

Diversidades alfa y beta en datos metagenomicos de shotgun de cultivo de fresa

Camila Silva

2023-04-14

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

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

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

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

```
## The following object is masked from 'package:kableExtra':
```

```
##
##   group_rows
```

```
## The following objects are masked from 'package:igraph':
```

```
##
##   as_data_frame, groups, union
```

```
## The following objects are masked from 'package:stats':
```

```
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
##   intersect, setdiff, setequal, union
```

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

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

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

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

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

```
## The following objects are masked from 'package:dplyr':
```

```
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
```

```
library("edgeR")
```

```
## Loading required package: limma
```

Datos kraken

Importamos los datos, salidas de kraken en un archivo BIOM. En R estos datos quedaran en un objeto Phyloseq, el cual puede contener: tabla de taxonomía, tabla de conteos, tabla de muestras o metadatos y el árbol filogenético.

```
setwd("/home/camila/GIT/Tesis_Maestria/Data/fresa_solena/Data1")
fresa_kraken <- import_biom("fresa_kraken.biom")
class(fresa_kraken) # objeto phyloseq
```

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

En este caso obtenemos solamente la tabla de abundancias o **otu_table** y la tabla de taxonomía, o **tax_table**.

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

Primero la tabla de taxonomía: en donde podemos ver los niveles taxonomicos existentes para cada OTU,

```
head(fresa_kraken@tax_table@.Data)
```

```
##      Rank1      Rank2      Rank3
## 2062 "k__Bacteria" "p__Actinobacteria" "c__Actinomycetia"
## 1883 "k__Bacteria" "p__Actinobacteria" "c__Actinomycetia"
## 2880933 "k__Bacteria" "p__Actinobacteria" "c__Actinomycetia"
## 1725411 "k__Bacteria" "p__Actinobacteria" "c__Actinomycetia"
## 2781734 "k__Bacteria" "p__Actinobacteria" "c__Actinomycetia"
## 659352 "k__Bacteria" "p__Actinobacteria" "c__Actinomycetia"
##      Rank4      Rank5      Rank6
## 2062 "o__Streptomycetales" "f__Streptomycetaceae" "g__"
## 1883 "o__Streptomycetales" "f__Streptomycetaceae" "g__Streptomyces"
## 2880933 "o__Streptomycetales" "f__Streptomycetaceae" "g__Streptomyces"
## 1725411 "o__Streptomycetales" "f__Streptomycetaceae" "g__Streptomyces"
## 2781734 "o__Streptomycetales" "f__Streptomycetaceae" "g__Streptomyces"
## 659352 "o__Streptomycetales" "f__Streptomycetaceae" "g__Streptomyces"
##      Rank7
## 2062 "s__"
## 1883 "s__"
## 2880933 "s__sp. BH-MK-02"
## 1725411 "s__sp. CdTB01"
## 2781734 "s__sp. A2-16"
## 659352 "s__sp. SN-593"
```

Cambiamos los nombres de las columnas según los niveles taxonomicos.

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

Tambien cortaremos la parte inicial nombres, ya que aparecen, por ejemplo: “B___Bacteria” y queremos que solo se vea “Bacteria”.

El comando **substring()** nos ayuda a extraer o reemplazar caracteres a conveniencia.

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

```
##      Kingdom      Phylum      Class      Order
## 2062  "Bacteria" "Actinobacteria" "Actinomycetia" "Streptomycetales"
## 1883  "Bacteria" "Actinobacteria" "Actinomycetia" "Streptomycetales"
## 2880933 "Bacteria" "Actinobacteria" "Actinomycetia" "Streptomycetales"
## 1725411 "Bacteria" "Actinobacteria" "Actinomycetia" "Streptomycetales"
## 2781734 "Bacteria" "Actinobacteria" "Actinomycetia" "Streptomycetales"
## 659352  "Bacteria" "Actinobacteria" "Actinomycetia" "Streptomycetales"
##      Family      Genus      Species
## 2062  "Streptomycetaceae" "" ""
## 1883  "Streptomycetaceae" "Streptomyces" ""
## 2880933 "Streptomycetaceae" "Streptomyces" "sp. BH-MK-02"
## 1725411 "Streptomycetaceae" "Streptomyces" "sp. CdTB01"
## 2781734 "Streptomycetaceae" "Streptomyces" "sp. A2-16"
## 659352  "Streptomycetaceae" "Streptomyces" "sp. SN-593"
```

Ahora queremos ver la tabla de OTUs, esta tabla contiene las abundancias de los OTUs para cada una de las muestras

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

```
##      MD2055.kraken2.report MD2056.kraken2.report MD2065.kraken2.report
## 2062      14978      16671      14434
## 1883     309650     369924     382080
## 2880933     24885     39404     19392
## 1725411     18762     18008     20447
## 2781734     15218     20492     16892
## 659352      12941     12642     12009
##      MD2066.kraken2.report MD2075.kraken2.report MD2076.kraken2.report
## 2062      19985      11740      14178
## 1883     470868     347565     391558
## 2880933     30749      8808     17675
## 1725411     35173     19992     10833
## 2781734     23422     25282      7588
## 659352      14695      9973     12879
##      MD2085.kraken2.report MD2086.kraken2.report MD2095.kraken2.report
## 2062      14014      9129     22767
## 1883     290224     202149     479760
## 2880933      9260      5945     30729
## 1725411     14935      9502     16057
## 2781734      7981      5138     14959
## 659352     13365     10479     15275
##      MD2096.kraken2.report MD2105.kraken2.report MD2106.kraken2.report
```

##	2062	19104	14868	4940
##	1883	408186	307824	100633
##	2880933	14509	15016	5732
##	1725411	18450	14369	4273
##	2781734	18562	11935	2538
##	659352	14166	9637	3142
##	MD2115.kraken2.report	MD2116.kraken2.report	MD2125.kraken2.report	
##	2062	15930	16647	8553
##	1883	327070	342009	202939
##	2880933	25734	18347	13173
##	1725411	19426	20289	13072
##	2781734	9389	12826	11619
##	659352	12613	12450	7116
##	MD2126.kraken2.report	MD2135.kraken2.report	MD2136.kraken2.report	
##	2062	17876	8239	8370
##	1883	409587	209898	188357
##	2880933	19148	8931	12668
##	1725411	35017	19699	12802
##	2781734	24599	17101	9577
##	659352	16401	7031	7821
##	MP2047.kraken2.report	MP2048.kraken2.report	MP2049.kraken2.report	
##	2062	11954	15902	17699
##	1883	266580	372843	405044
##	2880933	22207	30320	46649
##	1725411	18386	23309	23836
##	2781734	16844	19737	24247
##	659352	8262	12072	20157
##	MP2050.kraken2.report	MP2057.kraken2.report	MP2058.kraken2.report	
##	2062	26567	22846	27753
##	1883	475984	527148	750581
##	2880933	61499	33793	34519
##	1725411	18726	27569	60617
##	2781734	17282	33500	46613
##	659352	24266	17128	22909
##	MP2059.kraken2.report	MP2060.kraken2.report	MP2067.kraken2.report	
##	2062	22940	26981	21131
##	1883	438466	604749	481017
##	2880933	29838	32627	18138
##	1725411	25309	54020	24732
##	2781734	17270	31754	24172
##	659352	15093	15794	20061
##	MP2068.kraken2.report	MP2069.kraken2.report	MP2070.kraken2.report	
##	2062	29184	17415	15071
##	1883	583698	353252	298480
##	2880933	21247	15434	9944
##	1725411	28183	13726	11723
##	2781734	24470	11571	6597
##	659352	21102	10627	11300
##	MP2077.kraken2.report	MP2078.kraken2.report	MP2079.kraken2.report	
##	2062	26088	10789	1161
##	1883	293779	219233	24597
##	2880933	8648	6393	985
##	1725411	15507	9547	1202
##	2781734	6602	4916	713

## 659352	15175	10060	1110
##	MP2080.kraken2.report	MP2087.kraken2.report	MP2088.kraken2.report
## 2062	4512	27744	0
## 1883	96283	591983	0
## 2880933	3443	23886	0
## 1725411	4677	31963	0
## 2781734	2942	21732	0
## 659352	5409	27747	0
##	MP2089.kraken2.report	MP2090.kraken2.report	MP2097.kraken2.report
## 2062	22578	13129	15570
## 1883	500426	280636	323878
## 2880933	17543	9893	18427
## 1725411	18896	12400	16708
## 2781734	23800	12452	14494
## 659352	17421	10112	13223
##	MP2098.kraken2.report	MP2099.kraken2.report	MP2100.kraken2.report
## 2062	20459	17908	17823
## 1883	441849	381823	347101
## 2880933	26871	28428	16979
## 1725411	17172	17604	17832
## 2781734	14646	10650	9440
## 659352	12619	12267	15635
##	MP2107.kraken2.report	MP2108.kraken2.report	MP2109.kraken2.report
## 2062	14427	9408	1821
## 1883	340913	196757	43535
## 2880933	21746	13925	2387
## 1725411	23379	10078	2112
## 2781734	14349	5680	1384
## 659352	12030	6626	1425
##	MP2110.kraken2.report	MP2117.kraken2.report	MP2118.kraken2.report
## 2062	7995	17162	17011
## 1883	185697	373971	344457
## 2880933	14561	30177	24863
## 1725411	12573	19881	18899
## 2781734	9121	26676	16705
## 659352	6839	14136	12855
##	MP2119.kraken2.report	MP2120.kraken2.report	MP2127.kraken2.report
## 2062	12864	15126	10378
## 1883	278616	340307	223557
## 2880933	20282	23002	21811
## 1725411	13656	20846	13373
## 2781734	11258	20747	9515
## 659352	9862	13283	13243
##	MP2128.kraken2.report	MP2129.kraken2.report	MP2130.kraken2.report
## 2062	14087	11425	8383
## 1883	289910	249480	190633
## 2880933	16448	28085	19152
## 1725411	20144	13927	8668
## 2781734	15650	12163	6318
## 659352	12443	9377	12225
##	MP2137.kraken2.report	MP2138.kraken2.report	MP2139.kraken2.report
## 2062	4413	9744	7347
## 1883	81445	203351	130999
## 2880933	2477	5487	3194

```
## 1725411          4196          16516          5110
## 2781734          2218          10135          2391
## 659352          7081          11539          5731
##      MP2140.kraken2.report
## 2062          6630
## 1883          126168
## 2880933          4715
## 1725411          6029
## 2781734          2851
## 659352          5461
```

Como los nombres de las muestras aparecen con el formato completo de los archivos **fasta**, cortamos los nombres de las muestras para que coincidan con los metadatos

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

```
##      MD2055 MD2056 MD2065 MD2066 MD2075 MD2076 MD2085 MD2086 MD2095 MD2096
## 2062    14978  16671  14434  19985  11740  14178  14014   9129  22767  19104
## 1883    309650 369924 382080 470868 347565 391558 290224 202149 479760 408186
## 2880933 24885  39404  19392  30749   8808  17675   9260   5945  30729  14509
## 1725411 18762  18008  20447  35173  19992  10833  14935   9502  16057  18450
## 2781734 15218  20492  16892  23422  25282   7588   7981   5138  14959  18562
## 659352  12941  12642  12009  14695   9973  12879  13365  10479  15275  14166
##      MD2105 MD2106 MD2115 MD2116 MD2125 MD2126 MD2135 MD2136 MP2047 MP2048
## 2062    14868   4940  15930  16647   8553  17876   8239   8370  11954  15902
## 1883    307824 100633 327070 342009 202939 409587 209898 188357 266580 372843
## 2880933 15016   5732  25734  18347  13173  19148   8931  12668  22207  30320
## 1725411 14369   4273  19426  20289  13072  35017  19699  12802  18386  23309
## 2781734 11935   2538   9389  12826  11619  24599  17101   9577  16844  19737
## 659352   9637   3142  12613  12450   7116  16401   7031   7821   8262  12072
##      MP2049 MP2050 MP2057 MP2058 MP2059 MP2060 MP2067 MP2068 MP2069 MP2070
## 2062    17699  26567  22846  27753  22940  26981  21131  29184  17415  15071
## 1883    405044 475984 527148 750581 438466 604749 481017 583698 353252 298480
## 2880933 46649  61499  33793  34519  29838  32627  18138  21247  15434   9944
## 1725411 23836  18726  27569  60617  25309  54020  24732  28183  13726  11723
## 2781734 24247  17282  33500  46613  17270  31754  24172  24470  11571   6597
## 659352  20157  24266  17128  22909  15093  15794  20061  21102  10627  11300
##      MP2077 MP2078 MP2079 MP2080 MP2087 MP2088 MP2089 MP2090 MP2097 MP2098
## 2062    26088  10789   1161   4512  27744     0  22578  13129  15570  20459
## 1883    293779 219233 24597  96283 591983     0 500426 280636 323878 441849
## 2880933  8648   6393   985   3443  23886     0  17543   9893  18427  26871
## 1725411 15507   9547  1202   4677  31963     0  18896  12400  16708  17172
## 2781734  6602   4916   713   2942  21732     0  23800  12452  14494  14646
## 659352  15175  10060  1110   5409  27747     0  17421  10112  13223  12619
##      MP2099 MP2100 MP2107 MP2108 MP2109 MP2110 MP2117 MP2118 MP2119 MP2120
## 2062    17908  17823  14427   9408   1821   7995  17162  17011  12864  15126
## 1883    381823 347101 340913 196757 43535 185697 373971 344457 278616 340307
## 2880933 28428  16979  21746  13925  2387  14561  30177  24863  20282  23002
## 1725411 17604  17832  23379  10078  2112  12573  19881  18899  13656  20846
## 2781734 10650   9440  14349   5680  1384   9121  26676  16705  11258  20747
## 659352  12267  15635  12030   6626  1425   6839  14136  12855   9862  13283
##      MP2127 MP2128 MP2129 MP2130 MP2137 MP2138 MP2139 MP2140
```

```
## 2062      10378 14087 11425 8383 4413 9744 7347 6630
## 1883      223557 289910 249480 190633 81445 203351 130999 126168
## 2880933 21811 16448 28085 19152 2477 5487 3194 4715
## 1725411 13373 20144 13927 8668 4196 16516 5110 6029
## 2781734 9515 15650 12163 6318 2218 10135 2391 2851
## 659352 13243 12443 9377 12225 7081 11539 5731 5461
```

Cargar los metadatos

Revisando el archivo de metadatos, nos dimos cuenta que hay un desfase de dos muestras entre los metadatos y las muestras de la `otu_table`, por lo tanto se deben ver cuales son y quitarlas del archivo de metadatos, para que este coincida con nuestro objeto `phyloseq`

```
# $ 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/Data1/metadata.csv",head=1)
```

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 para el buen funcionamiento de nuestro análisis

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

```
##      Treatment Samples
## MP2068  healthy  MP2068
## MP2087  healthy  MP2087
## MD2095   wilted  MD2095
## MP2058  healthy  MP2058
## MP2099  healthy  MP2099
## MP2098  healthy  MP2098
```

Filtro de calidad

Luego de una breve revisión de los datos, ya que se pueden observar muestras con conteos en ceros, como **MP2088**; se obtuvo por medio de Solena, una tabla que nos muestra la calidad de las muestras, (`fastp_kraken_summary`), para ver que muestra podemos eliminar de nuestro dataset, que no cumpla ciertos estandares de calidad.

Esta tabla contiene la siguiente información:

- ID de la muestra
- Reads_B - Reads_Before -> total de reads crudos
- Reads_A - Reads_After -> total de reads después del análisis de calidad
- Reads_diff -> diferencia entre Reads_B y Reads_A
- Q30_B -> porcentaje arriba de 30 (escala fred) antes del análisis de calidad
- Q30_A -> porcentaje arriba de 30 (escala fred) después del análisis de calidad
- LowQua -> reads de baja calidad
- N_reads -> reads que contienen N y se descartan
- too_short -> no pasan el tamaño mínimo de calidad
- Duplication -> porcentaje de duplicados
- LengthR1 -> longitud promedio de los reads
- LengthR2 -> longitud promedio de los reads
- Classified -> porcentaje de clasificados del total después del filtrado

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

Eliminamos las muestras de baja calidad, usando el filtro de menos de 25 millones de reads luego del análisis de calidad, procedemos a eliminar las muestras por su nombre como se muestra aquí

```
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
```

Podemos hacer la cuenta de cuantas lecturas tenemos por muestra, así es posible también ver la calidad de las diferentes muestras,

```
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
```

como podemos ver la muestra **MP2088** contiene solo 2 lecturas, y es por esto nos genera un problema al analizar los datos y es una de las muestras eliminadas por el filtro de calidad.

```
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**, nos entrega un resumen general sobre las variables del data frame, con esto podemos darnos una idea de la uniformidad de los datos, ya que podemos ver datos estadísticos como mínimo, máximo, media, mediana, primer y tercer cuartil.

```
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

Normalizacion de los datos

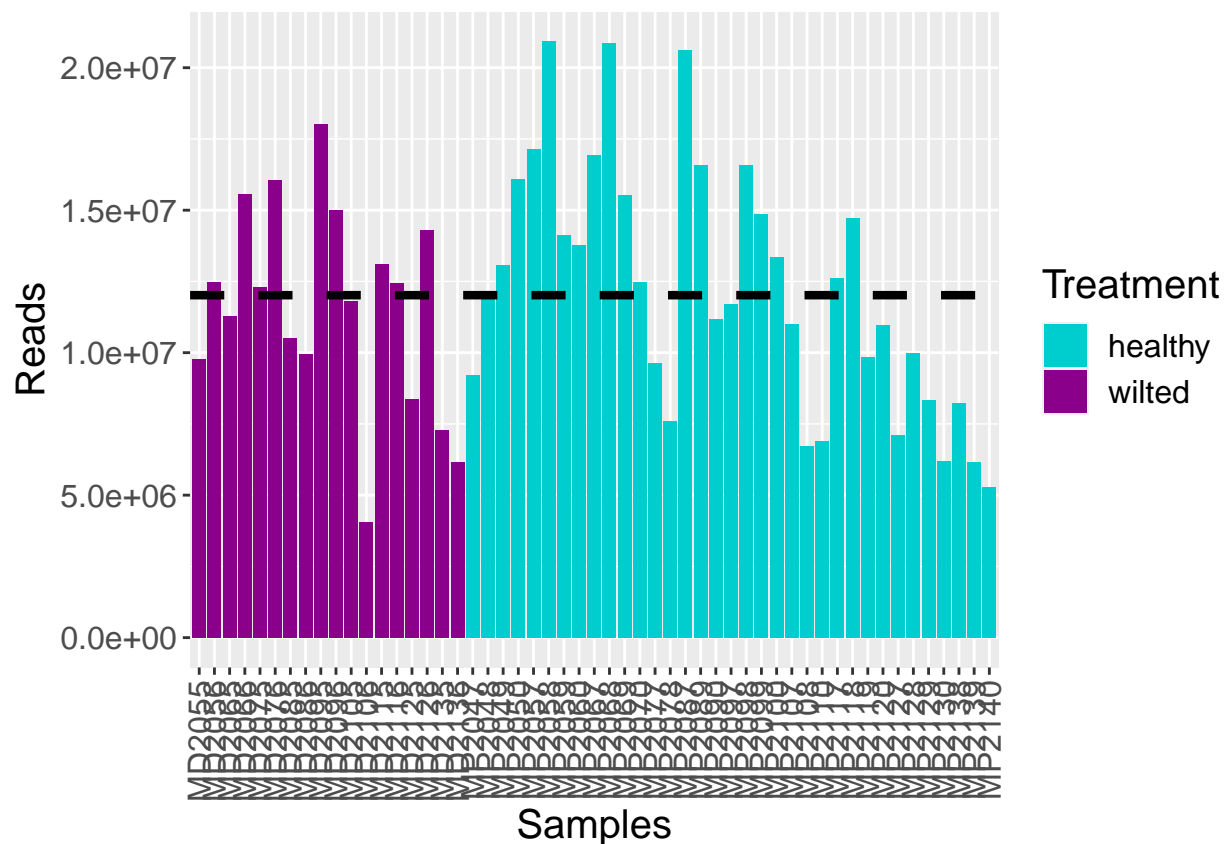
Queremos ver la profundidad de las muestras,

```
dprof <- data.frame(Samples = colnames(fresa_kraken_fil@otu_table@.Data),
                    Reads = sample_sums(fresa_kraken_fil),
                    Treatment = fresa_kraken_fil@sam_data@.Data[[1]])

mu_Samples <- mean(dprof$Reads)

ggplot(data = dprof, mapping = aes(x = Samples, y = Reads))+
  geom_bar(stat = "identity", aes( fill = Treatment)) +
  geom_hline(yintercept=mu_Samples, color = "black", linetype="dashed", size=1.5) +
  scale_fill_manual(values = c("cyan3","darkmagenta")) +
  theme(text = element_text(size = 15),
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
```



Podemos ver que nuestros datos tienen diferentes profundidades entre muestras. La línea marca la media de la profundidad de las muestras.

Esto deja en claro que los datos necesitan normalización. McMurdi et al. encontró una metodología para normalizar los datos. En su artículo “Waste Not, Want Not: Why Rarefying Microbiome Data Is Inadmissible”, hacen una interesante discusión sobre este tema constante en este tipo de análisis. Usamos su metodología:

Metodo de Normalización

Cada una de las funciones de normalización toma un objeto phloseq y devuelve un objeto physeq cuya tabla otu se transforma.

edgeRnorm

Esta función escala datos NGS normalizados utilizando la función de normalización provista en edgeR.

```
edgeRnorm = function(phy, ...){
  require("edgeR")
  require("phyloseq")
  if (!taxa_are_rows(phy)) {
    phy <- t(phy) #transpone el objeto phyoseq
  }
  x = as(otu_table(phy), "matrix")
  x = x + 1
  y = edgeR::DGEList(counts = x, remove.zeros = TRUE)
  z = edgeR::calcNormFactors(y, ...)
  if (!all(is.finite(z$samples$norm.factors))) {
    stop("Something wrong with edgeR::calcNormFactors on this data, non-finite $norm.factors")
  }
  return(z)
}
z <- edgeRnorm(fresa_kraken_fil, method = "TMM")
```

unimos z con el resto del objeto phyloseq

```
nor_fresa_kraken_fil <- merge_phyloseq(otu_table(z@.Data[[1]], taxa_are_rows = TRUE),
  tax_table(fresa_kraken_fil@tax_table@.Data),
  fresa_kraken_fil@sam_data)
```

method="TMM" es la media recortada ponderada de los valores M (a la referencia) propuesta por Robinson y Oshlack (2010), donde los pesos son del método delta en datos binomiales. Si refColumnno se especifica, se utiliza la biblioteca cuyo cuartil superior está más cerca del cuartil superior medio.

El método "TMM" es una media recortada ponderada de los valores M, que compara los niveles de expresión génica entre dos grupos (un grupo de referencia y un grupo de prueba). El método fue propuesto por Robinson y Oshlack en 2010 y utiliza pesos delta para datos binomiales. Si no se especifica la "refColumn" (la columna de referencia), se utiliza la biblioteca cuyo cuartil superior está más cerca del cuartil superior medio. En términos más simples, el método TMM es una técnica estadística utilizada para comparar los niveles de expresión génica y utiliza pesos y cálculos específicos para hacerlo. La "refColumn" se refiere a la columna de datos que sirve como referencia para la comparación

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

Queremos hacer un análisis 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 número de especies diferentes en ese ambiente o la abundancia de especies en ese ambiente. Para medir esta diversidad se tienen diferentes índices de medida, los cuales pueden representar cosas diferentes, en este caso usaremos los índices Shannon, Simpson, Chao1.

```
index = estimate_richness(nor_fresa_kraken_fil)
index
```

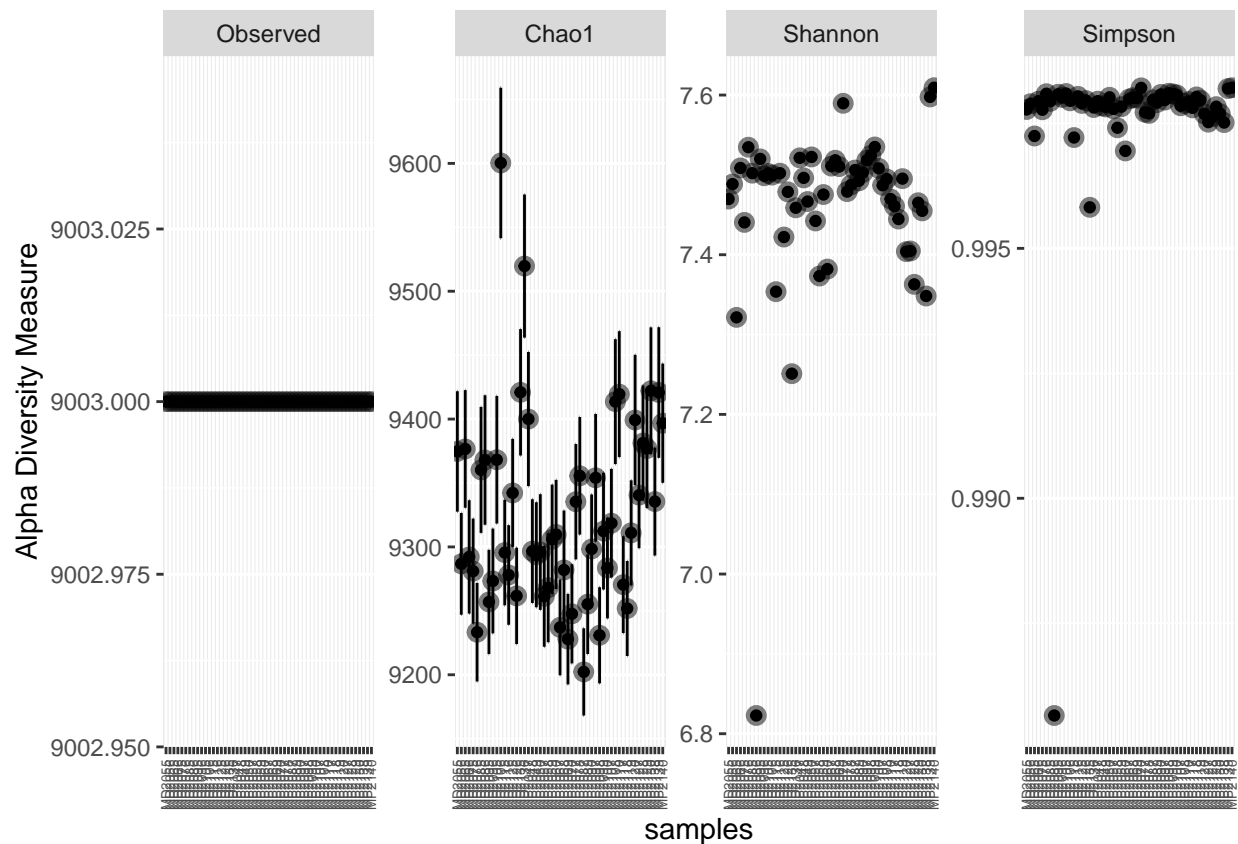
##	Observed	Chao1	se.chao1	ACE	se.ACE	Shannon	Simpson
## MD2055	9003	9374.684	47.11521	9237.352	38.12485	7.469756	0.9977780
## MD2056	9003	9286.766	39.79752	9190.820	35.38217	7.488525	0.9978945
## MD2065	9003	9376.642	46.00846	9252.668	38.83789	7.321537	0.9972478
## MD2066	9003	9292.184	44.33790	9150.809	33.55072	7.508710	0.9979434
## MD2075	9003	9281.045	41.10020	9165.906	34.11406	7.440511	0.9977728
## MD2076	9003	9233.286	38.53354	9147.509	29.34537	7.534661	0.9980936
## MD2085	9003	9360.239	49.26676	9195.634	35.91265	7.502162	0.9979419
## MD2086	9003	9368.006	50.64683	9196.199	35.29258	6.822819	0.9856544
## MD2095	9003	9256.951	40.75246	9145.014	31.75694	7.520042	0.9980824
## MD2096	9003	9273.359	40.98803	9159.286	33.35405	7.498915	0.9980195
## MD2105	9003	9368.188	49.89920	9199.202	35.98074	7.501835	0.9980973
## MD2106	9003	9600.322	58.86818	9449.070	43.28001	7.499797	0.9979520
## MD2115	9003	9295.574	41.26230	9188.630	35.34367	7.353726	0.9972167
## MD2116	9003	9278.052	38.95713	9186.604	35.40516	7.502120	0.9980411
## MD2125	9003	9342.231	42.26527	9244.494	38.51923	7.422170	0.9978989
## MD2126	9003	9261.721	37.70408	9182.242	34.27231	7.478727	0.9979561
## MD2135	9003	9420.945	49.42987	9273.358	39.79405	7.250991	0.9958214
## MD2136	9003	9519.701	55.97257	9356.135	41.31626	7.458775	0.9978177
## MP2047	9003	9400.030	52.43904	9218.746	36.80215	7.521372	0.9979447
## MP2048	9003	9296.727	40.56350	9200.439	35.82191	7.496231	0.9978517
## MP2049	9003	9293.882	41.07901	9183.847	35.38553	7.466646	0.9978296
## MP2050	9003	9296.000	45.22208	9161.406	32.14319	7.522510	0.9980222
## MP2057	9003	9261.343	39.44480	9156.050	33.42262	7.442303	0.9977906
## MP2058	9003	9268.073	42.64996	9141.044	32.01400	7.373309	0.9974124
## MP2059	9003	9305.970	42.78769	9187.649	35.14304	7.475393	0.9978343
## MP2060	9003	9309.748	42.67920	9195.331	35.48249	7.382022	0.9969488
## MP2067	9003	9237.065	37.48627	9147.662	31.38872	7.511019	0.9979887
## MP2068	9003	9281.974	46.46815	9137.884	30.55923	7.518364	0.9980149
## MP2069	9003	9227.859	35.40208	9145.938	32.83604	7.510397	0.9980045
## MP2070	9003	9247.643	38.78767	9143.010	32.55292	7.589684	0.9982105
## MP2077	9003	9335.346	45.10731	9208.045	36.15465	7.479034	0.9977220
## MP2078	9003	9355.629	46.01590	9208.275	38.01375	7.487454	0.9976997
## MP2087	9003	9202.239	34.10834	9125.212	30.41749	7.506069	0.9979704
## MP2089	9003	9255.218	38.97372	9155.197	33.00244	7.492632	0.9979163
## MP2090	9003	9298.409	42.62204	9171.732	35.42093	7.503030	0.9980800
## MP2097	9003	9354.058	49.92317	9197.711	34.09566	7.518359	0.9979664
## MP2098	9003	9230.879	37.61082	9135.749	31.01423	7.524400	0.9981017
## MP2099	9003	9312.333	45.66816	9178.710	33.51893	7.534950	0.9980900
## MP2100	9003	9283.519	39.44914	9195.586	34.77926	7.508206	0.9980717
## MP2107	9003	9318.555	42.49687	9212.660	36.28516	7.486946	0.9978525
## MP2108	9003	9413.645	48.85602	9265.085	40.11934	7.494958	0.9978990
## MP2110	9003	9419.503	49.25313	9274.376	39.85665	7.469821	0.9979358
## MP2117	9003	9270.482	37.86062	9185.118	35.49402	7.460812	0.9978104
## MP2118	9003	9251.832	37.14480	9156.332	34.69186	7.444708	0.9980298

##	MP2119	9003	9311.000	41.10703	9205.507	36.89236	7.495304	0.9979642
##	MP2120	9003	9399.249	51.04920	9221.600	37.80113	7.403900	0.9976897
##	MP2127	9003	9340.503	41.51759	9248.208	39.53778	7.404701	0.9975245
##	MP2128	9003	9381.272	44.99838	9277.179	39.29236	7.362911	0.9976342
##	MP2129	9003	9376.939	46.54215	9235.457	39.02220	7.465118	0.9978353
##	MP2130	9003	9422.146	49.63169	9275.945	39.89833	7.454929	0.9976932
##	MP2138	9003	9335.472	42.20980	9223.513	38.86997	7.348296	0.9975160
##	MP2139	9003	9420.785	51.06999	9271.465	38.47666	7.597651	0.9981971
##	MP2140	9003	9396.782	46.63558	9274.798	40.05717	7.609173	0.9982131
##		InvSimpson		Fisher				
##	MD2055	450.04240		977.2751				
##	MD2056	474.94044		949.3044				
##	MD2065	363.34073		960.4673				
##	MD2066	486.25002		925.0603				
##	MD2075	449.00101		950.7302				
##	MD2076	524.55981		921.8148				
##	MD2085	485.89542		968.6592				
##	MD2086	69.70758		975.4831				
##	MD2095	521.47537		909.9865				
##	MD2096	504.92350		929.1141				
##	MD2105	525.57607		955.5814				
##	MD2106	488.27441	1095.4397					
##	MD2115	359.28705		943.7928				
##	MD2116	510.48012		949.4560				
##	MD2125	475.93779		996.4010				
##	MD2126	489.26856		934.1824				
##	MD2135	239.31470	1013.7732					
##	MD2136	458.24134	1035.4693					
##	MP2047	486.53733		984.6425				
##	MP2048	465.48137		952.2352				
##	MP2049	460.75160		944.0186				
##	MP2050	505.62486		921.6129				
##	MP2057	452.60587		915.0705				
##	MP2058	386.45340		894.9195				
##	MP2059	461.73474		935.5324				
##	MP2060	327.73864		938.2074				
##	MP2067	497.18957		916.3920				
##	MP2068	503.75235		895.1544				
##	MP2069	501.12651		925.3559				
##	MP2070	558.81662		949.3597				
##	MP2077	438.99107		979.2965				
##	MP2078	434.72976	1008.4794					
##	MP2087	492.69979		896.3054				
##	MP2089	479.91333		918.5148				
##	MP2090	520.83607		961.7011				
##	MP2097	491.74835		956.3382				
##	MP2098	526.78226		918.4300				
##	MP2099	523.54732		930.2199				
##	MP2100	518.58684		941.7946				
##	MP2107	465.65848		963.3853				
##	MP2108	475.95669	1024.0241					
##	MP2110	484.43964	1020.6993					
##	MP2117	456.71028		947.9210				
##	MP2118	507.56844		931.1825				

```
## MP2119 491.21886 976.6345
## MP2120 432.83690 963.7920
## MP2127 403.95515 1016.8939
## MP2128 422.69223 974.9651
## MP2129 461.96570 996.5129
## MP2130 433.50492 1034.9405
## MP2138 402.58288 998.4290
## MP2139 554.65411 1035.7698
## MP2140 559.64012 1057.2413
```

Podemos ver una representación visual de la diversidad dentro de las muestras, esta diversidad la podemos ver por muestra, como esta a continuación, o como se mostrara mas adelante, dividiendo los datos por grupos.

```
p<-plot_richness(physeq = nor_fresa_kraken_fil, measures = c("Observed", "Chao1", "Shannon", "simpson"))
p + geom_point(size=3, alpha=0.5) +
  theme(axis.text.x = element_text(angle=90, size=5, hjust=1, vjust=0.5))
```

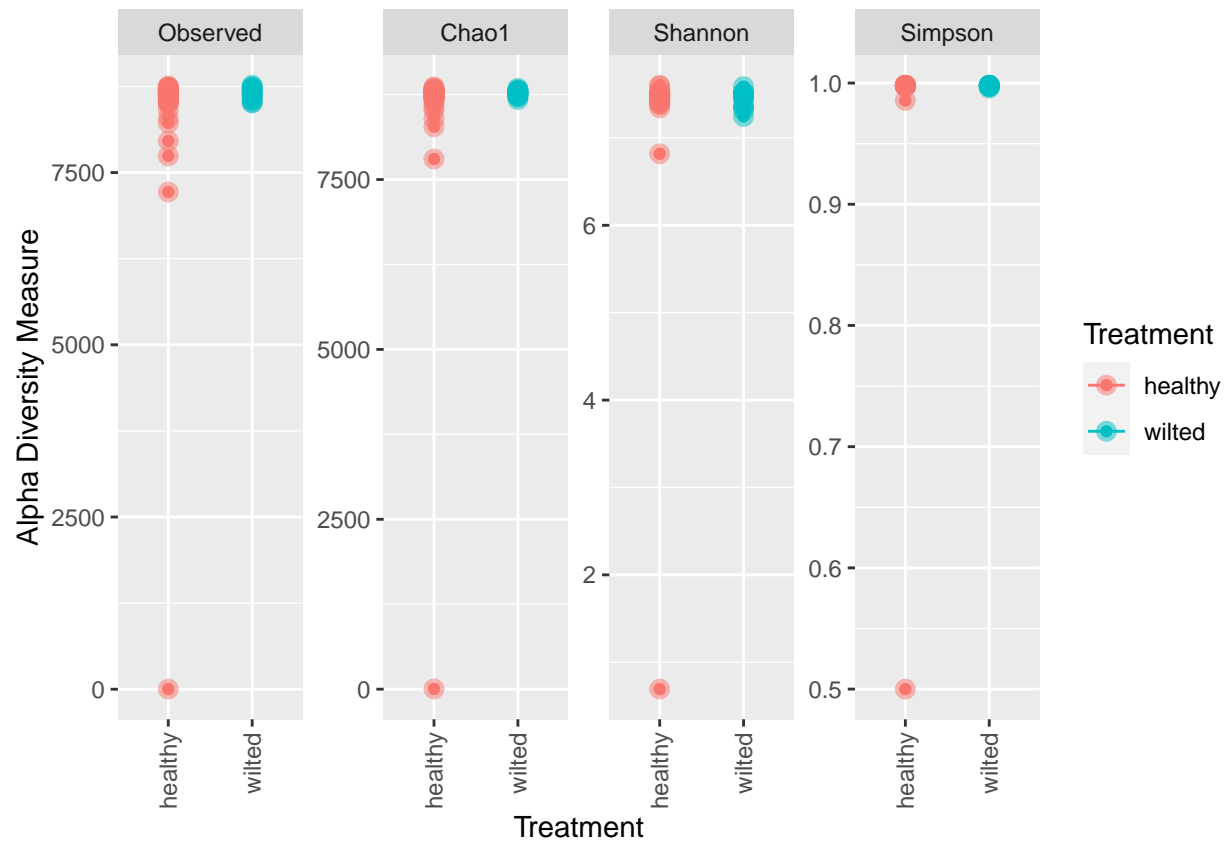


El indice Chao1 nos muestra la riqueza de especies en una comunidad

QUE NOS MUESTRA LA FIGURA??

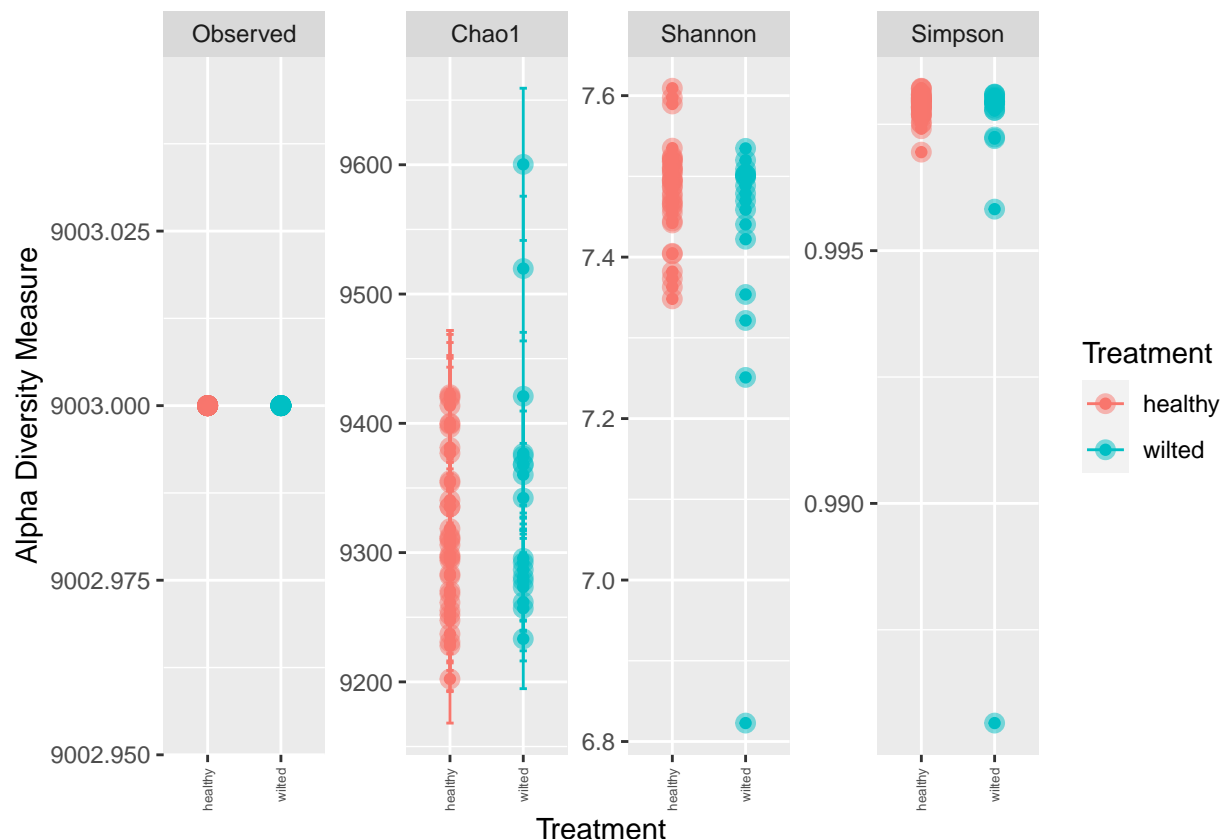
Ya que queremos diferenciar entre plantas sanas y enfermas, tomamos estos dos conjuntos por separado para mostrar la diversidad,

```
p<-plot_richness(physeq = fresa_kraken, measures = c("Observed","Chao1","Shannon","simpson"),x = "Treatment")
p + geom_point(size=3, alpha=0.5) +
  theme(axis.text.x = element_text(angle=90, size=9, hjust=1, vjust=0.5))
```



Comparando entre los datos en crudos y los datos filtrados por calidad,

```
p<-plot_richness(physeq = nor_fresa_kraken_fil, measures = c("Observed","Chao1","Shannon","simpson"),x = "Treatment")
p + geom_point(size=3, alpha=0.5) +
  theme(axis.text.x = element_text(angle=90, size=5, hjust=1, vjust=0.5))
```



Se puede ver una gran diferencia al desechar muestras de poca calidad.

QUE NOS MUESTRA LA FIGURA??

Ahora si podemos ver ***

Ahora queremos convertir las abundancias absolutas (número de lecturas por OTU) a relativas (porcentajes de lecturas asignadas a un OTU dentro de una muestra), ya que nuestros metagenomas tienen diferentes tamaños, calculando las abundancias relativas con la funcion de phyloseq 'transform_sample', tanto de los datos originales como de los 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
##	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
##	2062	0.1378134	0.1460339	0.1412250	0.1368541	0.1352811	0.2033890
##	1883	3.1005460	3.1457805	2.9064064	2.9883903	3.0763508	3.7536859
##	2880933	0.2095718	0.3069133	0.1648945	0.3364155	0.3090665	0.1141615
##	1725411	0.1899285	0.1881781	0.2019477	0.1668242	0.1398803	0.1933878
##	2781734	0.1890265	0.1338902	0.1568944	0.1456942	0.1019571	0.1022245
##	659352	0.1210218	0.1863488	0.1247436	0.1123222	0.1972816	0.3263534
##		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(nor_fresa_kraken_fil, function(x) x*100 / sum(x) )
head(percentages_fil@otu_table@.Data)
```

```
##          MD2055      MD2056      MD2065      MD2066      MD2075      MD2076
## 2062      0.1529806 0.1336162 0.1276688 0.12819788 0.09530200 0.08819518
## 1883      3.1624680 2.9647296 3.3792732 3.02033449 2.82120206 2.43554673
## 2880933 0.2541609 0.3158077 0.1715192 0.19724230 0.07150288 0.10994697
## 1725411 0.1916267 0.1443315 0.1808501 0.22561954 0.16228369 0.06738886
## 2781734 0.1554318 0.1642392 0.1494083 0.15024411 0.20522275 0.04720454
## 659352  0.1321767 0.1013262 0.1062211 0.09426578 0.08095921 0.08011524
##          MD2085      MD2086      MD2095      MD2096      MD2105      MD2106
## 2062      0.13304636 0.09184833 0.12635611 0.12730496 0.12599353 0.12163066
## 1883      2.75514666 2.03364084 2.66254100 2.71992823 2.60837697 2.47726779
## 2880933 0.08791597 0.05981711 0.17054301 0.09668647 0.12724761 0.14112702
## 1725411 0.14178954 0.09560074 0.08911746 0.12294707 0.12176521 0.10521139
## 2781734 0.07577425 0.05169864 0.08302387 0.12369338 0.10114054 0.06250157
## 659352  0.12688531 0.10542941 0.08477758 0.09440091 0.08166828 0.07737000
##          MD2115      MD2116      MD2125      MD2126      MD2135      MD2136
## 2062      0.12150349 0.13360469 0.10226117 0.1248715 0.11303536 0.1354230
## 1883      2.49452449 2.74472258 2.42610273 2.8609882 2.87937014 3.0471876
## 2880933 0.19627722 0.14724765 0.15749225 0.1337565 0.12252814 0.2049545
## 1725411 0.14816699 0.16283273 0.15628482 0.2446021 0.27024232 0.2071223
## 2781734 0.07161621 0.10294014 0.13891453 0.1718320 0.23460325 0.1549494
## 659352  0.09620520 0.09992264 0.08508216 0.1145686 0.09646416 0.1265415
##          MP2047      MP2048      MP2049      MP2050      MP2057      MP2058
## 2062      0.12983160 0.13082461 0.1352714 0.1649391 0.13321532 0.1325880
## 1883      2.89507630 3.06716782 3.0955362 2.9550043 3.07367804 3.5857247
## 2880933 0.24117943 0.24943299 0.3565203 0.3818036 0.19704462 0.1649110
## 1725411 0.19968328 0.19175763 0.1821731 0.1162607 0.16075399 0.2895879
## 2781734 0.18293712 0.16237289 0.1853141 0.1072961 0.19533621 0.2226872
## 659352  0.08973639 0.09931745 0.1540565 0.1506541 0.09987505 0.1094470
##          MP2059      MP2060      MP2067      MP2068      MP2069      MP2070
## 2062      0.1622535 0.1955836 0.1247961 0.1397562 0.11202498 0.12085291
## 1883      3.1011210 4.3836336 2.8406763 2.7951195 2.27223008 2.39333179
## 2880933 0.2110406 0.2365096 0.1071208 0.1017488 0.09928259 0.07974271
## 1725411 0.1790086 0.3915804 0.1460620 0.1349628 0.08829621 0.09400740
## 2781734 0.1221516 0.2301815 0.1427549 0.1171826 0.07443460 0.05290522
## 659352  0.1067545 0.1144928 0.1184772 0.1010545 0.06836251 0.09061563
##          MP2077      MP2078      MP2087      MP2089      MP2090      MP2097
## 2062      0.27100246 0.14201498 0.1344148 0.1360885 0.11738112 0.1328243
## 1883      3.05167318 2.88549697 2.8679557 3.0161818 2.50887175 2.7627648
## 2880933 0.08984247 0.08415605 0.1157242 0.1057415 0.08845155 0.1571952
## 1725411 0.16109111 0.12566812 0.1548544 0.1138963 0.11086392 0.1425317
## 2781734 0.06858941 0.06471619 0.1052888 0.1434538 0.11132880 0.1236458
## 659352  0.15764243 0.13242009 0.1344294 0.1050062 0.09040939 0.1128039
##          MP2098      MP2099      MP2100      MP2107      MP2108      MP2110
## 2062      0.12321664 0.12057464 0.13349982 0.1308844 0.13966109 0.11571187
## 1883      2.66096152 2.57067896 2.59975617 3.0926211 2.92054814 2.68727638
```

```
## 2880933 0.16183175 0.19140188 0.12717835 0.1972792 0.20670851 0.21072989
## 1725411 0.10342128 0.11852792 0.13356723 0.2120930 0.14960614 0.18196110
## 2781734 0.08820890 0.07170922 0.07071206 0.1301769 0.08432508 0.13200646
## 659352 0.07600166 0.08259588 0.11711194 0.1091399 0.09836689 0.09898314
##      MP2117      MP2118      MP2119      MP2120      MP2127      MP2128      MP2129
## 2062 0.1358584 0.11556768 0.1306844 0.1377095 0.1458632 0.1411077 0.1367186
## 1883 2.9602775 2.34000769 2.8302284 3.0980139 3.1418144 2.9037956 2.9851830
## 2880933 0.2388822 0.16890869 0.2060374 0.2094092 0.3065390 0.1647559 0.3360651
## 1725411 0.1573814 0.12839343 0.1387296 0.1897819 0.1879540 0.2017756 0.1666565
## 2781734 0.2111691 0.11348893 0.1143704 0.1888806 0.1337349 0.1567630 0.1455492
## 659352 0.1119053 0.08733471 0.1001897 0.1209317 0.1861270 0.1246411 0.1122131
##      MP2130      MP2138      MP2139      MP2140
## 2062 0.1351010 0.11841641 0.11913903 0.12567012
## 1883 3.0719039 2.47103264 2.12400836 2.39114354
## 2880933 0.3086342 0.06668745 0.05180310 0.08937721
## 1725411 0.1396935 0.20070639 0.08286875 0.11428002
## 2781734 0.1018253 0.12316764 0.03878342 0.05405085
## 659352 0.1970115 0.14022836 0.09293753 0.10351533
```

Ahora ya es posible una buena comparación de abundancias dadas por porcentajes con índices beta.

Diversidad Beta

La diversidad beta mide la diferencia entre dos o más 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. aquí 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"
##
## $designndist
## [1] "ANY"
```


Siendo las siguientes las mas usadas:

Disimilitud de Bray-Curtis

Este índice de disimilitud se basa en la composición y la abundancia de las especies en diferentes sitios. Mide la similitud entre dos muestras o poblaciones, en terminos de las especies que comparten, ponderando la abundancia de cada especie en cada población. La fórmula del índice de disimilitud de Bray-Curtis es:

$$d_{BC} = 1 - \frac{2S}{(S_a + S_b)}$$

donde d_{BC} es el índice de disimilitud de Bray-Curtis, S es el número de especies compartidas entre las poblaciones a y b , y S_a y S_b son los números de especies exclusivas de los sitios a y b , respectivamente.

Distancia Jaccard

Este índice de disimilitud se basa en la presencia o ausencia de especies en diferentes poblaciones. Compara la proporción de especies que son comunes entre dos poblaciones, en relación con el total de especies encontradas en ambas poblaciones. Este índice puede ser útil para comparar la diversidad de especies entre diferentes poblaciones o para evaluar la similitud de la composición de especies en diferentes comunidades. La fórmula del índice de disimilitud de Jaccard es:

$$d_{JC} = 1 - \frac{S}{(S_a + S_b - S)}$$

donde d_{JC} es el índice de disimilitud de Jaccard, S es el número de especies compartidas entre las poblaciones a y b , y S_a y S_b son los números de especies exclusivas de los sitios a y b , respectivamente.

Euclideana

Esta distancia se utiliza comúnmente en el análisis de datos numéricos y se basa en la diferencia de las abundancias o proporciones de las diferentes especies en diferentes muestras. La distancia Euclidiana entre dos muestras es la raíz cuadrada de la suma de las diferencias cuadráticas entre las proporciones de cada especie en ambas muestras.

La distancia euclideana entre dos muestras A y B es:

$$d_{euclidean}(A, B) = \sqrt{\sum (A_i - B_i)^2}$$

Donde A_i y B_i son las abundancias o proporciones de la especie i en las muestras A y B , respectivamente.

La distancia Euclidiana es simétrica y satisface la desigualdad del triángulo, lo que significa que cumple con las propiedades de una verdadera distancia.

Manhattan

Esta distancia también se utiliza en el análisis de datos numéricos y se basa en la diferencia de las abundancias o proporciones de las diferentes especies en diferentes muestras. La distancia de Manhattan entre dos muestras es la suma de las diferencias absolutas entre las proporciones de cada especie en ambas muestras.

La fórmula de la distancia de Manhattan entre dos muestras A y B se calcula como:

$$d_{Manhattan}(A, B) = \sum |A_i - B_i|$$

Donde A_i y B_i son las abundancias o proporciones de la especie i en las muestras A y B , respectivamente.

Esta distancia también es simétrica y cumple con la desigualdad del triángulo.

Divergencia de Jensen-Shannon (JSD)

Esta distancia se utiliza para comparar la similitud entre dos distribuciones de probabilidad. En el análisis de datos de diversidad, la distribución de probabilidad puede representar la proporción de diferentes especies en diferentes muestras. La distancia de JSD entre dos distribuciones de probabilidad se calcula como la raíz cuadrada de la divergencia de Kullback-Leibler entre las dos distribuciones, dividida por dos.

La fórmula de la distancia de JSD entre dos distribuciones de probabilidad P y Q se calcula como:

$$d_{JSD}(P, Q) = \frac{\sqrt{(D_{KL}(P, M) + D_{KL}(Q, M))}}{2}$$

Donde $D_{KL}(P, M)$ y $D_{KL}(Q, M)$ son las divergencias de Kullback-Leibler entre las distribuciones P y Q y la media M de ambas distribuciones, respectivamente.

Esta distancia es simétrica y también satisface la desigualdad del triángulo.

UniFrac

Este índice de disimilitud se basa en la filogenia de las especies presentes en diferentes sitios. Compara la similitud entre dos sitios en términos de la diversidad filogenética de las especies, considerando la contribución relativa de cada rama del árbol filogenético. Este índice puede ser útil para evaluar la similitud de la evolución de las especies en diferentes comunidades o para comparar la estructura filogenética de diferentes comunidades. La fórmula del índice de disimilitud de UniFrac es más compleja que las de Bray-Curtis y Jaccard, y se basa en un análisis de la distribución de ramas filogenéticas únicas o compartidas entre los sitios.

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

Usando la distancia de Bray-Curtis

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. En este caso usaremos NMDS (Non-metric Multidimensional Scaling), es una herramienta de análisis exploratorio de datos que se utiliza para visualizar la similitud o disimilitud de una colección de objetos (por ejemplo, especies, sitios, genes) en un espacio de baja dimensionalidad.

Podemos ver la diversidad beta entre las muestras, tanto para los datos en crudo,

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

```
## Wisconsin double standardization
## Run 0 stress 8.456249e-05
## Run 1 stress 9.875392e-05
## ... Procrustes: rmse 6.986823e-05 max resid 0.0002935905
## ... Similar to previous best
## Run 2 stress 8.77576e-05
## ... Procrustes: rmse 5.447404e-05 max resid 0.0002110028
## ... Similar to previous best
## Run 3 stress 9.949391e-05
```

```

## ... Procrustes: rmse 7.398994e-05  max resid 0.0003405072
## ... Similar to previous best
## Run 4 stress 8.625175e-05
## ... Procrustes: rmse 4.375498e-05  max resid 0.0001234933
## ... Similar to previous best
## Run 5 stress 9.430932e-05
## ... Procrustes: rmse 7.138228e-05  max resid 0.0003132232
## ... Similar to previous best
## Run 6 stress 9.618639e-05
## ... Procrustes: rmse 7.665885e-05  max resid 0.0002719477
## ... Similar to previous best
## Run 7 stress 9.170298e-05
## ... Procrustes: rmse 6.907414e-05  max resid 0.0003231745
## ... Similar to previous best
## Run 8 stress 9.407518e-05
## ... Procrustes: rmse 6.472702e-05  max resid 0.0002161202
## ... Similar to previous best
## Run 9 stress 8.531122e-05
## ... Procrustes: rmse 4.72716e-05   max resid 0.0001349753
## ... Similar to previous best
## Run 10 stress 9.697347e-05
## ... Procrustes: rmse 4.412511e-05  max resid 0.0001875611
## ... Similar to previous best
## Run 11 stress 9.309507e-05
## ... Procrustes: rmse 6.988189e-05  max resid 0.0003162781
## ... Similar to previous best
## Run 12 stress 9.114731e-05
## ... Procrustes: rmse 6.567973e-05  max resid 0.0002816173
## ... Similar to previous best
## Run 13 stress 9.438447e-05
## ... Procrustes: rmse 7.602945e-05  max resid 0.0003122474
## ... Similar to previous best
## Run 14 stress 9.193484e-05
## ... Procrustes: rmse 7.276739e-05  max resid 0.0003144731
## ... Similar to previous best
## Run 15 stress 9.150757e-05
## ... Procrustes: rmse 5.140836e-05  max resid 0.0001790468
## ... Similar to previous best
## Run 16 stress 9.775658e-05
## ... Procrustes: rmse 7.048376e-05  max resid 0.0002873929
## ... Similar to previous best
## Run 17 stress 9.638636e-05
## ... Procrustes: rmse 6.867623e-05  max resid 0.000230144
## ... Similar to previous best
## Run 18 stress 9.142268e-05
## ... Procrustes: rmse 7.486504e-05  max resid 0.000254439
## ... Similar to previous best
## Run 19 stress 9.386887e-05
## ... Procrustes: rmse 6.053598e-05  max resid 0.0002127678
## ... Similar to previous best
## Run 20 stress 9.760333e-05
## ... Procrustes: rmse 7.145882e-05  max resid 0.0002795787
## ... Similar to previous best
## *** Best solution repeated 20 times

```

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

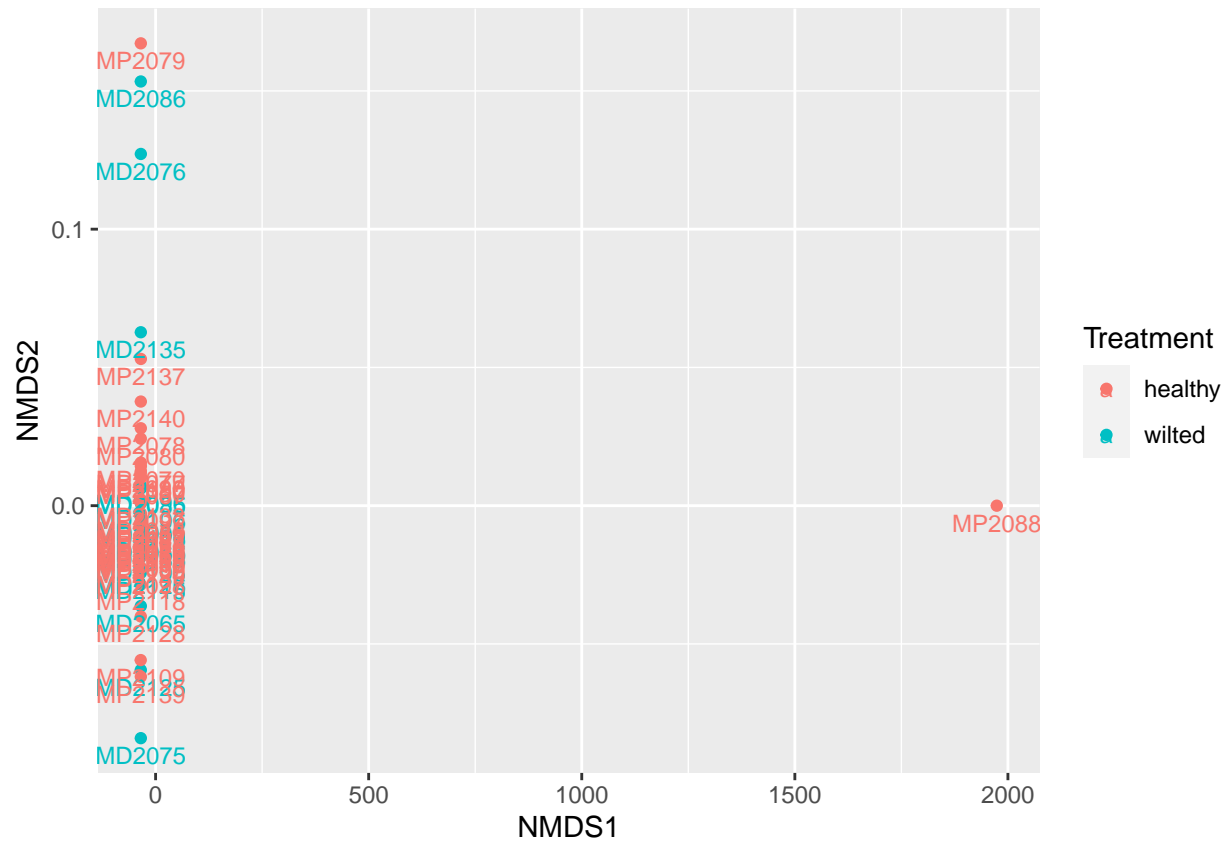
Como para los datos ya filtrados por calidad

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

```
## Wisconsin double standardization  
## Run 0 stress 0.1628248  
## Run 1 stress 0.1552463  
## ... New best solution  
## ... Procrustes: rmse 0.06102624 max resid 0.3504903  
## Run 2 stress 0.1562696  
## Run 3 stress 0.1552467  
## ... Procrustes: rmse 0.0007592585 max resid 0.003118551  
## ... Similar to previous best  
## Run 4 stress 0.1562701  
## Run 5 stress 0.1555662  
## ... Procrustes: rmse 0.01139096 max resid 0.06689494  
## Run 6 stress 0.1651551  
## Run 7 stress 0.1817245  
## Run 8 stress 0.1633759  
## Run 9 stress 0.1661768  
## Run 10 stress 0.1631324  
## Run 11 stress 0.171304  
## Run 12 stress 0.1703594  
## Run 13 stress 0.163606  
## Run 14 stress 0.1679944  
## Run 15 stress 0.1551777  
## ... New best solution  
## ... Procrustes: rmse 0.01064628 max resid 0.06323831  
## Run 16 stress 0.1596502  
## Run 17 stress 0.1721083  
## Run 18 stress 0.154614  
## ... New best solution  
## ... Procrustes: rmse 0.01661016 max resid 0.07370043  
## Run 19 stress 0.1631516  
## Run 20 stress 0.1562704  
## *** Best solution was not repeated -- monoMDS stopping criteria:  
##      2: no. of iterations >= maxit  
##     18: stress ratio > sratmax
```

Ademas lo queremos diferenciar por color entre plantas sanas y enfermas, para los datos en crudo,

```
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)
```



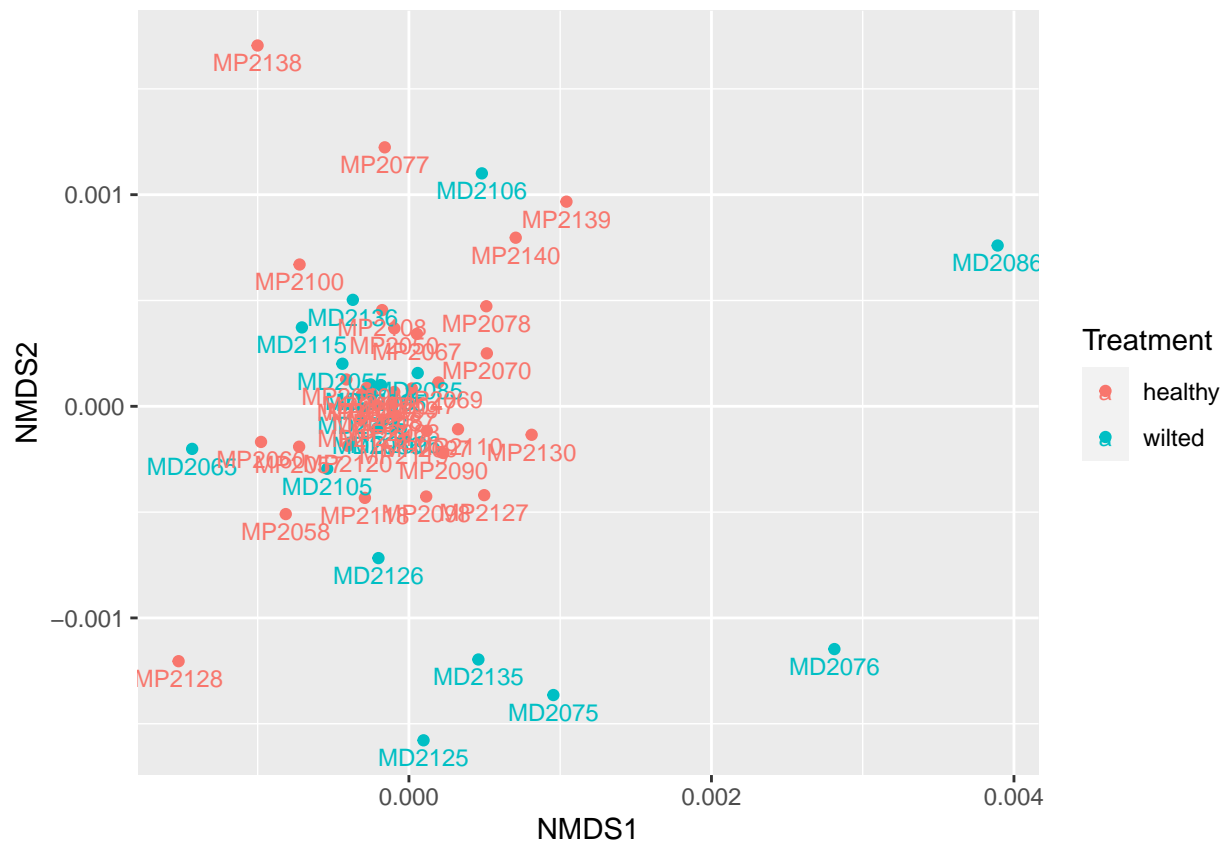
Y para los datos ya filtrados por calidad,

```
plot_ordination(physeq = percentages_fil, ordination = meta_ord_fil, color = "Treatment") +  
  geom_text(mapping = aes(label = colnames(nor_fresa_kraken_fil@otu_table@.Data)), size = 3, vjust = 1.1)
```



```
## Run 7 stress 0.1959436
## Run 8 stress 0.182807
## Run 9 stress 0.1903724
## Run 10 stress 0.1674833
## Run 11 stress 0.1768367
## Run 12 stress 0.1878911
## Run 13 stress 0.1822127
## Run 14 stress 0.1865104
## Run 15 stress 0.1876896
## Run 16 stress 0.1705372
## Run 17 stress 0.1866009
## Run 18 stress 0.1990039
## Run 19 stress 0.1796643
## Run 20 stress 0.1827531
## *** Best solution repeated 1 times
```

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



Usando la distancia de Jaccard

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

## Wisconsin double standardization
```

```

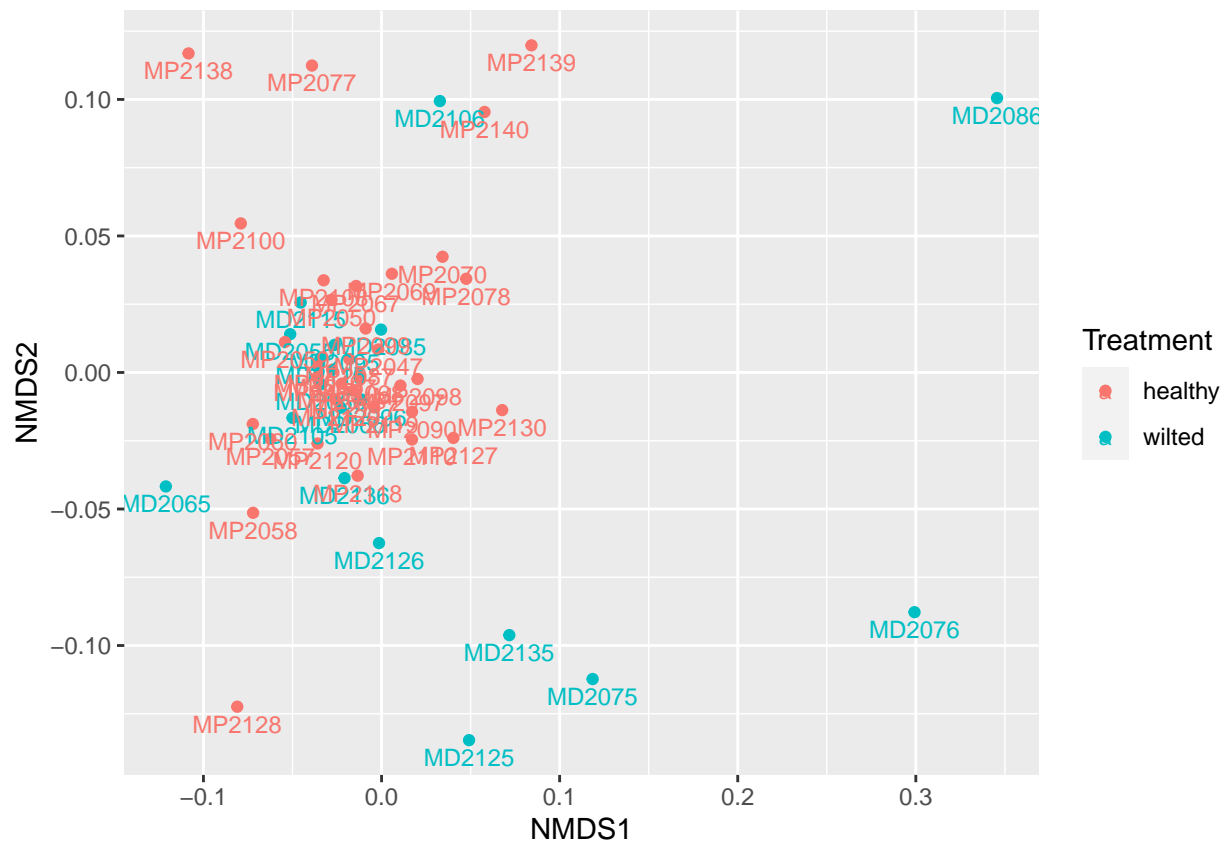
## Run 0 stress 0.1628248
## Run 1 stress 0.1691754
## Run 2 stress 0.1562697
## ... New best solution
## ... Procrustes: rmse 0.05774505  max resid 0.338685
## Run 3 stress 0.1625829
## Run 4 stress 0.1755155
## Run 5 stress 0.1555661
## ... New best solution
## ... Procrustes: rmse 0.009865285  max resid 0.065214
## Run 6 stress 0.185058
## Run 7 stress 0.1619741
## Run 8 stress 0.1602791
## Run 9 stress 0.1551784
## ... New best solution
## ... Procrustes: rmse 0.01642229  max resid 0.07231938
## Run 10 stress 0.1731075
## Run 11 stress 0.1661768
## Run 12 stress 0.1630647
## Run 13 stress 0.1651629
## Run 14 stress 0.1711671
## Run 15 stress 0.4042247
## Run 16 stress 0.1685324
## Run 17 stress 0.1833495
## Run 18 stress 0.1552463
## ... Procrustes: rmse 0.01058254  max resid 0.0628366
## Run 19 stress 0.1601099
## Run 20 stress 0.1546141
## ... New best solution
## ... Procrustes: rmse 0.01676281  max resid 0.0739621
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      3: no. of iterations >= maxit
##     17: stress ratio > sratmax

```

```

plot_ordination(physeq = percentages_fil, ordination = meta_ord_fil_4, color = "Treatment") +
  geom_text(mapping = aes(label = colnames(nor_fresa_kraken_fil@otu_table@.Data)), size = 3, vjust = 1.1)

```

Usando la distancia Manhattan

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

## Wisconsin double standardization
## Run 0 stress 0.1628248
## Run 1 stress 0.1631947
## ... Procrustes: rmse 0.08486974 max resid 0.366807
## Run 2 stress 0.1631959
## ... Procrustes: rmse 0.05409252 max resid 0.3559524
## Run 3 stress 0.1630615
## ... Procrustes: rmse 0.08373087 max resid 0.3669406
## Run 4 stress 0.1739775
## Run 5 stress 0.1628194
## ... New best solution
## ... Procrustes: rmse 0.002486551 max resid 0.01554611
## Run 6 stress 0.1664371
## Run 7 stress 0.1914741
## Run 8 stress 0.1635879
## Run 9 stress 0.1619741
## ... New best solution
## ... Procrustes: rmse 0.06090922 max resid 0.356157
## Run 10 stress 0.1696558
## Run 11 stress 0.1562707
```

```
## ... New best solution
## ... Procrustes: rmse 0.0511813  max resid 0.3221243
## Run 12 stress 0.1806602
## Run 13 stress 0.1602793
## Run 14 stress 0.1800755
## Run 15 stress 0.1575003
## Run 16 stress 0.1626586
## Run 17 stress 0.1665724
## Run 18 stress 0.1769639
## Run 19 stress 0.182292
## Run 20 stress 0.1755855
## *** 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(nor_fresa_kraken_fil@otu_table@.Data)), size = 3, vjust = 1.1)
```



Usando la distancia de Divergencia de Jensen-Shannon (JSD)

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

```
## Run 0 stress 8.881142e-05
```

```

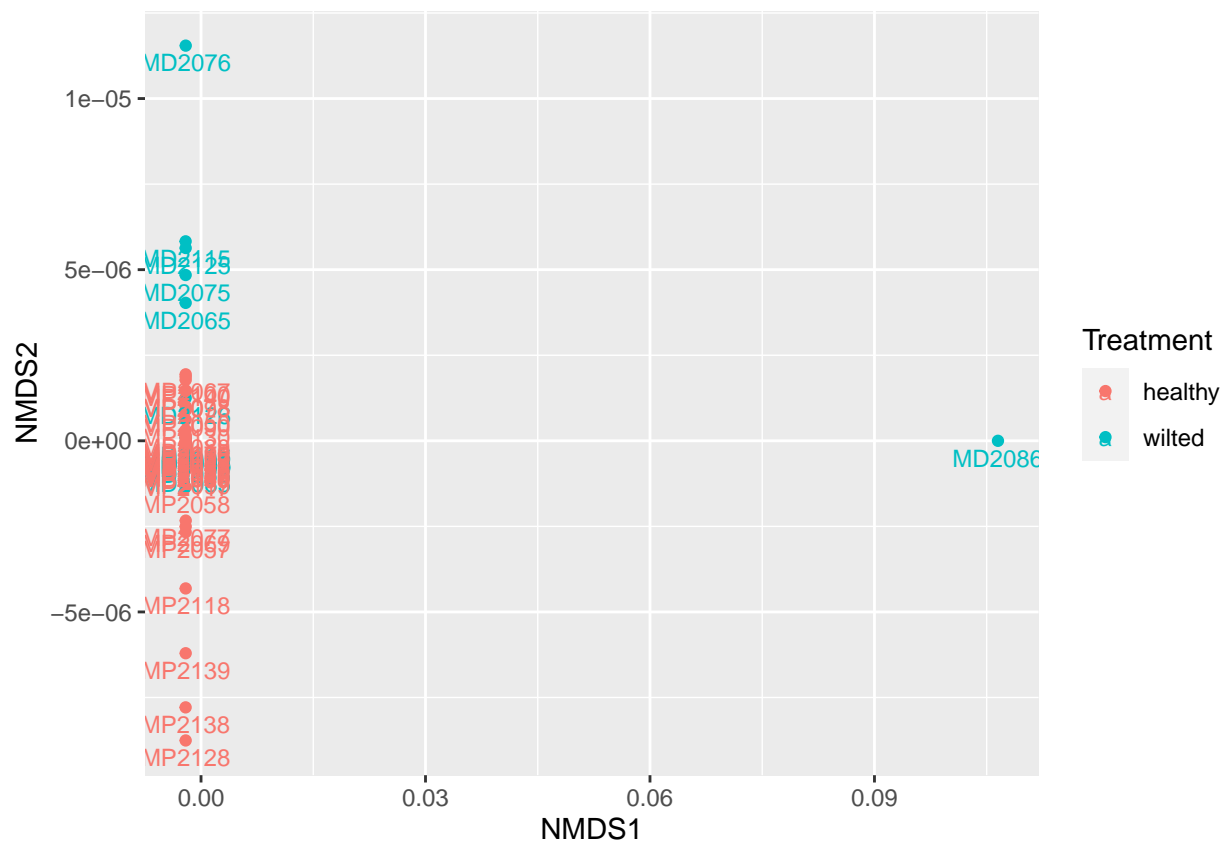
## Run 1 stress 9.522029e-05
## ... Procrustes: rmse 8.009684e-05  max resid 0.000350472
## ... Similar to previous best
## Run 2 stress 9.389937e-05
## ... Procrustes: rmse 7.53497e-05  max resid 0.000338647
## ... Similar to previous best
## Run 3 stress 9.504416e-05
## ... Procrustes: rmse 6.633353e-05  max resid 0.0002405734
## ... Similar to previous best
## Run 4 stress 8.606736e-05
## ... New best solution
## ... Procrustes: rmse 5.751668e-05  max resid 0.0002157293
## ... Similar to previous best
## Run 5 stress 9.393403e-05
## ... Procrustes: rmse 8.664957e-05  max resid 0.0002969483
## ... Similar to previous best
## Run 6 stress 8.032653e-05
## ... New best solution
## ... Procrustes: rmse 7.645736e-05  max resid 0.0002948915
## ... Similar to previous best
## Run 7 stress 9.831316e-05
## ... Procrustes: rmse 8.201595e-05  max resid 0.0002600698
## ... Similar to previous best
## Run 8 stress 9.328752e-05
## ... Procrustes: rmse 8.918931e-05  max resid 0.0003584347
## ... Similar to previous best
## Run 9 stress 9.267592e-05
## ... Procrustes: rmse 7.236628e-05  max resid 0.0002556499
## ... Similar to previous best
## Run 10 stress 9.650808e-05
## ... Procrustes: rmse 6.152325e-05  max resid 0.0002439672
## ... Similar to previous best
## Run 11 stress 9.529835e-05
## ... Procrustes: rmse 7.123609e-05  max resid 0.0003069813
## ... Similar to previous best
## Run 12 stress 9.546697e-05
## ... Procrustes: rmse 7.562737e-05  max resid 0.0002750821
## ... Similar to previous best
## Run 13 stress 9.015416e-05
## ... Procrustes: rmse 6.797954e-05  max resid 0.0002806556
## ... Similar to previous best
## Run 14 stress 7.370607e-05
## ... New best solution
## ... Procrustes: rmse 6.301358e-05  max resid 0.0002569669
## ... Similar to previous best
## Run 15 stress 8.878167e-05
## ... Procrustes: rmse 7.156551e-05  max resid 0.0002150006
## ... Similar to previous best
## Run 16 stress 9.094259e-05
## ... Procrustes: rmse 8.367138e-05  max resid 0.0002841757
## ... Similar to previous best
## Run 17 stress 9.529884e-05
## ... Procrustes: rmse 6.167584e-05  max resid 0.0002539095
## ... Similar to previous best

```

```
## Run 18 stress 9.404883e-05
## ... Procrustes: rmse 5.549699e-05  max resid 0.0002489926
## ... Similar to previous best
## Run 19 stress 9.175076e-05
## ... Procrustes: rmse 7.165683e-05  max resid 0.0003016803
## ... Similar to previous best
## Run 20 stress 9.439157e-05
## ... Procrustes: rmse 7.436347e-05  max resid 0.0002371317
## ... Similar to previous best
## *** Best solution repeated 7 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(nor_fresa_kraken_fil@otu_table@.Data)), size = 3, vjust = 1.5)
```

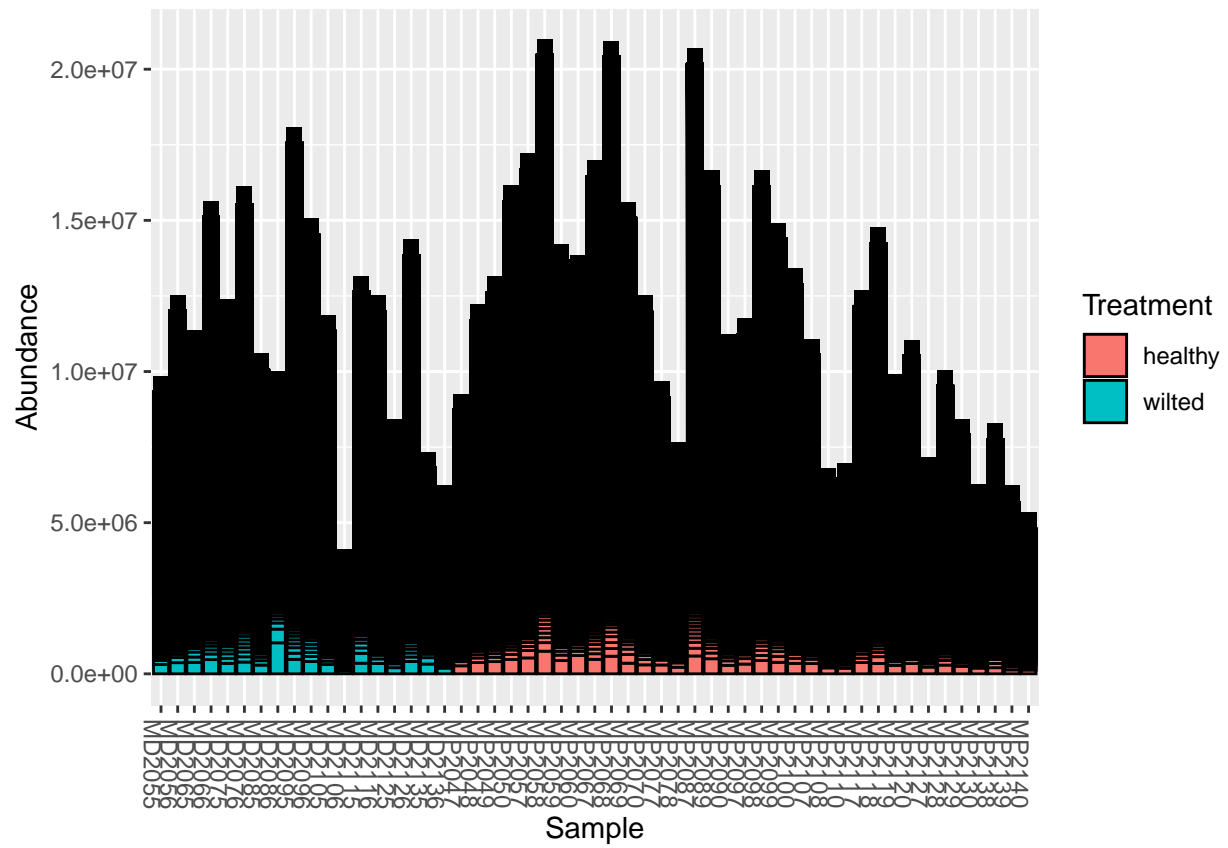


Con ninguna distancia vemos una diferenciación clara, por lo que lo veremos por los distintos niveles taxonómicos más adelante.

Grafico de barras de abundancia

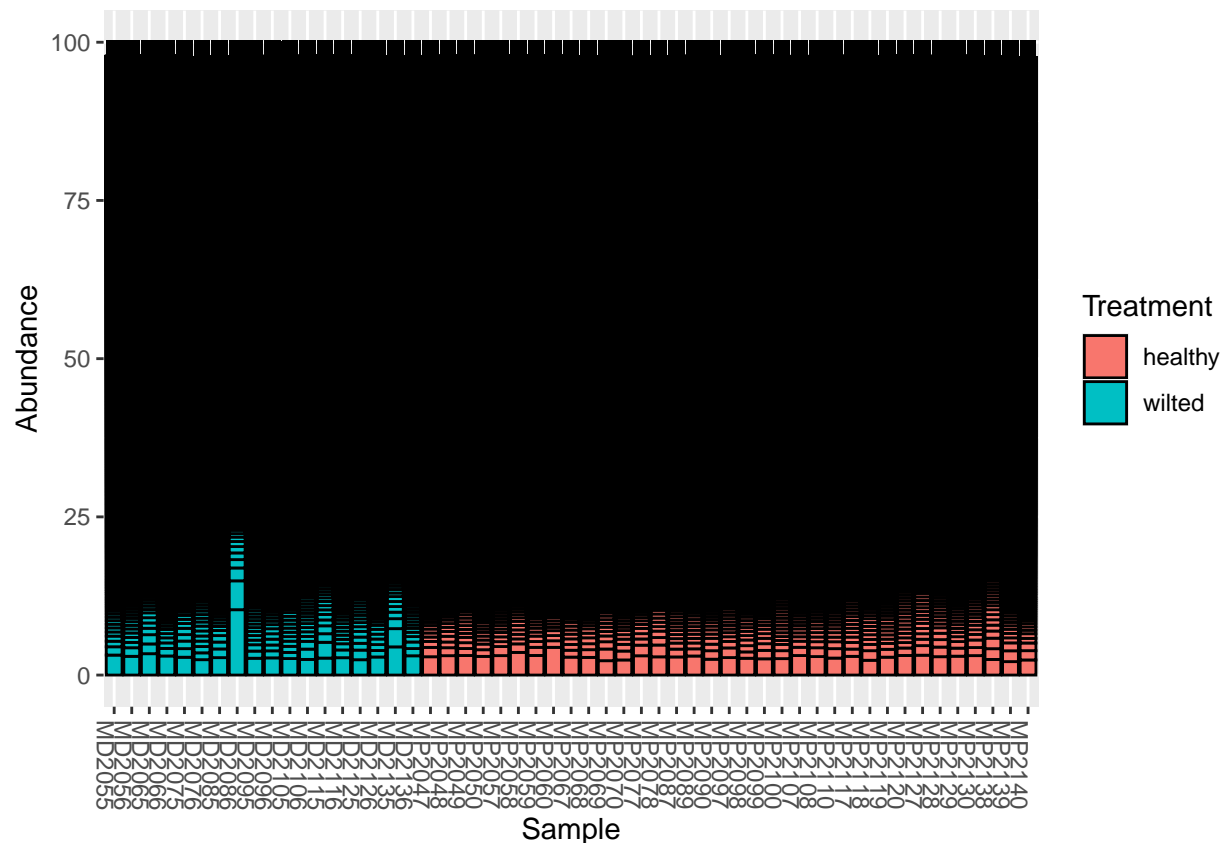
Trazando las muestras en el eje x y las abundancias en el eje y. Podemos ver las abundancias absolutas de las muestras,

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



y las abundancias relativas,

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



REDES

Para poder visualizar las redes de nuestros datos, podemos hacer un data.frame uniendo toda la informacion del objeto phyloseq.

```
df <- psmelt(nor_fresa_kraken_fil)
```

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

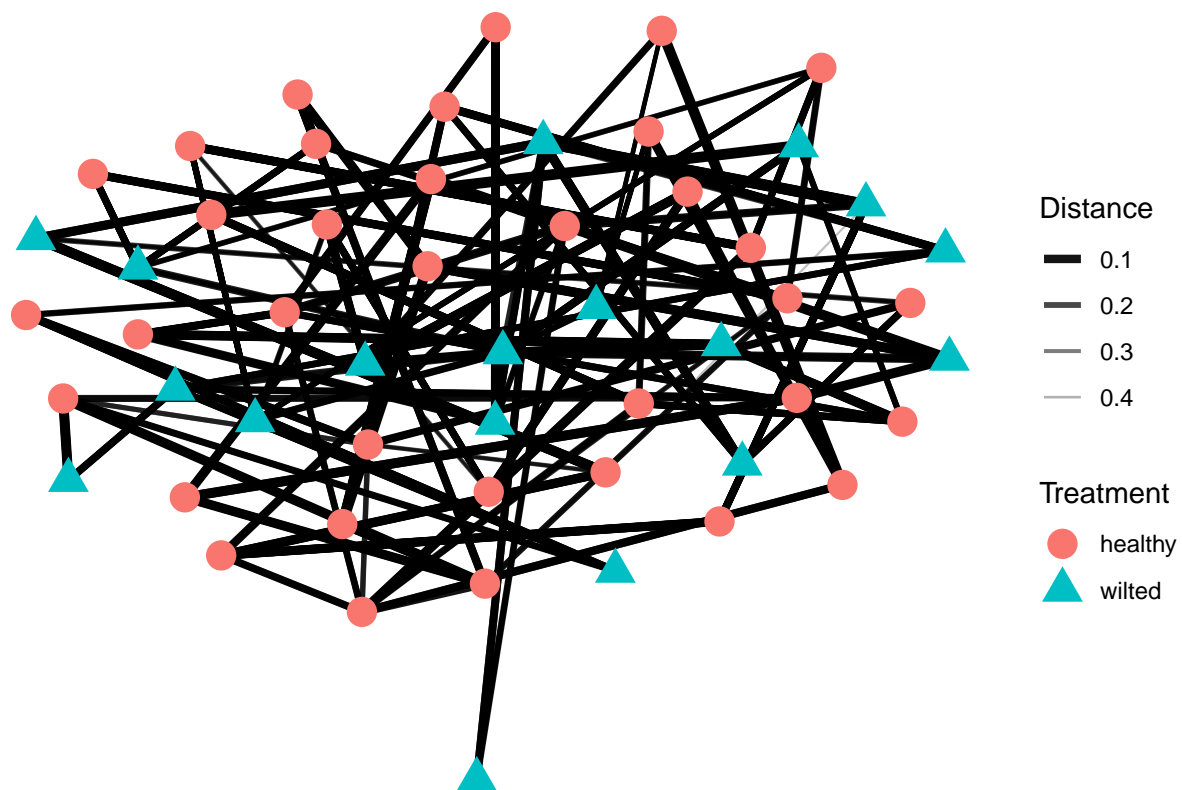
Se crea un grafo basado en “igraph”, basado en el método de distancia por defecto, Jaccard y una distancia máxima entre nodos conectados de 0,8. El “Treatment” se utiliza para los mapeos de color y forma para visualizar la estructura de las muestras.

Hacemos un grafo a partir de el objeto phyloseq

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

Y luego lo graficamos.

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

En conclusion para esta observación general de los datos, no es posible ver una separación entre muestras sanas y enfermas claramente con la diversidad beta, y con los gráficos de barras y redes, no es posible identificar datos especiales, por lo que en el próximo reporte empezaremos a crear subconjuntos de los datos, como cortar a distintos niveles taxonómicos y separar por “Kingdom”, y así ver más detalladamente el comportamiento de los datos.