Diversidades alfa y beta en datos metagenomicos de shotgun de fresa

Camila Silva

2023-03-09

Pre-procesamiento de los datos

DIVERSIDADES CON TODO EL CONJUNTO DE DATOS Y LOS DATOS FILTRADOS POR CALIDAD

Los datos fueron entregados en una carpeta de drive https://drive.google.com/drive/folders/1x0106TYUr54gfqE6uod3g5qN8D
Q5x. Para poderlos usar en el servidor, se descargaron en mi maquina local y luego se pasaron por ssh al servidor.

En el equipo local

```
# $ scp Downloads/kraken_results-20230203T201634Z-001.zip camila@132.248.196.39:/home/camila/GIT/Tesis_
#  camila@132.248.196.39's password:
#  kraken_results-20230203T201634Z-001.zip          100%   12MB   2.2MB/s   00:05
# $ scp Downloads/fastp_results-20230203T175936Z-001.zip camila@132.248.196.39:/home/camila/GIT/Tesis_M
#  camila@132.248.196.39's password:
#  fastp_results-20230203T175936Z-001.zip           100% 2728KB   2.1MB/s   00:01
# $ scp Downloads/bracken_results-20230203T203724Z-001.zip camila@132.248.196.39:/home/camila/GIT/Tesis
#  camila@132.248.196.39's password:
#  bracken_results-20230203T203724Z-001.zip         100% 3798KB   2.4MB/s   00:01
```

En el servidor

```
# $ unzip kraken_results-20230203T201634Z-001.zip
# $ unzip bracken_results-20230203T203724Z-001.zip
# $ unzip fastp_results-20230203T175936Z-001.zip
```

Luego de tener las muestras Kraken y braken en el servidor con RStudio; es necesario generar el archivo **.biom**; para esto, se debe activar el ambiente de conda **metagenomis**, **kraken-biom** es un programa ampliamente utilizado para M a partir de la salida de kraken.

Con esto se obtiene una matriz de abundancia a partir de los archivos de salida de Kraken, que permite el uso de el paquete Phyloseq en R, que nos permiten analizar la diversidad y la abundancia mediante la manipulación de datos de asignación taxonómica.

```
# $ conda activate metagenomics

# $ kraken-biom kraken_results/* --fmt json -o fresa_kraken.biom
```

```
getwd()
```

```
## [1] "/home/camila/GIT/Tesis_Maestria/Analisis_Comparativo/Fresa_Solena"
```

```
# if (!requireNamespace("BiocManager", quietly = TRUE))
#   +     install.packages("BiocManager")
# BiocManager::install("phyloseq")

# install.packages(c("ggplot2", "readr", "patchwork"))
```

Phyloseq es un paquete de Bioconductor (Open Source Software For Bioinformatics) para la manipulación y análisis (herramienta para importar, guardar, analizar y visualizar) de datos metagenómicos generados por metodologías de secuenciación de alto rendimiento.

```
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("pbkrtest")
```

```
## Loading required package: lme4
```

```
## Loading required package: Matrix
```

```
#library("BiodiversityR")
library("kableExtra")
```

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

```
library("RColorBrewer")
```

Datos kraken

Importamos los datos kraken en un archivo biom

```
setwd("/home/camila/GIT/Tesis_Maestria/Data/fresa_solena")
fresa_kraken <- import_biom("fresa_kraken.biom")
class(fresa_kraken) # objeto phyloseq
```

```
## [1] "phyloseq"
## attr(,"package")
## [1] "phyloseq"
```

Queremos acceder a los datos que contiene nuestro objeto phyloseq **fresa_kraken**

Primero la tabla de taxonomia

```
#fresa_kraken@tax_table@.Data)
```

Cambiamos los nombres de las columnas a los niveles taxonomicos

```
colnames(fresa_kraken@tax_table@.Data) <- c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus", "S
```

Queremos cortar la parte inicial nombres, ya que aparecen,por ejemplo: "B___Bacteria" y queremos que solo se vea "Bacteria" substring(). Este comando ayuda a extraer o reemplazar caracteres en un vector.

```
fresa_kraken@tax_table@.Data <- substr(fresa_kraken@tax_table@.Data,4,100)
#View(fresa_kraken@tax_table@.Data)
```

Queremos ver la tabla de OTUs, esta tabla contiene las abundancias de los otus de cada muestra

```
#fresa_kraken@otu_table@.Data
```

Queremos cortar los nombres de las muestras para que coincida con los metadatos

```
colnames(fresa_kraken@otu_table@.Data) <- substr(colnames(fresa_kraken@otu_table@.Data),1,6)
#View(fresa_kraken@otu_table@.Data)
```

Cargar los metadatos

Ya que hay un desface de dos muestras entre los metadatos y las muestras de la otu_table se deben ver cuales son y quitarlas del archivo de metadatos

```
# $ ls kraken_results |cut -d'.' -f1 > lista_kraken.txt
# $ ls metadata.csv |cut -d',' -f1 > lista_metadata.txt
# $ wc *txt
# $ cat lista_metadata.txt lista_kraken.txt | sort | uniq -c
# $ cat lista_metadata.txt lista_kraken.txt | sort | uniq -c | sort | head
# $  1 MD2145
# $  1 MD2146
# $  2 MD2055
# $  2 MD2056
```

Eliminamos las dos muestras que no estaban en nuestra otu_table y cargamos los metadatos

```
metadata_fresa <- read.csv2("/home/camila/GIT/Tesis_Maestria/Data/fresa_solena/metadata.csv",header =
#metadata_fresa <- sample_data(metadata_fresa)
```

luego hacemos que los metadatos pertenezcan al objeto phyloseq en la seccion de **sam_data**

```
fresa_kraken@sam_data <- sample_data(metadata_fresa)
```

Creamos una columna extra en sam_data por necesidad de funcionamiento de mas adelante

```
fresa_kraken@sam_data$Sample<-row.names(fresa_kraken@sam_data)
colnames(fresa_kraken@sam_data)<-c('Treatment','Samples')
#View(fresa_kraken)
```

Ahora tenemos tambien la tabla de datos de calidad (fastp_kraken_summary),

para ver que muestra podemos eliminar de nuestro dataset, que no cumpla ciertos estandares de calidad + ID de la muestra + Reads_B - Reads_Before -> total de reads crudos + Reads_A - Reads_After -> total de reads despues del analisis de calidad + Reads_diff -> diferencia en tre Reads_B y Reads_A + Q30_B -> porcentaje arriba de 30 (escala fred) antes del analisis de calidad + Q30_A -> porcentaje arriba de 30 (escala fred) despues del analisis de calidad + LowQua -> reads de baja calidad + N_reads -> readas que contienen N y se descartan + too_short -> no pasan el tamaño minimo de calidad + Duplication -> porcentaje de duplicados + LengthR1 -> longitud promedio de los reads + LengthR2 -> longitud promedio de los reads + Classified -> porcentaje de clasificados del total despues del filtrado

ejemplo muestra MD2055 -> contienen 97millones de reads antes del filtrado de calidad, y despues queda con 79millones filtro usado para eliminar muestras, que luego del filtrado de calidad contengan menos de 25millones de reads los que nos da 5 muestras a eliminar (MP2079,MP2080,MP2088,MP2109,MP2137)

Eliminiar las muestras de baja calidad,usando filtro de menos de 25 millones de reads luegodel analisis de calidad se eliminan las muestras por su nombre

```
samples_to_remove <- c("MP2079","MP2080","MP2088","MP2109","MP2137")
fresa_kraken_fil <- prune_samples(!(sample_names(fresa_kraken) %in% samples_to_remove), fresa_kraken)
```

podemos comprobar el número de muestras antes y después del filtrado

```
nsamples(fresa_kraken) # 58
```

```
## [1] 58
```

```
nsamples(fresa_kraken_fil) # 53
```

```
## [1] 53
```

Queremos hacer un analisis de diversidad de nuestras muestras, para esto las dos metricas las usadas son: Diversidad Alfa y Beta

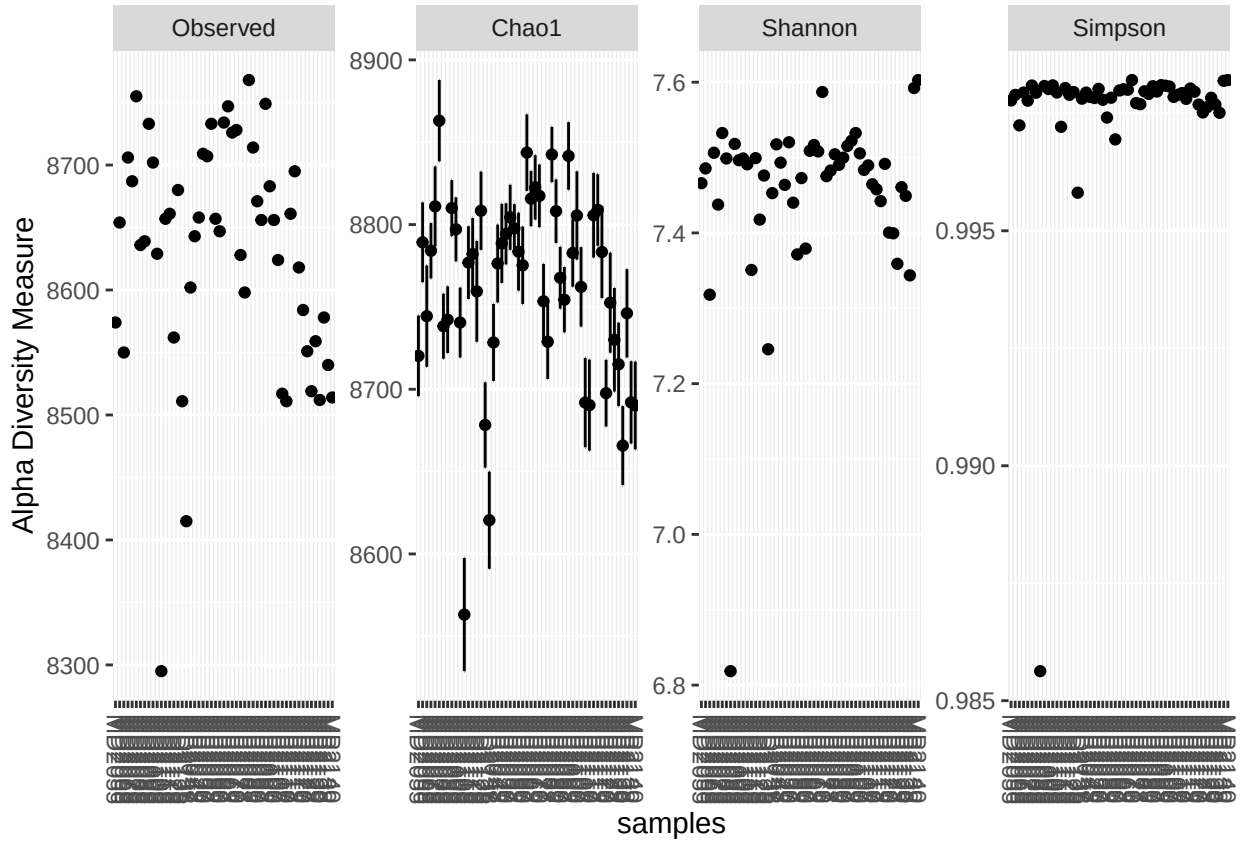
Diversidad Alfa

Esta representa la riqueza de las muestras, es decir el numero de especies diferentes en ese ambiente o la abundancia de especies en ese ambiente. Paramedier esta divercidad se tienen diferentes indices de medida, entre ellos losindices Shannon,Simpson, Chao1, entre otros.

```
index = estimate_richness(fresa_kraken_fil)
```

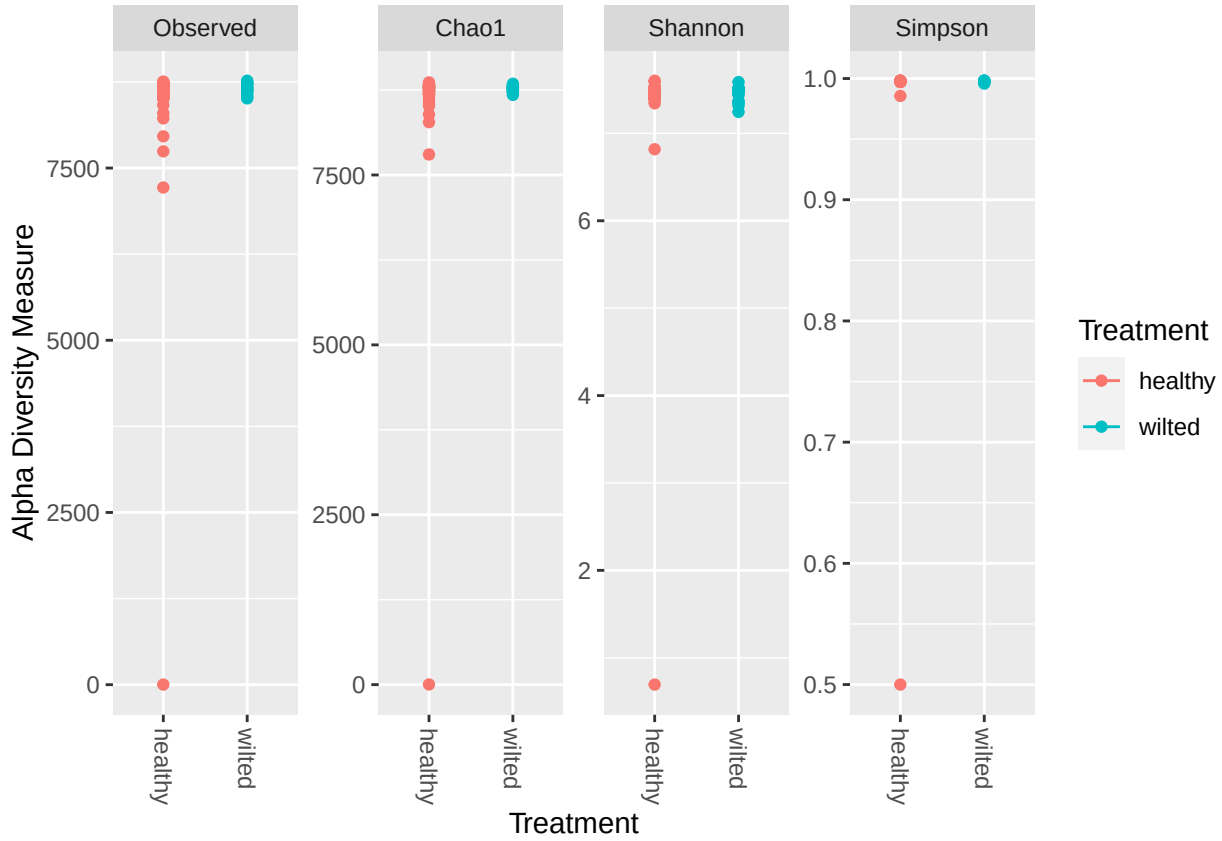
podemos ver una representación visual de la diversidad dentro de las muestras esta diversidad la podemos ver por muestra,

```
plot_richness(physeq = fresa_kraken_fil, measures = c("Observed","Chao1","Shannon","simpson"))
```



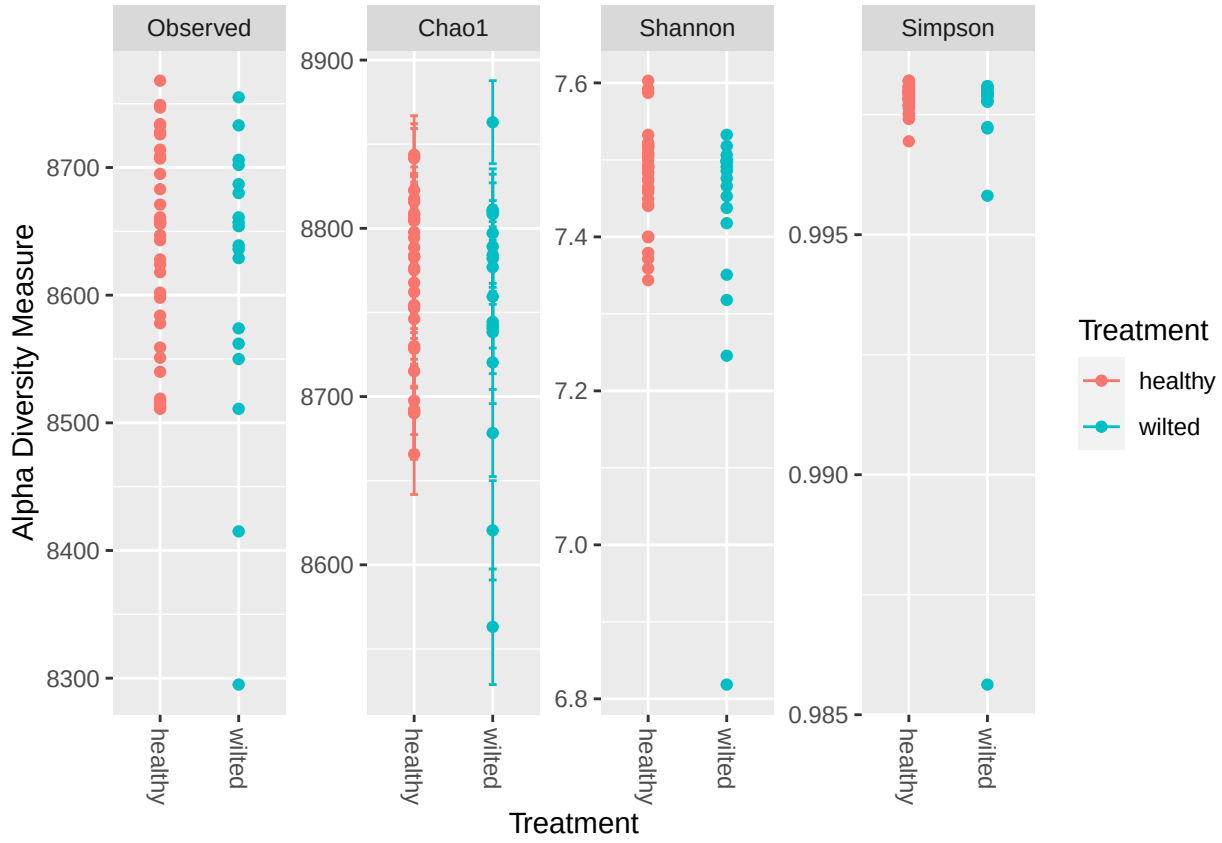
pero ya que queremos diferenciar entre plantas sanas y enfermas, tomamos estos dos conjuntos por ceparado para mostrar la diversidad

```
plot_richness(physeq = fresa_kraken, measures = c("Observed","Chao1","Shannon","simpson"),x = "Treatmen
```



comparando entre los datos en crudo,y los datos filtrados por calidad

```
plot_richness(physeq = fresa_kraken_fil, measures = c("Observed","Chao1","Shannon","simpson"),x = "Trea
```



Shannon

Simpson

Chao1

Aqui podemos ver cuantos reads tenemos por muestra

```
sample_sums(fresa_kraken)
```

```
##    MD2055    MD2056    MD2065    MD2066    MD2075    MD2076    MD2085    MD2086
##   9782432  12468526  11297600  15580959  12310781  16067839  10524919   9931297
##    MD2095    MD2096    MD2105    MD2106    MD2115    MD2116    MD2125    MD2126
## 18009912  14998268  11792397   4053295  13102554  12451637   8355853  14307309
##    MD2135    MD2136    MP2047    MP2048    MP2049    MP2050    MP2057    MP2058
##   7280751   6172369   9199079  12146967  13075806  16098757  17141427  20923502
##    MP2059    MP2060    MP2067    MP2068    MP2069    MP2070    MP2077    MP2078
```

```
## 14129981 13786630 16924218 20873789 15537530 12462356  9617847  7588787
##   MP2079   MP2080   MP2087   MP2088   MP2089   MP2090   MP2097   MP2098
##   745830  3125701 20632320        2 16582404 11176782 11714000 16595897
##   MP2099   MP2100   MP2107   MP2108   MP2109   MP2110   MP2117   MP2118
## 14844038 13342326 11014462  6728020  1405462  6901265 12624002 14711376
##   MP2119   MP2120   MP2127   MP2128   MP2129   MP2130   MP2137   MP2138
##  9835326 10975712  7106567  9974861  8348307  6196725  2169734  8220431
##   MP2139   MP2140
##  6158581  5267510
```

```
sample_sums(fresa_kraken_fil)
```

```
##   MD2055   MD2056   MD2065   MD2066   MD2075   MD2076   MD2085   MD2086
##  9782432 12468526 11297600 15580959 12310781 16067839 10524919  9931297
##   MD2095   MD2096   MD2105   MD2106   MD2115   MD2116   MD2125   MD2126
## 18009912 14998268 11792397  4053295 13102554 12451637  8355853 14307309
##   MD2135   MD2136   MP2047   MP2048   MP2049   MP2050   MP2057   MP2058
##  7280751  6172369  9199079 12146967 13075806 16098757 17141427 20923502
##   MP2059   MP2060   MP2067   MP2068   MP2069   MP2070   MP2077   MP2078
## 14129981 13786630 16924218 20873789 15537530 12462356  9617847  7588787
##   MP2087   MP2089   MP2090   MP2097   MP2098   MP2099   MP2100   MP2107
## 20632320 16582404 11176782 11714000 16595897 14844038 13342326 11014462
##   MP2108   MP2110   MP2117   MP2118   MP2119   MP2120   MP2127   MP2128
##  6728020  6901265 12624002 14711376  9835326 10975712  7106567  9974861
##   MP2129   MP2130   MP2138   MP2139   MP2140
##  8348307  6196725  8220431  6158581  5267510
```

y con **summary** podemos darnos una idea dela uniformidad de los datos, ya que podemos ver datos estadisticos como elmaximo, el minimo y la media

```
summary(fresa_kraken@otu_table@.Data)
```

```
##      MD2055             MD2056           MD2065             MD2066
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0   Min.   :     0
##  1st Qu.:    14.0   1st Qu.:    18   1st Qu.:    13.0   1st Qu.:    24
##  Median :    95.0   Median :   124   Median :    87.0   Median :   155
##  Mean   :  1086.6   Mean   :  1385   Mean   :  1254.9   Mean   :  1731
##  3rd Qu.:   854.5   3rd Qu.:  1100   3rd Qu.:   846.5   3rd Qu.:  1412
##  Max.   :309650.0   Max.   :369924   Max.   :382080.0   Max.   :470868
##      MD2075           MD2076           MD2085           MD2086
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :      0
##  1st Qu.:    22   1st Qu.:    40   1st Qu.:    18   1st Qu.:     18
##  Median :   124   Median :   210   Median :   112   Median :     90
##  Mean   :  1367   Mean   :  1785   Mean   :  1169   Mean   :   1103
##  3rd Qu.:  1027   3rd Qu.:  1470   3rd Qu.:   934   3rd Qu.:    675
##  Max.   :347565   Max.   :391558   Max.   :290224   Max.   :1026738
##      MD2095             MD2096           MD2105           MD2106
##  Min.   :     0.0   Min.   :     0   Min.   :     0   Min.   :     0.0
##  1st Qu.:    28.5   1st Qu.:    24   1st Qu.:    18   1st Qu.:     6.0
##  Median :   188.0   Median :   153   Median :   115   Median :    43.0
##  Mean   :  2000.4   Mean   :  1666   Mean   :  1310   Mean   :   450.2
##  3rd Qu.:  1667.0   3rd Qu.:  1333   3rd Qu.:  1056   3rd Qu.:   375.5
```

```
##  Max.   :479760.0   Max.   :408186   Max.   :307824   Max.   :100633.0
##      MD2115           MD2116           MD2125           MD2126
##  Min.   :     0   Min.   :     0   Min.   :     0.0   Min.   :     0
##  1st Qu.:    19   1st Qu.:    18   1st Qu.:    13.0   1st Qu.:    23
##  Median :   126   Median :   123   Median :    80.0   Median :   146
##  Mean   :  1455   Mean   :  1383   Mean   :   928.1   Mean   :  1589
##  3rd Qu.:  1112   3rd Qu.:  1106   3rd Qu.:   705.0   3rd Qu.:  1260
##  Max.   :352456   Max.   :342009   Max.   :202939.0   Max.   :409587
##      MD2135           MD2136           MP2047           MP2048
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0
##  1st Qu.:    11.0   1st Qu.:     9.0   1st Qu.:    15.0   1st Qu.:    18
##  Median :    66.0   Median :    59.0   Median :   100.0   Median :   124
##  Mean   :   808.7   Mean   :   685.6   Mean   :  1021.8   Mean   :  1349
##  3rd Qu.:   570.0   3rd Qu.:   540.5   3rd Qu.:   849.5   3rd Qu.:  1096
##  Max.   :324929.0   Max.   :188357.0   Max.   :266580.0   Max.   :372843
##      MP2049           MP2050           MP2057           MP2058
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :     0.0
##  1st Qu.:    19   1st Qu.:    28   1st Qu.:    23   1st Qu.:    28.5
##  Median :   128   Median :   173   Median :   157   Median :   177.0
##  Mean   :  1452   Mean   :  1788   Mean   :  1904   Mean   :  2324.1
##  3rd Qu.:  1120   3rd Qu.:  1444   3rd Qu.:  1422   3rd Qu.:  1613.5
##  Max.   :405044   Max.   :475984   Max.   :527148   Max.   :750581.0
##      MP2059           MP2060           MP2067           MP2068
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :     0
##  1st Qu.:    20   1st Qu.:    19   1st Qu.:    29   1st Qu.:    34
##  Median :   134   Median :   119   Median :   184   Median :   220
##  Mean   :  1570   Mean   :  1531   Mean   :  1880   Mean   :  2318
##  3rd Qu.:  1238   3rd Qu.:  1108   3rd Qu.:  1543   3rd Qu.:  1930
##  Max.   :438466   Max.   :604749   Max.   :481017   Max.   :583698
##      MP2069           MP2070           MP2077           MP2078
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:    27.5   1st Qu.:    27   1st Qu.:    17.0   1st Qu.:    14.0
##  Median :   173.0   Median :   155   Median :   107.0   Median :    85.0
##  Mean   :  1725.8   Mean   :  1384   Mean   :  1068.3   Mean   :   842.9
##  3rd Qu.:  1454.5   3rd Qu.:  1226   3rd Qu.:   819.5   3rd Qu.:   693.0
##  Max.   :353252.0   Max.   :298480   Max.   :293779.0   Max.   :219233.0
##      MP2079           MP2080           MP2087           MP2088
##  Min.   :    0.00   Min.   :    0.0   Min.   :     0   Min.   :0.0000000
##  1st Qu.:    1.00   1st Qu.:    5.0   1st Qu.:    35   1st Qu.:0.0000000
##  Median :    8.00   Median :   32.0   Median :   217   Median :0.0000000
##  Mean   :   82.84   Mean   :  347.2   Mean   :  2292   Mean   :0.0002221
##  3rd Qu.:   66.00   3rd Qu.:  268.5   3rd Qu.:  1844   3rd Qu.:0.0000000
##  Max.   :24597.00   Max.   :96283.0   Max.   :591983   Max.   :1.0000000
##      MP2089           MP2090           MP2097           MP2098
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :     0
##  1st Qu.:    26   1st Qu.:    19   1st Qu.:    21   1st Qu.:    31
##  Median :   162   Median :   117   Median :   131   Median :   181
##  Mean   :  1842   Mean   :  1241   Mean   :  1301   Mean   :  1843
##  3rd Qu.:  1466   3rd Qu.:   988   3rd Qu.:  1068   3rd Qu.:  1484
##  Max.   :500426   Max.   :280636   Max.   :323878   Max.   :441849
##      MP2099           MP2100           MP2107           MP2108
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :     0.0
##  1st Qu.:    24   1st Qu.:    20   1st Qu.:    16   1st Qu.:    10.0
##  Median :   163   Median :   141   Median :   108   Median :    67.0
```

```
##  Mean   :  1649   Mean   :  1482   Mean   :  1223   Mean   :   747.3
##  3rd Qu.:  1404   3rd Qu.:  1211   3rd Qu.:   966   3rd Qu.:   605.0
##  Max.   :381823   Max.   :347101   Max.   :340913   Max.   :196757.0
##      MP2109            MP2110            MP2117           MP2118
##  Min.   :    0.0   Min.   :     0.0   Min.   :     0   Min.   :     0
##  1st Qu.:    2.0   1st Qu.:    11.0   1st Qu.:    19   1st Qu.:    21
##  Median :   15.0   Median :    68.0   Median :   131   Median :   143
##  Mean   :  156.1   Mean   :   766.5   Mean   :  1402   Mean   :  1634
##  3rd Qu.:  129.0   3rd Qu.:   598.0   3rd Qu.:  1108   3rd Qu.:  1266
##  Max.   :43535.0   Max.   :185697.0   Max.   :373971   Max.   :344457
##      MP2119            MP2120           MP2127            MP2128
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:    16.0   1st Qu.:    14   1st Qu.:    11.0   1st Qu.:    12.0
##  Median :   100.0   Median :    98   Median :    71.0   Median :    83.0
##  Mean   :  1092.5   Mean   :  1219   Mean   :   789.4   Mean   :  1108.0
##  3rd Qu.:   882.5   3rd Qu.:   907   3rd Qu.:   599.0   3rd Qu.:   773.5
##  Max.   :278616.0   Max.   :340307   Max.   :223557.0   Max.   :289910.0
##      MP2129            MP2130            MP2137          MP2138
##  Min.   :     0.0   Min.   :     0.0   Min.   :    0   Min.   :     0.0
##  1st Qu.:    12.0   1st Qu.:    10.0   1st Qu.:    3   1st Qu.:    12.0
##  Median :    80.0   Median :    66.0   Median :   21   Median :    78.0
##  Mean   :   927.3   Mean   :   688.3   Mean   :  241   Mean   :   913.1
##  3rd Qu.:   717.0   3rd Qu.:   539.0   3rd Qu.:  177   3rd Qu.:   650.5
##  Max.   :249480.0   Max.   :190633.0   Max.   :81445   Max.   :203351.0
##      MP2139            MP2140
##  Min.   :     0.0   Min.   :     0.0
##  1st Qu.:    13.0   1st Qu.:    10.0
##  Median :    81.0   Median :    68.0
##  Mean   :   684.1   Mean   :   585.1
##  3rd Qu.:   622.5   3rd Qu.:   535.0
##  Max.   :130999.0   Max.   :126168.0
```

```
summary(fresa_kraken_fil@otu_table@.Data)
```

```
##      MD2055            MD2056           MD2065            MD2066
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0   Min.   :     0
##  1st Qu.:    14.0   1st Qu.:    18   1st Qu.:    13.0   1st Qu.:    24
##  Median :    95.0   Median :   124   Median :    87.0   Median :   155
##  Mean   :  1086.6   Mean   :  1385   Mean   :  1254.9   Mean   :  1731
##  3rd Qu.:   854.5   3rd Qu.:  1100   3rd Qu.:   846.5   3rd Qu.:  1412
##  Max.   :309650.0   Max.   :369924   Max.   :382080.0   Max.   :470868
##      MD2075           MD2076           MD2085           MD2086
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :      0
##  1st Qu.:    22   1st Qu.:    40   1st Qu.:    18   1st Qu.:     18
##  Median :   124   Median :   210   Median :   112   Median :     90
##  Mean   :  1367   Mean   :  1785   Mean   :  1169   Mean   :   1103
##  3rd Qu.:  1027   3rd Qu.:  1470   3rd Qu.:   934   3rd Qu.:    675
##  Max.   :347565   Max.   :391558   Max.   :290224   Max.   :1026738
##      MD2095            MD2096           MD2105           MD2106
##  Min.   :     0.0   Min.   :     0   Min.   :     0   Min.   :     0.0
##  1st Qu.:    28.5   1st Qu.:    24   1st Qu.:    18   1st Qu.:     6.0
##  Median :   188.0   Median :   153   Median :   115   Median :    43.0
##  Mean   :  2000.4   Mean   :  1666   Mean   :  1310   Mean   :   450.2
##  3rd Qu.:  1667.0   3rd Qu.:  1333   3rd Qu.:  1056   3rd Qu.:   375.5
```

```
##  Max.   :479760.0   Max.   :408186   Max.   :307824   Max.   :100633.0
##      MD2115           MD2116           MD2125           MD2126
##  Min.   :     0   Min.   :     0   Min.   :     0.0   Min.   :     0
##  1st Qu.:    19   1st Qu.:    18   1st Qu.:    13.0   1st Qu.:    23
##  Median :   126   Median :   123   Median :    80.0   Median :   146
##  Mean   :  1455   Mean   :  1383   Mean   :   928.1   Mean   :  1589
##  3rd Qu.:  1112   3rd Qu.:  1106   3rd Qu.:   705.0   3rd Qu.:  1260
##  Max.   :352456   Max.   :342009   Max.   :202939.0   Max.   :409587
##      MD2135           MD2136           MP2047           MP2048
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0
##  1st Qu.:    11.0   1st Qu.:     9.0   1st Qu.:    15.0   1st Qu.:    18
##  Median :    66.0   Median :    59.0   Median :   100.0   Median :   124
##  Mean   :   808.7   Mean   :   685.6   Mean   :  1021.8   Mean   :  1349
##  3rd Qu.:   570.0   3rd Qu.:   540.5   3rd Qu.:   849.5   3rd Qu.:  1096
##  Max.   :324929.0   Max.   :188357.0   Max.   :266580.0   Max.   :372843
##      MP2049           MP2050           MP2057           MP2058
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :     0.0
##  1st Qu.:    19   1st Qu.:    28   1st Qu.:    23   1st Qu.:    28.5
##  Median :   128   Median :   173   Median :   157   Median :   177.0
##  Mean   :  1452   Mean   :  1788   Mean   :  1904   Mean   :  2324.1
##  3rd Qu.:  1120   3rd Qu.:  1444   3rd Qu.:  1422   3rd Qu.:  1613.5
##  Max.   :405044   Max.   :475984   Max.   :527148   Max.   :750581.0
##      MP2059           MP2060           MP2067           MP2068
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :     0
##  1st Qu.:    20   1st Qu.:    19   1st Qu.:    29   1st Qu.:    34
##  Median :   134   Median :   119   Median :   184   Median :   220
##  Mean   :  1570   Mean   :  1531   Mean   :  1880   Mean   :  2318
##  3rd Qu.:  1238   3rd Qu.:  1108   3rd Qu.:  1543   3rd Qu.:  1930
##  Max.   :438466   Max.   :604749   Max.   :481017   Max.   :583698
##      MP2069           MP2070           MP2077           MP2078
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:    27.5   1st Qu.:    27   1st Qu.:    17.0   1st Qu.:    14.0
##  Median :   173.0   Median :   155   Median :   107.0   Median :    85.0
##  Mean   :  1725.8   Mean   :  1384   Mean   :  1068.3   Mean   :   842.9
##  3rd Qu.:  1454.5   3rd Qu.:  1226   3rd Qu.:   819.5   3rd Qu.:   693.0
##  Max.   :353252.0   Max.   :298480   Max.   :293779.0   Max.   :219233.0
##      MP2087           MP2089           MP2090           MP2097
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :     0
##  1st Qu.:    35   1st Qu.:    26   1st Qu.:    19   1st Qu.:    21
##  Median :   217   Median :   162   Median :   117   Median :   131
##  Mean   :  2292   Mean   :  1842   Mean   :  1241   Mean   :  1301
##  3rd Qu.:  1844   3rd Qu.:  1466   3rd Qu.:   988   3rd Qu.:  1068
##  Max.   :591983   Max.   :500426   Max.   :280636   Max.   :323878
##      MP2098           MP2099           MP2100           MP2107
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   :     0
##  1st Qu.:    31   1st Qu.:    24   1st Qu.:    20   1st Qu.:    16
##  Median :   181   Median :   163   Median :   141   Median :   108
##  Mean   :  1843   Mean   :  1649   Mean   :  1482   Mean   :  1223
##  3rd Qu.:  1484   3rd Qu.:  1404   3rd Qu.:  1211   3rd Qu.:   966
##  Max.   :441849   Max.   :381823   Max.   :347101   Max.   :340913
##      MP2108           MP2110           MP2117           MP2118
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0   Min.   :     0
##  1st Qu.:    10.0   1st Qu.:    11.0   1st Qu.:    19   1st Qu.:    21
##  Median :    67.0   Median :    68.0   Median :   131   Median :   143
```

```
##  Mean   :   747.3   Mean   :   766.5   Mean   :  1402   Mean   :  1634
##  3rd Qu.:   605.0   3rd Qu.:   598.0   3rd Qu.:  1108   3rd Qu.:  1266
##  Max.   :196757.0   Max.   :185697.0   Max.   :373971   Max.   :344457
##      MP2119             MP2120             MP2127             MP2128
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:    16.0   1st Qu.:    14   1st Qu.:    11.0   1st Qu.:    12.0
##  Median :   100.0   Median :    98   Median :    71.0   Median :    83.0
##  Mean   :  1092.5   Mean   :  1219   Mean   :   789.4   Mean   :  1108.0
##  3rd Qu.:   882.5   3rd Qu.:   907   3rd Qu.:   599.0   3rd Qu.:   773.5
##  Max.   :278616.0   Max.   :340307   Max.   :223557.0   Max.   :289910.0
##      MP2129             MP2130             MP2138             MP2139
##  Min.   :     0.0   Min.   :     0.0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:    12.0   1st Qu.:    10.0   1st Qu.:    12.0   1st Qu.:    13.0
##  Median :    80.0   Median :    66.0   Median :    78.0   Median :    81.0
##  Mean   :   927.3   Mean   :   688.3   Mean   :   913.1   Mean   :   684.1
##  3rd Qu.:   717.0   3rd Qu.:   539.0   3rd Qu.:   650.5   3rd Qu.:   622.5
##  Max.   :249480.0   Max.   :190633.0   Max.   :203351.0   Max.   :130999.0
##      MP2140
##  Min.   :     0.0
##  1st Qu.:    10.0
##  Median :    68.0
##  Mean   :   585.1
##  3rd Qu.:   535.0
##  Max.   :126168.0
```

Con el siguiente comando podemos ver si tenemos muestras no identificadas taxonomicamente, esto se puede ver identificando los espacios en blanco (" ") en los diferentes niveles taxonomicos.

```
summary(fresa_kraken_fil@tax_table@.Data== "")
```

```
##   Kingdom          Phylum          Class           Order
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:9003      FALSE:9001      FALSE:8720      FALSE:8926
##                  TRUE :2         TRUE :283       TRUE :77
##    Family          Genus           Species
##  Mode :logical   Mode :logical   Mode :logical
##  FALSE:8780      FALSE:8629      FALSE:7904
##  TRUE :223       TRUE :374       TRUE :1099
```

podemos ver los **TRUE** de cada nivel taxonomico,por ejemplo a nivel de "Phylum" tenemos solo 2 sin clasificar, y a nivel de "Specie" tenemos 1099 sin clasificar

ESTO ES LO QUE QUEREMOS COMPARAR CON LAS SALIDAS DE BRACKEN, YA QUE PROMETE HACER UNA REASISGNACION DE TODO LO QUE QUEDE SIN CLASIFICAR CON KRAKEN Queremos convertir las abundancias absolutas a relativas, Calculamos las abundancias relativas con la funcion, tanto de los datos originales como delos filtrados

```
percentages <- transform_sample_counts(fresa_kraken, function(x) x*100 / sum(x) )
head(percentages@otu_table@.Data)
```

```
##            MD2055    MD2056    MD2065     MD2066     MD2075     MD2076
## 2062    0.1531112 0.1337047 0.1277616 0.12826553 0.09536357 0.08823837
```

```
## 1883    3.1653683 2.9668623 3.3819572 3.02207329 2.82325711 2.43690517
## 2880933 0.2543846 0.3160277 0.1716471 0.19734986 0.07154704 0.11000235
## 1725411 0.1917928 0.1444277 0.1809853 0.22574349 0.16239425 0.06742039
## 2781734 0.1555646 0.1643498 0.1495185 0.15032451 0.20536471 0.04722477
## 659352  0.1322882 0.1013913 0.1062969 0.09431384 0.08101029 0.08015390
##             MD2085     MD2086     MD2095     MD2096     MD2105     MD2106
## 2062    0.13315067 0.09192153 0.12641372 0.12737471 0.12608124 0.12187615
## 1883    2.75749391 2.03547432 2.66386643 2.72155425 2.61035988 2.48274552
## 2880933 0.08798167 0.05986126 0.17062271 0.09673784 0.12733628 0.14141581
## 1725411 0.14190133 0.09567733 0.08915646 0.12301420 0.12184970 0.10542040
## 2781734 0.07582956 0.05173544 0.08305982 0.12376096 0.10120928 0.06261572
## 659352  0.12698435 0.10551492 0.08481441 0.09445091 0.08172215 0.07751718
##             MD2115     MD2116     MD2125    MD2126     MD2135    MD2136
## 2062    0.12157935 0.13369326 0.10235939 0.1249431 0.11316140 0.1356043
## 1883    2.49623089 2.74669909 2.42870477 2.8627815 2.88291689 3.0516160
## 2880933 0.19640446 0.14734609 0.15764997 0.1338337 0.12266592 0.2052372
## 1725411 0.14826117 0.16294243 0.15644124 0.2447490 0.27056275 0.2074082
## 2781734 0.07165779 0.10300654 0.13905223 0.1719331 0.23487962 0.1551592
## 659352  0.09626368 0.09998685 0.08516186 0.1146337 0.09656971 0.1267099
##             MP2047     MP2048    MP2049    MP2050     MP2057    MP2058
## 2062    0.12994779 0.13091334 0.1353569 0.1650252 0.13327945 0.1326403
## 1883    2.89789880 3.06943289 3.0976599 2.9566506 3.07528656 3.5872628
## 2880933 0.24140460 0.24960964 0.3567581 0.3820109 0.19714228 0.1649772
## 1725411 0.19986783 0.19189152 0.1822909 0.1163195 0.16083258 0.2897077
## 2781734 0.18310529 0.16248501 0.1854341 0.1073499 0.19543297 0.2227782
## 659352  0.08981334 0.09938283 0.1541549 0.1507321 0.09992167 0.1094893
##            MP2059    MP2060    MP2067    MP2068     MP2069     MP2070
## 2062    0.1623498 0.1957041 0.1248566 0.1398117 0.11208345 0.12093219
## 1883    3.1030898 4.3864889 2.8421815 2.7963203 2.27354026 2.39505275
## 2880933 0.2111680 0.2366568 0.1071719 0.1017879 0.09933368 0.07979230
## 1725411 0.1791156 0.3918289 0.1461338 0.1350162 0.08834094 0.09406729
## 2781734 0.1222224 0.2303246 0.1428249 0.1172284 0.07447130 0.05293542
## 659352  0.1068154 0.1145603 0.1185343 0.1010933 0.06839568 0.09067306
##             MP2077     MP2078     MP2079     MP2080    MP2087 MP2088    MP2089
## 2062    0.27124574 0.14217028 0.15566550 0.14435162 0.1344686      0 0.1361564
## 1883    3.05451937 2.88890702 3.29793653 3.08036501 2.8692023      0 3.0178133
## 2880933 0.08991617 0.08424271 0.13206763 0.11015129 0.1157698      0 0.1057929
## 1725411 0.16123151 0.12580403 0.16116273 0.14963043 0.1549171      0 0.1139521
## 2781734 0.06864322 0.06477979 0.09559819 0.09412289 0.1053299      0 0.1435256
## 659352  0.15777959 0.13256401 0.14882748 0.17304918 0.1344832      0 0.1050571
##             MP2090    MP2097     MP2098     MP2099     MP2100    MP2107
## 2062    0.11746673 0.1329179 0.12327746 0.12064103 0.13358241 0.1309823
## 1883    2.51088372 2.7648796 2.66239903 2.57223136 2.60150292 3.0951398
## 2880933 0.08851385 0.1573075 0.16191351 0.19151123 0.12725667 0.1974313
## 1725411 0.11094428 0.1426327 0.10347136 0.11859307 0.13364986 0.2122573
## 2781734 0.11140953 0.1237323 0.08825073 0.07174598 0.07075228 0.1302742
## 659352  0.09047327 0.1128820 0.07603687 0.08263924 0.11718347 0.1092200
##             MP2108     MP2109     MP2110    MP2117     MP2118    MP2119
## 2062    0.13983312 0.12956594 0.11584833 0.1359474 0.11563160 0.1307938
## 1883    2.92444137 3.09755796 2.69076756 2.9623807 2.34143292 2.8328090
## 2880933 0.20697025 0.16983739 0.21099030 0.2390446 0.16900527 0.2062158
## 1725411 0.14979147 0.15027087 0.18218399 0.1574857 0.12846521 0.1388464
## 2781734 0.08442305 0.09847296 0.13216418 0.2113118 0.11355158 0.1144649
## 659352  0.09848365 0.10139015 0.09909777 0.1119772 0.08738136 0.1002712
```

```
##            MP2120    MP2127    MP2128    MP2129    MP2130    MP2137     MP2138
## 2062    0.1378134 0.1460339 0.1412250 0.1368541 0.1352811 0.2033890 0.11853393
## 1883    3.1005460 3.1457805 2.9064064 2.9883903 3.0763508 3.7536859 2.47372674
## 2880933 0.2095718 0.3069133 0.1648945 0.3364155 0.3090665 0.1141615 0.06674832
## 1725411 0.1899285 0.1881781 0.2019477 0.1668242 0.1398803 0.1933878 0.20091404
## 2781734 0.1890265 0.1338902 0.1568944 0.1456942 0.1019571 0.1022245 0.12329037
## 659352  0.1210218 0.1863488 0.1247436 0.1123222 0.1972816 0.3263534 0.14036977
##             MP2139     MP2140
## 2062    0.11929696 0.12586592
## 1883    2.12709713 2.39521140
## 2880933 0.05186260 0.08951098
## 1725411 0.08297366 0.11445636
## 2781734 0.03882388 0.05412424
## 659352  0.09305715 0.10367327
```

```
percentages_fil <- transform_sample_counts(fresa_kraken_fil, function(x) x*100 / sum(x) )
head(percentages_fil@otu_table@.Data)
```

```
##            MD2055    MD2056    MD2065     MD2066     MD2075     MD2076
## 2062    0.1531112 0.1337047 0.1277616 0.12826553 0.09536357 0.08823837
## 1883    3.1653683 2.9668623 3.3819572 3.02207329 2.82325711 2.43690517
## 2880933 0.2543846 0.3160277 0.1716471 0.19734986 0.07154704 0.11000235
## 1725411 0.1917928 0.1444277 0.1809853 0.22574349 0.16239425 0.06742039
## 2781734 0.1555646 0.1643498 0.1495185 0.15032451 0.20536471 0.04722477
## 659352  0.1322882 0.1013913 0.1062969 0.09431384 0.08101029 0.08015390
##             MD2085     MD2086     MD2095     MD2096     MD2105     MD2106
## 2062    0.13315067 0.09192153 0.12641372 0.12737471 0.12608124 0.12187615
## 1883    2.75749391 2.03547432 2.66386643 2.72155425 2.61035988 2.48274552
## 2880933 0.08798167 0.05986126 0.17062271 0.09673784 0.12733628 0.14141581
## 1725411 0.14190133 0.09567733 0.08915646 0.12301420 0.12184970 0.10542040
## 2781734 0.07582956 0.05173544 0.08305982 0.12376096 0.10120928 0.06261572
## 659352  0.12698435 0.10551492 0.08481441 0.09445091 0.08172215 0.07751718
##             MD2115     MD2116     MD2125    MD2126     MD2135    MD2136
## 2062    0.12157935 0.13369326 0.10235939 0.1249431 0.11316140 0.1356043
## 1883    2.49623089 2.74669909 2.42870477 2.8627815 2.88291689 3.0516160
## 2880933 0.19640446 0.14734609 0.15764997 0.1338337 0.12266592 0.2052372
## 1725411 0.14826117 0.16294243 0.15644124 0.2447490 0.27056275 0.2074082
## 2781734 0.07165779 0.10300654 0.13905223 0.1719331 0.23487962 0.1551592
## 659352  0.09626368 0.09998685 0.08516186 0.1146337 0.09656971 0.1267099
##             MP2047     MP2048    MP2049    MP2050     MP2057    MP2058
## 2062    0.12994779 0.13091334 0.1353569 0.1650252 0.13327945 0.1326403
## 1883    2.89789880 3.06943289 3.0976599 2.9566506 3.07528656 3.5872628
## 2880933 0.24140460 0.24960964 0.3567581 0.3820109 0.19714228 0.1649772
## 1725411 0.19986783 0.19189152 0.1822909 0.1163195 0.16083258 0.2897077
## 2781734 0.18310529 0.16248501 0.1854341 0.1073499 0.19543297 0.2227782
## 659352  0.08981334 0.09938283 0.1541549 0.1507321 0.09992167 0.1094893
##            MP2059    MP2060    MP2067    MP2068     MP2069     MP2070
## 2062    0.1623498 0.1957041 0.1248566 0.1398117 0.11208345 0.12093219
## 1883    3.1030898 4.3864889 2.8421815 2.7963203 2.27354026 2.39505275
## 2880933 0.2111680 0.2366568 0.1071719 0.1017879 0.09933368 0.07979230
## 1725411 0.1791156 0.3918289 0.1461338 0.1350162 0.08834094 0.09406729
## 2781734 0.1222224 0.2303246 0.1428249 0.1172284 0.07447130 0.05293542
## 659352  0.1068154 0.1145603 0.1185343 0.1010933 0.06839568 0.09067306
##             MP2077     MP2078    MP2087    MP2089     MP2090    MP2097
```

```
## 2062    0.27124574 0.14217028 0.1344686 0.1361564 0.11746673 0.1329179
## 1883    3.05451937 2.88890702 2.8692023 3.0178133 2.51088372 2.7648796
## 2880933 0.08991617 0.08424271 0.1157698 0.1057929 0.08851385 0.1573075
## 1725411 0.16123151 0.12580403 0.1549171 0.1139521 0.11094428 0.1426327
## 2781734 0.06864322 0.06477979 0.1053299 0.1435256 0.11140953 0.1237323
## 659352  0.15777959 0.13256401 0.1344832 0.1050571 0.09047327 0.1128820
##             MP2098     MP2099     MP2100    MP2107     MP2108     MP2110
## 2062    0.12327746 0.12064103 0.13358241 0.1309823 0.13983312 0.11584833
## 1883    2.66239903 2.57223136 2.60150292 3.0951398 2.92444137 2.69076756
## 2880933 0.16191351 0.19151123 0.12725667 0.1974313 0.20697025 0.21099030
## 1725411 0.10347136 0.11859307 0.13364986 0.2122573 0.14979147 0.18218399
## 2781734 0.08825073 0.07174598 0.07075228 0.1302742 0.08442305 0.13216418
## 659352  0.07603687 0.08263924 0.11718347 0.1092200 0.09848365 0.09909777
##            MP2117     MP2118    MP2119    MP2120    MP2127    MP2128    MP2129
## 2062    0.1359474 0.11563160 0.1307938 0.1378134 0.1460339 0.1412250 0.1368541
## 1883    2.9623807 2.34143292 2.8328090 3.1005460 3.1457805 2.9064064 2.9883903
## 2880933 0.2390446 0.16900527 0.2062158 0.2095718 0.3069133 0.1648945 0.3364155
## 1725411 0.1574857 0.12846521 0.1388464 0.1899285 0.1881781 0.2019477 0.1668242
## 2781734 0.2113118 0.11355158 0.1144649 0.1890265 0.1338902 0.1568944 0.1456942
## 659352  0.1119772 0.08738136 0.1002712 0.1210218 0.1863488 0.1247436 0.1123222
##            MP2130     MP2138     MP2139     MP2140
## 2062    0.1352811 0.11853393 0.11929696 0.12586592
## 1883    3.0763508 2.47372674 2.12709713 2.39521140
## 2880933 0.3090665 0.06674832 0.05186260 0.08951098
## 1725411 0.1398803 0.20091404 0.08297366 0.11445636
## 2781734 0.1019571 0.12329037 0.03882388 0.05412424
## 659352  0.1972816 0.14036977 0.09305715 0.10367327
```

Ahora ya es posibleuna buena comparacion de abundancias dadas por porcentajes con indices beta

Diversidad Beta

La diversidad beta mide la diferencia entre dos o mas entornos. Se puede medir con métricas como la disimilitud de Bray-Curtis, la distancia Jaccard o la distancia UniFrac.

Mide que tan similares o diferentes son un par de especies, muestras o conjuntos de muestras. aqui podemos ver una lista de distancias disponibles, que Phyloseq puede usar

```
distanceMethodList
```

```
## $UniFrac
## [1] "unifrac"  "wunifrac"
##
## $DPCoA
## [1] "dpcoa"
##
## $JSD
## [1] "jsd"
##
## $vegdist
##  [1] "manhattan"  "euclidean"  "canberra"   "bray"       "kulczynski"
##  [6] "jaccard"    "gower"      "altGower"   "morisita"   "horn"
## [11] "mountford"  "raup"       "binomial"   "chao"       "cao"
```

```
##
## $betadiver
##  [1] "w"   "-1"  "c"   "wb"  "r"   "I"   "e"   "t"   "me"  "j"   "sor" "m"
## [13] "-2"  "co"  "cc"  "g"   "-3"  "l"   "19"  "hk"  "rlb" "sim" "gl"  "z"
##
## $dist
## [1] "maximum"   "binary"    "minkowski"
##
## $designdist
## [1] "ANY"
```

Siendo las siguientes las mas usadas: + Disimilitud de Bray-Curtis

- Distancia Jaccard
- UniFrac

```
#vegdist(meta_ord_fil,"bray")# usa la libreria Vegan,  REVISAR COMO FUNCIONA PARA IMPRIMIR LA TABLA
```

Usamos "ordinate" para asignar las distancias entre muestras,usando "Bray-Curtis", ya que es una de las metricas mas completas y mayormente utilizadas para medir la diversidad beta

Hay diferentes formas de trazar y mostrar los resultados de dicho análisis. Entre otros, se utilizan ampliamente los análisis PCA, PCoA o NMDS.

```
meta_ord <- ordinate(physeq = percentages, method = "NMDS", distance = "bray")
```

```
## Wisconsin double standardization
## Run 0 stress 8.456249e-05
## Run 1 stress 8.730444e-05
## ... Procrustes: rmse 5.846526e-05  max resid 0.0002085156
## ... Similar to previous best
## Run 2 stress 9.346884e-05
## ... Procrustes: rmse 5.528965e-05  max resid 0.0001282282
## ... Similar to previous best
## Run 3 stress 6.626938e-05
## ... New best solution
## ... Procrustes: rmse 3.757661e-05  max resid 0.0001847309
## ... Similar to previous best
## Run 4 stress 9.35411e-05
## ... Procrustes: rmse 6.408873e-05  max resid 0.00020861
## ... Similar to previous best
## Run 5 stress 9.460451e-05
## ... Procrustes: rmse 5.258747e-05  max resid 0.0001477321
## ... Similar to previous best
## Run 6 stress 9.416872e-05
## ... Procrustes: rmse 4.99617e-05  max resid 0.0001476149
## ... Similar to previous best
## Run 7 stress 9.861422e-05
## ... Procrustes: rmse 5.588722e-05  max resid 0.0001753003
## ... Similar to previous best
## Run 8 stress 9.20651e-05
## ... Procrustes: rmse 4.040013e-05  max resid 0.0001907229
```

```
## ... Similar to previous best
## Run 9 stress 9.359642e-05
## ... Procrustes: rmse 6.307103e-05  max resid 0.0002722143
## ... Similar to previous best
## Run 10 stress 9.300692e-05
## ... Procrustes: rmse 5.323925e-05  max resid 0.0001645599
## ... Similar to previous best
## Run 11 stress 8.874184e-05
## ... Procrustes: rmse 5.797053e-05  max resid 0.000195058
## ... Similar to previous best
## Run 12 stress 9.706551e-05
## ... Procrustes: rmse 4.587631e-05  max resid 0.0001442516
## ... Similar to previous best
## Run 13 stress 9.835124e-05
## ... Procrustes: rmse 5.595098e-05  max resid 0.0001620877
## ... Similar to previous best
## Run 14 stress 9.513722e-05
## ... Procrustes: rmse 6.185097e-05  max resid 0.0002543875
## ... Similar to previous best
## Run 15 stress 9.767676e-05
## ... Procrustes: rmse 4.769726e-05  max resid 0.0001773738
## ... Similar to previous best
## Run 16 stress 9.037714e-05
## ... Procrustes: rmse 6.313906e-05  max resid 0.0001957713
## ... Similar to previous best
## Run 17 stress 9.720465e-05
## ... Procrustes: rmse 6.969577e-05  max resid 0.0002828839
## ... Similar to previous best
## Run 18 stress 9.669563e-05
## ... Procrustes: rmse 6.232526e-05  max resid 0.0001738937
## ... Similar to previous best
## Run 19 stress 9.570455e-05
## ... Procrustes: rmse 5.551977e-05  max resid 0.0001644574
## ... Similar to previous best
## Run 20 stress 9.363498e-05
## ... Procrustes: rmse 5.804502e-05  max resid 0.0001658404
## ... Similar to previous best
## *** Best solution repeated 18 times

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

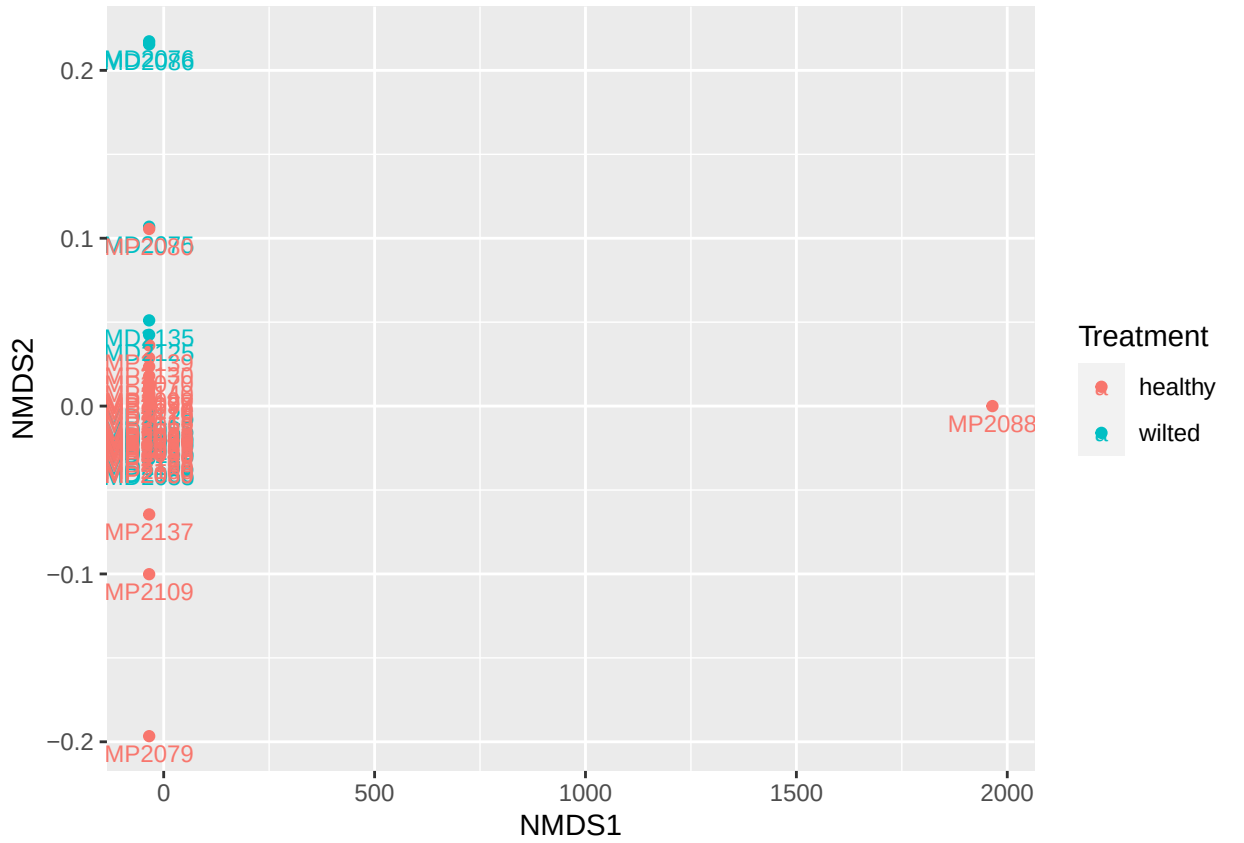
```
meta_ord_fil <- ordinate(physeq = percentages_fil, method = "NMDS", distance = "bray")
```

```
## Wisconsin double standardization
## Run 0 stress 0.1686115
## Run 1 stress 0.1720959
## Run 2 stress 0.2021837
## Run 3 stress 0.1646876
## ... New best solution
## ... Procrustes: rmse 0.05972879  max resid 0.2657492
## Run 4 stress 0.1665887
## Run 5 stress 0.1668561
```

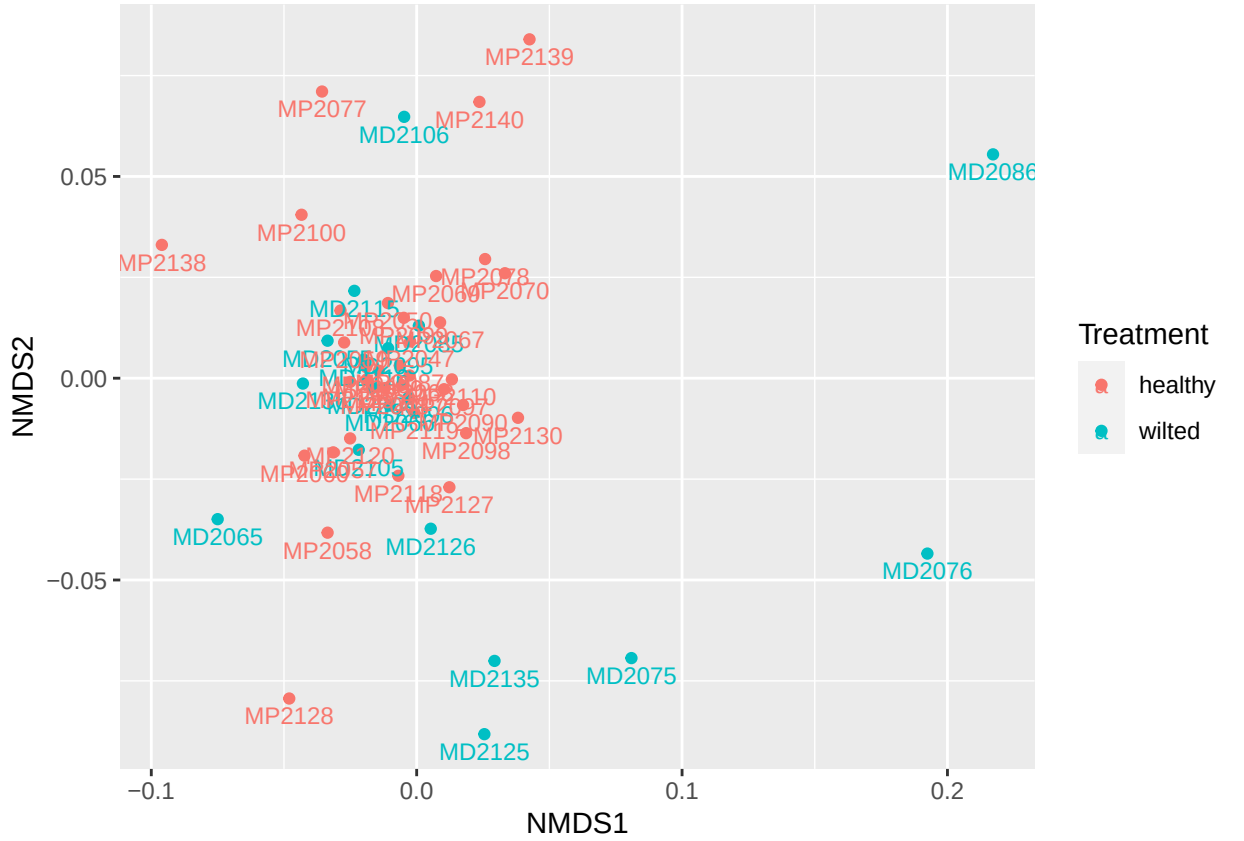
```
## Run 6 stress 0.1939999
## Run 7 stress 0.1639277
## ... New best solution
## ... Procrustes: rmse 0.0442278  max resid 0.1798766
## Run 8 stress 0.1668603
## Run 9 stress 0.170307
## Run 10 stress 0.1639272
## ... New best solution
## ... Procrustes: rmse 0.003506486  max resid 0.016453
## Run 11 stress 0.1798568
## Run 12 stress 0.1635176
## ... New best solution
## ... Procrustes: rmse 0.03481015  max resid 0.1533124
## Run 13 stress 0.1768353
## Run 14 stress 0.1788165
## Run 15 stress 0.1697222
## Run 16 stress 0.1636949
## ... Procrustes: rmse 0.0165617  max resid 0.0518228
## Run 17 stress 0.1663009
## Run 18 stress 0.1800728
## Run 19 stress 0.16661
## Run 20 stress 0.1637885
## ... Procrustes: rmse 0.007302641  max resid 0.04717434
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      4: no. of iterations >= maxit
##     16: stress ratio > sratmax
```

Ademas lo queremos diferenciar por color entre plantas sanas y enfermas,

```
plot_ordination(physeq = percentages, ordination = meta_ord, color = "Treatment") +
  geom_text(mapping = aes(label = colnames(fresa_kraken@otu_table@.Data)), size = 3, vjust = 1.5)
```




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



En este gráfico NMDS, cada punto representa la abundancia combinada de todas sus OTU. No se puiede ver una diferencia entre los conjuntos de muestras de plantas sanas y muestras enfermas, por lo tanto: Probaremos varias distancias ya con los datos filtrados podemos ver varios ejemplos y llegar a la posibilidad que se diferencien un poco los dos conjuntos de datos

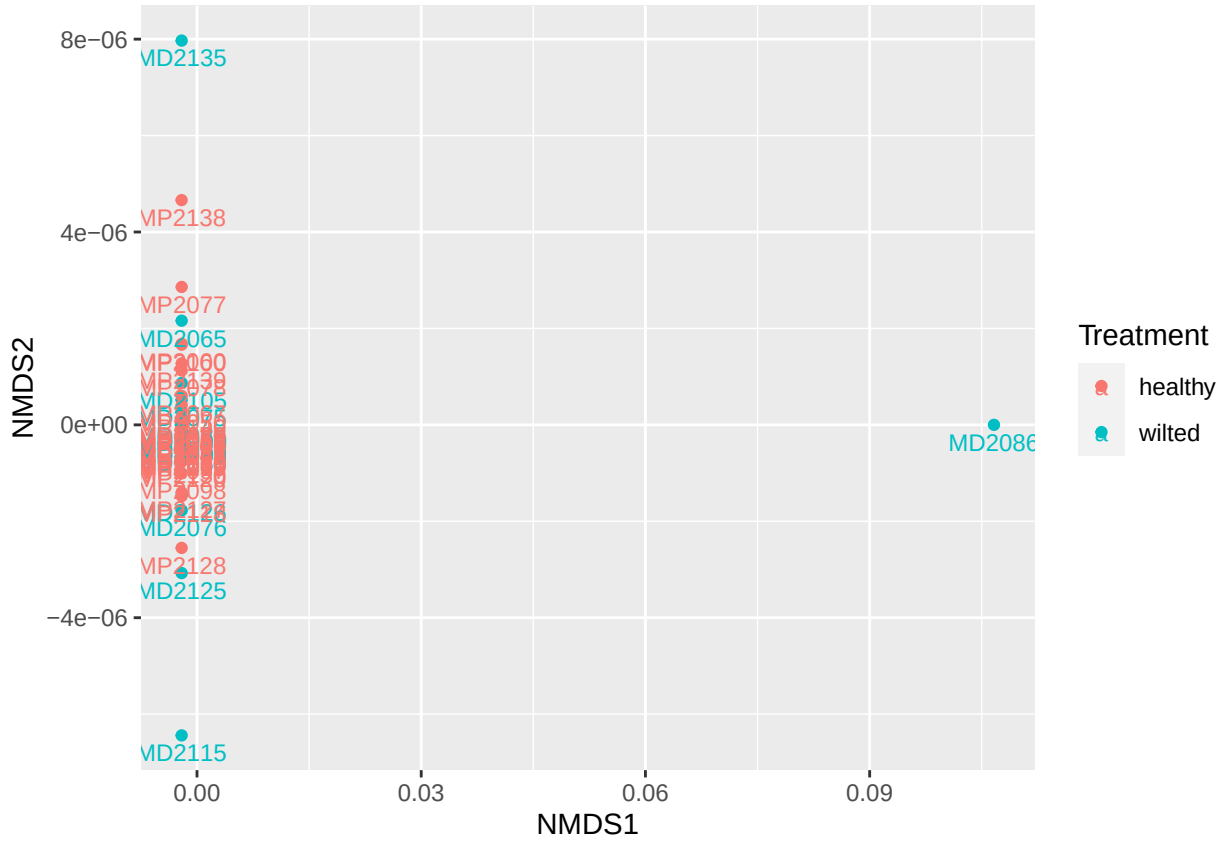
Usando

```
meta_ord_fil_2 <- ordinate(physeq = percentages_fil, method = "NMDS", distance = "jsd")
```

```
## Run 0 stress 9.277469e-05
## Run 1 stress 9.453648e-05
## ... Procrustes: rmse 6.849104e-05  max resid 0.0002878395
## ... Similar to previous best
## Run 2 stress 9.236093e-05
## ... New best solution
## ... Procrustes: rmse 5.475432e-05  max resid 0.0001708646
## ... Similar to previous best
## Run 3 stress 9.508889e-05
## ... Procrustes: rmse 7.576757e-05  max resid 0.0003657582
## ... Similar to previous best
## Run 4 stress 9.373873e-05
## ... Procrustes: rmse 7.313094e-05  max resid 0.0002071405
## ... Similar to previous best
## Run 5 stress 9.787758e-05
```

```
## ... Procrustes: rmse 8.831411e-05  max resid 0.0002825702
## ... Similar to previous best
## Run 6 stress 9.463442e-05
## ... Procrustes: rmse 9.062191e-05  max resid 0.0003763376
## ... Similar to previous best
## Run 7 stress 9.154183e-05
## ... New best solution
## ... Procrustes: rmse 8.821241e-05  max resid 0.0003372005
## ... Similar to previous best
## Run 8 stress 0.4030179
## Run 9 stress 9.354169e-05
## ... Procrustes: rmse 8.064795e-05  max resid 0.0003218632
## ... Similar to previous best
## Run 10 stress 9.821504e-05
## ... Procrustes: rmse 7.190303e-05  max resid 0.0002683282
## ... Similar to previous best
## Run 11 stress 6.322483e-05
## ... New best solution
## ... Procrustes: rmse 6.264448e-05  max resid 0.0002140281
## ... Similar to previous best
## Run 12 stress 9.429718e-05
## ... Procrustes: rmse 5.499954e-05  max resid 0.0001841124
## ... Similar to previous best
## Run 13 stress 9.102691e-05
## ... Procrustes: rmse 6.85014e-05  max resid 0.0002222926
## ... Similar to previous best
## Run 14 stress 9.927494e-05
## ... Procrustes: rmse 5.268635e-05  max resid 0.0001937807
## ... Similar to previous best
## Run 15 stress 8.927591e-05
## ... Procrustes: rmse 6.346448e-05  max resid 0.0002516416
## ... Similar to previous best
## Run 16 stress 8.888119e-05
## ... Procrustes: rmse 5.620376e-05  max resid 0.0001832829
## ... Similar to previous best
## Run 17 stress 9.732597e-05
## ... Procrustes: rmse 6.883855e-05  max resid 0.0002839787
## ... Similar to previous best
## Run 18 stress 9.821188e-05
## ... Procrustes: rmse 8.005894e-05  max resid 0.0002698677
## ... Similar to previous best
## Run 19 stress 9.052606e-05
## ... Procrustes: rmse 6.720451e-05  max resid 0.0002285255
## ... Similar to previous best
## Run 20 stress 9.063654e-05
## ... Procrustes: rmse 5.500878e-05  max resid 0.0001959764
## ... Similar to previous best
## *** Best solution repeated 10 times

## Warning in metaMDS(ps.dist): stress is (nearly) zero: you may have insufficient
## data
```

```
plot_ordination(physeq = percentages_fil, ordination = meta_ord_fil_2, color = "Treatment") +
  geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@.Data)), size = 3, vjust = 1.5)
```



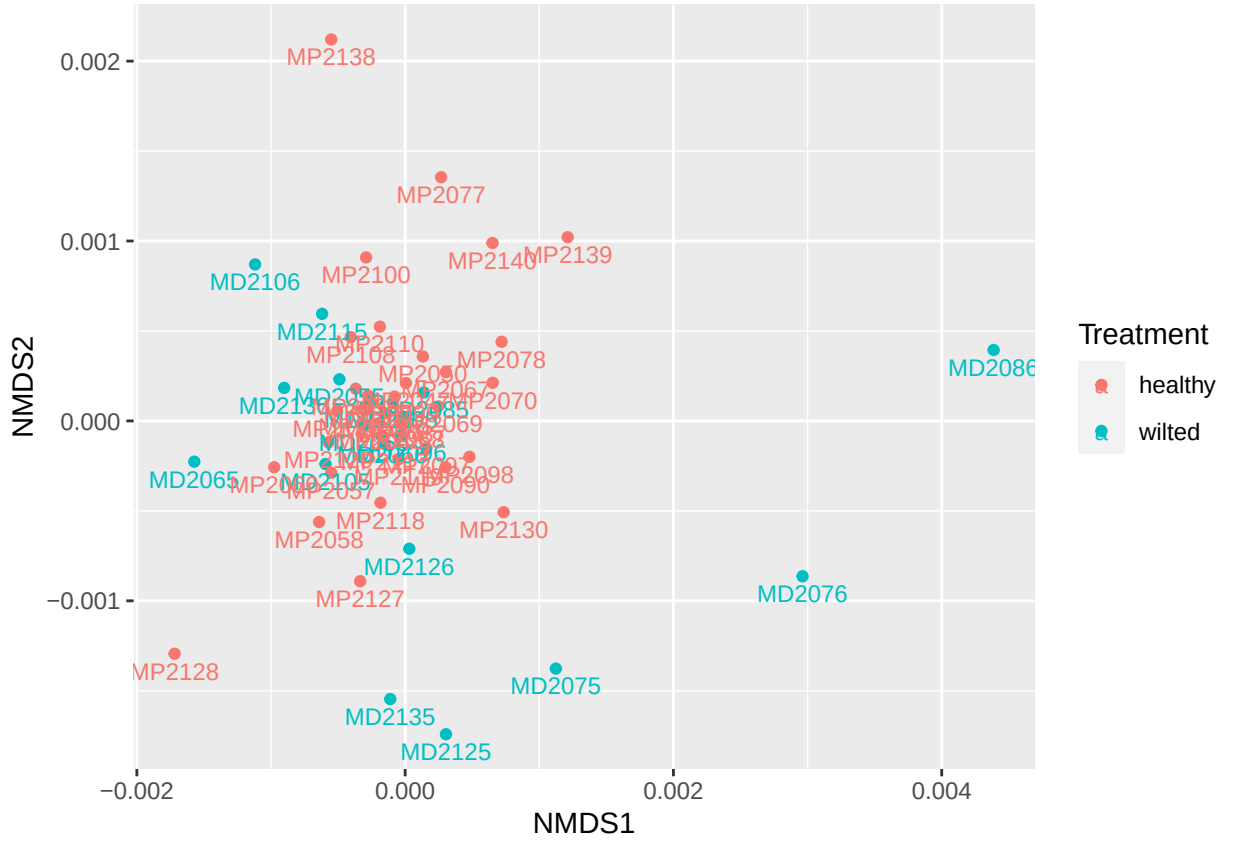
Usando

```
meta_ord_fil_3 <- ordinate(physeq = percentages_fil, method = "NMDS", distance = "euclidean")
```

```
## Wisconsin double standardization
## Run 0 stress 0.1866907
## Run 1 stress 0.1961377
## Run 2 stress 0.1979378
## Run 3 stress 0.1876519
## Run 4 stress 0.1810222
## ... New best solution
## ... Procrustes: rmse 0.09243898  max resid 0.4393109
## Run 5 stress 0.1898393
## Run 6 stress 0.1779441
## ... New best solution
## ... Procrustes: rmse 0.05095054  max resid 0.2104657
## Run 7 stress 0.1891111
## Run 8 stress 0.1901222
## Run 9 stress 0.1771694
## ... New best solution
## ... Procrustes: rmse 0.044194  max resid 0.2059813
## Run 10 stress 0.1867635
## Run 11 stress 0.2014674
```

```
## Run 12 stress 0.2079774
## Run 13 stress 0.2065766
## Run 14 stress 0.2050305
## Run 15 stress 0.1914003
## Run 16 stress 0.1779186
## Run 17 stress 0.1812592
## Run 18 stress 0.1836356
## Run 19 stress 0.1765558
## ... New best solution
## ... Procrustes: rmse 0.02898278  max resid 0.1375525
## Run 20 stress 0.1913343
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      2: no. of iterations >= maxit
##     18: stress ratio > sratmax
```

```
plot_ordination(physeq = percentages_fil, ordination = meta_ord_fil_3, color = "Treatment") +
  geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@.Data)), size = 3, vjust = 1.5)
```



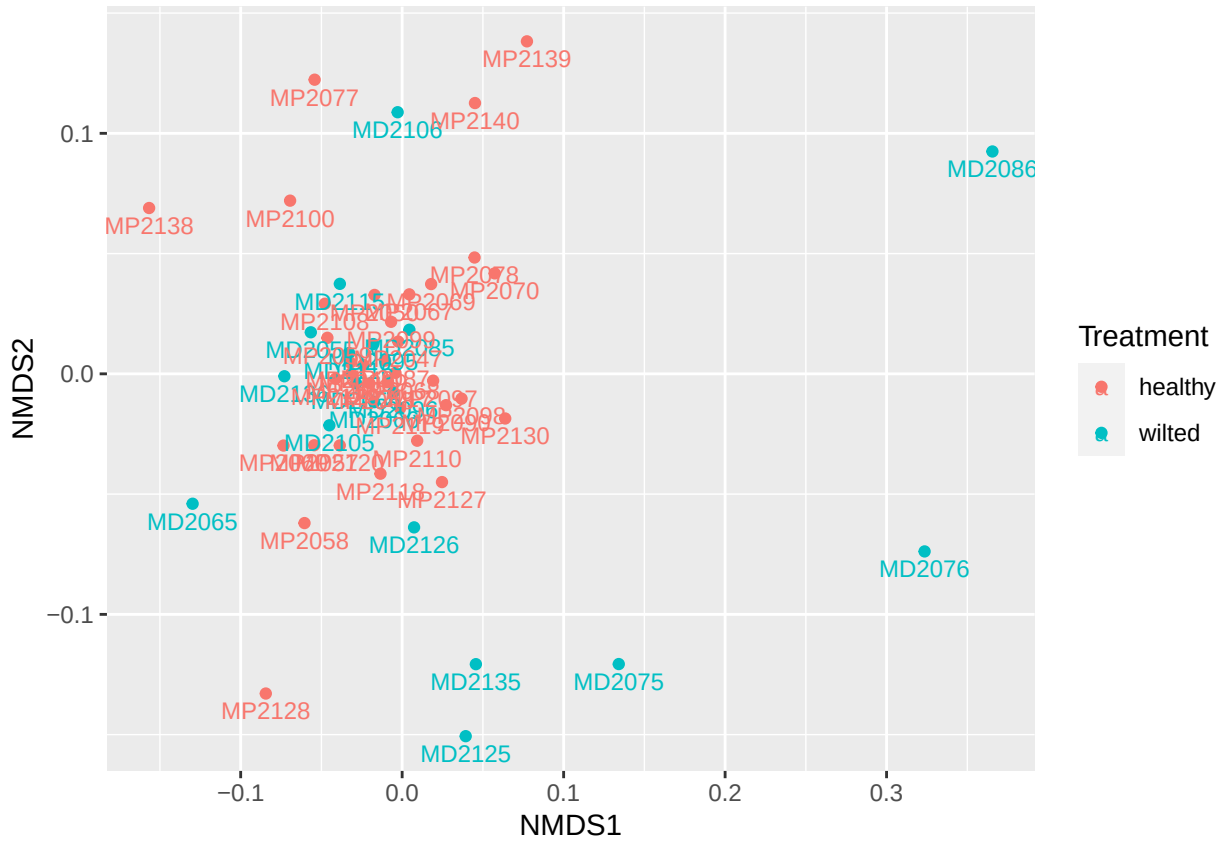
Usando

```
meta_ord_fil_4 <- ordinate(physeq = percentages_fil, method = "NMDS", distance = "jaccard")
```

```
## Wisconsin double standardization
## Run 0 stress 0.168611
## Run 1 stress 0.169946
## Run 2 stress 0.1677187
```

```
## ... New best solution
## ... Procrustes: rmse 0.03554263  max resid 0.1588795
## Run 3 stress 0.1676929
## ... New best solution
## ... Procrustes: rmse 0.04220422  max resid 0.2179595
## Run 4 stress 0.1665993
## ... New best solution
## ... Procrustes: rmse 0.08434842  max resid 0.3423987
## Run 5 stress 0.1797529
## Run 6 stress 0.1781481
## Run 7 stress 0.1651799
## ... New best solution
## ... Procrustes: rmse 0.04704966  max resid 0.2148221
## Run 8 stress 0.1665887
## Run 9 stress 0.1705237
## Run 10 stress 0.1675346
## Run 11 stress 0.163308
## ... New best solution
## ... Procrustes: rmse 0.03803754  max resid 0.1521945
## Run 12 stress 0.171941
## Run 13 stress 0.1657738
## Run 14 stress 0.1680284
## Run 15 stress 0.1766145
## Run 16 stress 0.1737255
## Run 17 stress 0.1746764
## Run 18 stress 0.1704801
## Run 19 stress 0.1649569
## Run 20 stress 0.1678345
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      1: no. of iterations >= maxit
##     19: stress ratio > sratmax
```

```
plot_ordination(physeq = percentages_fil, ordination = meta_ord_fil_4, color = "Treatment") +
  geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@.Data)), size = 3, vjust = 1.5)
```



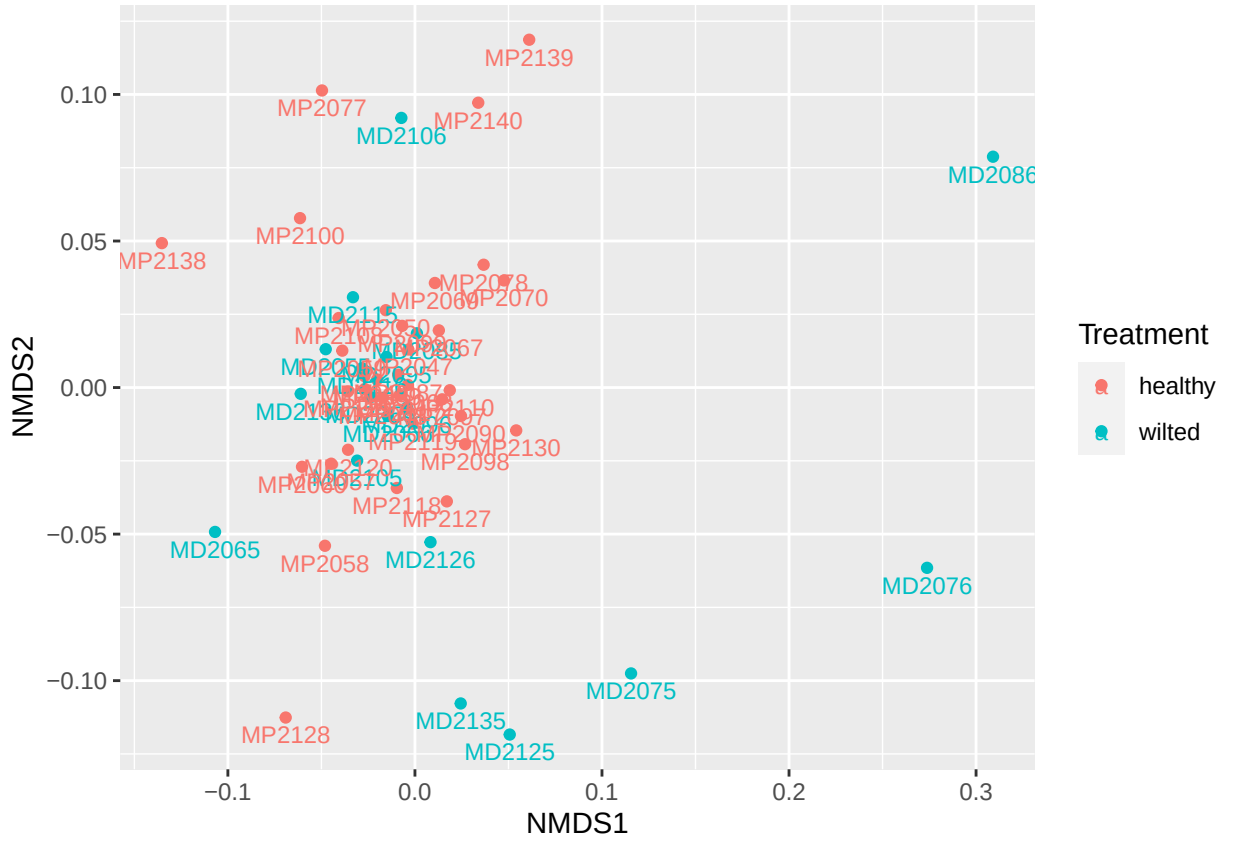
Usando

```
meta_ord_fil_5 <- ordinate(physeq = percentages_fil, method = "NMDS", distance = "manhattan")
```

```
## Wisconsin double standardization
## Run 0 stress 0.1686118
## Run 1 stress 0.1660961
## ... New best solution
## ... Procrustes: rmse 0.065342  max resid 0.3487895
## Run 2 stress 0.2078356
## Run 3 stress 0.1701508
## Run 4 stress 0.1707507
## Run 5 stress 0.1777985
## Run 6 stress 0.1674247
## Run 7 stress 0.1775214
## Run 8 stress 0.1665992
## Run 9 stress 0.1666117
## Run 10 stress 0.1634885
## ... New best solution
## ... Procrustes: rmse 0.06032723  max resid 0.3426419
## Run 11 stress 0.1790928
## Run 12 stress 0.1687607
## Run 13 stress 0.167619
## Run 14 stress 0.1693476
## Run 15 stress 0.1955763
## Run 16 stress 0.1688576
```

```
## Run 17 stress 0.1723263
## Run 18 stress 0.1824605
## Run 19 stress 0.1775322
## Run 20 stress 0.1734123
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      2: no. of iterations >= maxit
##     18: stress ratio > sratmax
```

```
plot_ordination(physeq = percentages_fil, ordination = meta_ord_fil_5, color = "Treatment") +
  geom_text(mapping = aes(label = colnames(fresa_kraken_fil@otu_table@.Data)), size = 3, vjust = 1.5)
```



Con ninguna distancia vemos una diferenciacion clara, por lo que lo veremos por los distintos niveles taxonomicos

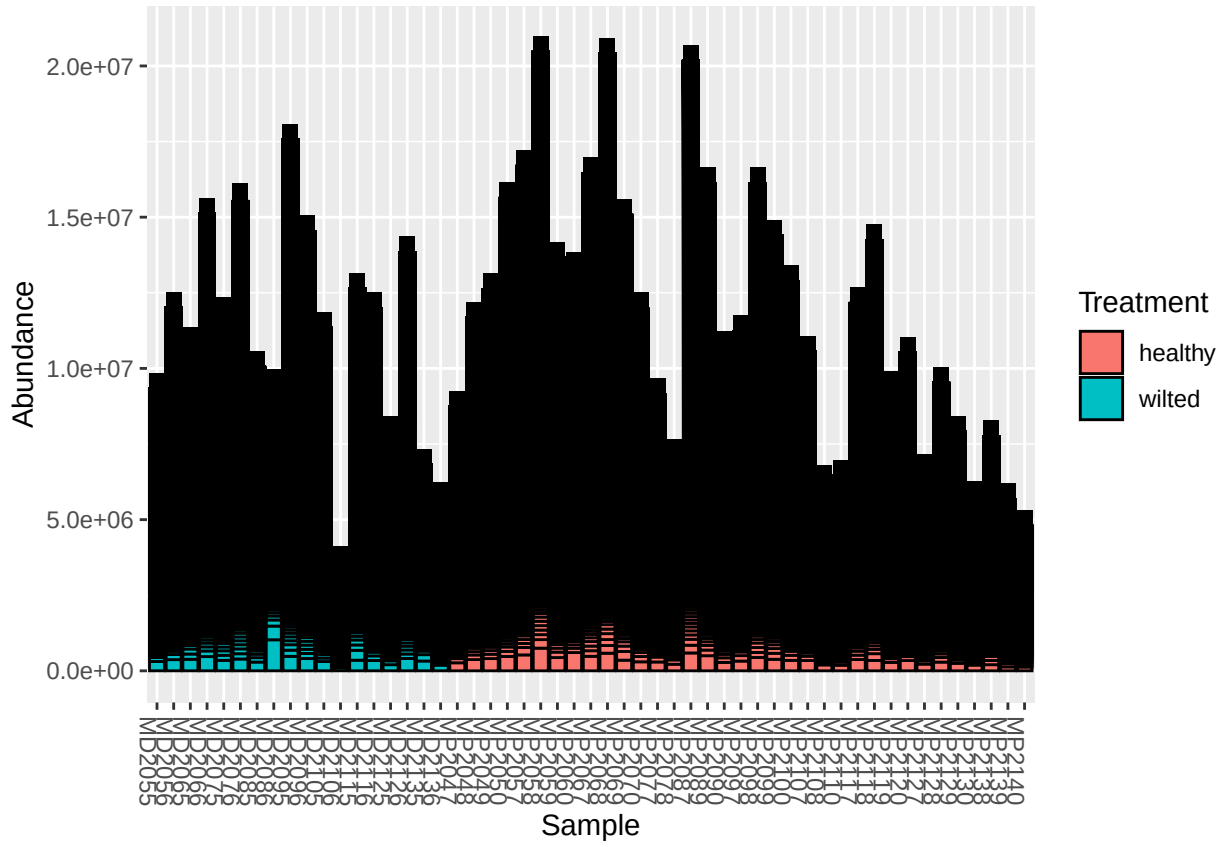
AHORA QUEREMOS VER DIFERENCIAS ENTRE DIFERENTES NIVELES TAXONOMICOS

De aqui en adelante solo trabajaremos con los datos filtrados,

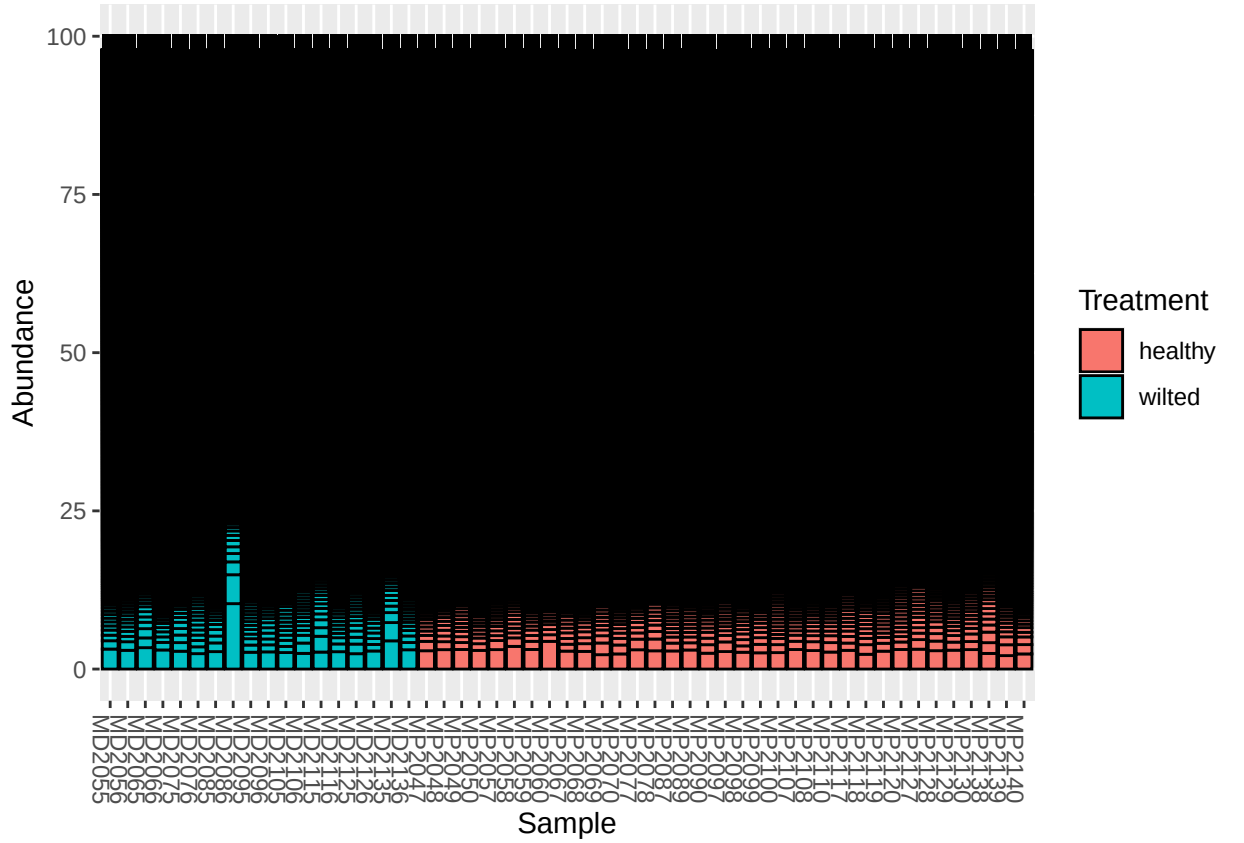
Grafico de barras de abundancia

Trazando las muestras en el eje x y las abundancias en el eje y Los valores de abundancia para cada OTUen cada muestra se apilan en el orden de mayor a menor, separados por una fina línea horizontal.

```
plot_bar(fresa_kraken_fil,fill="Treatment")
```




```
plot_bar(percentages_fil,fill="Treatment")
```



Queremos explorar nuestras muestras a diferentes niveles taxonómicos específicos
A nivel de Kingdom

```
percentages_glom_kingdom <- tax_glom(percentages_fil, taxrank = 'Kingdom')
##View(percentages_glom_kingdom@tax_table@.Data)
```

A nivel de Phylum

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

Creamos un dataframe con los porcentajes de phylum

```
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" ...
```

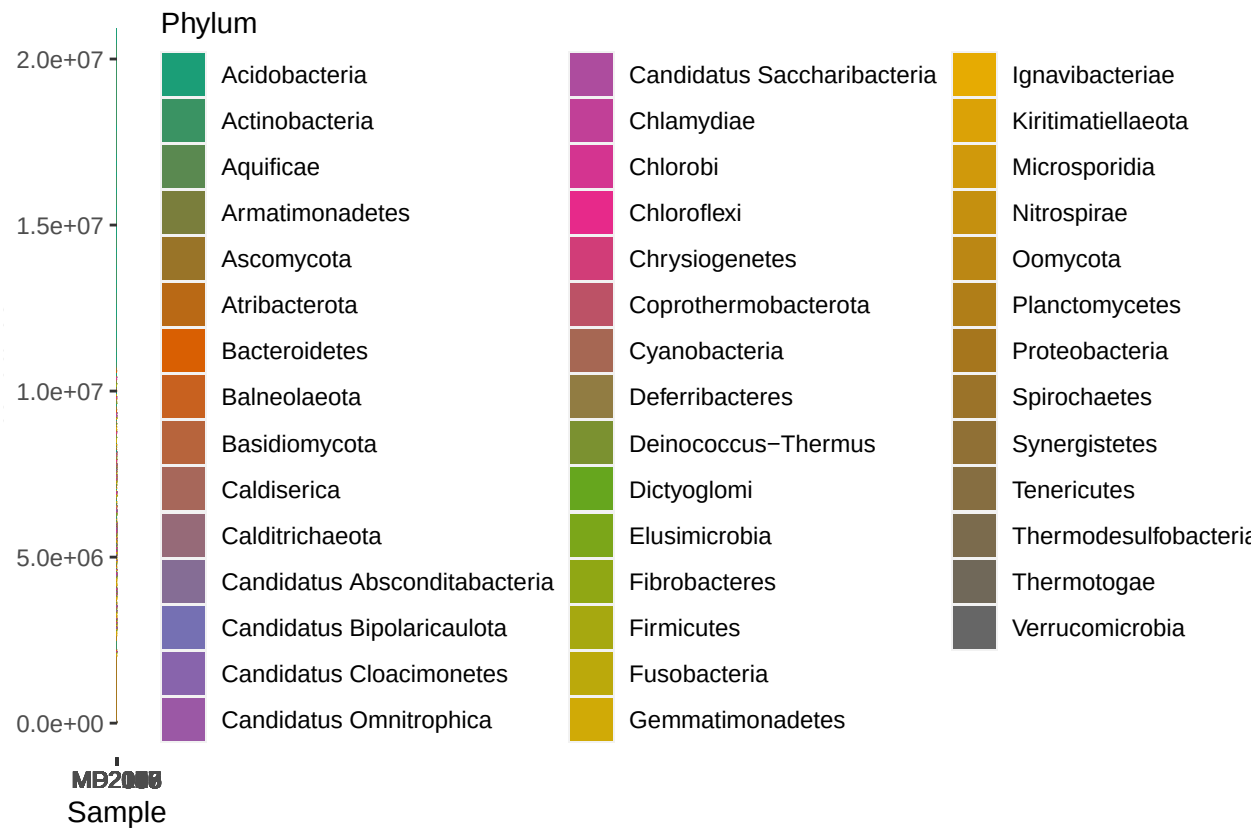
```
absolute_glom_phylum <- tax_glom(physeq = fresa_kraken_fil, taxrank = "Phylum")
absolute_df_phylum <- psmelt(absolute_glom_phylum)
str(absolute_df_phylum)
```

```
## '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" ...
```

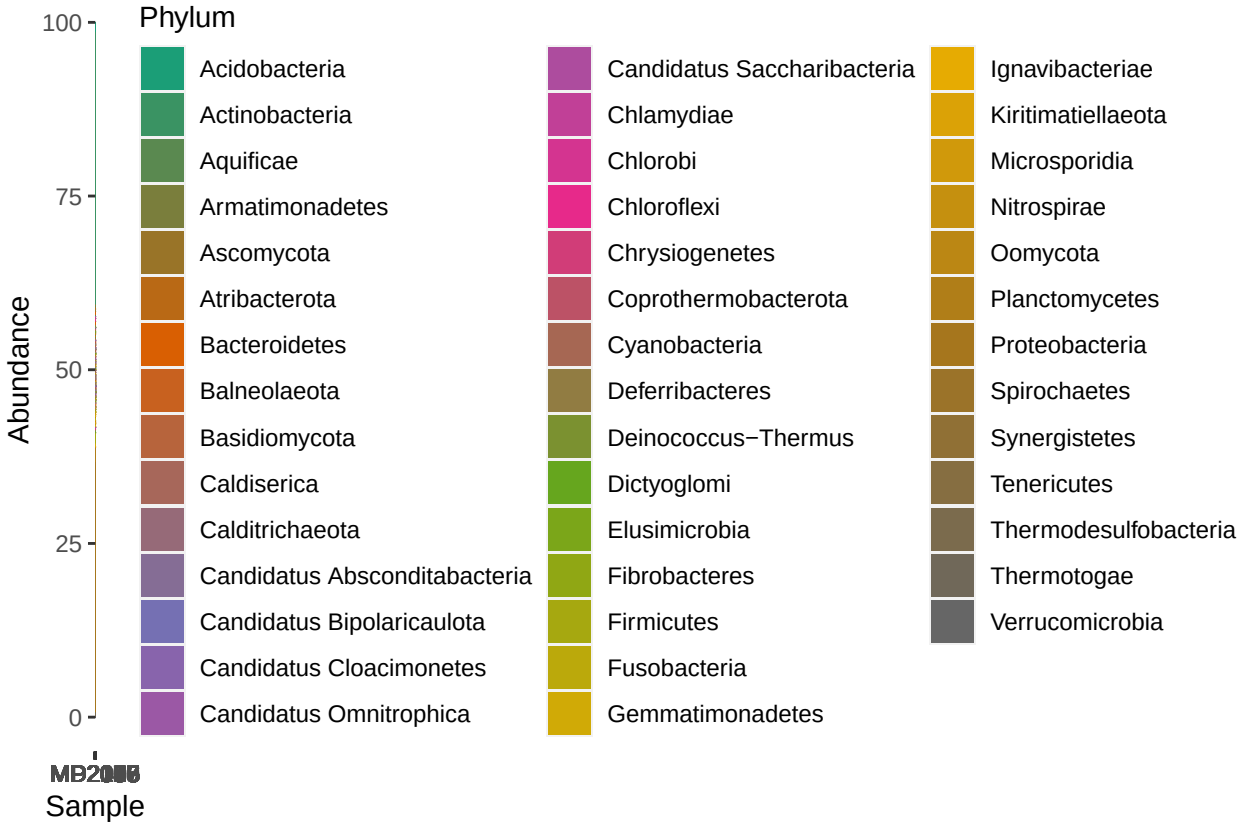
```
absolute_df_phylum$Phylum <- as.factor(absolute_df_phylum$Phylum)
phylum_colors_abs <- colorRampPalette(brewer.pal(8,"Dark2")) (length(levels(absolute_df_phylum$Phylum)))
```

```
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)
```

```
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)

absolute_plot
```




```
relative_plot
```



Usaremos 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"                   "Aquificae"
## [31] "Thermodesulfobacteria"         "Deferribacteres"
## [33] "Fusobacteria"                  "Chrysiogenetes"
```

```
## [35] "Calditrichaeota"              "Elusimicrobia"
## [37] "Caldiserica"                  "Coprothermobacterota"
## [39] "Atribacterota"                "Dictyoglomi"
## [41] "Ascomycota"                   "Basidiomycota"
## [43] "Microsporidia"                "Oomycota"
```

con esto 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
```

Diversidad Beta

veremos aqui 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)
```

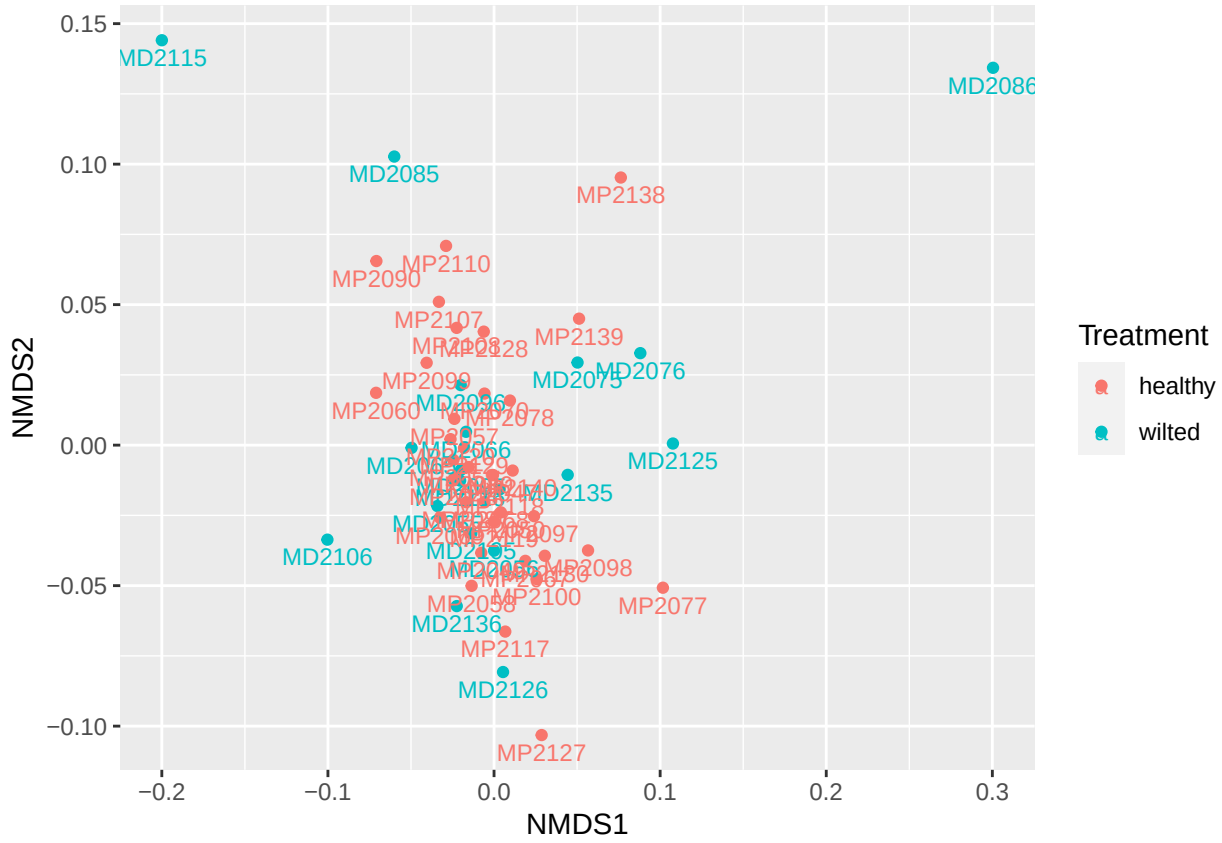
beta diversidad de Eukaryota

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

```
## Wisconsin double standardization
## Run 0 stress 0.1341389
## Run 1 stress 0.1558514
## Run 2 stress 0.1636495
## Run 3 stress 0.1368379
## Run 4 stress 0.1392503
## Run 5 stress 0.1611309
## Run 6 stress 0.1711303
## Run 7 stress 0.1348056
## Run 8 stress 0.1394637
## Run 9 stress 0.1394555
## Run 10 stress 0.1392747
## Run 11 stress 0.1365988
## Run 12 stress 0.1359144
## Run 13 stress 0.1644409
```

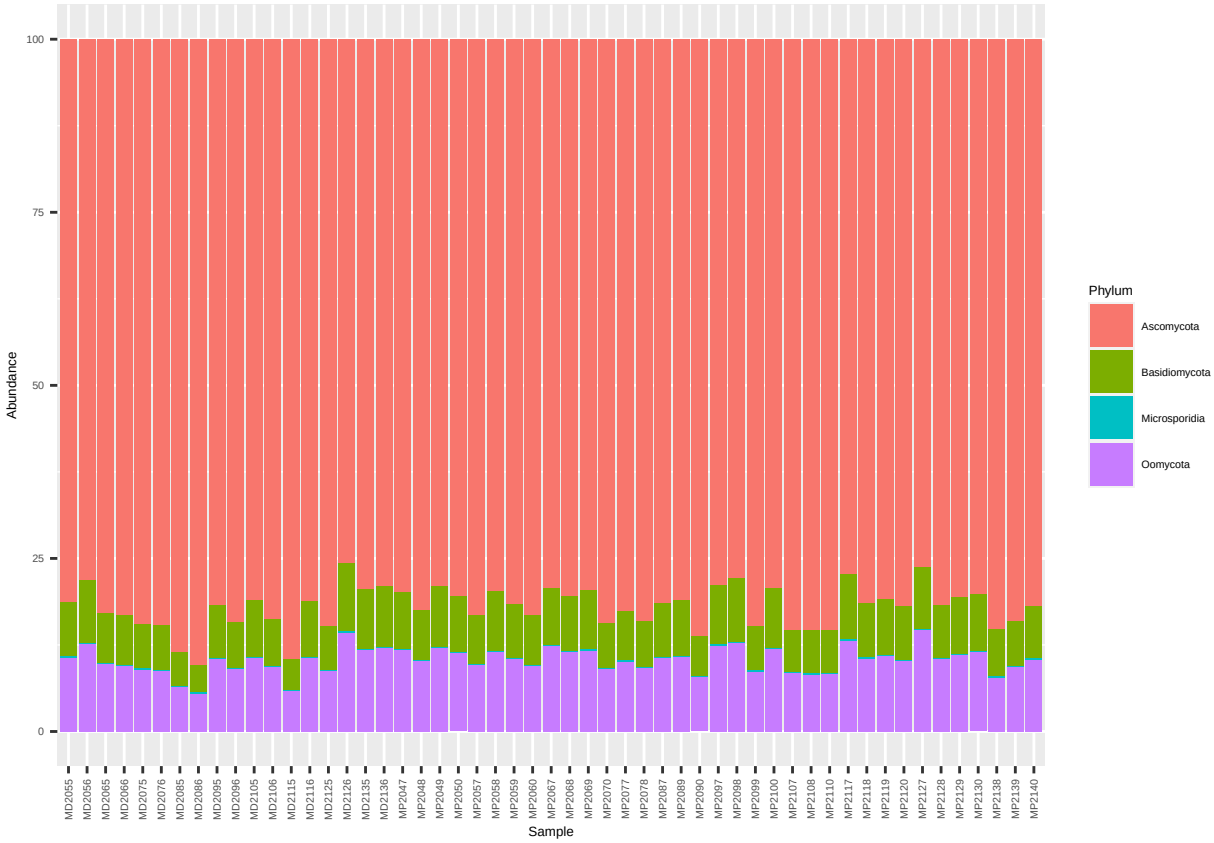
```
## Run 14 stress 0.1341975
## ... Procrustes: rmse 0.01193448  max resid 0.08063325
## Run 15 stress 0.1394645
## Run 16 stress 0.1490905
## Run 17 stress 0.1398812
## Run 18 stress 0.1657792
## Run 19 stress 0.1616037
## Run 20 stress 0.1708762
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      3: no. of iterations >= maxit
##     17: stress ratio > sratmax
```

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

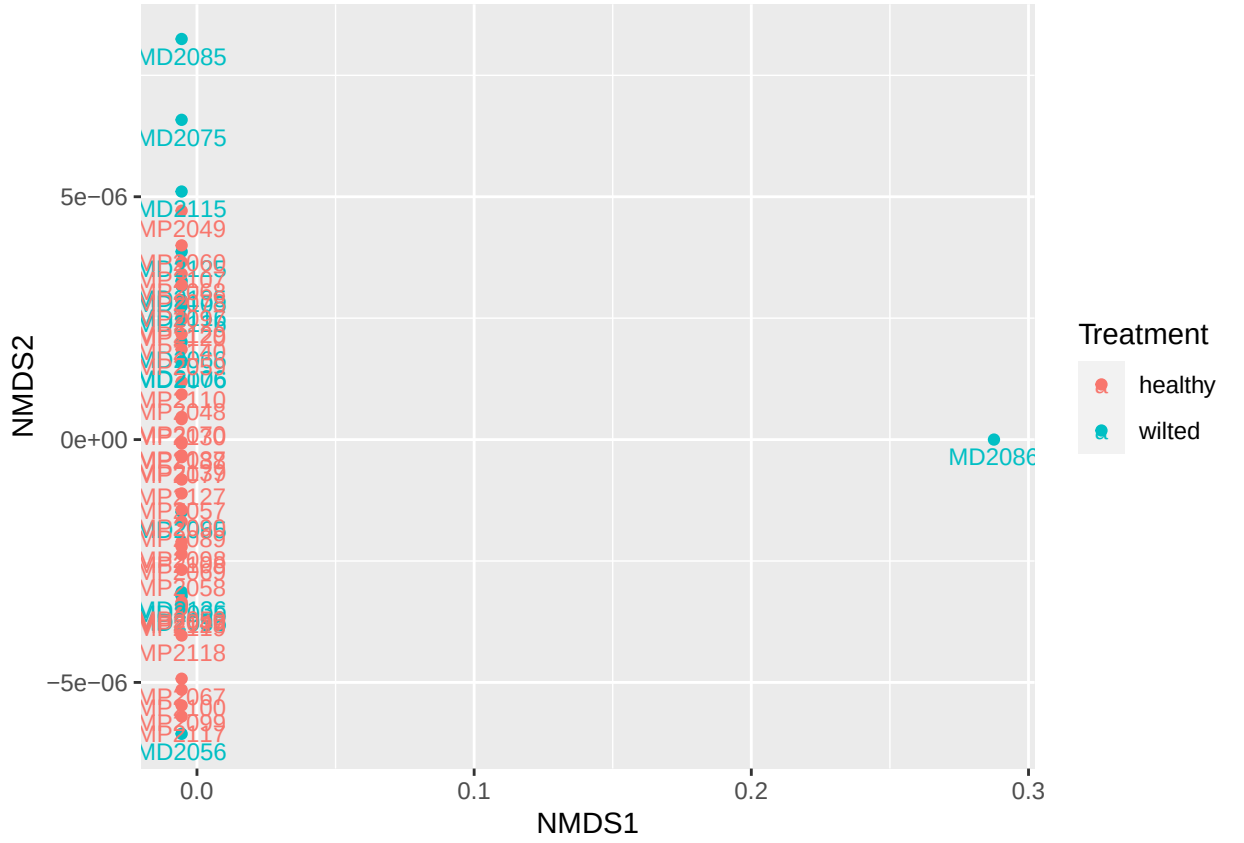
sacamos las abundancias relativas

```
percentages_Eukaryota_Phylum <- transform_sample_counts(merge_Eukaryota_Phylum, function(x) x*100 / sum
percentages_Eukaryota_Phylum_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 9.833358e-05
## ... New best solution
## ... Procrustes: rmse 0.03458091  max resid 0.0906527
## Run 2 stress 9.964059e-05
## ... Procrustes: rmse 0.0001593859  max resid 0.0003895394
## ... Similar to previous best
## Run 3 stress 0.0001807176
## ... Procrustes: rmse 0.000450952  max resid 0.00112692
## ... Similar to previous best
## Run 4 stress 8.208215e-05
## ... New best solution
## ... Procrustes: rmse 0.0001064449  max resid 0.0003268099
## ... Similar to previous best
## Run 5 stress 9.186741e-05
```

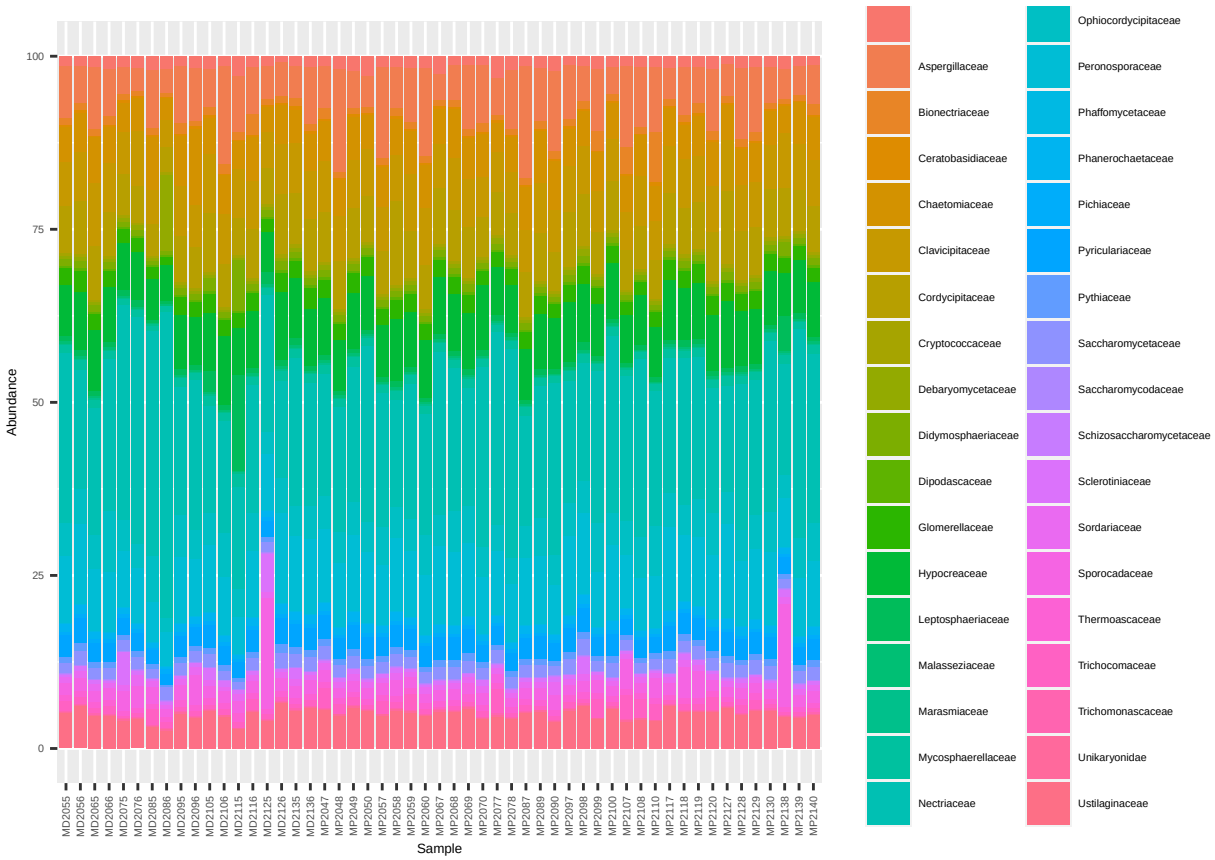
```
## ... Procrustes: rmse 0.0001412208  max resid 0.0004309775
## ... Similar to previous best
## Run 6 stress 7.462886e-05
## ... New best solution
## ... Procrustes: rmse 3.211791e-05  max resid 8.559031e-05
## ... Similar to previous best
## Run 7 stress 0.0001048768
## ... Procrustes: rmse 0.0002993268  max resid 0.0007891489
## ... Similar to previous best
## Run 8 stress 7.996713e-05
## ... Procrustes: rmse 5.233094e-05  max resid 0.0001331529
## ... Similar to previous best
## Run 9 stress 7.32546e-05
## ... New best solution
## ... Procrustes: rmse 4.60509e-05  max resid 0.0001679312
## ... Similar to previous best
## Run 10 stress 6.222568e-05
## ... New best solution
## ... Procrustes: rmse 4.352739e-05  max resid 0.0001077313
## ... Similar to previous best
## Run 11 stress 0.001023353
## Run 12 stress 9.984771e-05
## ... Procrustes: rmse 0.0001075881  max resid 0.0003100759
## ... Similar to previous best
## Run 13 stress 9.721e-05
## ... Procrustes: rmse 0.0001820262  max resid 0.0004477894
## ... Similar to previous best
## Run 14 stress 8.409742e-05
## ... Procrustes: rmse 3.708914e-05  max resid 0.0001118275
## ... Similar to previous best
## Run 15 stress 9.869134e-05
## ... Procrustes: rmse 0.0001608863  max resid 0.0004522027
## ... Similar to previous best
## Run 16 stress 9.797609e-05
## ... Procrustes: rmse 0.0001113431  max resid 0.0003237375
## ... Similar to previous best
## Run 17 stress 9.955898e-05
## ... Procrustes: rmse 0.0001629702  max resid 0.0004561932
## ... Similar to previous best
## Run 18 stress 9.530357e-05
## ... Procrustes: rmse 9.445489e-05  max resid 0.0002854999
## ... Similar to previous best
## Run 19 stress 4.705652e-05
## ... New best solution
## ... Procrustes: rmse 3.033326e-05  max resid 7.031517e-05
## ... Similar to previous best
## Run 20 stress 0.001748362
## *** Best solution repeated 1 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))+
  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_Family<-tax_glom(merge_Eukaryota,taxrank="Family")
```

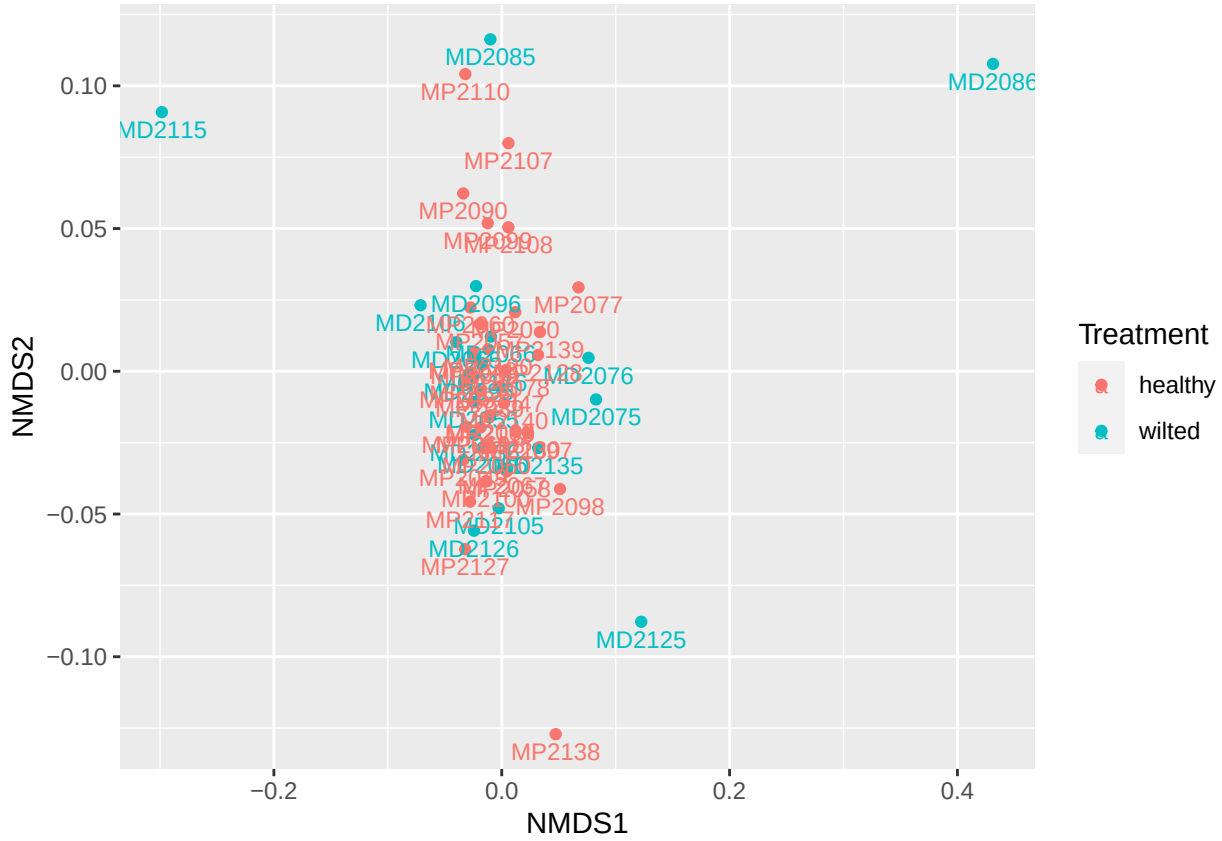
sacamos las abundancias relativas

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

```
## Wisconsin double standardization
## Run 0 stress 0.1089081
## Run 1 stress 0.108888
## ... New best solution
## ... Procrustes: rmse 0.001692112  max resid 0.009361591
## ... Similar to previous best
## Run 2 stress 0.1089296
## ... Procrustes: rmse 0.008412076  max resid 0.05476102
## Run 3 stress 0.1089294
## ... Procrustes: rmse 0.008515315  max resid 0.05452947
## Run 4 stress 0.10893
## ... Procrustes: rmse 0.008409098  max resid 0.05478302
## Run 5 stress 0.1089298
## ... Procrustes: rmse 0.008410544  max resid 0.05477326
## Run 6 stress 0.1352079
## Run 7 stress 0.1352087
## Run 8 stress 0.1089299
```

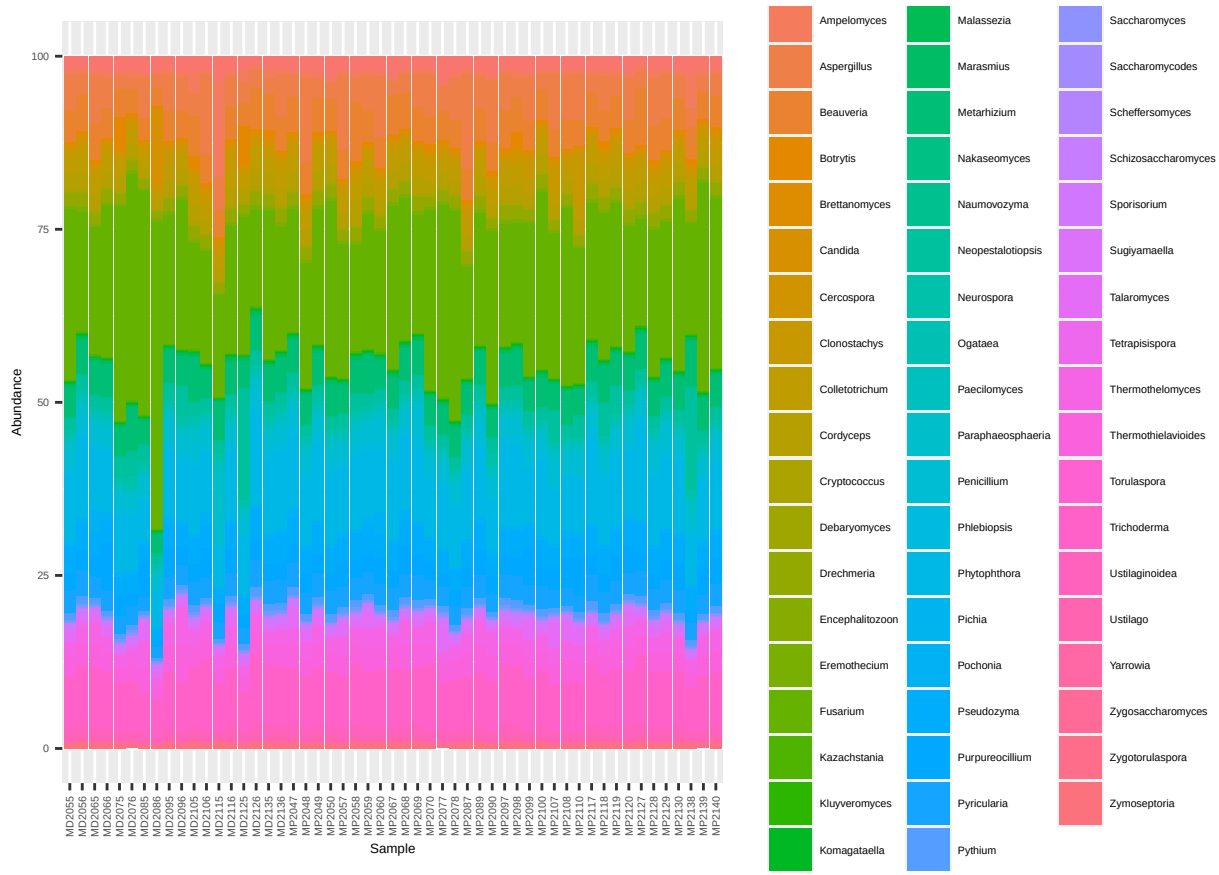
```
## ... Procrustes: rmse 0.008409391  max resid 0.05478159
## Run 9 stress 0.108949
## ... Procrustes: rmse 0.008246563  max resid 0.05400406
## Run 10 stress 0.1089484
## ... Procrustes: rmse 0.008204904  max resid 0.05405395
## Run 11 stress 0.1092107
## ... Procrustes: rmse 0.01213051  max resid 0.05289255
## Run 12 stress 0.1090933
## ... Procrustes: rmse 0.005793809  max resid 0.0287062
## Run 13 stress 0.1089087
## ... Procrustes: rmse 0.001706961  max resid 0.009327453
## ... Similar to previous best
## Run 14 stress 0.1089293
## ... Procrustes: rmse 0.008417598  max resid 0.05473245
## Run 15 stress 0.109113
## ... Procrustes: rmse 0.00989047  max resid 0.0541286
## Run 16 stress 0.1352077
## Run 17 stress 0.1089296
## ... Procrustes: rmse 0.008412506  max resid 0.05475463
## Run 18 stress 0.1089489
## ... Procrustes: rmse 0.008233157  max resid 0.05401687
## Run 19 stress 0.1088897
## ... Procrustes: rmse 0.00156262  max resid 0.008901338
## ... Similar to previous best
## Run 20 stress 0.1352081
## *** Best solution repeated 3 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")
```

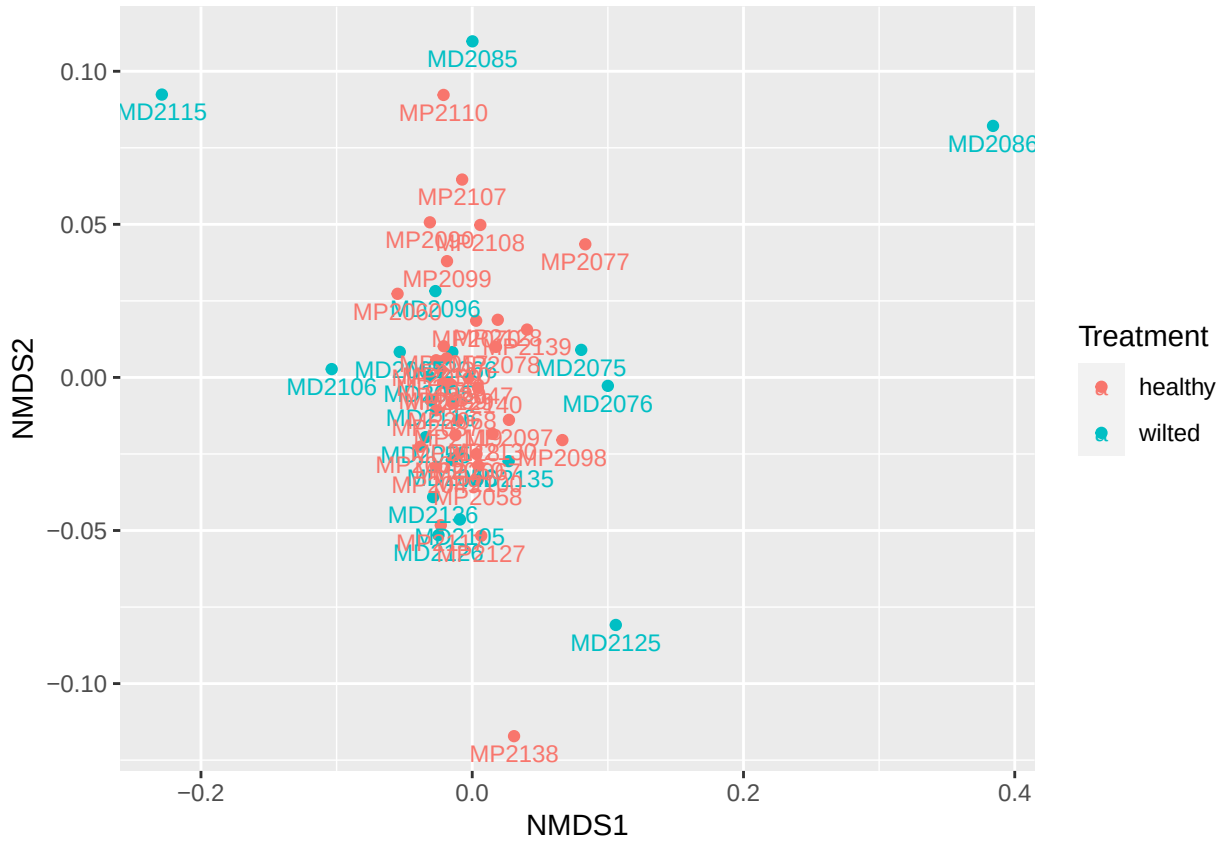
sacamos las abundancias relativas

```
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 0.1119817
## Run 2 stress 0.1176789
## Run 3 stress 0.1107094
## ... Procrustes: rmse 0.001110875  max resid 0.006357406
## ... Similar to previous best
## Run 4 stress 0.1107123
## ... Procrustes: rmse 0.0006081088  max resid 0.003382023
## ... Similar to previous best
## Run 5 stress 0.1119822
## Run 6 stress 0.111982
## Run 7 stress 0.1107085
## ... New best solution
## ... Procrustes: rmse 0.0006905962  max resid 0.003953207
## ... Similar to previous best
## Run 8 stress 0.1107097
```

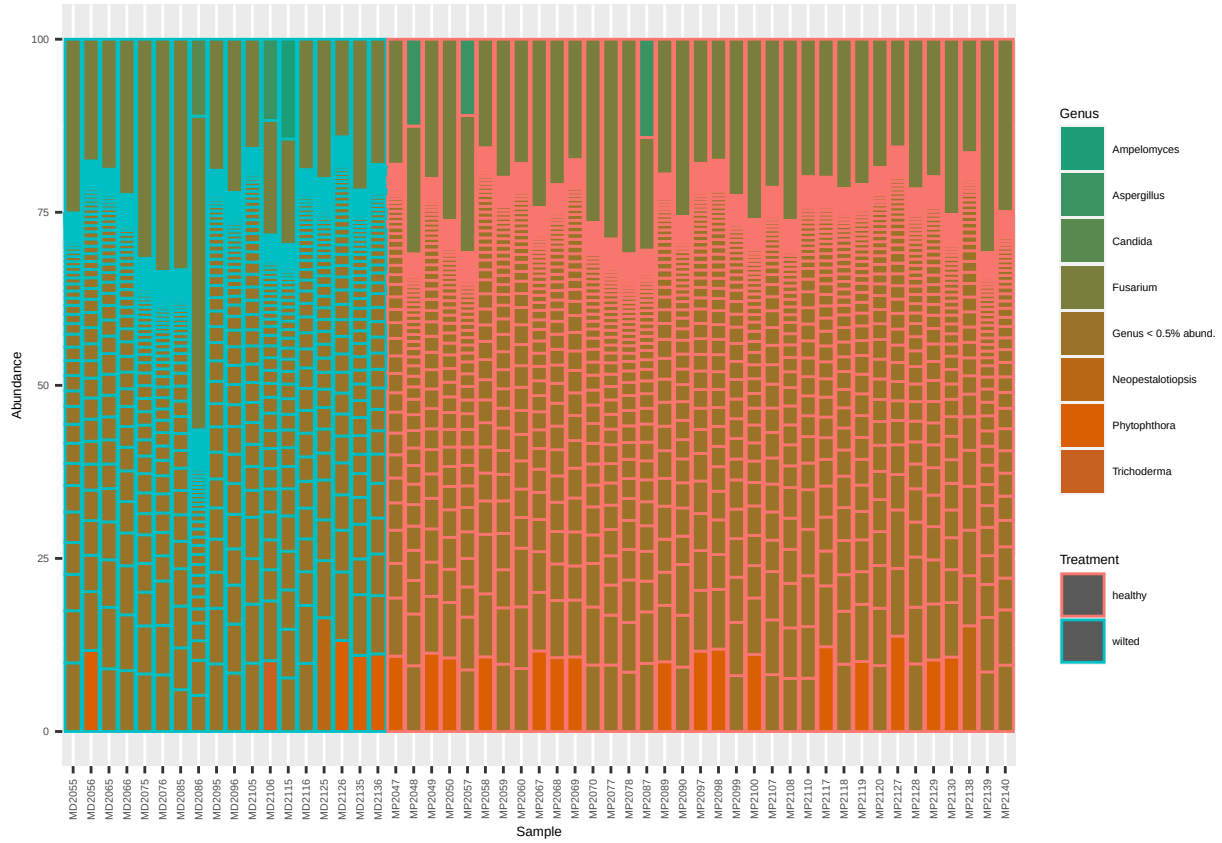
```
## ... Procrustes: rmse 0.000498978  max resid 0.00285821
## ... Similar to previous best
## Run 9 stress 0.1107094
## ... Procrustes: rmse 0.0004042803  max resid 0.002318586
## ... Similar to previous best
## Run 10 stress 0.1145078
## Run 11 stress 0.1236307
## Run 12 stress 0.1119823
## Run 13 stress 0.1107093
## ... Procrustes: rmse 0.0003766442  max resid 0.002156754
## ... Similar to previous best
## Run 14 stress 0.11071
## ... Procrustes: rmse 0.0009845455  max resid 0.005603379
## ... Similar to previous best
## Run 15 stress 0.1119828
## Run 16 stress 0.1119826
## Run 17 stress 0.1107099
## ... Procrustes: rmse 0.000956729  max resid 0.005442851
## ... Similar to previous best
## Run 18 stress 0.1107096
## ... Procrustes: rmse 0.0008634493  max resid 0.004912192
## ... Similar to previous best
## Run 19 stress 0.1119812
## Run 20 stress 0.1119789
## *** Best solution repeated 7 times
```

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

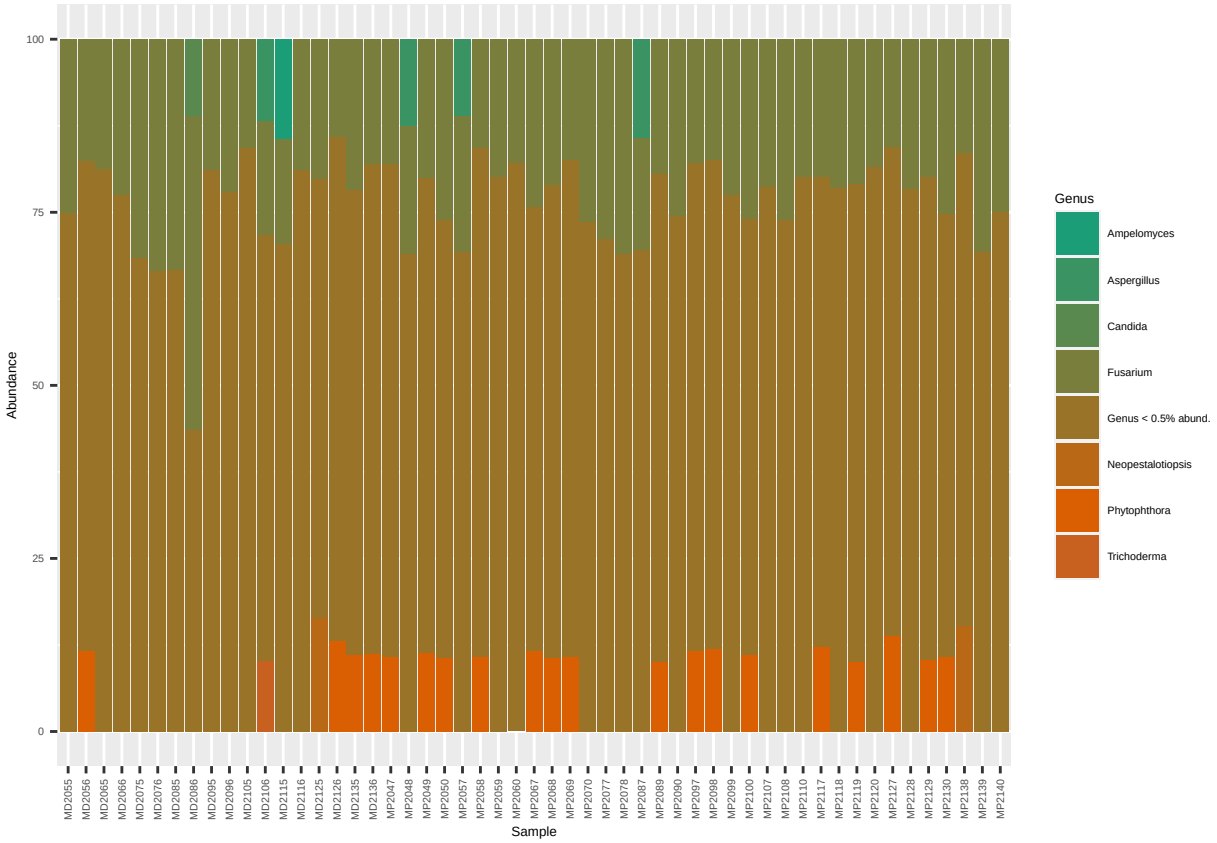


tomando diferentes porcentajes de abundancia

```
percentages_Eukaryota_Genus_df$Genus[percentages_Eukaryota_Genus_df$Abundance < 10.0] <- "Genus < 0.5% a
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_d
relative_plot <- ggplot(data=percentages_Eukaryota_Genus_df, aes(x=Sample, y=Abundance, fill=Genus ,colo
  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
```



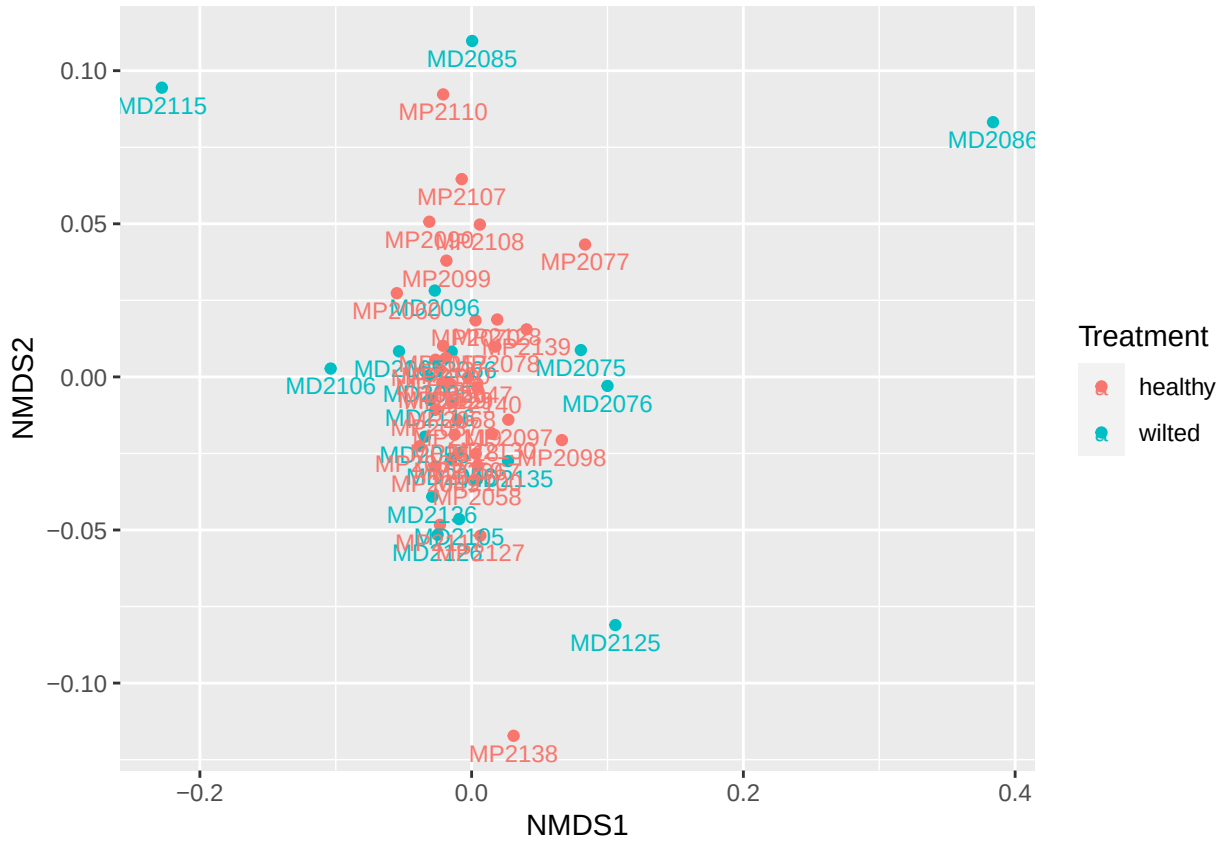
sacando la beta diversidad con aglomerado de 10%

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

```
## Wisconsin double standardization
## Run 0 stress 0.110709
## Run 1 stress 0.1223848
## Run 2 stress 0.1119784
## Run 3 stress 0.1107097
## ... Procrustes: rmse 0.0002124151  max resid 0.001183418
## ... Similar to previous best
## Run 4 stress 0.1315555
## Run 5 stress 0.1107095
## ... Procrustes: rmse 0.000161435  max resid 0.0008925069
## ... Similar to previous best
## Run 6 stress 0.1223841
## Run 7 stress 0.1222462
## Run 8 stress 0.1119813
## Run 9 stress 0.1119815
## Run 10 stress 0.1235025
## Run 11 stress 0.1119824
## Run 12 stress 0.1236308
## Run 13 stress 0.122247
## Run 14 stress 0.1341447
## Run 15 stress 0.1235034
## Run 16 stress 0.1119786
```

```
## Run 17 stress 0.1235018
## Run 18 stress 0.1116557
## Run 19 stress 0.1107088
## ... New best solution
## ... Procrustes: rmse 0.0001141873  max resid 0.0006636301
## ... Similar to previous best
## Run 20 stress 0.1119817
## *** Best solution repeated 1 times
```

```
plot_ordination(physeq = percentages_Eukaryota_Genus, ordination = meta_ord_Eukaryota_Genus, color = "Tr
  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

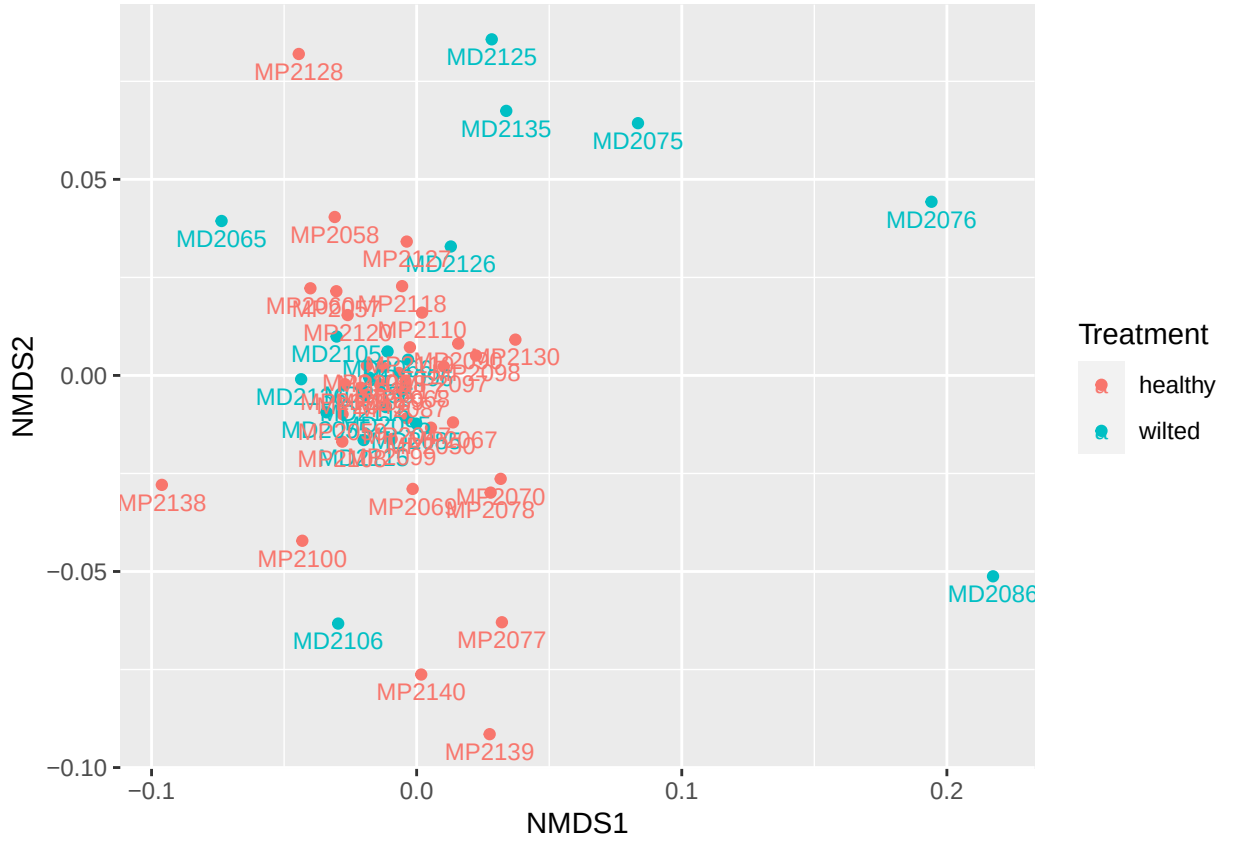
```
percentages_Bacteria <- transform_sample_counts(merge_Bacteria, function(x) x*100 / sum(x) )
percentages_Bacteria_df <- psmelt(percentages_Bacteria)
```

beta diversidad de Bacteria

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

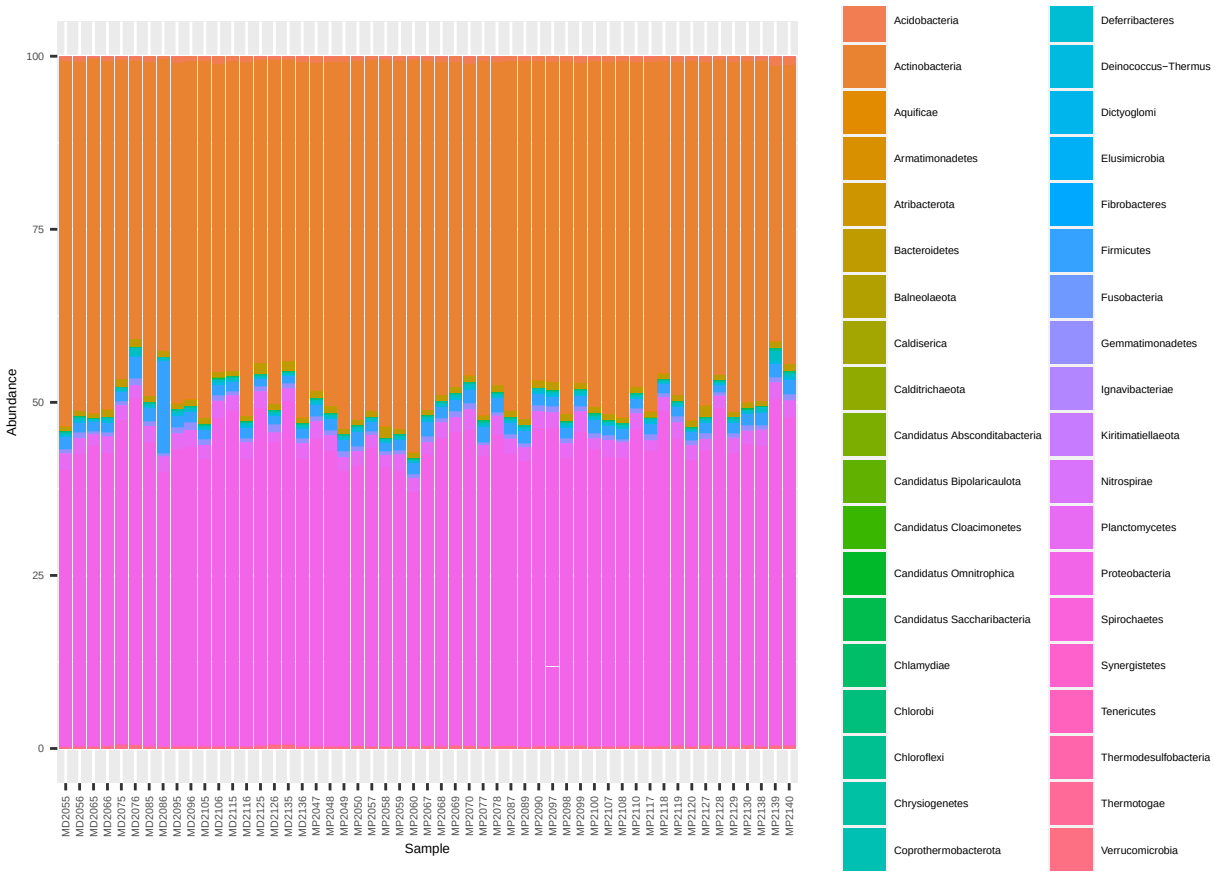
```
## Wisconsin double standardization
## Run 0 stress 0.1680232
## Run 1 stress 0.1658268
## ... New best solution
## ... Procrustes: rmse 0.06625423  max resid 0.3497755
## Run 2 stress 0.1687377
## Run 3 stress 0.1691068
## Run 4 stress 0.1686396
## Run 5 stress 0.1665102
## Run 6 stress 0.1797448
## Run 7 stress 0.1742406
## Run 8 stress 0.1692049
## Run 9 stress 0.1992948
## Run 10 stress 0.1661233
## ... Procrustes: rmse 0.07083586  max resid 0.3407577
## Run 11 stress 0.1708893
## Run 12 stress 0.1658877
## ... Procrustes: rmse 0.06560222  max resid 0.3474303
## Run 13 stress 0.1689047
## Run 14 stress 0.1657577
## ... New best solution
## ... Procrustes: rmse 0.01952603  max resid 0.1283628
## Run 15 stress 0.1717097
## Run 16 stress 0.1710615
## Run 17 stress 0.1800279
## Run 18 stress 0.1680232
## Run 19 stress 0.1640079
## ... New best solution
## ... Procrustes: rmse 0.05965582  max resid 0.3561727
## Run 20 stress 0.1799721
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      2: no. of iterations >= maxit
##     18: stress ratio > sratmax
```

```r
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_glom(merge_Bacteria,taxrank="Phylum")
```

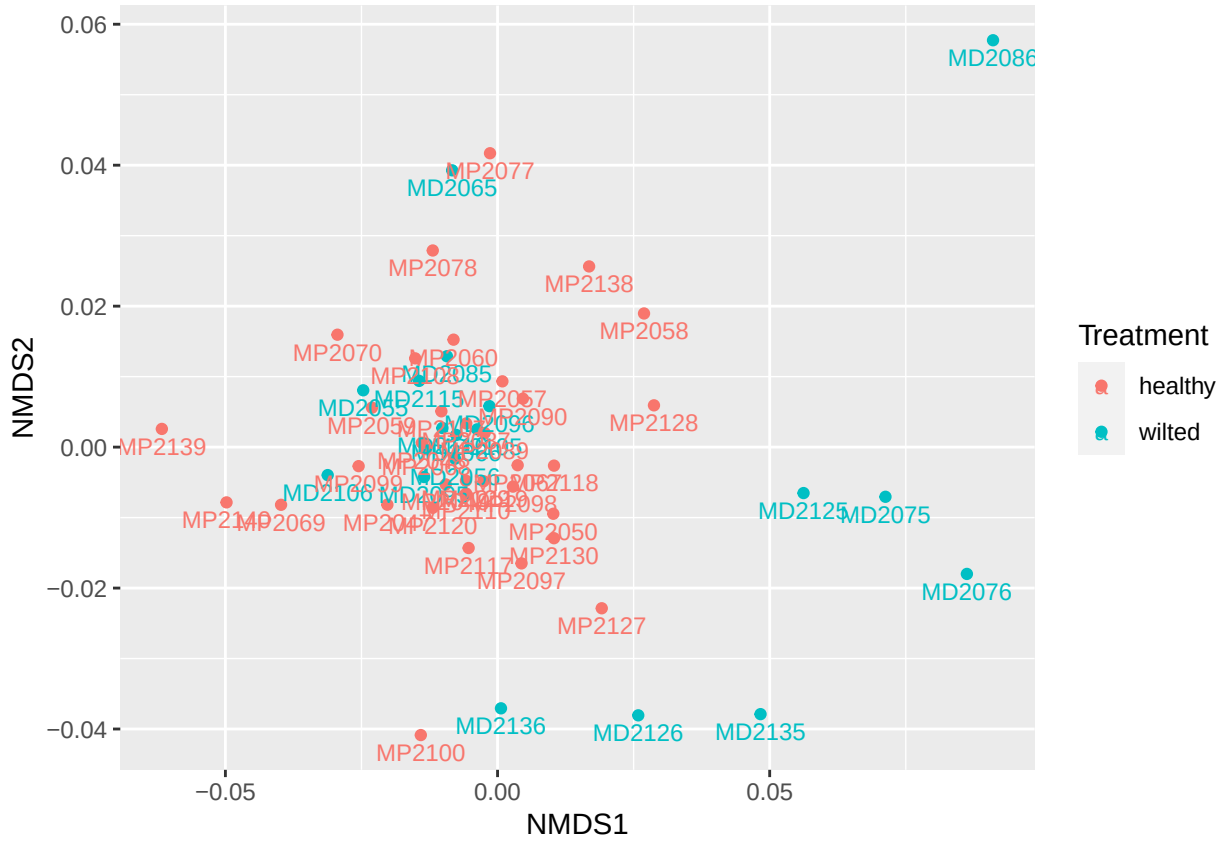
sacamos las abundancias relativas

```
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 0.1524009
## ... Procrustes: rmse 0.04800039  max resid 0.315377
## Run 2 stress 0.1646847
## Run 3 stress 0.1519998
## ... New best solution
## ... Procrustes: rmse 0.0002224863  max resid 0.001257028
## ... Similar to previous best
## Run 4 stress 0.1550553
## Run 5 stress 0.1597146
## Run 6 stress 0.1550552
## Run 7 stress 0.1619444
## Run 8 stress 0.155422
## Run 9 stress 0.174961
## Run 10 stress 0.1548791
```

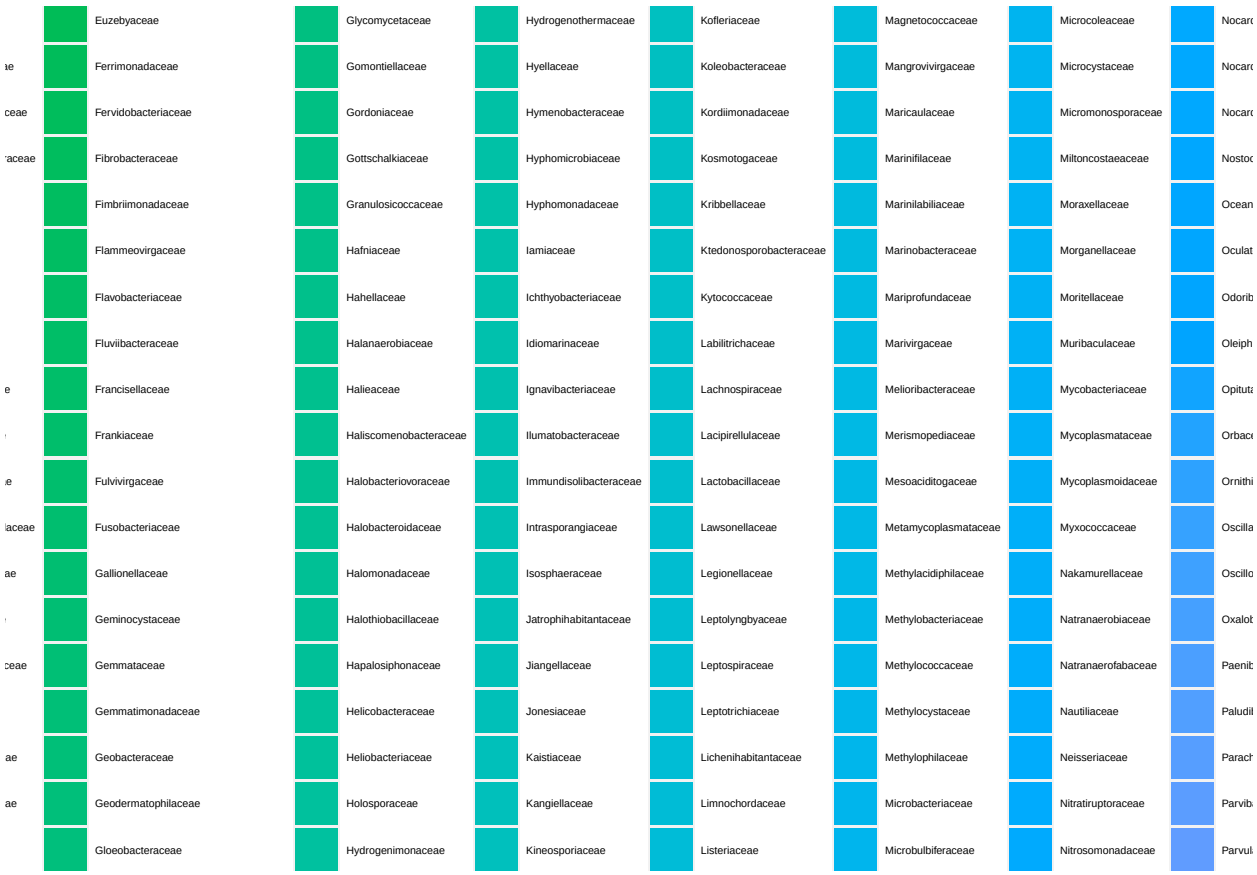
```
## Run 11 stress 0.1749744
## Run 12 stress 0.1652839
## Run 13 stress 0.1587859
## Run 14 stress 0.1578492
## Run 15 stress 0.1710106
## Run 16 stress 0.1597147
## Run 17 stress 0.1572805
## Run 18 stress 0.1693713
## Run 19 stress 0.164222
## Run 20 stress 0.1630501
## *** Best solution repeated 1 times
```

```
plot_ordination(physeq = percentages_Bacteria_Phylum, ordination = meta_ord_Bacteria_Phylum, color = "Ti
  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")
```

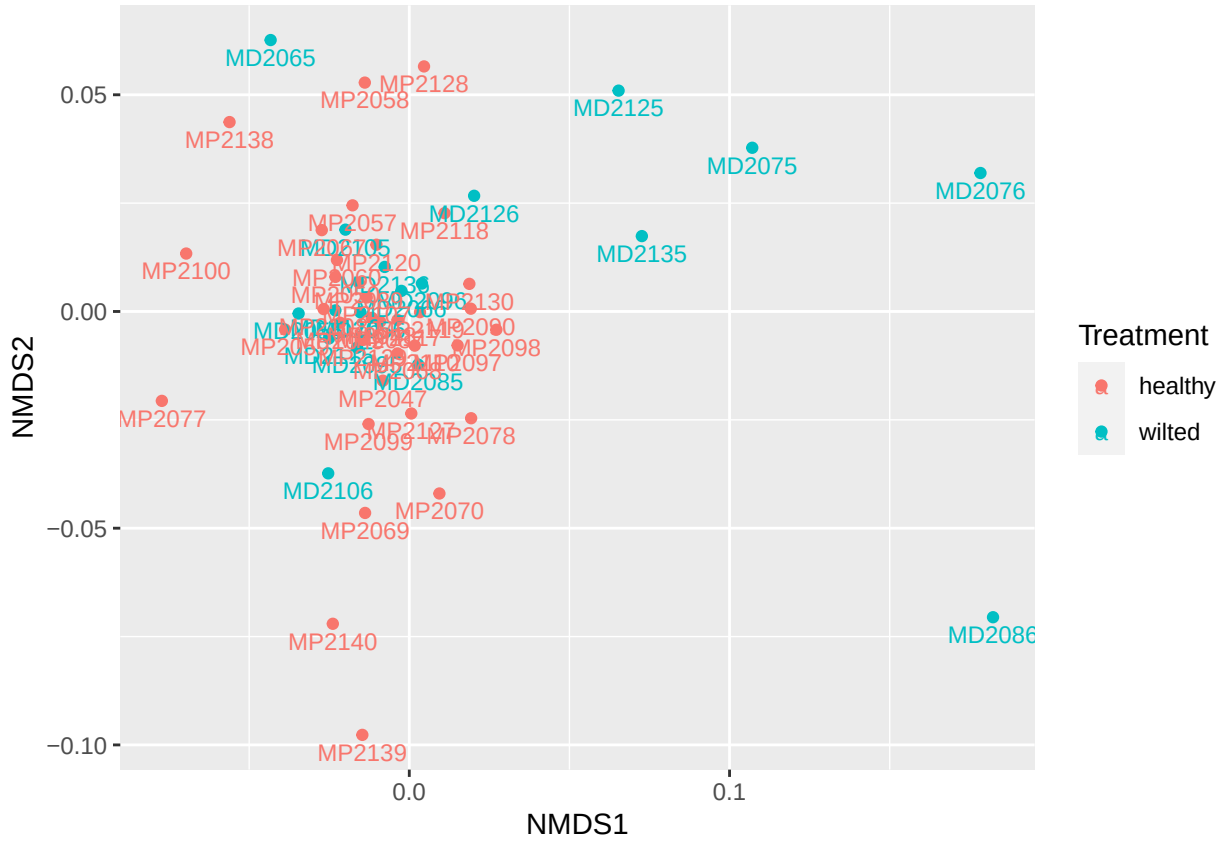
sacamos las abundancias relativas

```
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 0.1841839
## Run 2 stress 0.171039
## Run 3 stress 0.1373771
## ... New best solution
## ... Procrustes: rmse 0.06365222  max resid 0.3610578
## Run 4 stress 0.1396571
## Run 5 stress 0.1569678
## Run 6 stress 0.1423481
## Run 7 stress 0.137038
## ... New best solution
## ... Procrustes: rmse 0.01628313  max resid 0.05956157
## Run 8 stress 0.1369861
## ... New best solution
```

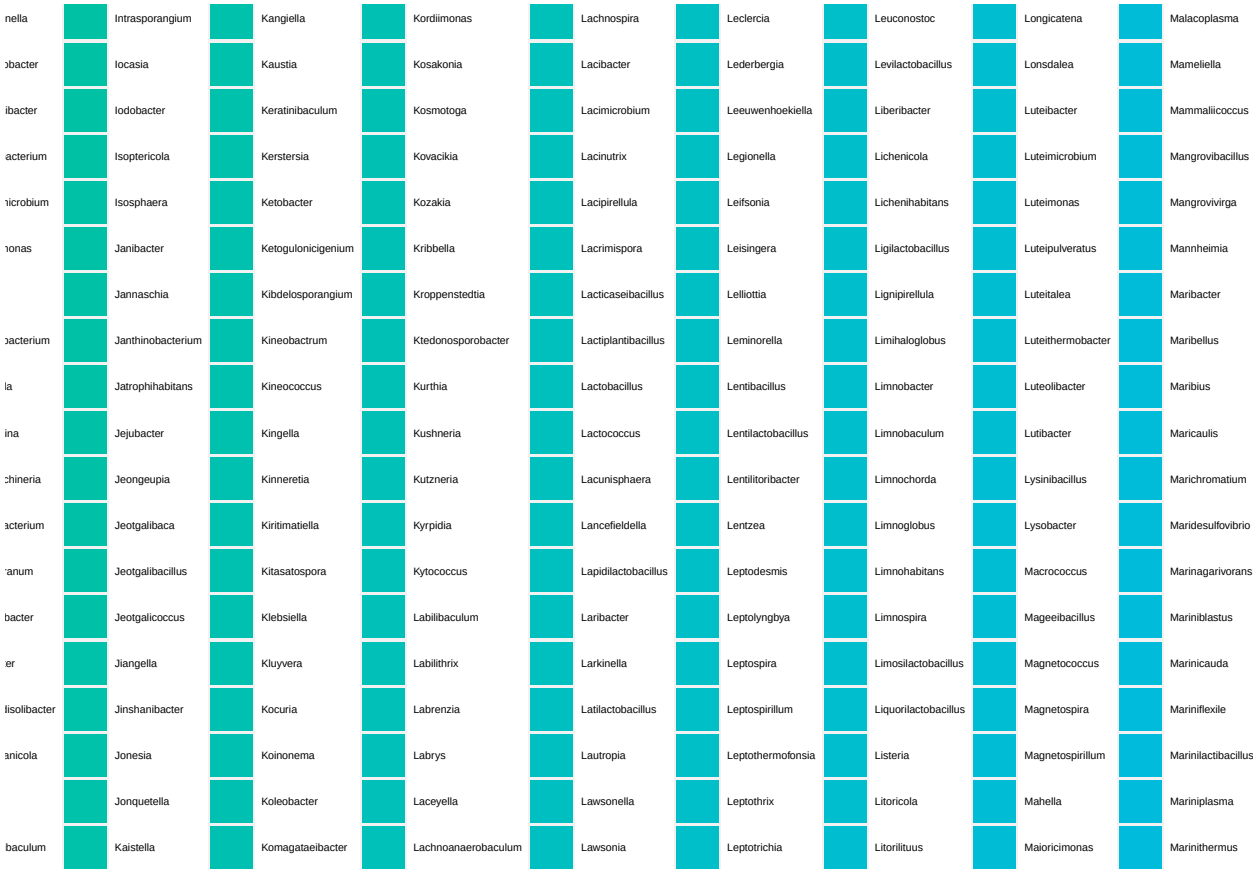
```
## ... Procrustes: rmse 0.00239762  max resid 0.01201976
## Run 9 stress 0.137038
## ... Procrustes: rmse 0.002397831  max resid 0.0120207
## Run 10 stress 0.1731639
## Run 11 stress 0.1412064
## Run 12 stress 0.1374744
## ... Procrustes: rmse 0.0155282  max resid 0.05889792
## Run 13 stress 0.1799551
## Run 14 stress 0.1370382
## ... Procrustes: rmse 0.002410986  max resid 0.01199184
## Run 15 stress 0.1522704
## Run 16 stress 0.1373768
## ... Procrustes: rmse 0.01527212  max resid 0.05900154
## Run 17 stress 0.1401715
## Run 18 stress 0.139046
## Run 19 stress 0.139501
## Run 20 stress 0.1373772
## ... Procrustes: rmse 0.01518985  max resid 0.05894204
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      3: no. of iterations >= maxit
##     17: stress ratio > sratmax
```

```
plot_ordination(physeq = percentages_Bacteria_Family, ordination = meta_ord_Bacteria_Family, color = "T
  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")
```

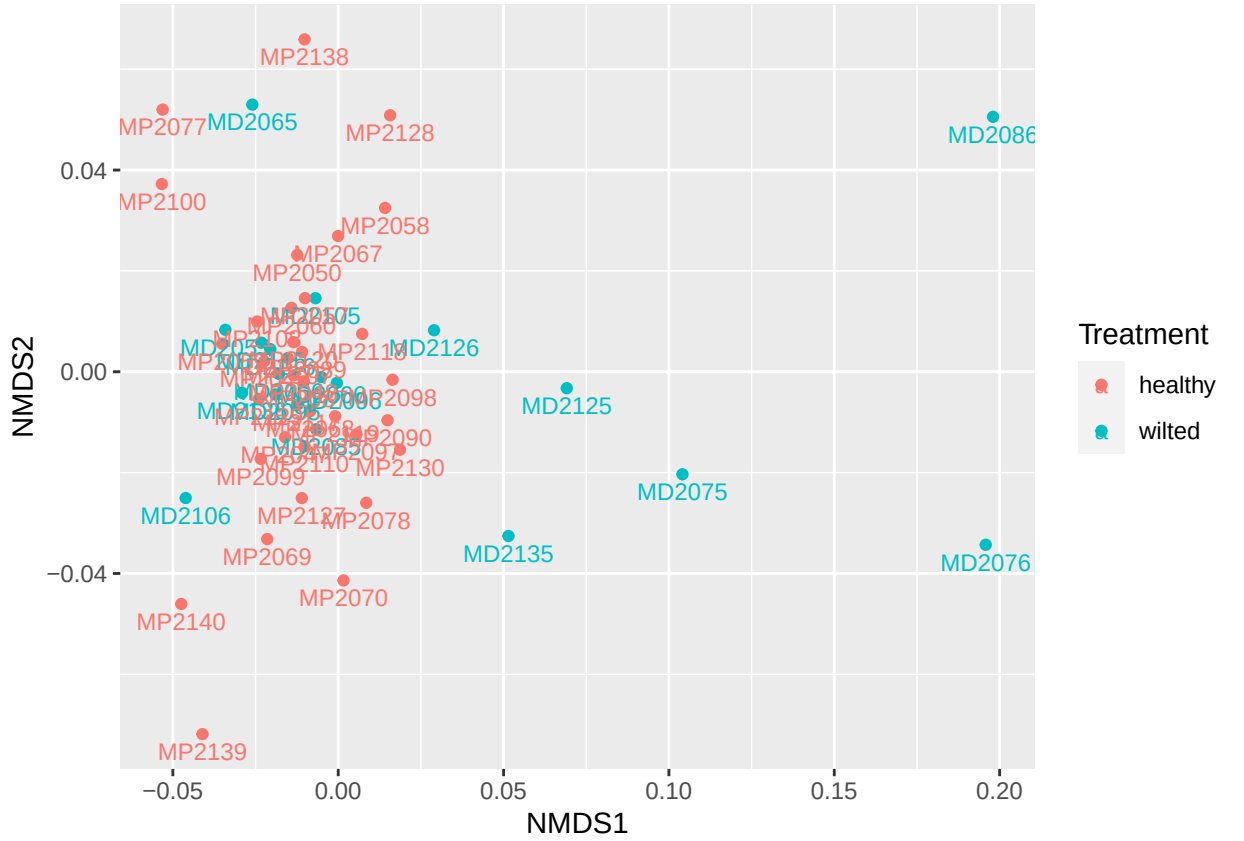
sacamos las abundancias relativas

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

```
## Wisconsin double standardization
## Run 0 stress 0.1467945
## Run 1 stress 0.1467691
## ... New best solution
## ... Procrustes: rmse 0.002700535  max resid 0.01690071
## Run 2 stress 0.1660091
## Run 3 stress 0.1514
## Run 4 stress 0.1527708
## Run 5 stress 0.1453935
## ... New best solution
## ... Procrustes: rmse 0.08909263  max resid 0.2984409
```

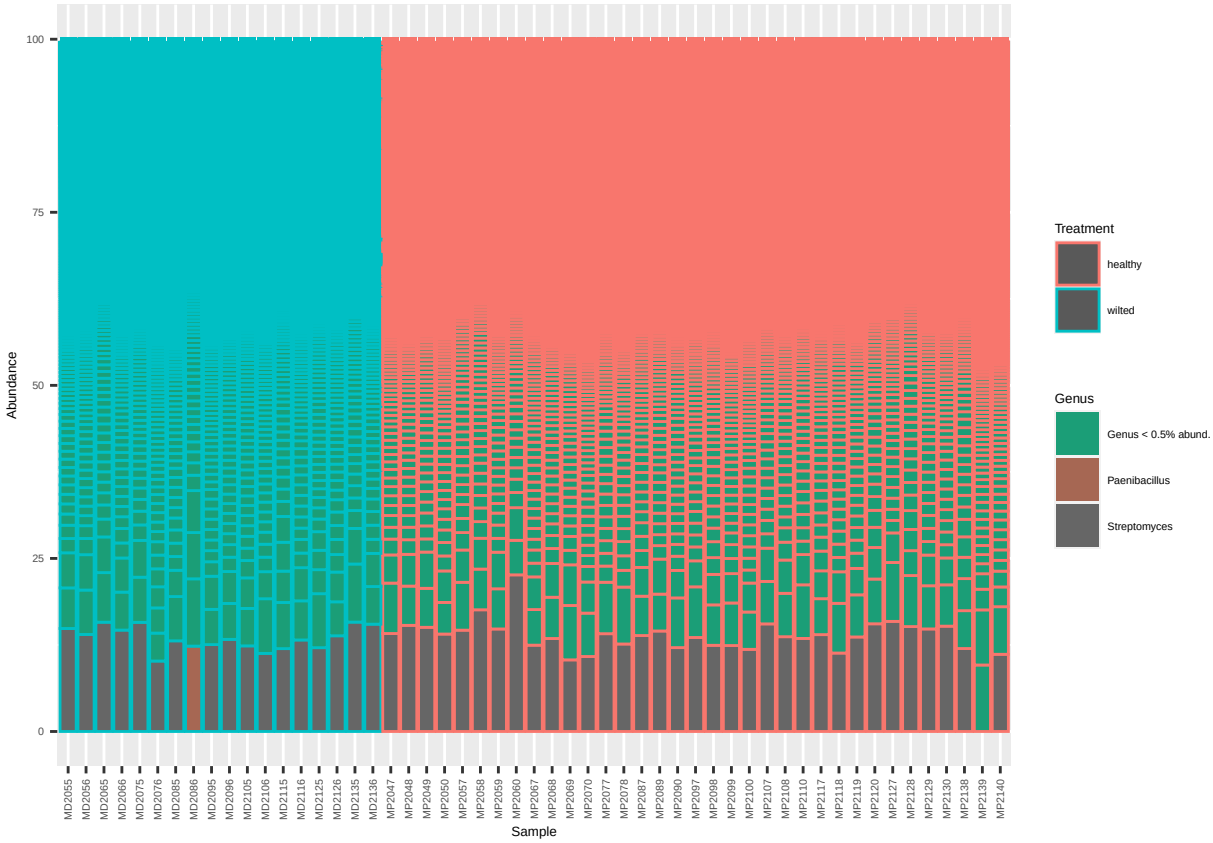
```
## Run 6 stress 0.1439574
## ... New best solution
## ... Procrustes: rmse 0.02430511  max resid 0.1319898
## Run 7 stress 0.1504571
## Run 8 stress 0.1456405
## Run 9 stress 0.1502732
## Run 10 stress 0.1538257
## Run 11 stress 0.143814
## ... New best solution
## ... Procrustes: rmse 0.05706939  max resid 0.3103235
## Run 12 stress 0.143284
## ... New best solution
## ... Procrustes: rmse 0.05080514  max resid 0.3015335
## Run 13 stress 0.1432845
## ... Procrustes: rmse 0.001290418  max resid 0.005619273
## ... Similar to previous best
## Run 14 stress 0.154184
## Run 15 stress 0.1490354
## Run 16 stress 0.1482214
## Run 17 stress 0.1484507
## Run 18 stress 0.1432838
## ... New best solution
## ... Procrustes: rmse 6.320535e-05  max resid 0.0003107312
## ... Similar to previous best
## Run 19 stress 0.1433099
## ... Procrustes: rmse 0.003956295  max resid 0.02546086
## Run 20 stress 0.1438152
## *** Best solution repeated 1 times
```

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

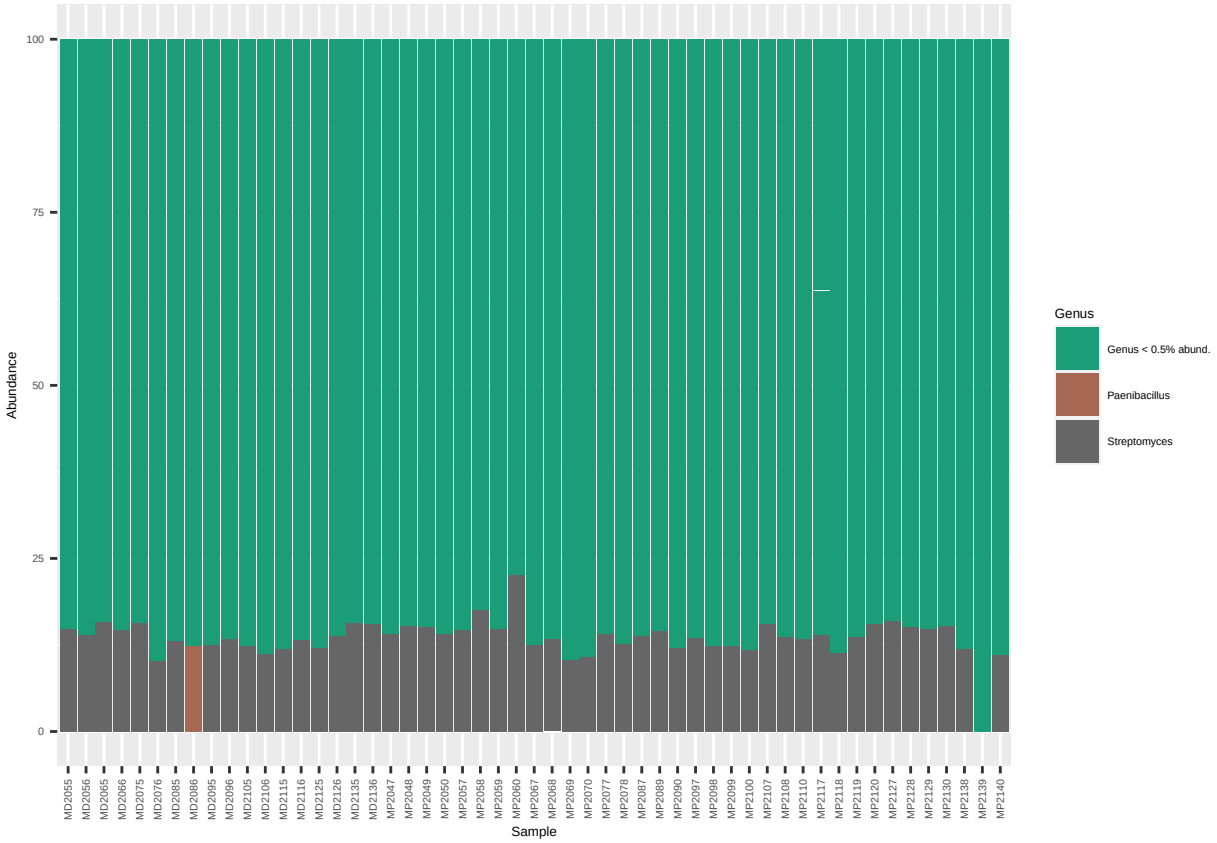


tomando diferentes porcentajes de abundancia

```
percentages_Bacteria_Genus_df$Genus[percentages_Bacteria_Genus_df$Abundance < 10.0] <- "Genus < 0.5% abu
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
relative_plot <- ggplot(data=percentages_Bacteria_Genus_df, aes(x=Sample, y=Abundance, fill=Genus ,colo
  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
```



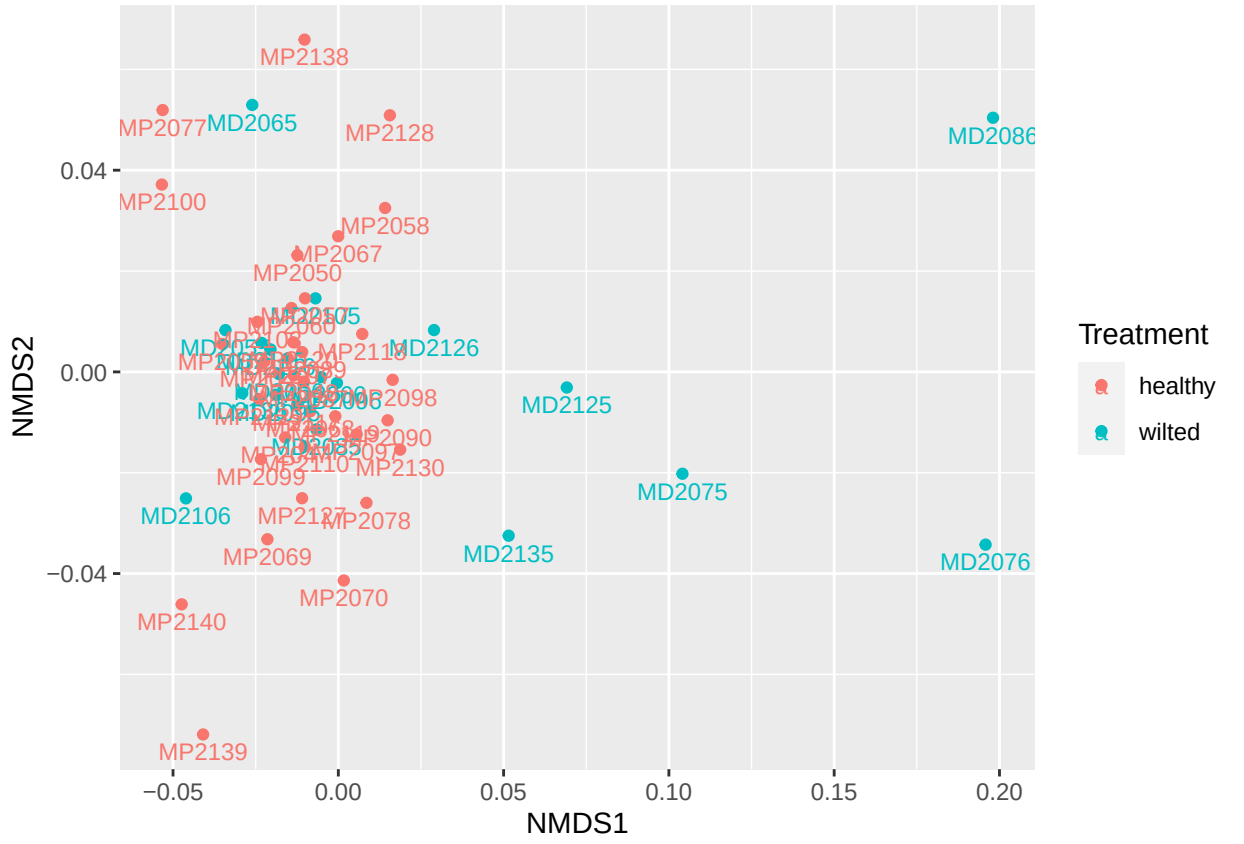
sacando la beta diversidad con aglomerado de 10%

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

```
## Wisconsin double standardization
## Run 0 stress 0.1467945
## Run 1 stress 0.1561836
## Run 2 stress 0.1497397
## Run 3 stress 0.1482215
## Run 4 stress 0.1458785
## ... New best solution
## ... Procrustes: rmse 0.05080048  max resid 0.2810505
## Run 5 stress 0.1433354
## ... New best solution
## ... Procrustes: rmse 0.07338854  max resid 0.3700134
## Run 6 stress 0.1541723
## Run 7 stress 0.1502742
## Run 8 stress 0.1476415
## Run 9 stress 0.1544768
## Run 10 stress 0.1432845
## ... New best solution
## ... Procrustes: rmse 0.005055307  max resid 0.03013918
## Run 11 stress 0.1513999
## Run 12 stress 0.1438163
## Run 13 stress 0.1437857
## Run 14 stress 0.1437856
```

```
## Run 15 stress 0.1492682
## Run 16 stress 0.1432835
## ... New best solution
## ... Procrustes: rmse 0.0003142473  max resid 0.001426629
## ... Similar to previous best
## Run 17 stress 0.1505676
## Run 18 stress 0.4024306
## Run 19 stress 0.1454658
## Run 20 stress 0.1441635
## *** Best solution repeated 1 times
```

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



REDES podemos hacer un data.frame uniendo toda la informacion del objeto phyloseq

```
df <- psmelt(fresa_kraken_fil)
```

Hay dos funciones en el paquete phyloseq para trazar la red del microbioma usando "ggplot2": plot_network() y plot_net()

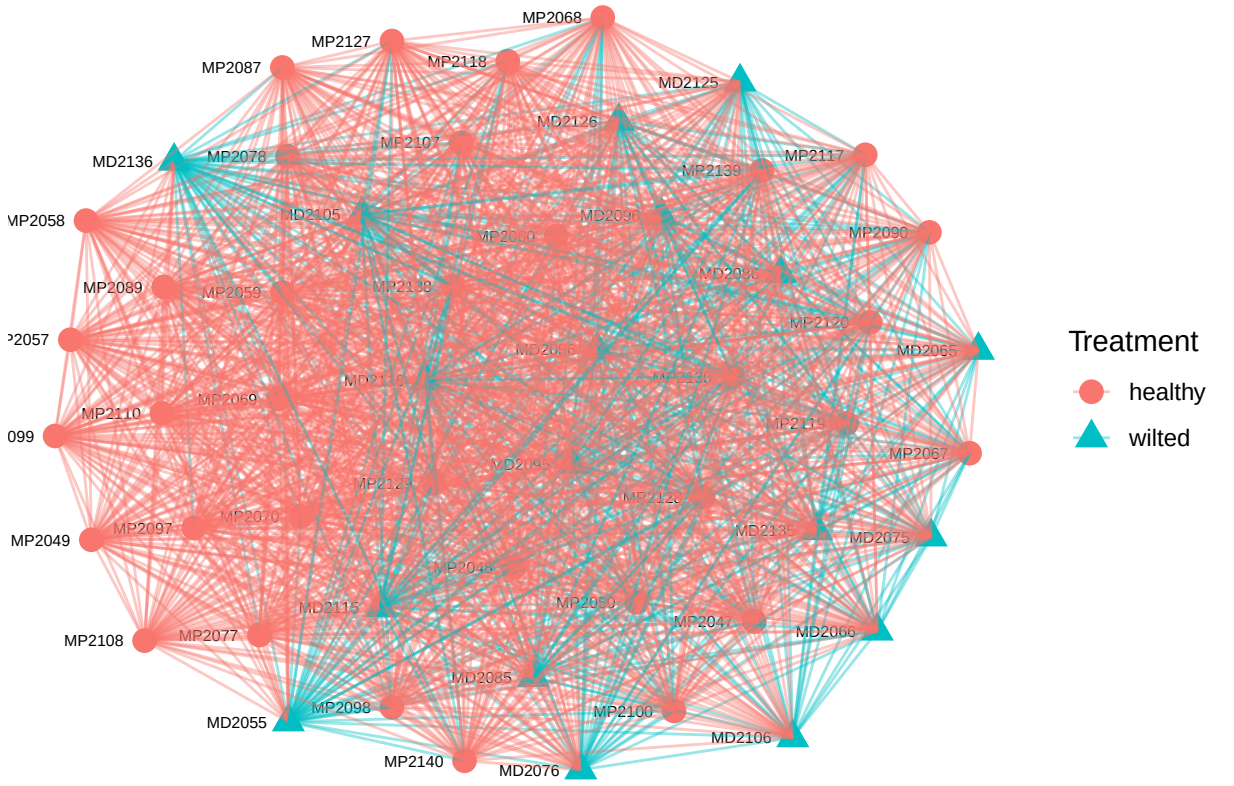
Se crean un grafo basado en igraph, basado en el método de distancia por defecto, Jaccard y una distancia máxima entre nodos conectados de 0,8. El "Treatment" se utiliza para los mapeos de color y forma para visualizar la estructura de las muestras

Hacemos una gráfica a partir de el objeto phyloseq

```
ig <- make_network(fresa_kraken_fil, max.dist=0.8)
```

Graficamos

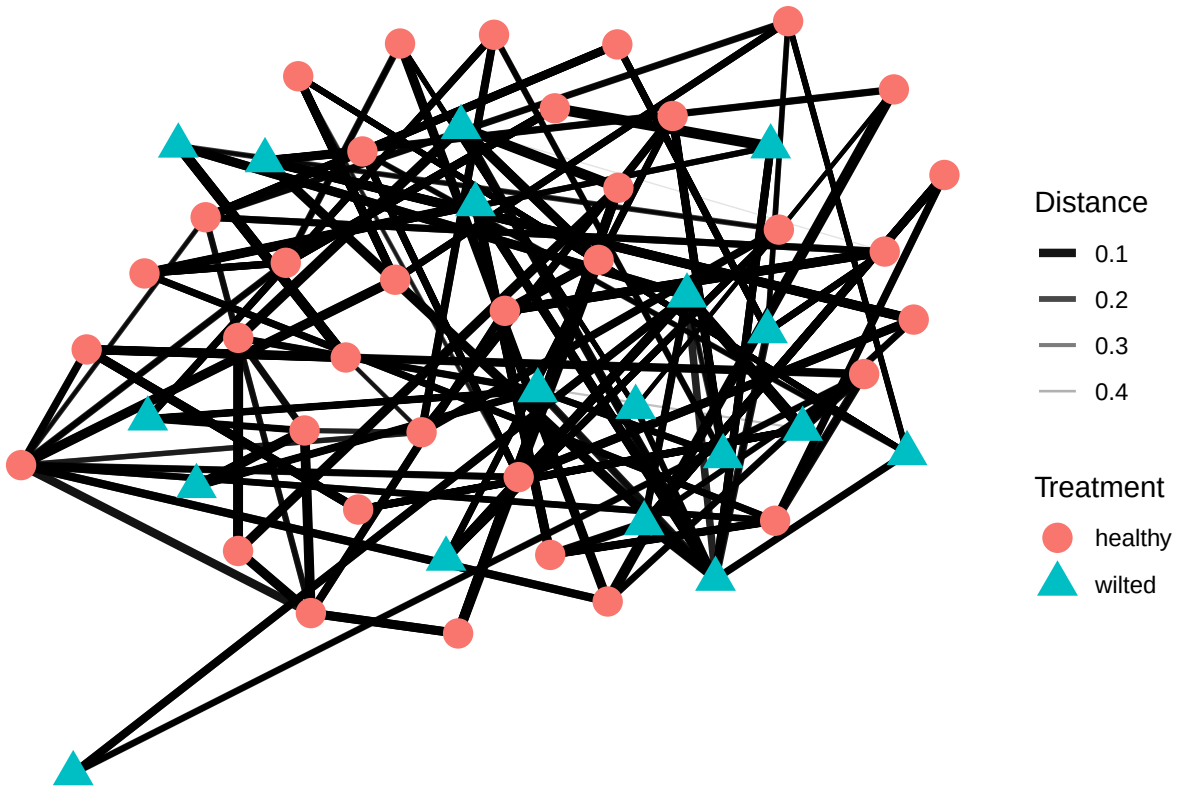
```
plot_network(ig, fresa_kraken_fil, color="Treatment", shape="Treatment")
```



Los siguientes códigos crean una red basada en una distancia máxima entre los nodos conectados de 0,5

Graficamos sin necesidad de hacer el objeto anterior

```
plot_net(fresa_kraken_fil, maxdist = 0.5, color = "Treatment", shape="Treatment")
```


Distance
0.1
0.2
0.3
0.4
Treatment
healthy
wilted