

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

Camila Silva

2023-05-12

## Analisis exploratorio de datos

Obtuvimos un conjunto de datos metagenómicos de cultivo de fresa, entregado por Solena. Con el objetivo de encontrar características diferenciadoras entre frutos sanos y enfermos, y para esto se realizo un análisis exploratorio de los datos.

## Pre-procesamiento de los datos

Los datos fueron entregados en una carpeta de drive (<https://drive.google.com/drive/folders/1x0106TYUr54gfqE6uod3g5qN8lQ5x>). Para poderlos usar en el servidor, se descargaron en una maquina local y luego se pasaron por **\$ssh** al servidor.

En el equipo local se realizaron los siguientes procedimientos para la descarga de los datos comprimidos y luego fueron enviados al servidor,

```
# $ scp Downloads/kraken_results-20230203T201634Z-001.zip camila@132.248.196.39:/home/camila/GIT/Tesis_M
# 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_M
# camila@132.248.196.39's password:
# bracken_results-20230203T203724Z-001.zip        100% 3798KB  2.4MB/s  00:01
```

Ya teniendo los datos en el servidor, los descomprimimos,

```
# $ 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 bracken en el servidor con RStudio; es necesario generar el archivo **.BIOM**; para esto, se debe activar el ambiente de conda **metagenomis;kraken-biom** (Dabdou, SM (2016). kraken-biom: Habilidad de la conversión de formato interoperativo para los resultados de Kraken (Versión 1.2) [Software]. Disponible en <https://github.com/smdabdou/kraken-biom> . ) es un programa ampliamente utilizado para hacer tablas en formato **BIOM** a partir de la salida de **kraken**.

```
# $ conda activate metagenomics

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

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

```
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("kableExtra")
```

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

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

## 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=
```

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 estándares 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 “Species” tenemos 1099 sin clasificar.

## 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.

### Chao1

Es una medida de la riqueza de especies que estima la riqueza de especies en una comunidad. Es útil cuando la muestra es pequeña o la proporción de especies raras es alta. Este índice también es menos sensible a la presencia de especies raras que otros índices, lo que puede ser útil en estudios de conservación de especies raras.

$$S_{Chao1} = S_{Obs} + \frac{F_1 \ddot{O}(F_1 - 1)}{F_2 \ddot{O}(F_2 + 1)}$$

Donde  $S_{Obs}$  es el número de de especies observadas,  $F_1$  y  $F_2$  son el recuento de singletons y doubletons respectivamente. Nos muestra en recuento de singletons y doubletons respectivamente.

### Shannon

Estima la diversidad de especies considerando tanto la abundancia como la uniformidad de las especies en una comunidad. Este índice mide la incertidumbre asociada con la identificación de una especie seleccionada aleatoriamente. Cuanto mayor sea el índice de Shannon, mayor será la diversidad de especies en la comunidad.

$$D_{SH} = \frac{1}{\sum_{i_1}^S P_i^2}$$

Donde  $S$  es el número total de especies en la comunidad,  $P_i$  es la proporción de la comunidad representada por  $OTU_i$

### Simpson

Es una medida de la dominancia relativa de una o unas pocas especies en una comunidad. Este índice mide la probabilidad de que dos individuos seleccionados aleatoriamente pertenezcan a la misma especie. Cuanto menor sea el índice de Simpson, mayor será la diversidad de especies en la comunidad.

$$H = - \sum_{i_1}^S P_i \ln(P_i)$$

Donde  $S$  es el número de OTU's y  $P_i$  es la proporción de la comunidad representada por  $OTU_i$ .

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

##	Observed	Chao1	se.chao1	ACE	se.ACE	Shannon	Simpson
## MD2055	8574	8720.286	24.47547	8677.413	35.19357	7.466004	0.9977741
## MD2056	8654	8789.198	24.26666	8745.673	32.87577	7.485615	0.9978916
## MD2065	8550	8744.387	30.77128	8664.496	35.54419	7.318039	0.9972436
## MD2066	8706	8784.103	16.80939	8761.941	31.20163	7.506399	0.9979412
## MD2075	8687	8811.039	24.32975	8755.774	31.40644	7.437571	0.9977697
## MD2076	8755	8863.075	24.67599	8807.968	26.71026	7.532560	0.9980916
## MD2085	8636	8738.300	19.71192	8708.837	32.99597	7.498782	0.9979386
## MD2086	8639	8742.269	20.23632	8707.247	32.67625	6.818533	0.9856285
## MD2095	8733	8810.008	17.04429	8788.485	29.47769	7.518061	0.9980806
## MD2096	8702	8797.104	19.41002	8766.701	30.82408	7.496519	0.9980173
## MD2105	8629	8740.522	21.24880	8702.952	33.26369	7.498767	0.9980946
## MD2106	8295	8563.163	34.32997	8506.578	39.30231	7.491047	0.9979434
## MD2115	8657	8776.901	22.09429	8741.454	32.50139	7.350835	0.9972130
## MD2116	8661	8782.066	21.90662	8753.027	32.93621	7.499214	0.9980384
## MD2125	8562	8759.415	30.58607	8688.434	35.50199	7.417748	0.9978946
## MD2126	8680	8808.387	23.77747	8766.526	31.31394	7.476194	0.9979537
## MD2135	8511	8678.320	25.89195	8637.671	36.73668	7.245656	0.9958113
## MD2136	8415	8620.494	29.50701	8567.984	37.75486	7.452826	0.9978117
## MP2047	8602	8728.415	23.36468	8678.009	33.90179	7.517547	0.9979409
## MP2048	8643	8776.358	23.63204	8736.855	33.06283	7.493255	0.9978487
## MP2049	8658	8788.592	24.00498	8741.035	32.62028	7.463841	0.9978268
## MP2050	8709	8794.363	18.56534	8764.037	29.40382	7.520308	0.9980202
## MP2057	8707	8804.417	19.75270	8773.933	31.24870	7.440127	0.9977884
## MP2058	8733	8797.648	14.75489	8784.840	29.73999	7.371476	0.9974102
## MP2059	8657	8783.556	23.67422	8736.193	32.38400	7.472789	0.9978316
## MP2060	8647	8775.282	23.47971	8731.028	32.55745	7.379259	0.9969449
## MP2067	8734	8843.698	23.21274	8796.198	29.05599	7.508925	0.9979867
## MP2068	8747	8815.763	16.77715	8788.732	27.95643	7.516656	0.9980133
## MP2069	8726	8822.571	19.53542	8795.846	30.55664	7.508116	0.9980023
## MP2070	8728	8817.446	19.00159	8787.508	30.00127	7.587001	0.9982081
## MP2077	8628	8753.400	22.70916	8715.247	33.40400	7.475328	0.9977180
## MP2078	8598	8728.862	22.64135	8688.905	35.31764	7.482809	0.9976945
## MP2087	8768	8842.528	16.79212	8822.699	27.93595	7.504340	0.9979687
## MP2089	8714	8808.127	19.32620	8780.325	30.50478	7.490445	0.9979141
## MP2090	8671	8767.705	18.78399	8743.911	32.88782	7.499854	0.9980771
## MP2097	8656	8754.390	19.83362	8724.324	31.18153	7.515360	0.9979635
## MP2098	8749	8841.667	20.51302	8802.235	28.52034	7.522290	0.9980997
## MP2099	8683	8782.746	20.43521	8748.062	30.73221	7.532568	0.9980878
## MP2100	8656	8805.523	26.80371	8747.377	31.84662	7.505509	0.9980692
## MP2107	8624	8762.179	24.10034	8721.837	33.41679	7.483650	0.9978492
## MP2108	8517	8691.914	27.11206	8637.478	37.03501	7.489635	0.9978936
## MP2110	8511	8690.379	27.64739	8636.503	36.78420	7.464569	0.9979307
## MP2117	8661	8805.667	25.70258	8753.372	32.92052	7.457922	0.9978075
## MP2118	8695	8808.885	21.73577	8771.157	32.46190	7.442191	0.9980275
## MP2119	8618	8783.355	27.98089	8718.685	34.18521	7.491656	0.9979607
## MP2120	8584	8697.632	20.25075	8671.818	34.85759	7.400446	0.9976861
## MP2127	8551	8752.562	30.39450	8687.282	36.71417	7.399520	0.9975185
## MP2128	8519	8730.062	31.45584	8661.570	35.99706	7.359011	0.9976301
## MP2129	8559	8715.123	25.30752	8670.249	36.21551	7.460731	0.9978309

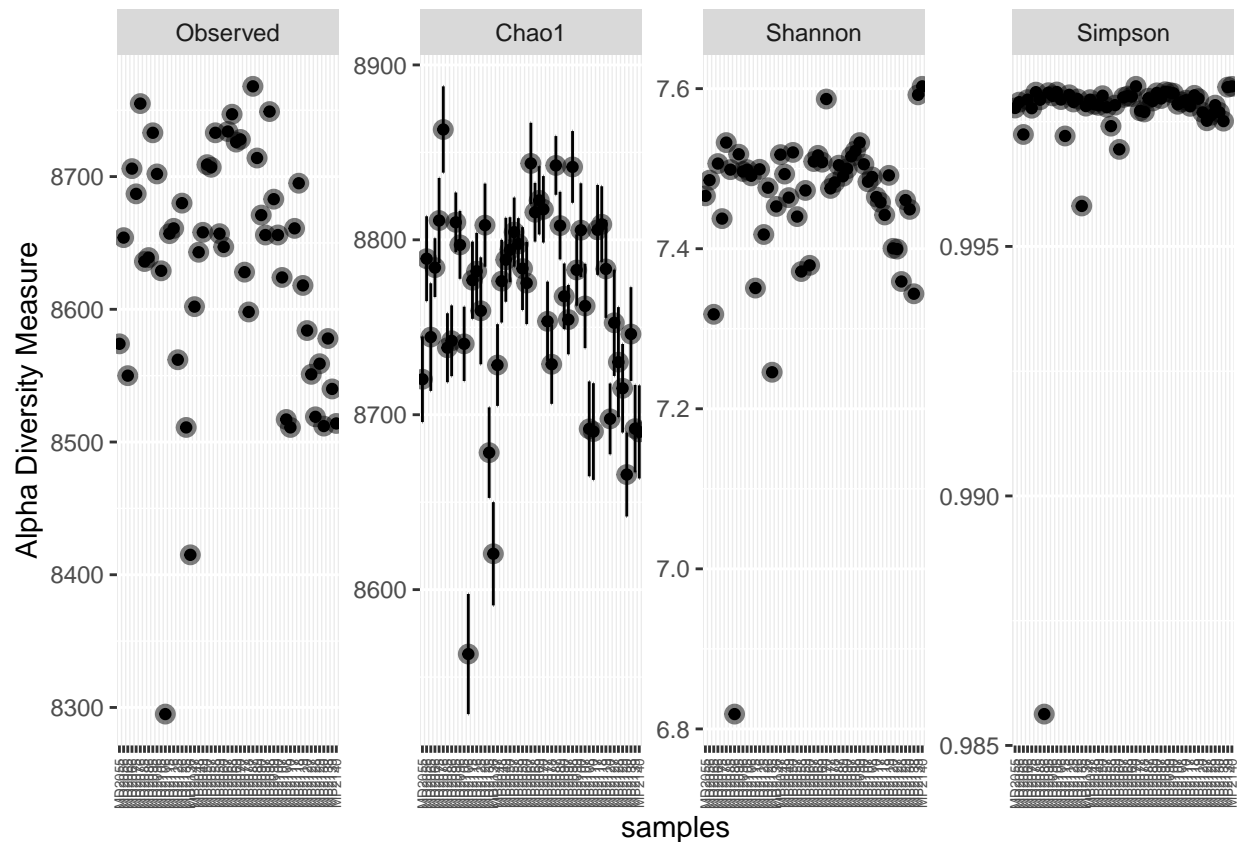


##	MP2130	8512	8665.792	23.97226	8638.625	37.01356	7.449130	0.9976868
##	MP2138	8578	8746.125	26.80847	8691.902	36.10410	7.343699	0.9975108
##	MP2139	8540	8692.042	24.98216	8648.647	35.38417	7.592256	0.9981921
##	MP2140	8514	8690.167	26.61601	8649.404	37.35319	7.602862	0.9982074
##		InvSimpson	Fisher					
##	MD2055	449.25645	925.3121					
##	MD2056	474.29141	908.3504					
##	MD2065	362.78564	906.6388					
##	MD2066	485.71887	891.1799					
##	MD2075	448.37768	913.5861					
##	MD2076	524.00665	893.6326					
##	MD2085	485.10995	924.6302					
##	MD2086	69.58234	931.4763					
##	MD2095	520.98455	879.7342					
##	MD2096	504.35179	894.6127					
##	MD2105	524.82120	911.3775					
##	MD2106	486.22931	998.2853					
##	MD2115	358.81348	903.4491					
##	MD2116	509.78448	909.3159					
##	MD2125	474.96787	941.8360					
##	MD2126	488.68678	896.9392					
##	MD2135	238.73962	951.7382					
##	MD2136	456.97519	959.6058					
##	MP2047	485.63768	935.6653					
##	MP2048	464.82772	909.8482					
##	MP2049	460.15017	903.7808					
##	MP2050	505.09150	888.2114					
##	MP2057	452.15467	881.7033					
##	MP2058	386.13530	865.2148					
##	MP2059	461.17702	895.5759					
##	MP2060	327.32656	896.9680					
##	MP2067	496.69018	886.0196					
##	MP2068	503.34236	866.9814					
##	MP2069	500.57854	893.7455					
##	MP2070	558.06009	917.0866					
##	MP2077	438.21032	933.7660					
##	MP2078	433.74971	957.6983					
##	MP2087	492.29358	870.4068					
##	MP2089	479.42037	885.8020					
##	MP2090	520.04642	922.1831					
##	MP2097	491.03452	915.2878					
##	MP2098	526.24456	889.6815					
##	MP2099	522.94968	893.4939					
##	MP2100	517.92792	901.4283					
##	MP2107	464.93743	918.1892					
##	MP2108	474.75249	962.0625					
##	MP2110	483.24587	958.1978					
##	MP2117	456.09252	907.8520					
##	MP2118	506.98272	895.7930					
##	MP2119	490.36971	930.0299					
##	MP2120	432.16173	913.8080					
##	MP2127	402.97931	959.7039					
##	MP2128	421.96596	916.5040					
##	MP2129	461.02186	941.5704					

```
## MP2130 432.30837 971.6079
## MP2138 401.74180 945.7243
## MP2139 553.13542 975.9934
## MP2140 557.85022 992.6713
```

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.

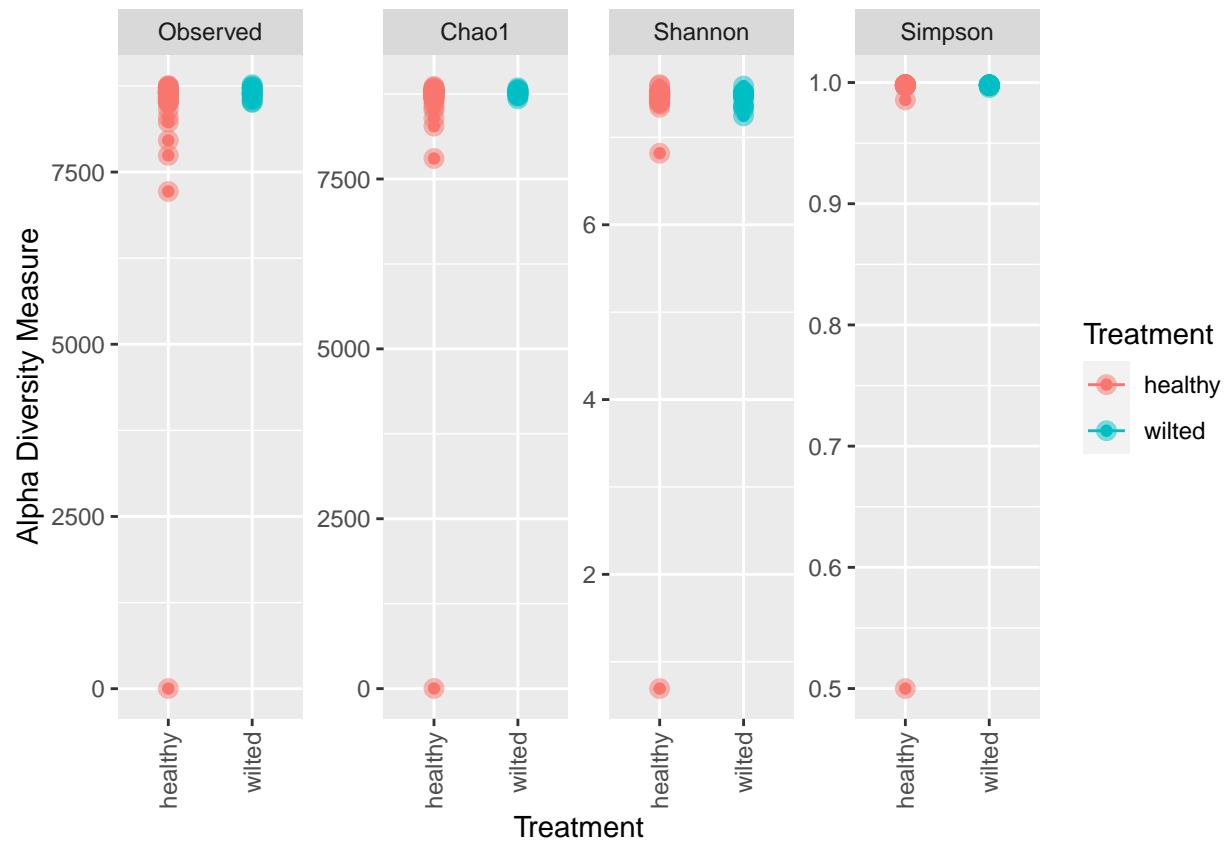
```
p<-plot_richness(physeq = fresa_kraken_fil, measures = c("Observed","Chao1","Shannon","simpson"))
p + geom_point(size=3, alpha=0.5) +
  theme(axis.text.x = element_text(angle=90, size=5, hjust=1, vjust=0.5))
```



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

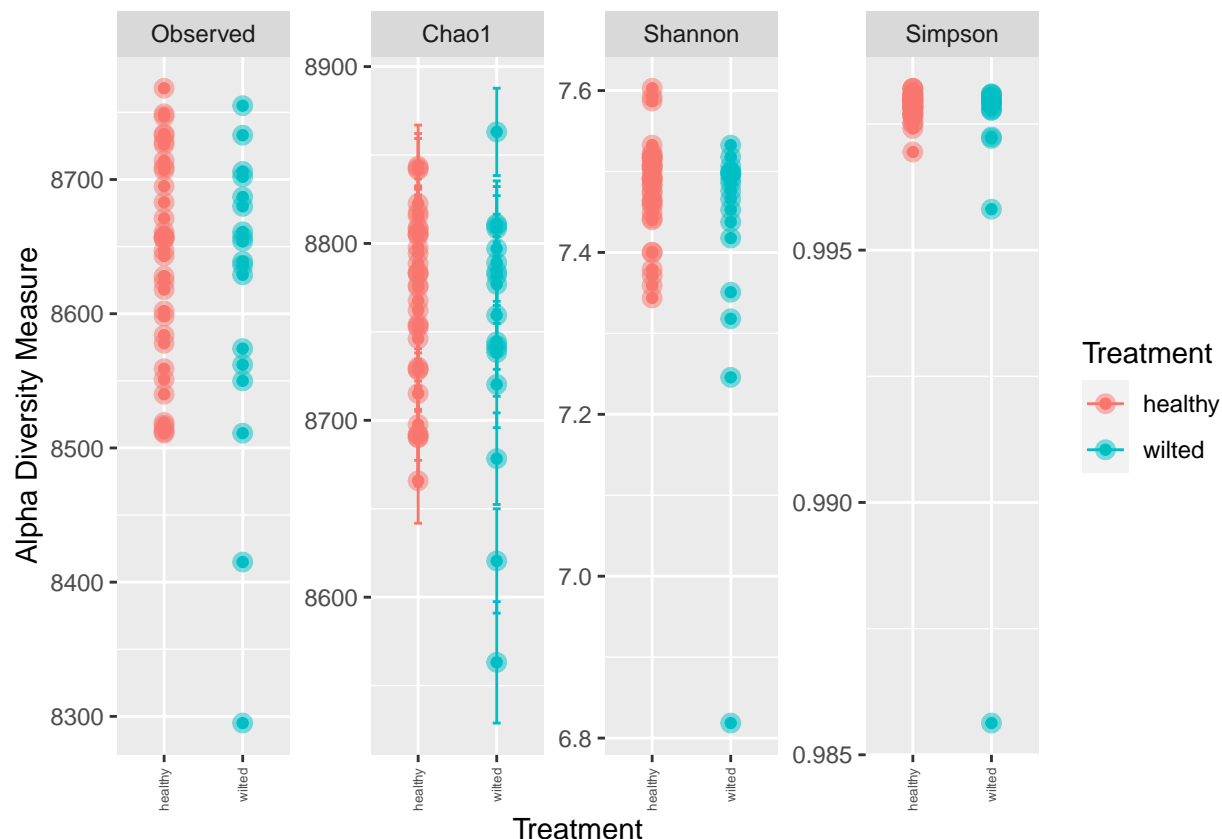
Ya que queremos diferenciar entre plantas sanas y enfermas, tomamos estos dos conjuntos por separado para mostrar la diversidad,

```
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 = 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.

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 función 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(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 posible una buena comparación de abundancias dadas por porcentajes con índices 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"
##
## $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.

## Euclidean

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 euclidean 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 9.398085e-05
## ... Procrustes: rmse 6.842023e-05 max resid 0.0002689006
## ... Similar to previous best
## Run 2 stress 9.44672e-05
## ... Procrustes: rmse 8.090121e-05 max resid 0.0003436899
## ... Similar to previous best
## Run 3 stress 9.216207e-05
```

```

## ... Procrustes: rmse 7.112519e-05  max resid 0.0003044466
## ... Similar to previous best
## Run 4 stress 9.726638e-05
## ... Procrustes: rmse 7.34743e-05  max resid 0.0003265482
## ... Similar to previous best
## Run 5 stress 9.364209e-05
## ... Procrustes: rmse 7.596066e-05  max resid 0.0002543088
## ... Similar to previous best
## Run 6 stress 8.39259e-05
## ... New best solution
## ... Procrustes: rmse 6.757807e-05  max resid 0.0002249863
## ... Similar to previous best
## Run 7 stress 9.731806e-05
## ... Procrustes: rmse 7.4671e-05  max resid 0.0002447202
## ... Similar to previous best
## Run 8 stress 6.210341e-05
## ... New best solution
## ... Procrustes: rmse 5.308999e-05  max resid 0.0001624942
## ... Similar to previous best
## Run 9 stress 9.70134e-05
## ... Procrustes: rmse 5.584742e-05  max resid 0.0002117803
## ... Similar to previous best
## Run 10 stress 6.557461e-05
## ... Procrustes: rmse 4.602941e-05  max resid 0.0001521527
## ... Similar to previous best
## Run 11 stress 9.617187e-05
## ... Procrustes: rmse 6.15211e-05  max resid 0.0002106753
## ... Similar to previous best
## Run 12 stress 7.914113e-05
## ... Procrustes: rmse 4.582252e-05  max resid 0.0001433328
## ... Similar to previous best
## Run 13 stress 9.966979e-05
## ... Procrustes: rmse 7.793462e-05  max resid 0.0002980683
## ... Similar to previous best
## Run 14 stress 9.845816e-05
## ... Procrustes: rmse 7.880046e-05  max resid 0.0002656432
## ... Similar to previous best
## Run 15 stress 9.883471e-05
## ... Procrustes: rmse 6.988643e-05  max resid 0.0002372233
## ... Similar to previous best
## Run 16 stress 8.656538e-05
## ... Procrustes: rmse 6.791441e-05  max resid 0.0002806575
## ... Similar to previous best
## Run 17 stress 9.317352e-05
## ... Procrustes: rmse 7.785507e-05  max resid 0.0002838914
## ... Similar to previous best
## Run 18 stress 9.954234e-05
## ... Procrustes: rmse 7.758895e-05  max resid 0.0002727762
## ... Similar to previous best
## Run 19 stress 9.086608e-05
## ... Procrustes: rmse 4.784188e-05  max resid 0.0001866935
## ... Similar to previous best
## Run 20 stress 9.909891e-05
## ... Procrustes: rmse 5.840472e-05  max resid 0.0002078172

```

```
## ... Similar to previous best
## *** Best solution repeated 13 times

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

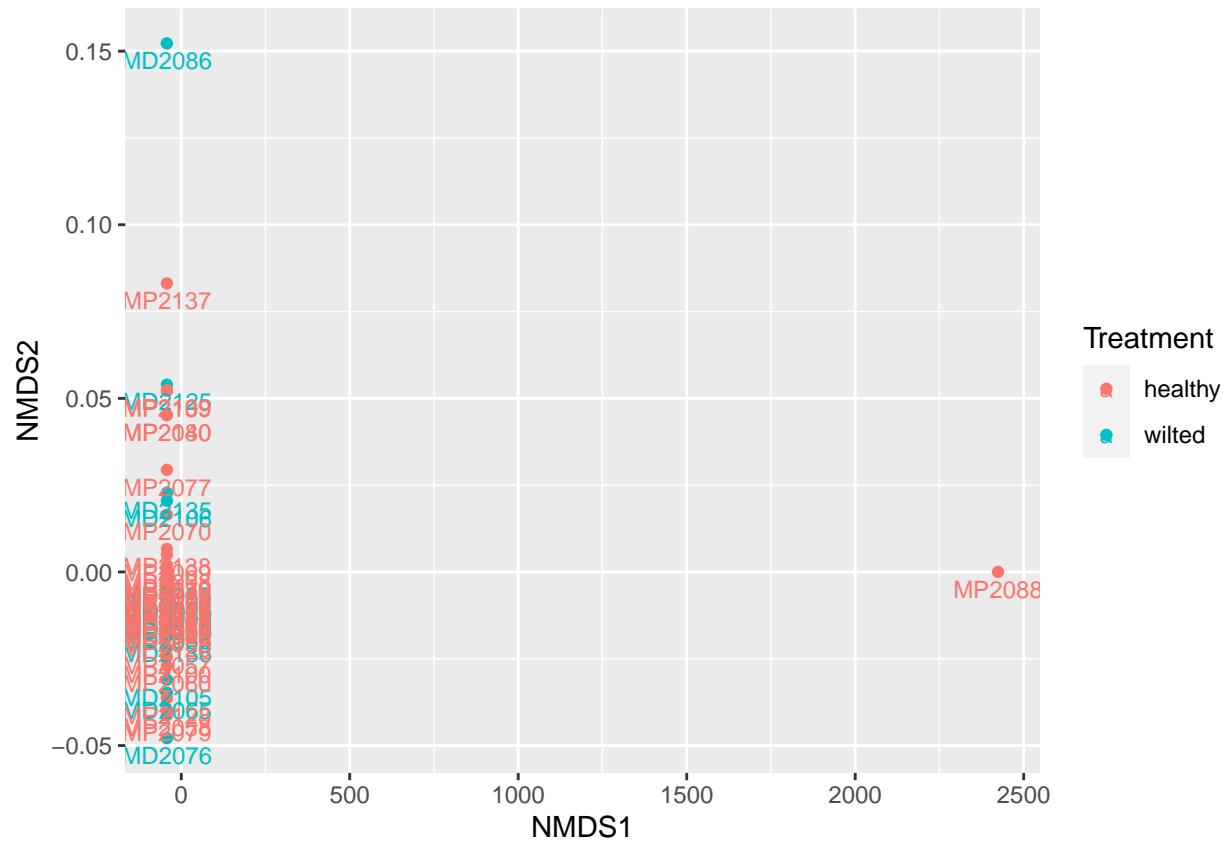
Como para los datos ya filtrados por calidad

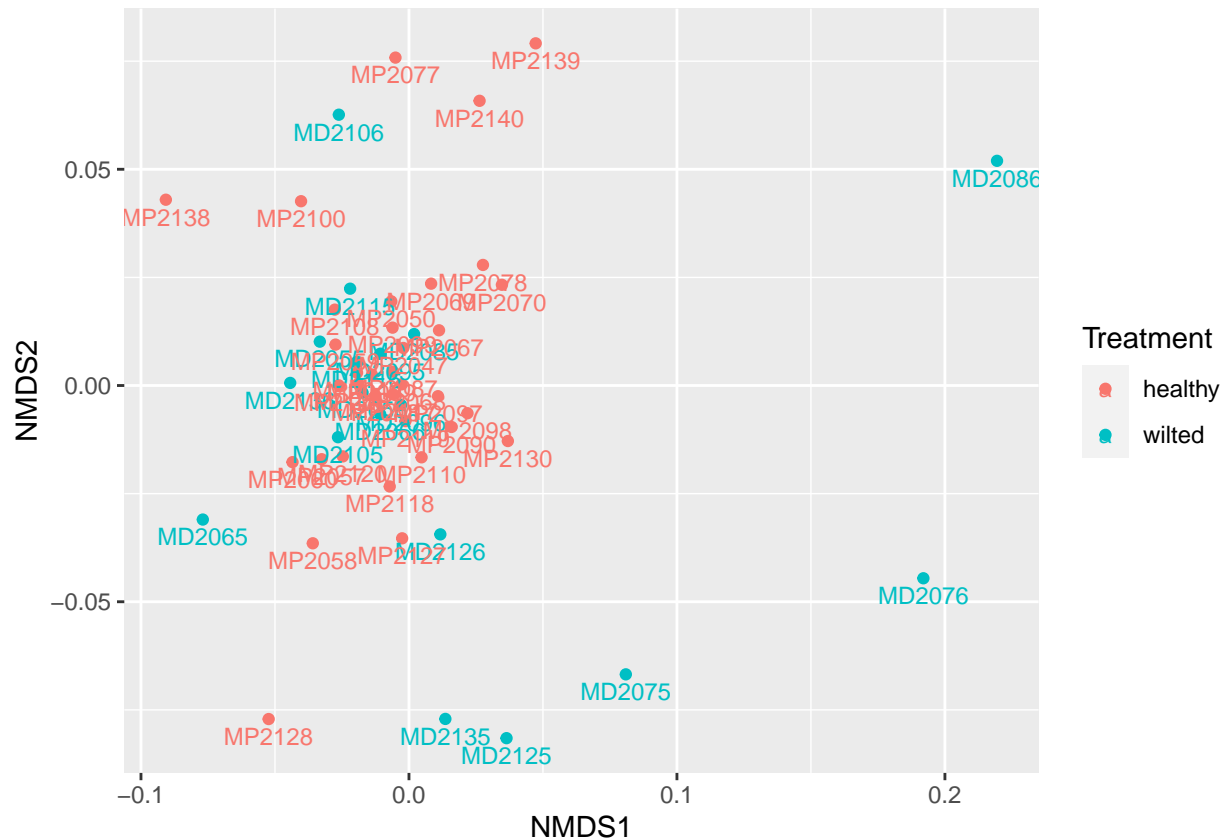
```
meta_ord_fil <- ordinate(physeq = percentages_fil, method = "NMDS", distance = "bray")
```

```
## Wisconsin double standardization
## Run 0 stress 0.1686115
## Run 1 stress 0.1666037
## ... New best solution
## ... Procrustes: rmse 0.06085191 max resid 0.3418445
## Run 2 stress 0.1721013
## Run 3 stress 0.1867868
## Run 4 stress 0.1676195
## Run 5 stress 0.1634614
## ... New best solution
## ... Procrustes: rmse 0.0608054 max resid 0.3422877
## Run 6 stress 0.1676982
## Run 7 stress 0.1737623
## Run 8 stress 0.1666386
## Run 9 stress 0.1665993
## Run 10 stress 0.1817674
## Run 11 stress 0.1649716
## Run 12 stress 0.1663025
## Run 13 stress 0.4035789
## Run 14 stress 0.1867866
## Run 15 stress 0.1644885
## Run 16 stress 0.1726624
## Run 17 stress 0.165276
## Run 18 stress 0.1738573
## Run 19 stress 0.1665675
## Run 20 stress 0.1686555
## *** 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)
```





En este gráfico NMDS, cada punto representa la abundancia combinada de todos sus OTU's. No se puede ver una diferencia entre los conjuntos de muestras de muestras 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 la distancia Euclideana

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

## Wisconsin double standardization
## Run 0 stress 0.1866907
## Run 1 stress 0.1846076
## ... New best solution
## ... Procrustes: rmse 0.06112944   max resid 0.2266818
## Run 2 stress 0.1825491
## ... New best solution
## ... Procrustes: rmse 0.07905421   max resid 0.4705931
## Run 3 stress 0.1924179
## Run 4 stress 0.1822099
## ... New best solution
## ... Procrustes: rmse 0.0478424   max resid 0.2076126
## Run 5 stress 0.180767
## ... New best solution
## ... Procrustes: rmse 0.04660276   max resid 0.2010978
```

```

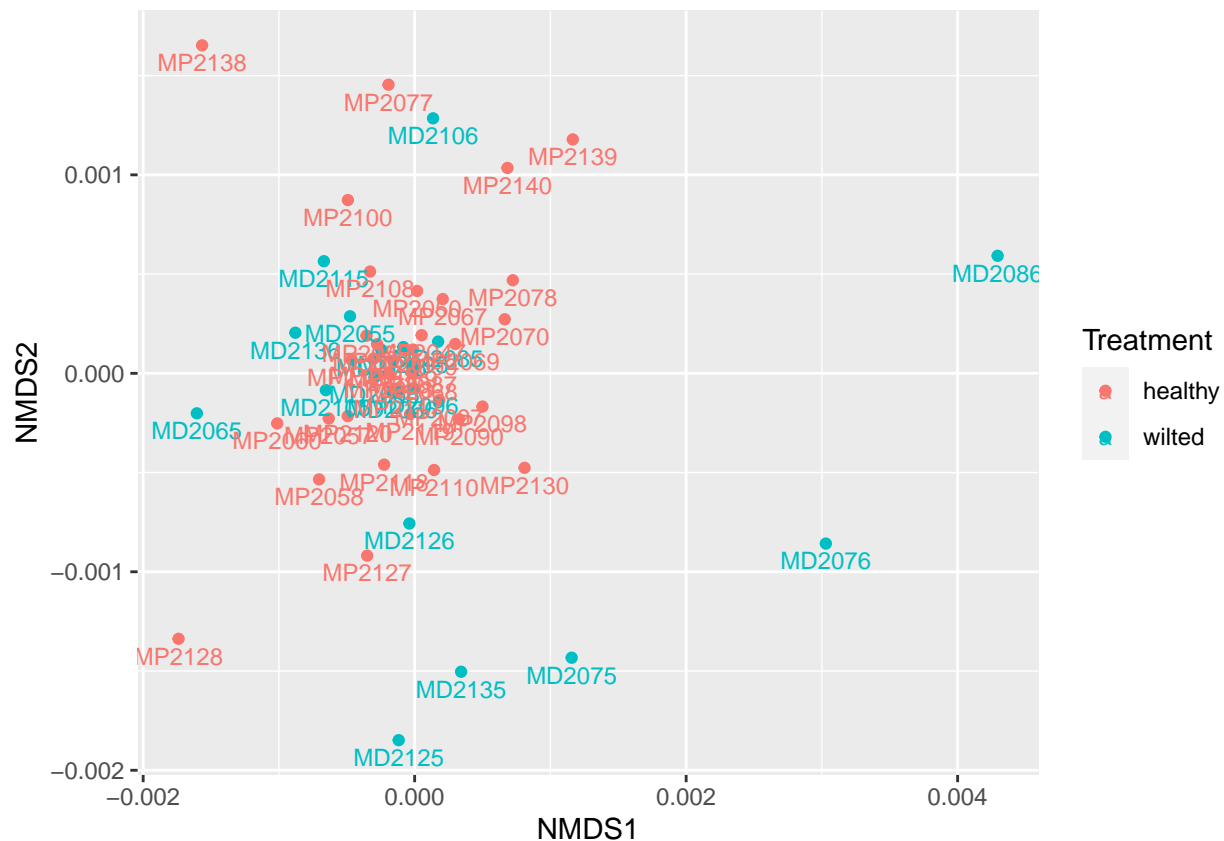
## Run 6 stress 0.1992504
## Run 7 stress 0.1787637
## ... New best solution
## ... Procrustes: rmse 0.06415087  max resid 0.3249119
## Run 8 stress 0.178079
## ... New best solution
## ... Procrustes: rmse 0.02953879  max resid 0.1607046
## Run 9 stress 0.1768281
## ... New best solution
## ... Procrustes: rmse 0.03633343  max resid 0.2123835
## Run 10 stress 0.1937849
## Run 11 stress 0.1768313
## ... Procrustes: rmse 0.03150578  max resid 0.1544945
## Run 12 stress 0.1823531
## Run 13 stress 0.1764459
## ... New best solution
## ... Procrustes: rmse 0.04024569  max resid 0.1615784
## Run 14 stress 0.1834694
## Run 15 stress 0.2015631
## Run 16 stress 0.1949269
## Run 17 stress 0.1889664
## Run 18 stress 0.1768329
## ... Procrustes: rmse 0.03647047  max resid 0.1506305
## Run 19 stress 0.1846313
## Run 20 stress 0.1839135
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      4: no. of iterations >= maxit
##     16: 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)

```



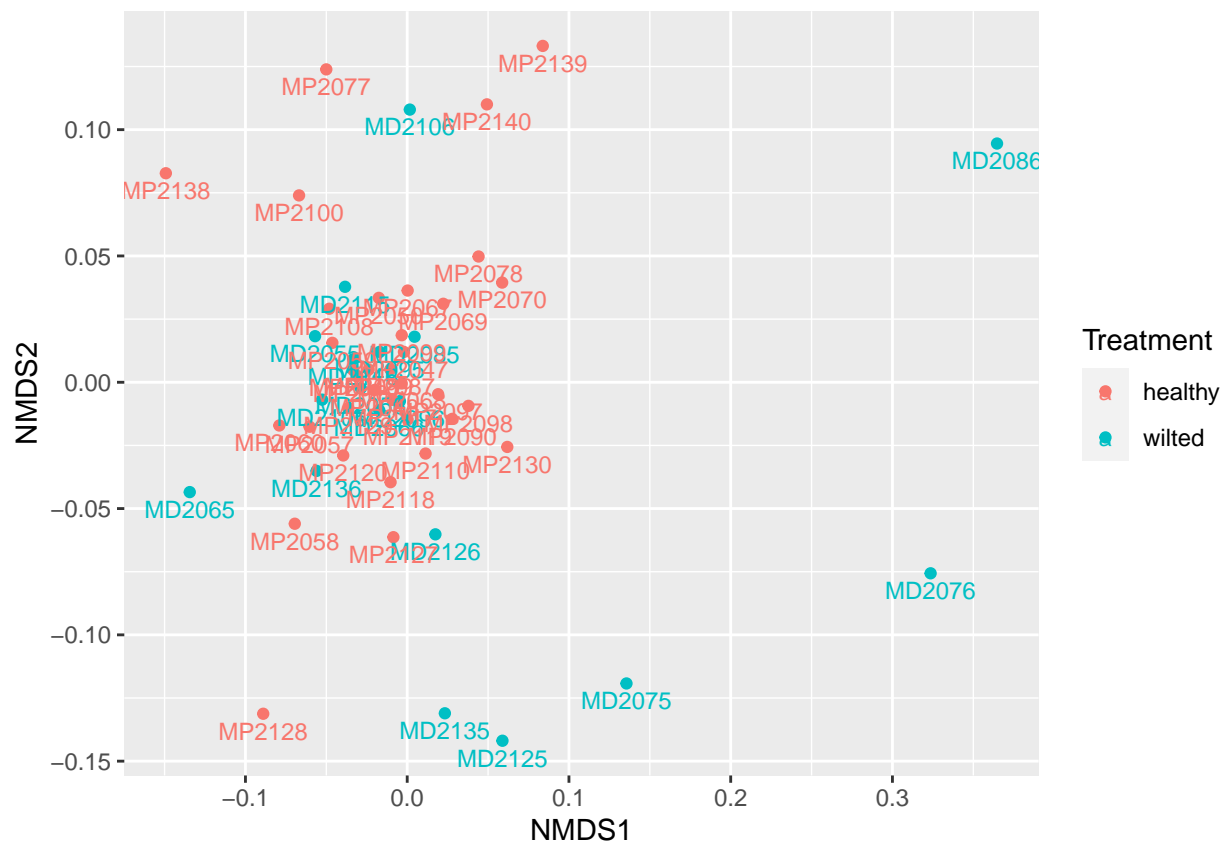
Usando la distancia de Jaccard

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

## Wisconsin double standardization
## Run 0 stress 0.168611
## Run 1 stress 0.1639279
## ... New best solution
## ... Procrustes: rmse 0.07213925 max resid 0.3627468
## Run 2 stress 0.1665993
## Run 3 stress 0.1649562
## Run 4 stress 0.1674751
## Run 5 stress 0.1707504
## Run 6 stress 0.1851409
## Run 7 stress 0.1666101
## Run 8 stress 0.1668607
## Run 9 stress 0.1652145
## Run 10 stress 0.167173
## Run 11 stress 0.1665361
## Run 12 stress 0.164512
## Run 13 stress 0.1665306
## Run 14 stress 0.1663024
## Run 15 stress 0.1681196
## Run 16 stress 0.1660361
```

```
## Run 17 stress 0.1636955
## ... New best solution
## ... Procrustes: rmse 0.03136718 max resid 0.1595674
## Run 18 stress 0.1724444
## Run 19 stress 0.1718818
## Run 20 stress 0.1645113
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      20: 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 la distancia Manhattan

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

```
## Wisconsin double standardization
## Run 0 stress 0.1686118
## Run 1 stress 0.1665887
## ... New best solution
## ... Procrustes: rmse 0.0629001 max resid 0.3447616
## Run 2 stress 0.1650632
## ... New best solution
```

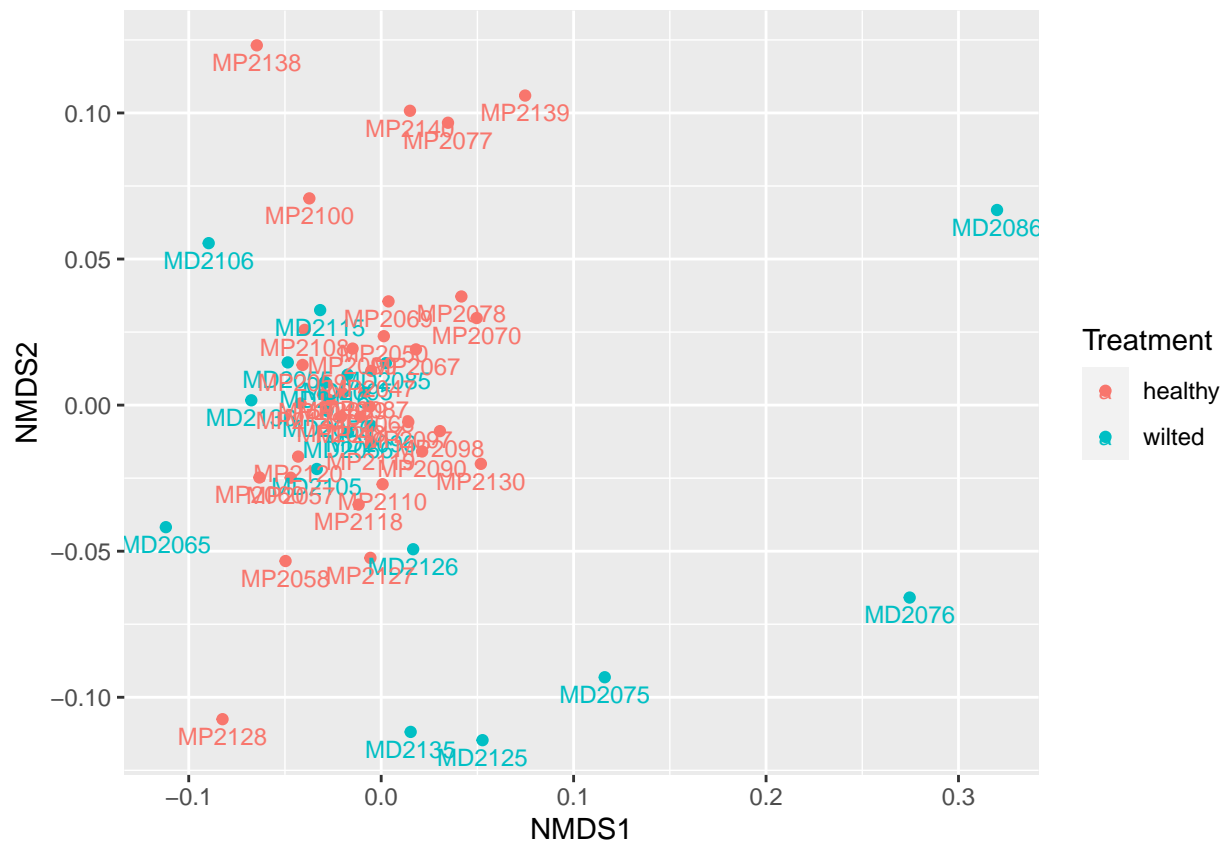


```

## ... Procrustes: rmse 0.04456959  max resid 0.2657224
## Run 3 stress 0.1737969
## Run 4 stress 0.1655724
## Run 5 stress 0.1917244
## Run 6 stress 0.1837192
## Run 7 stress 0.1674751
## Run 8 stress 0.1750513
## Run 9 stress 0.1665988
## Run 10 stress 0.1666803
## Run 11 stress 0.1681192
## Run 12 stress 0.1651726
## ... Procrustes: rmse 0.05382359  max resid 0.2213658
## Run 13 stress 0.1666076
## Run 14 stress 0.167225
## Run 15 stress 0.1652804
## ... Procrustes: rmse 0.04682408  max resid 0.2245007
## Run 16 stress 0.2013121
## Run 17 stress 0.1681192
## Run 18 stress 0.1725474
## Run 19 stress 0.1667639
## Run 20 stress 0.1725791
## *** 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)

```



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

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

```
## Run 0 stress 9.277469e-05
## Run 1 stress 9.466428e-05
## ... Procrustes: rmse 6.544977e-05 max resid 0.0002291464
## ... Similar to previous best
## Run 2 stress 8.343701e-05
## ... New best solution
## ... Procrustes: rmse 4.720369e-05 max resid 0.0001461449
## ... Similar to previous best
## Run 3 stress 8.364971e-05
## ... Procrustes: rmse 7.042457e-05 max resid 0.0002804699
## ... Similar to previous best
## Run 4 stress 7.179723e-05
## ... New best solution
## ... Procrustes: rmse 8.173691e-05 max resid 0.0002976145
## ... Similar to previous best
## Run 5 stress 6.284265e-05
## ... New best solution
## ... Procrustes: rmse 4.948852e-05 max resid 0.000202213
## ... Similar to previous best
## Run 6 stress 9.385458e-05
```

```

## ... Procrustes: rmse 5.875868e-05  max resid 0.0002374417
## ... Similar to previous best
## Run 7 stress 9.844749e-05
## ... Procrustes: rmse 6.000901e-05  max resid 0.000201355
## ... Similar to previous best
## Run 8 stress 8.726018e-05
## ... Procrustes: rmse 3.42797e-05  max resid 0.0001431635
## ... Similar to previous best
## Run 9 stress 9.190641e-05
## ... Procrustes: rmse 7.054795e-05  max resid 0.0002423212
## ... Similar to previous best
## Run 10 stress 8.897234e-05
## ... Procrustes: rmse 4.536218e-05  max resid 0.0001885403
## ... Similar to previous best
## Run 11 stress 9.369327e-05
## ... Procrustes: rmse 6.371548e-05  max resid 0.0002070419
## ... Similar to previous best
## Run 12 stress 9.82439e-05
## ... Procrustes: rmse 7.019971e-05  max resid 0.0002470617
## ... Similar to previous best
## Run 13 stress 9.476954e-05
## ... Procrustes: rmse 6.290208e-05  max resid 0.000247852
## ... Similar to previous best
## Run 14 stress 9.659007e-05
## ... Procrustes: rmse 5.78511e-05  max resid 0.0001648571
## ... Similar to previous best
## Run 15 stress 8.878668e-05
## ... Procrustes: rmse 3.111853e-05  max resid 9.783399e-05
## ... Similar to previous best
## Run 16 stress 7.547831e-05
## ... Procrustes: rmse 3.170447e-05  max resid 0.000157037
## ... Similar to previous best
## Run 17 stress 9.177352e-05
## ... Procrustes: rmse 6.800932e-05  max resid 0.0002003719
## ... Similar to previous best
## Run 18 stress 9.001978e-05
## ... Procrustes: rmse 5.798646e-05  max resid 0.0002335133
## ... Similar to previous best
## Run 19 stress 8.987808e-05
## ... Procrustes: rmse 6.590092e-05  max resid 0.0002171766
## ... Similar to previous best
## Run 20 stress 8.836339e-05
## ... Procrustes: rmse 6.08863e-05  max resid 0.0002152048
## ... Similar to previous best
## *** Best solution repeated 16 times

```

```

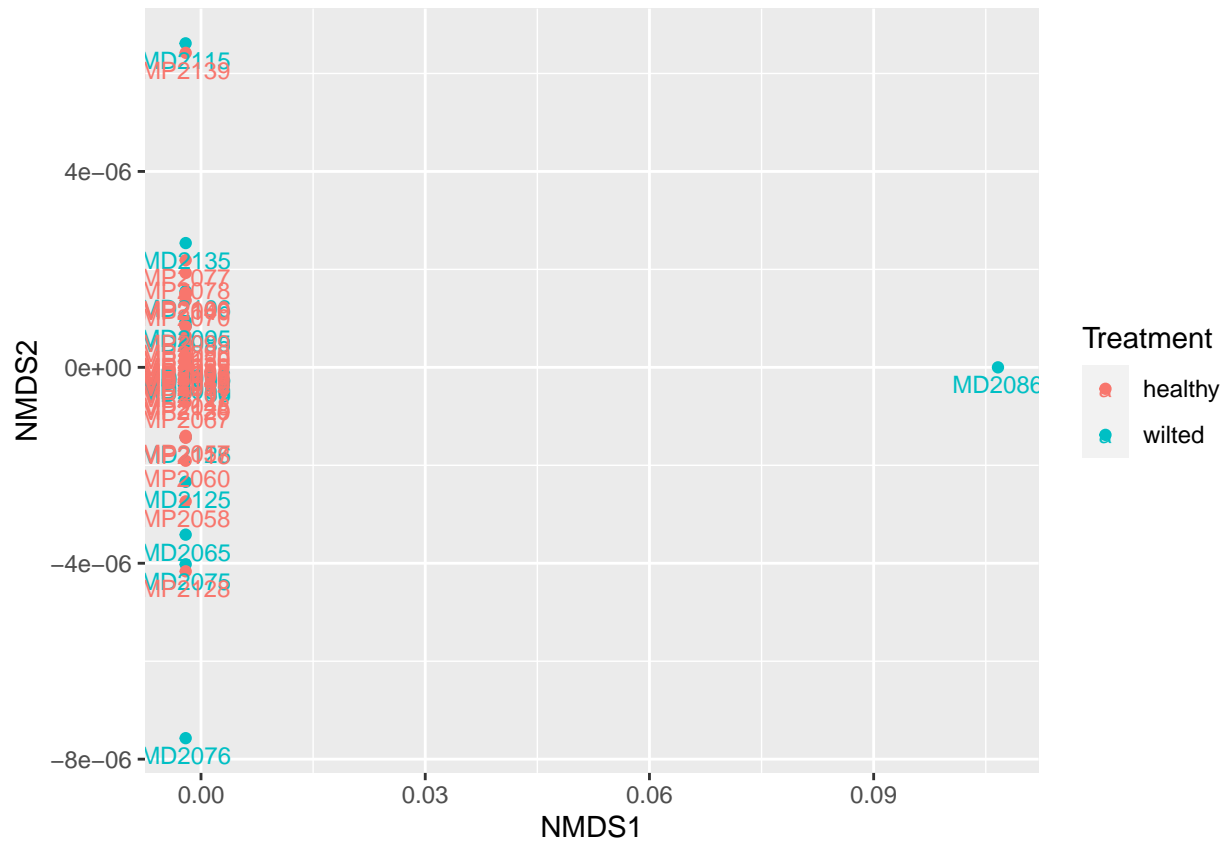
## Warning in metaMDS(ps.dist): stress is (nearly) zero: you may have insufficient
## data

```

```

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)

```

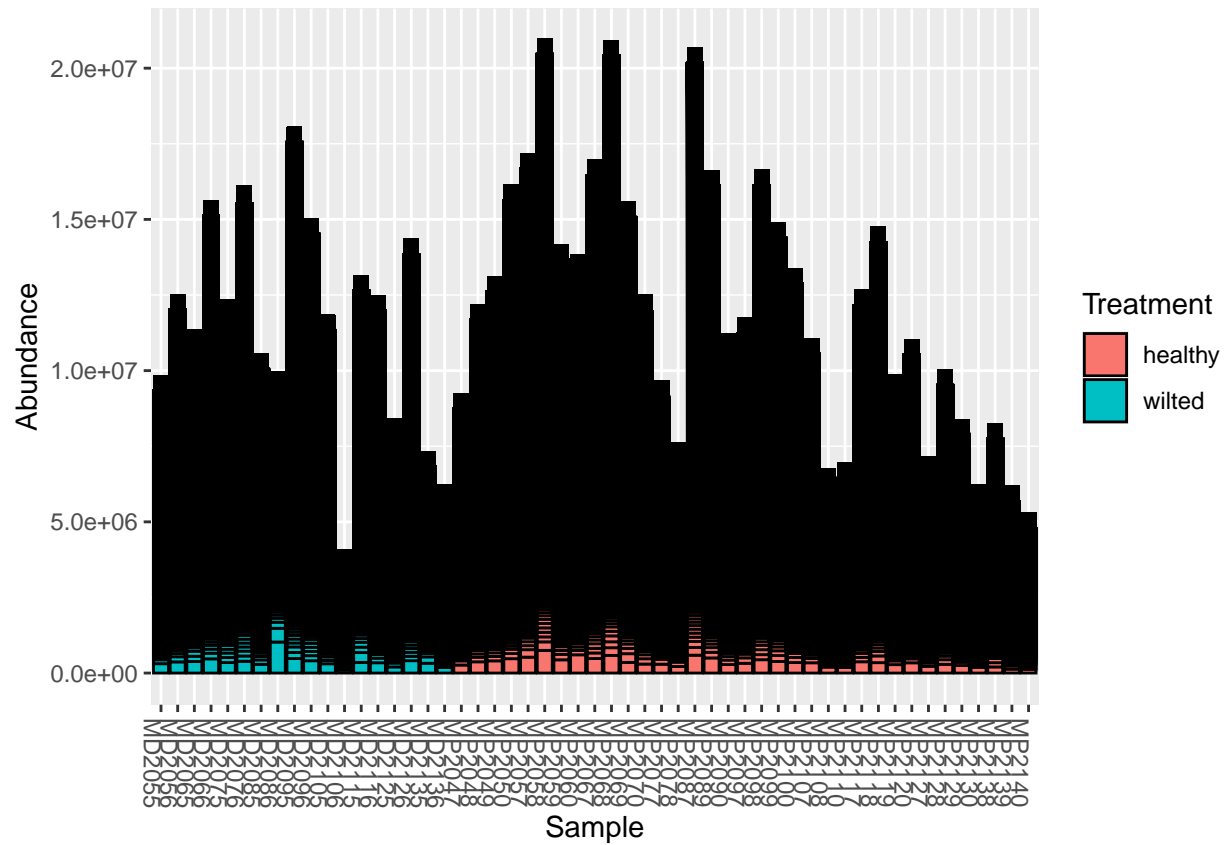


Con ninguna distancia vemos una diferenciación clara, por lo que lo veremos por los distintos niveles taxonómicos más adelante.

## Grafico de barras de abundancia

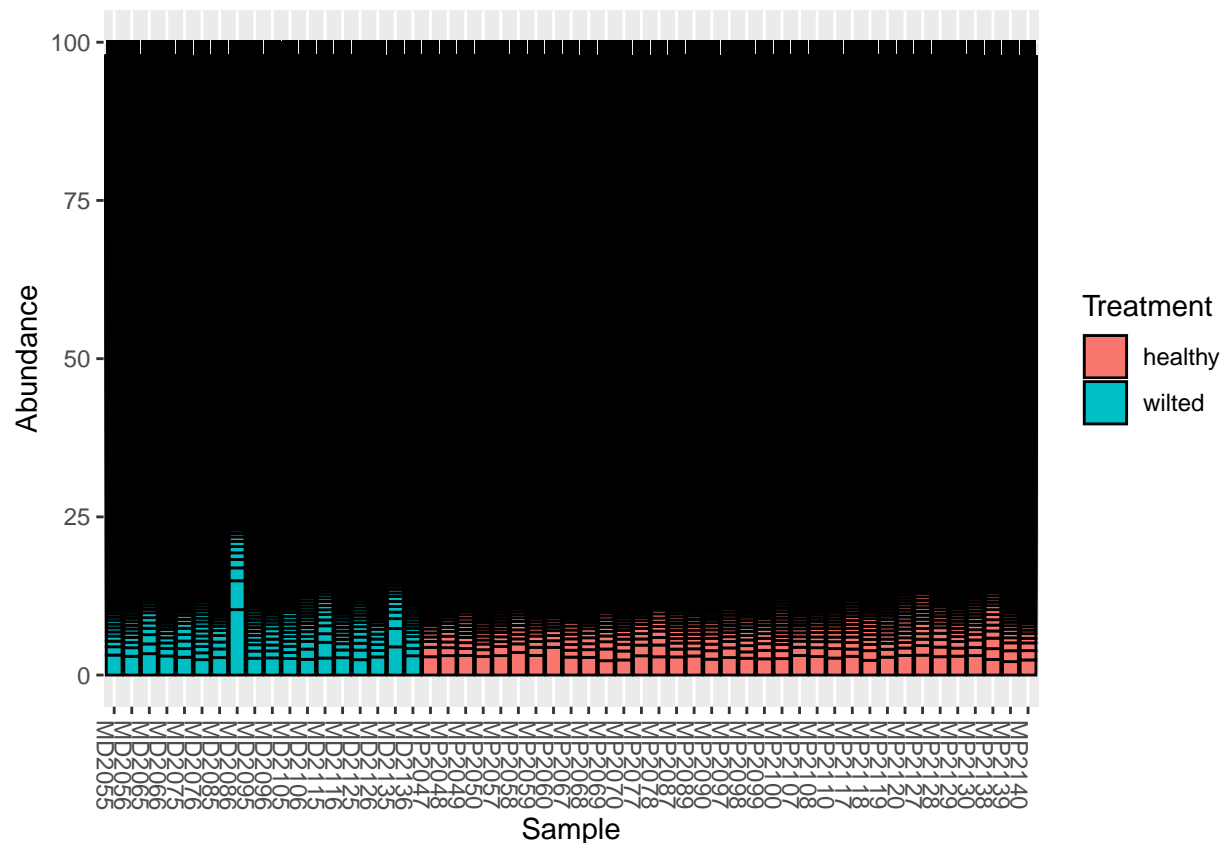
Trazando las muestras en el eje x y las abundancias en el eje y. Podemos ver las abundancias absolutas de las muestras,

```
plot_bar(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(fresa_kraken_fil)
```

Hay dos funciones en el paquete phyloseq para trazar la red del microbioma usando “ggplot2”: plot\_network() y plot\_net().

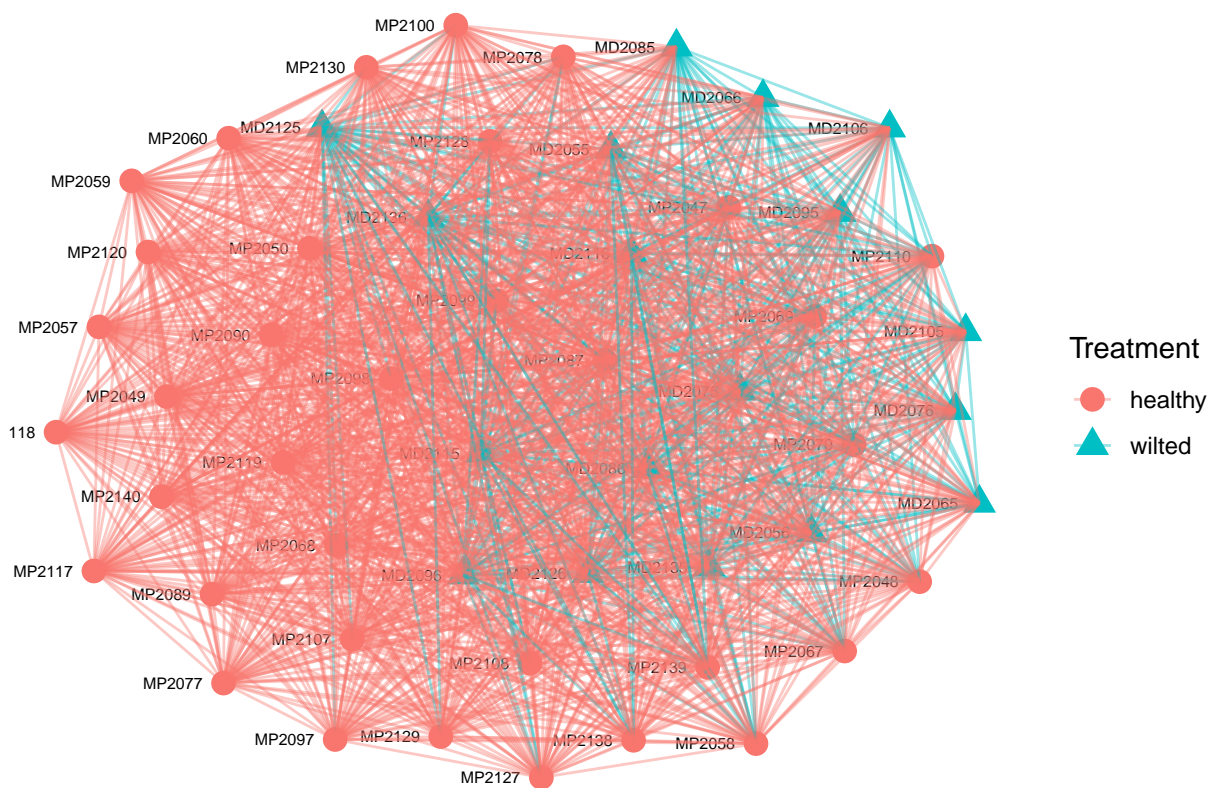
Se crea un grafo basado en “igraph”, basado en el método de distancia por defecto, Jaccard y una distancia máxima entre nodos conectados de 0,8. El “Treatment” se utiliza para los mapeos de color y forma para visualizar la estructura de las muestras.

Hacemos un grafo a partir de el objeto phyloseq

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

Y luego lo graficamos.

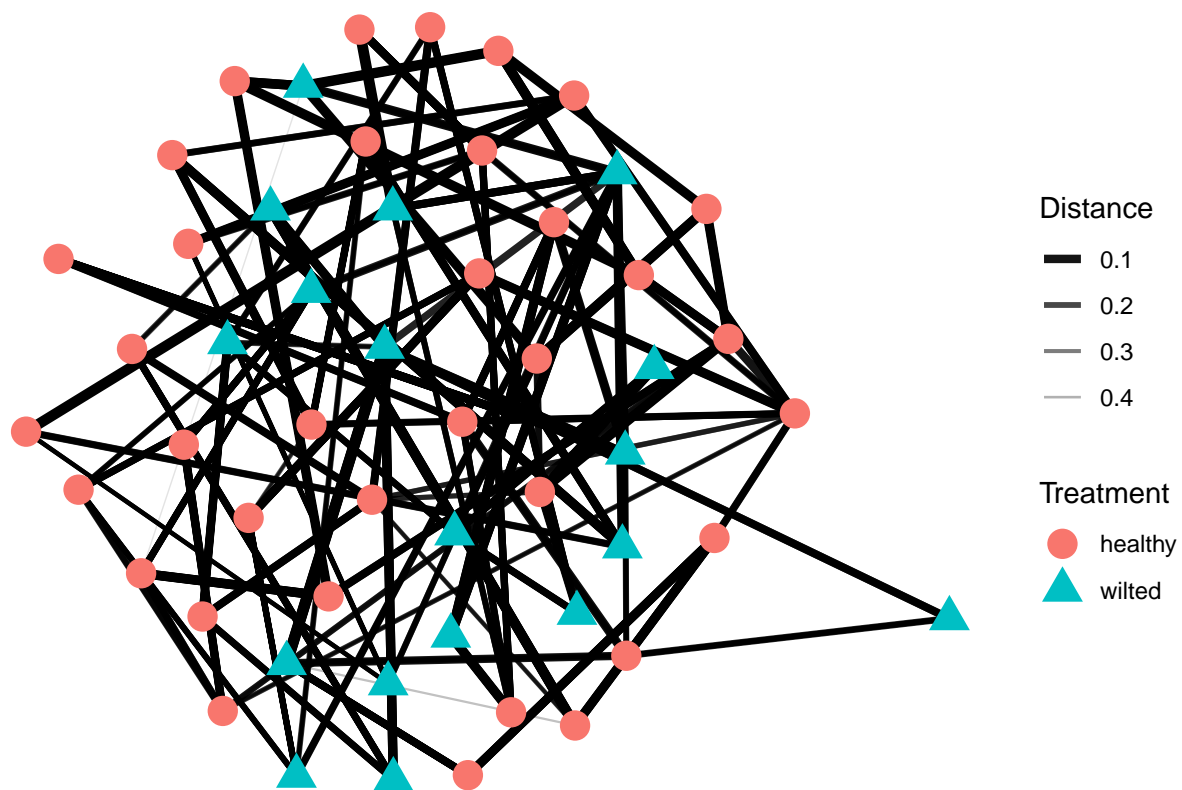
```
plot_network(ig, fresa_kraken_fil, color="Treatment", shape="Treatment")
```



En este grafo podemos ver la complejidad de los datos y las conexiones entre nuestras muestras.

En comparacion con la función `plot_network()`, la nueva función `plot_net()` no requiere una llamada separada a la función `make_network()`, o un objeto `igraph` separado. Los siguientes códigos crean una red basada en una distancia máxima entre los nodos conectados de 0,5.

```
plot_net(fresa_kraken_fil, maxdist = 0.5, 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.