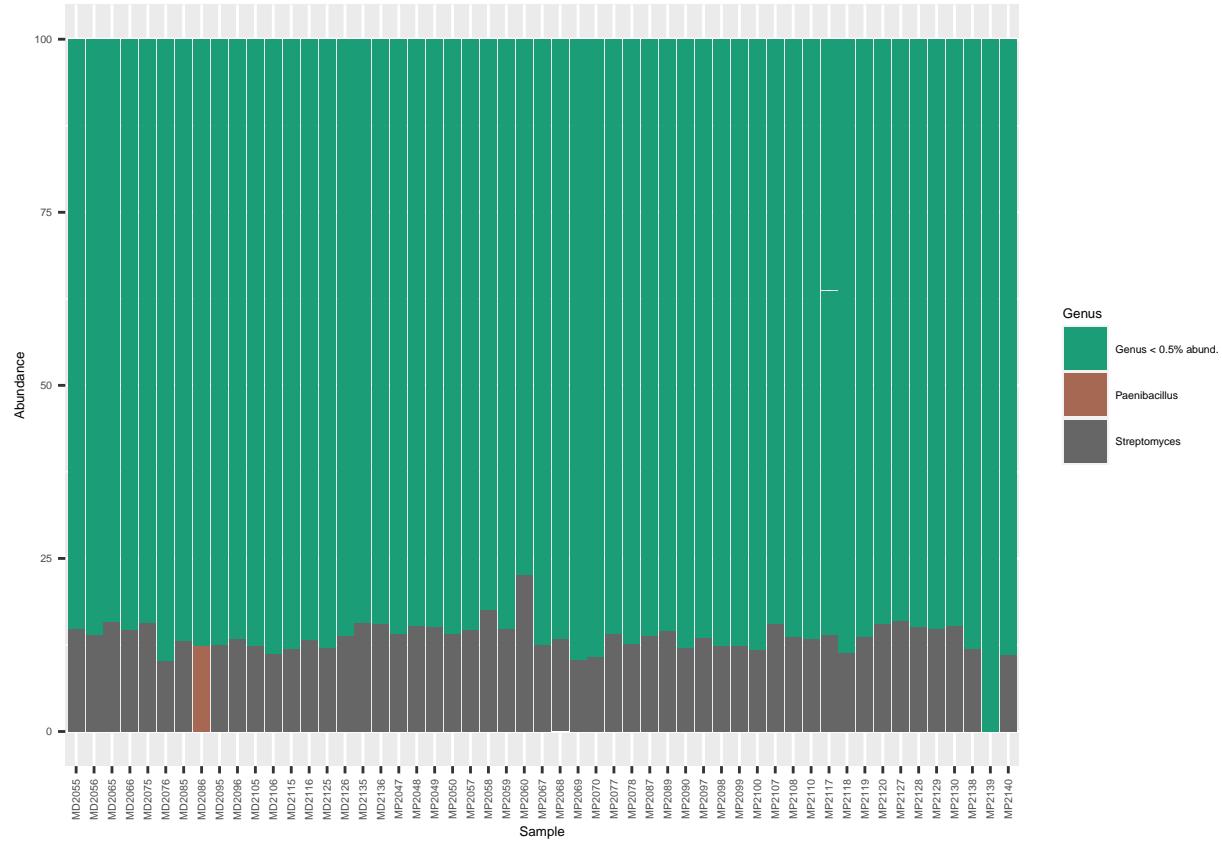


```

relative_plot <- ggplot(data=percentages_Bacteria_Genus_df, aes(x=Sample, y=Abundance, fill=Genus)) +
  geom_bar(aes(), stat="identity", position="stack") +
  scale_fill_manual(values = genus_colors_rel) +
  theme(text = element_text(size = 5),
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))

relative_plot

```



sacando la beta diversidad con aglomerado de 10%

```
meta_ord_Bacteria_Genus <- ordinate(physeq = percentages_Bacteria_Genus, method = "NMDS", distance = "b
```

```
## Wisconsin double standardization
## Run 0 stress 0.1467945
## Run 1 stress 0.1530208
## Run 2 stress 0.1495486
## Run 3 stress 0.1501965
## Run 4 stress 0.1492427
## Run 5 stress 0.1474936
## Run 6 stress 0.1437863
## ... New best solution
## ... Procrustes: rmse 0.09211544 max resid 0.339202
## Run 7 stress 0.1438319
## ... Procrustes: rmse 0.05413051 max resid 0.3063047
## Run 8 stress 0.1438182
## ... Procrustes: rmse 0.05539379 max resid 0.312997
## Run 9 stress 0.1432891
## ... New best solution
## ... Procrustes: rmse 0.01459527 max resid 0.08671625
## Run 10 stress 0.1597643
## Run 11 stress 0.1438134
## Run 12 stress 0.1454656
## Run 13 stress 0.150477
## Run 14 stress 0.1527695
```

```

## Run 15 stress 0.1467425
## Run 16 stress 0.1641878
## Run 17 stress 0.1438167
## Run 18 stress 0.1537524
## Run 19 stress 0.1562419
## Run 20 stress 0.1528352
## *** Best solution was not repeated -- monoMDS stopping criteria:
##     4: no. of iterations >= maxit
##     16: stress ratio > sratmax

plot_ordination(physeq = percentages_Bacteria_Genus, ordination = meta_ord_Bacteria_Genus, color = "Treatment",
geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@Data))), size = 3, vjust = 1.5)

```

