

Project_IRAS

Martijn Melissen

2023-12-07

guides used: <https://www.nicholas-ollberding.com/post/introduction-to-the-statistical-analysis-of-microbiome-data-in-r/>, <https://rpubs.com/lconteville/713954>, <https://mibwurrepo.github.io/Microbial-bioinformatics-introductory-course-Material-2018/beta-diversity-metrics.html>, <https://rfunctions.blogspot.com/2019/03/betadisper-and-adonis-homogeneity-of.html>, <https://david-barnett.github.io/microViz/articles/web-only/compositions.html>, <https://microbiome.github.io/OMA/viz-chapter.html>, <https://microbiome.github.io/OMA/clustering.html>, https://microucph.github.io/amplicon_data_analysis/html/cluster.html, <https://www.datacamp.com/tutorial/hierarchical-clustering-R>, <https://rpubs.com/TBrach/68544>

```
library(phyloseq) # Data analysis and visualisation, also the basis of data object.
library(DT) # Interactive tables in html and markdown.
library(data.table) # Giving overview of data.
library(tidyverse) # Data handling and much more.
library(readxl) # Reading in excel files.
library(ape) # Phylogenetic package, used for creating random trees and as dependency for other packages.
library(magrittr) # Data handling, specifically assignment pipes.
library(microViz) # Both analysis and visualisation.
library(plyr) # to apply functions, transform data.
library(microbiome) # For data analysis and visualisation, reading phyloseq object.
library(ggpubr) # Publication quality figures, based on ggplot2.
library(RColorBrewer) # Color options.
library(microbiomeutilities) # Some utility tools for microbiome package.
library(mia) # microbiome analysis package, making tse objects.
library(sechm) # Used for plotting heatmaps.
library(ggtree) # For creating trees, hierarchical clustering for heatmaps
library(pheatmap) # Creating heatmaps.
library(viridis) # Creating colour pallettes.
library(patchwork) # Used to add plots together into the same plot.
library(data.table) # Alternative to data.frame
library(picante) # Used for calculating Phylogenetic diversities
library(lme4) # Repeated measures, add to report if used
library(QsRutils) # For the goods() function, to estimate coverage
library(scater) # plotReducedDim
library(vegan) # used to run simper
library(nlme) # for usage of llply(), to apply functions over lists
library(mia) # Broad package, includes clustering functions.
library(bluster) # Used for clustering.
library(scater) # visualisation, reduced dimensions.
library(scran) # A wrapper for bluster and tse objects.
```

```

library(NbClust) # To find out the optimal number of clusters.
library(dendextend) # For creating dendrograms with additional options, labeling etc.
library(factoextra) # Visualize optomimal number of clusters.
library(cluster) # For clustering algorithms, specifically used for PAM.

```

Load packages

R Markdown

Metataxonomics

Loading in metataxonomic data

```

pseq <- read_phyloseq(otu.file= "ASV.biom1",
                       taxonomy.file = NULL,
                       metadata.file = "MetaData.csv",
                       type="biom", sep =";" )

treefile <- read_tree("all_asvTREE.tree")
ps <- merge_phyloseq(pseq, treefile)
ps # 180 samples

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 6249 taxa and 180 samples ]
## sample_data() Sample Data: [ 180 samples by 26 sample variables ]
## tax_table() Taxonomy Table: [ 6249 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 6249 tips and 6248 internal nodes ]

sort(sample_sums(ps))

##          14.F2S1.20.06          4.F2S1.11.07          13.F2S2.11.07
##                 46850                  51537                  56421
##          5.F2S2.20.06  8.F5.S2.CA.21.9.17          24.F2S1.11.07
##                 57164                  58212                  59365
##    7.F4.S2.CA.8.8.2017          24.F2S2.20.06          7.F2S2.11.07
##                 60691                  64111                  65002
##          26.F2S2.11.07          17.F2S1.11.07  2.F5.S1.CA.21.9.17
##                 72476                  73085                  76540
##  9.F4.S2.CA.29.8.2017          4.F6.S2.CA.21.9.17          3.F2S2.11.07
##                 77277                  81063                  82316
##    3.F6.S1.CA.21.9.17          3.F6.S2.CA.21.9.17          27.F1S2.31.08
##                 85263                  85370                  85864
##          4.F1S1.21.09          34.F1S1.21.09  9.F5.S1.CA.21.9.17
##                 86303                  90167                  94098
##          20.F1S1.31.08          2.F6.S2.CA.21.9.17  4.F6.S1.CA.21.9.17
##                 94757                  94800                  95509
##  6.F4.S2.CA.8.8.2017          19.F2S2.11.07  7.F5.S1.CA.31.8.17
##                 97472                  99631                  100430
##    28.F1S1.21.09          5.F5.S1.CA.21.9.17  1.F6.S1.CA.21.9.17
##                 101299                 104417                 107531

```

##	3.F5.S1.CA.21.9.17	33.F1S2.21.09	7.F4.S1.CA.29.8.2017
##	107684	110305	115371
##	31.F1S2.31.08	9.F4.S2.CA.8.8.2017	3.F1S2.31.08
##	117379	117431	118155
##	35.F1S2.31.08	4.F5.S2.CA.31.8.17	1.F1S2.21.09
##	119253	119720	121074
##	18.F2S1.20.06	4.F4.S1.CA.29.8.2017	10.F2S1.20.06
##	121143	122331	123221
##	11.F1S2.21.09	6.F5.S1.CA.21.9.17	14.F1S1.21.09
##	123322	123520	126120
##	8.F4.S2.CA.8.8.2017	12.F1S1.21.09	6.F5.S2.CA.21.9.17
##	126501	128514	128935
##	25.F2S2.20.06	6.F1S1.21.09	10.F2S1.11.07
##	129532	130408	132808
##	8.F4.S1.CA.29.8.2017	14.F1S1.31.08	9.F2S2.20.06
##	135194	135303	135373
##	21.F1S2.31.08	15.F1S2.21.09	6.F4.S1.CA.29.8.2017
##	135703	135968	136635
##	8.F6.S1.CA.21.9.17	9.F4.S1.CA.29.8.2017	6.F4.S1.CA.8.8.2017
##	138151	138463	138738
##	4.F5.S1.CA.21.9.17	7.F4.S1.CA.8.8.2017	1.F5.S1.CA.21.9.17
##	140635	140868	142741
##	19.F2S2.20.06	1.F4.S1.CA.8.8.2017	1.F5.S2.CA.21.9.17
##	144528	146133	146908
##	5.F4.S1.CA.8.8.2017	25.F2S1.11.07	7.F5.S1.CA.21.9.17
##	148633	149461	150709
##	2.F6.S1.CA.21.9.17	2.F5.S2.CA.21.9.17	8.F4.S1.CA.8.8.2017
##	150730	151425	153407
##	8.F4.S2.CA.29.8.2017	6.F4.S2.CA.29.8.2017	2.F2S2.20.06
##	153433	153692	153925
##	7.F1S2.21.09	2.F5.S2.CA.31.8.17	19.F1S2.21.09
##	155378	156224	156652
##	1.F4.S1.CA.29.8.2017	2.F1S1.31.08	18.F2S2.11.07
##	157292	158039	158392
##	2.F4.S2.CA.29.8.2017	30.F1S1.21.09	4.F2S2.11.07
##	158954	160633	160749
##	3.F5.S1.CA.31.8.17	8.F5.S1.CA.21.9.17	21.F1S2.21.09
##	161067	163383	163995
##	4.F5.S2.CA.21.9.17	5.F6.S1.CA.21.9.17	8.F6.S1.CA.31.8.17
##	166497	166953	167207
##	5.F4.S2.CA.8.8.2017	22.F1S1.21.09	9.F5.S2.CA.21.9.17
##	167899	169419	170464
##	13.F2S1.11.07	1.F6.S2.CA.21.9.17	5.F4.S1.CA.29.8.2017
##	171477	173496	173499
##	9.F1S2.31.08	3.F4.S2.CA.29.8.2017	20.F2S1.11.07
##	173638	175652	175734
##	6.F1S1.31.08	9.F2S1.11.07	10.F1S1.31.08
##	176305	178651	178746
##	10.F2S2.11.07	3.F4.S1.CA.29.8.2017	5.F5.S2.CA.21.9.17
##	179318	180443	180662
##	8.F2S1.20.06	2.F4.S1.CA.8.8.2017	25.F2S1.20.06
##	182277	188508	189251
##	17.F2S2.20.06	1.F5.S2.CA.31.8.17	9.F4.S1.CA.8.8.2017
##	189516	189756	192242

```

##      1.F2S1.11.07 1.F4.S2.CA.29.8.2017 4.F4.S2.CA.8.8.2017
##      192332          192909          194069
## 3.F5.S2.CA.21.9.17      2.F2S1.20.06 3.F4.S2.CA.8.8.2017
##      194734          195641          199046
## 7.F5.S2.CA.21.9.17 7.F5.S2.CA.31.8.17 2.F4.S2.CA.8.8.2017
##      199400          199807          201158
## 22.F1S1.31.08 5.F4.S2.CA.29.8.2017 2.F4.S1.CA.29.8.2017
##      201793          203638          204571
## 34.F1S1.31.08      5.F1S2.31.08 7.F4.S2.CA.29.8.2017
##      204694          207273          209008
## 1.F6.S2.CA.31.8.17 9.F6.S2.CA.31.8.17 3.F6.S2.CA.31.8.17
##      209323          221264          221983
## 1.F4.S2.CA.8.8.2017      22.F2S2.11.07 3.F5.S2.CA.31.8.17
##      221999          226798          228397
## 5.F5.S1.CA.31.8.17      20.F1S1.21.09 3.F6.S1.CA.31.8.17
##      228481          230243          231433
## 2.F6.S1.CA.31.8.17      13.F1S2.31.08 19.F1S2.31.08
##      232543          233573          235914
## 8.F5.S1.CA.31.8.17      27.F1S2.21.09 6.F6.S2.CA.31.8.17
##      239846          242410          244741
## 3.F4.S1.CA.8.8.2017 9.F6.S1.CA.21.9.17 5.F6.S1.CA.31.8.17
##      246719          247130          248915
## 6.F5.S1.CA.31.8.17      32.F1S1.31.08 2.F5.S1.CA.31.8.17
##      249576          249739          251208
## 4.F4.S1.CA.8.8.2017      22.F2S1.20.06 9.F6.S1.CA.31.8.17
##      251679          251715          257402
## 5.F5.S2.CA.31.8.17 6.F6.S1.CA.21.9.17 9.F5.S2.CA.31.8.17
##      257528          258237          259497
## 6.F5.S2.CA.31.8.17      31.F1S2.21.09 7.F6.S2.CA.31.8.17
##      260095          260175          261493
## 4.F6.S1.CA.31.8.17 4.F5.S1.CA.31.8.17 7.F6.S2.CA.21.9.17
##      263338          264570          269513
## 7.F6.S1.CA.21.9.17 9.F5.S1.CA.31.8.17 8.F5.S2.CA.31.8.17
##      269892          277381          278536
## 8.F6.S2.CA.21.9.17 6.F6.S1.CA.31.8.17 12.F2S2.20.06
##      279210          281640          282992
## 1.F6.S1.CA.31.8.17 8.F6.S2.CA.31.8.17 14.F2S2.20.06
##      283033          283726          289533
## 4.F6.S2.CA.31.8.17      4.F2S1.20.06 4.F4.S2.CA.29.8.2017
##      296588          297958          303543
## 1.F5.S1.CA.31.8.17      28.F1S1.31.08 7.F6.S1.CA.31.8.17
##      304766          305042          307360
## 19.F2S1.20.06 9.F6.S2.CA.21.9.17 5.F6.S2.CA.31.8.17
##      308959          320077          338720
## 2.F6.S2.CA.31.8.17 5.F6.S2.CA.21.9.17 6.F6.S2.CA.21.9.17
##      368702          393697          395258

```

```

### overview data
datatable(tax_table(ps))

```

```

### remove some contamination to filter out plant and eukaryote data that had chloroplast and mitochondrial DNA
subset <- subset_taxa(ps, Domain != "NA")
subset <- subset_taxa(subset, Family != "f__Mitochondria")
subset <- subset_taxa(subset, Family != "f__Mitochondria")

```

```
subset <- subset_taxa(subset, Order != "o__Chloroplast")
subset <- subset_taxa(subset, Domain != "k__Archaea")
```

```
### remove taxa with zeros
subset <- prune_taxa(taxa_sums(subset) > 0, subset)
```

```
### subset phyloseq object n=120 metagenomics data
subset16S <- subset_samples(subset, Metagenomics == "yes" )      #n=120
subset16S <- prune_taxa(taxa_sums(subset16S) > 0, subset16S)
subset16S # 120 samples
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 1536 taxa and 120 samples ]
## sample_data() Sample Data: [ 120 samples by 26 sample variables ]
## tax_table() Taxonomy Table: [ 1536 taxa by 6 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 1536 tips and 1535 internal nodes ]
```

```
# cleaning out all kinds of overlapping names from taxonomy table, removing ~*, *= and =<empty>, this h
subset16S@tax_table = gsub("=\\"*|~\\*|\\\\*|<empty>","",subset16S@tax_table)
```

```
# overview data
datatable(tax_table(subset16S))
```

```
rank_names(subset16S) # Shows classes and ARGs
```

```
## [1] "Domain" "Phylum" "Class" "Order" "Family" "Genus"
```

```
sort(get_taxa_unique(subset16S, "Genus")) # Shows unique genera
```

```
## [1] "g__"
## [2] "g__[Eubacterium]_coprostanoligenes_group"
## [3] "g__[Eubacterium]_hallii_group"
## [4] "g__[Eubacterium]_nodatum_group"
## [5] "g__[Eubacterium]_ventriosum_group"
## [6] "g__[Ruminococcus]_gauvreauii_group"
## [7] "g__[Ruminococcus]_torques_group"
## [8] "g__Akermansia"
## [9] "g__Alistipes"
## [10] "g__Anaerofilum"
## [11] "g__Anaerofustis"
## [12] "g__Anaeroplasma"
## [13] "g__Anaerostipes"
## [14] "g__Anaerotruncus"
## [15] "g__Angelakisella"
## [16] "g__ASF356"
## [17] "g__Bacillus"
## [18] "g__Bacteroides"
## [19] "g__Bifidobacterium"
## [20] "g__Bilophila"
## [21] "g__Blautia"
## [22] "g__Brachybacterium"
```

```

## [23] "g__Butyricicoccus"
## [24] "g__Caldalkalibacillus"
## [25] "g__Candidatus_Soleaferrea"
## [26] "g__CHKCI001"
## [27] "g__CHKCI002"
## [28] "g__Christensenellaceae_R-7_group"
## [29] "g__Coprobacter"
## [30] "g__Coproccus_3"
## [31] "g__Corynebacterium_1"
## [32] "g__Defluviitaleaceae_UCG-011"
## [33] "g__Dielma"
## [34] "g__DTU089"
## [35] "g__Eisenbergiella"
## [36] "g__Enterococcus"
## [37] "g__Erysipelatoclostridium"
## [38] "g__Escherichia-Shigella"
## [39] "g__Faecalibacterium"
## [40] "g__Family_XIII_AD3011_group"
## [41] "g__Flavonifractor"
## [42] "g__Fournierella"
## [43] "g__Fusicatenibacter"
## [44] "g__GCA-900066225"
## [45] "g__GCA-900066575"
## [46] "g__Gordonibacter"
## [47] "g__Hydrogenoanaerobacterium"
## [48] "g__Intestinimonas"
## [49] "g__Lachnoclostridium"
## [50] "g__Lachnospiraceae_FCS020_group"
## [51] "g__Lachnospiraceae_FE2018_group"
## [52] "g__Lachnospiraceae_UCG-004"
## [53] "g__Lachnospiraceae_UCG-006"
## [54] "g__Lachnospiraceae_UCG-008"
## [55] "g__Lactobacillus"
## [56] "g__Marvinbryantia"
## [57] "g__Merdibacter"
## [58] "g__Negativibacillus"
## [59] "g__Odoribacter"
## [60] "g__Oscillibacter"
## [61] "g__Oscillospira"
## [62] "g__Parabacteroides"
## [63] "g__Paraeggerthella"
## [64] "g__Parasutterella"
## [65] "g__Phascolarctobacterium"
## [66] "g__Phoea"
## [67] "g__Ralstonia"
## [68] "g__Romboutsia"
## [69] "g__Roseburia"
## [70] "g__Ruminiclostridium"
## [71] "g__Ruminiclostridium_5"
## [72] "g__Ruminiclostridium_9"
## [73] "g__Ruminococcaceae_NK4A214_group"
## [74] "g__Ruminococcaceae_UCG-005"
## [75] "g__Ruminococcaceae_UCG-009"
## [76] "g__Ruminococcaceae_UCG-010"

```

```

## [77] "g__Ruminococcaceae_UCG-013"
## [78] "g__Ruminococcaceae_UCG-014"
## [79] "g__Ruminococcus_1"
## [80] "g__Ruminococcus_2"
## [81] "g__Sellimonas"
## [82] "g__Shuttleworthia"
## [83] "g__Staphylococcus"
## [84] "g__Streptococcus"
## [85] "g__Subdoligranulum"
## [86] "g__Tyzzerella"
## [87] "g__Tyzzerella_3"
## [88] "g__UBA1819"
## [89] "g__UC5-1-2E3"
## [90] "g__uncultured"
## [91] "g__uncultured_bacterium"
## [92] "g__uncultured_organism"
## [93] "g__Weissella"

```

```
sort(sample_sums(subset16S)) # Amount of unique taxa"per sample, the min is 46731 and max 393697, which
```

##	14.F2S1.20.06	4.F2S1.11.07	13.F2S2.11.07
##	46731	51537	55002
##	5.F2S2.20.06	24.F2S1.11.07	7.F2S2.11.07
##	57164	59138	65002
##	2.F5.S1.CA.21.9.17	4.F6.S2.CA.21.9.17	3.F6.S1.CA.21.9.17
##	76540	80838	85041
##	3.F6.S2.CA.21.9.17	27.F1S2.31.08	4.F1S1.21.09
##	85370	85751	86303
##	20.F1S1.31.08	2.F6.S2.CA.21.9.17	4.F6.S1.CA.21.9.17
##	94222	94800	95509
##	6.F4.S2.CA.8.8.2017	19.F2S2.11.07	7.F5.S1.CA.31.8.17
##	96497	99631	100277
##	5.F5.S1.CA.21.9.17	1.F6.S1.CA.21.9.17	3.F5.S1.CA.21.9.17
##	104268	107531	107684
##	10.F2S1.20.06	31.F1S2.31.08	35.F1S2.31.08
##	112768	117379	119253
##	4.F5.S2.CA.31.8.17	18.F2S1.20.06	1.F1S2.21.09
##	119720	120688	121074
##	4.F4.S1.CA.29.8.2017	6.F5.S1.CA.21.9.17	11.F1S2.21.09
##	121879	123314	123322
##	14.F1S1.21.09	12.F1S1.21.09	6.F5.S2.CA.21.9.17
##	126120	128514	128935
##	6.F1S1.21.09	10.F2S1.11.07	8.F4.S1.CA.29.8.2017
##	130408	132808	134681
##	9.F2S2.20.06	14.F1S1.31.08	21.F1S2.31.08
##	134990	135080	135703
##	15.F1S2.21.09	6.F4.S1.CA.29.8.2017	8.F6.S1.CA.21.9.17
##	135968	136635	137890
##	6.F4.S1.CA.8.8.2017	4.F5.S1.CA.21.9.17	7.F4.S1.CA.8.8.2017
##	138088	140635	140868
##	1.F5.S1.CA.21.9.17	1.F4.S1.CA.8.8.2017	1.F5.S2.CA.21.9.17
##	141444	146133	146908
##	2.F6.S1.CA.21.9.17	2.F5.S2.CA.21.9.17	7.F1S2.21.09
##	150121	151425	153367

```

## 6.F4.S2.CA.29.8.2017      2.F2S2.20.06      2.F5.S2.CA.31.8.17
##          153497           153925           155053
## 19.F1S2.21.09 1.F4.S1.CA.29.8.2017      2.F1S1.31.08
##          156652           157292           158039
## 18.F2S2.11.07 2.F4.S2.CA.29.8.2017      3.F5.S1.CA.31.8.17
##          158392           158632           159579
## 21.F1S2.21.09 4.F5.S2.CA.21.9.17      8.F6.S1.CA.31.8.17
##          163719           165742           166565
## 5.F4.S2.CA.8.8.2017      22.F1S1.21.09      20.F2S1.11.07
##          167240           169419           171242
## 1.F6.S2.CA.21.9.17      9.F1S2.31.08 3.F4.S2.CA.29.8.2017
##          173063           173638           175120
## 6.F1S1.31.08          9.F2S1.11.07      10.F1S1.31.08
##          175950           177762           178746
## 3.F4.S1.CA.29.8.2017      10.F2S2.11.07      5.F5.S2.CA.21.9.17
##          179206           179318           180662
## 8.F2S1.20.06          2.F4.S1.CA.8.8.2017      1.F5.S2.CA.31.8.17
##          182277           188508           189411
## 17.F2S2.20.06          1.F2S1.11.07 1.F4.S2.CA.29.8.2017
##          189516           191885           192096
## 4.F4.S2.CA.8.8.2017      3.F5.S2.CA.21.9.17      2.F2S1.20.06
##          194069           194327           194902
## 3.F4.S2.CA.8.8.2017      2.F4.S2.CA.8.8.2017 5.F4.S2.CA.29.8.2017
##          199046           201158           203364
## 2.F4.S1.CA.29.8.2017      5.F1S2.31.08      1.F6.S2.CA.31.8.17
##          203912           207273           209323
## 3.F6.S2.CA.31.8.17      1.F4.S2.CA.8.8.2017      22.F2S2.11.07
##          221983           221999           225672
## 3.F5.S2.CA.31.8.17      20.F1S1.21.09      3.F6.S1.CA.31.8.17
##          227443           230243           231433
## 2.F6.S1.CA.31.8.17      8.F5.S1.CA.31.8.17      6.F6.S2.CA.31.8.17
##          232543           238256           244741
## 3.F4.S1.CA.8.8.2017      2.F5.S1.CA.31.8.17 4.F4.S1.CA.8.8.2017
##          246719           250484           251679
## 5.F5.S2.CA.31.8.17      6.F5.S2.CA.31.8.17 4.F6.S1.CA.31.8.17
##          257528           260095           263338
## 4.F5.S1.CA.31.8.17      7.F6.S1.CA.21.9.17 7.F6.S2.CA.21.9.17
##          264570           269468           269513
## 12.F2S2.20.06          1.F6.S1.CA.31.8.17 14.F2S2.20.06
##          282992           283033           289533
## 4.F6.S2.CA.31.8.17      4.F2S1.20.06 4.F4.S2.CA.29.8.2017
##          296588           297958           303543
## 28.F1S1.31.08          1.F5.S1.CA.31.8.17 7.F6.S1.CA.31.8.17
##          304246           304446           307360
## 5.F6.S2.CA.31.8.17      2.F6.S2.CA.31.8.17 5.F6.S2.CA.21.9.17
##          338720           368702           393697

```

```
summary(sample_sums(subset16S)) # summary of the sampling depths
```

```

##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
## 46731 125421 161649 173742 212488 393697

```

```
sample_variables(subset16S) # metadata variables
```

```
## [1] "LibraryNumber"      "Sample_Unique"      "LibraryName"      "Farm"
## [5] "Farm2"                "Stable"           "FarmRoundStable"  "Days"
## [9] "Age"                  "Sname"            "WeightAnimal"    "Gender"
## [13] "AgeParentStock"     "Hatchery"         "Researcher"       "AB"
## [17] "Abday"               "FlockSize"        "FeedF"            "FeedType"
## [21] "FeedProducent"      "Cox"              "OPG"              "Cluster"
## [25] "LitterType"          "Metagenomics"
```

```
# Rewriting sampleIDs as sample_unique rownames to align with the other datasets
```

```
sample_names(subset16S) = sample_data(subset16S)$Sample_Unique
sample_names(subset16S)
```

```
## [1] "2_23"   "2_24"   "2_25"   "2_26"   "2_27"   "2_29"   "2_36"   "2_39"   "2_40"
## [10] "2_41"   "2_42"   "2_47"   "2_48"   "2_49"   "2_50"   "2_51"   "2_52"   "2_56"
## [19] "2_57"   "2_58"   "2_59"   "2_60"   "2_61"   "4_36"   "4_37"   "4_38"   "4_39"
## [28] "4_40"   "4_41"   "4_54"   "4_55"   "4_56"   "4_57"   "4_65"   "5_39"   "5_40"
## [37] "5_41"   "5_54"   "5_55"   "5_59"   "6_36"   "6_37"   "6_38"   "6_54"   "6_55"
## [46] "6_56"   "6_57"   "6_58"   "9_16"   "9_17"   "9_18"   "9_19"   "9_21"   "9_22"
## [55] "9_34"   "9_35"   "9_36"   "9_37"   "9_38"   "9_39"   "10_1"  "10_2"  "10_3"
## [64] "10_4"   "10_7"   "10_8"   "10_10"  "10_11"  "10_12"  "10_13"  "10_14"  "10_15"
## [73] "10_19"  "10_20"  "10_21"  "10_22"  "10_25"  "10_26"  "10_28"  "10_29"  "10_30"
## [82] "10_33"  "10_34"  "10_35"  "10_39"  "10_40"  "10_41"  "10_42"  "10_43"  "10_44"
## [91] "10_48"  "10_49"  "10_50"  "10_51"  "10_52"  "10_53"  "10_57"  "10_58"  "10_59"
## [100] "10_60"  "10_63"  "10_64"  "10_66"  "10_67"  "10_68"  "10_69"  "11_1"  "11_3"
## [109] "14_20"  "14_21"  "14_22"  "14_23"  "14_25"  "14_27"  "14_29"  "14_30"  "14_33"
## [118] "14_34"  "14_35"  "14_36"
```

```
# Stable "Farm2R1S1" has the three lowest sampling depths of the dataset, the other nine samples are f
subset16S %>% ps_filter(FarmRoundStable == c("Farm2R1S1")) %>% sample_sums() %>% sort()
```

```
##   4_40   4_55   5_55   4_39   4_41   4_57   5_54   4_56   4_38   4_54   4_36
## 46731 51537 59138 112768 120688 132808 171242 177762 182277 191885 194902
##   4_37
## 297958
```

```
# Amount of different taxa present.
sort(table(tax_table(subset16S)[, "Phylum"]))
```

```
##
## p__Verrucomicrobia  p__Cyanobacteria  p__Proteobacteria  p__Bacteroidetes
##                      2                  13                  14                  22
## p__Actinobacteria    p__Tenericutes     p__Firmicutes
##                      48                  81                  1356
```

```
sort(table(tax_table(subset16S)[, "Order"]))
```

```
##
```

```

##          o__Selenomonadales      o__Corynebacteriales    o__Desulfovibrionales
##                           1                      2                      2
##          o__Izimaplasmatales      o__Micrococcales       o__Verrucomicrobiales
##                           2                      2                      2
##          o__                    o__Anaeroplasmatales   o__Enterobacteriales
##                           4                      4                      5
##          o__Bacillales          o__Betaproteobacteriales o__Coriobacteriales
##                           6                      7                      12
##          o__Gastranaerophilales  o__Bacteroidales        o__Bifidobacteriales
##                           13                     22                     32
##          o__Erysipelotrichales  o__Lactobacillales      o__Mollicutes_RF39
##                           39                     45                     71
##          o__Clostridiales        1265

```

```
sort(table(tax_table(subset16S) [, "Family"]))
```

```

##
##          f__Acidaminococcaceae      f__Barnesiellaceae
##                           1                      1
##          f__Dermabacteraceae        f__Eubacteriaceae
##                           1                      1
##          f__F082                  f__Gracilibacteraceae
##                           1                      1
##          f__Leuconostocaceae        f__Marinifilaceae
##                           1                      1
##          f__Micrococcaceae         f__Akkermansiaceae
##                           1                      2
##          f__Corynebacteriaceae     f__Desulfovibrionaceae
##                           2                      2
##          f__Peptococcaceae         f__Streptococcaceae
##                           2                      2
##          f__uncultured_Firmicutes_bacterium f__Bacillaceae
##                           2                      3
##          f__Rikenellaceae          f__Staphylococcaceae
##                           3                      3
##          f__Tannerellaceae         f__uncultured_rumen_bacterium
##                           3                      3
##          f__Anaeroplasmataceae    f__Defluvitaleaceae
##                           4                      4
##          f__Family_XIII           f__Enterobacteriaceae
##                           4                      5
##          f__Enterococcaceae       f__Burkholderiaceae
##                           5                      7
##          f__Peptostreptococcaceae  f__Eggerthellaceae
##                           11                     12
##          f__Bacteroidaceae        f__Christensenellaceae
##                           13                     24
##          f__Clostridiales_vadinBB60_group f__Bifidobacteriaceae
##                           28                     32
##          f__                         f__Lactobacillaceae
##                           33                     37
##          f__Erysipelotrichaceae    f__uncultured_bacterium
##                           39                     62

```

```

##          f__Ruminococcaceae      f__Lachnospiraceae
##                                549                      631

# factorizing variables as not to create problems with visualization later down the line
sample_data(subset16S)$Cluster = as.factor(sample_data(subset16S)$Cluster)
sample_data(subset16S)$FlockSize = as.factor(sample_data(subset16S)$FlockSize)
sample_data(subset16S)$AgeParentStock = as.factor(sample_data(subset16S)$AgeParentStock)
sample_data(subset16S)$Age = as.factor(sample_data(subset16S)$Age)
sample_data(subset16S)$LibraryNumber = as.factor(sample_data(subset16S)$LibraryNumber)

# add stable column with shorter names
sample_data(subset16S)$FarmRoundStable = as.factor(sample_data(subset16S)$FarmRoundStable)
subset16S@sam_data$Stables = revalue(sample_data(subset16S)$FarmRoundStable, c("Farm1R1S1"="Stable1", "Farm1R2S1"="Stable2", "Farm2R1S1"="Stable3", "Farm2R2S1"="Stable4", "Farm3R1S1"="Stable5", "Farm3R2S1"="Stable6", "Farm4R1S1"="Stable7", "Farm4R2S1"="Stable8", "Farm5R1S1"="Stable9", "Farm5R2S1"="Stable10"))

# Shortening agent names
subset16S@sam_data$Cox[subset16S@sam_data$Cox == "narasinandnicarbazin(maxiban)"] = "Maxiban"
subset16S@sam_data$Cox[subset16S@sam_data$Cox == "narasin(monteban)"] = "Monteban"
subset16S@sam_data$Cox[subset16S@sam_data$Cox == "salinomycin(Sacox120microGranulate)"] = "Sacox"

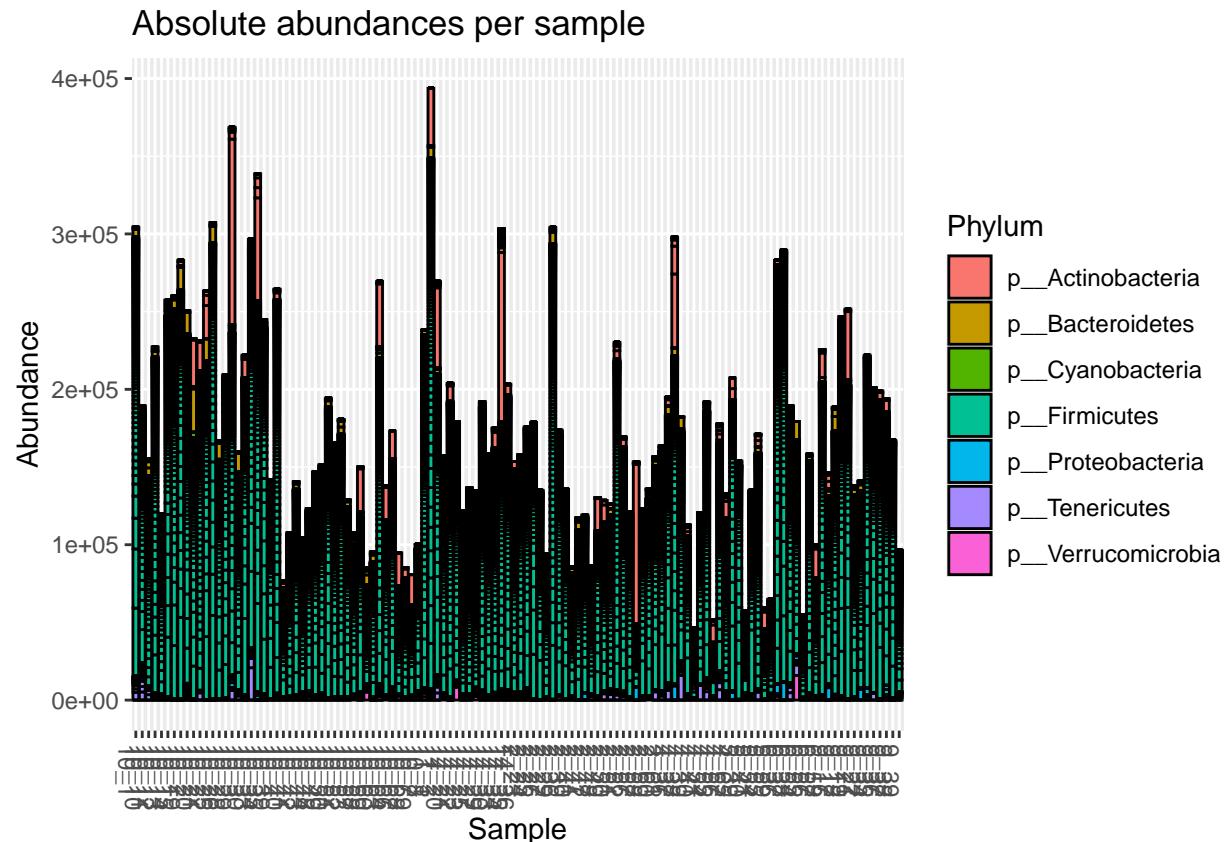
```

Abundances and heatmaps

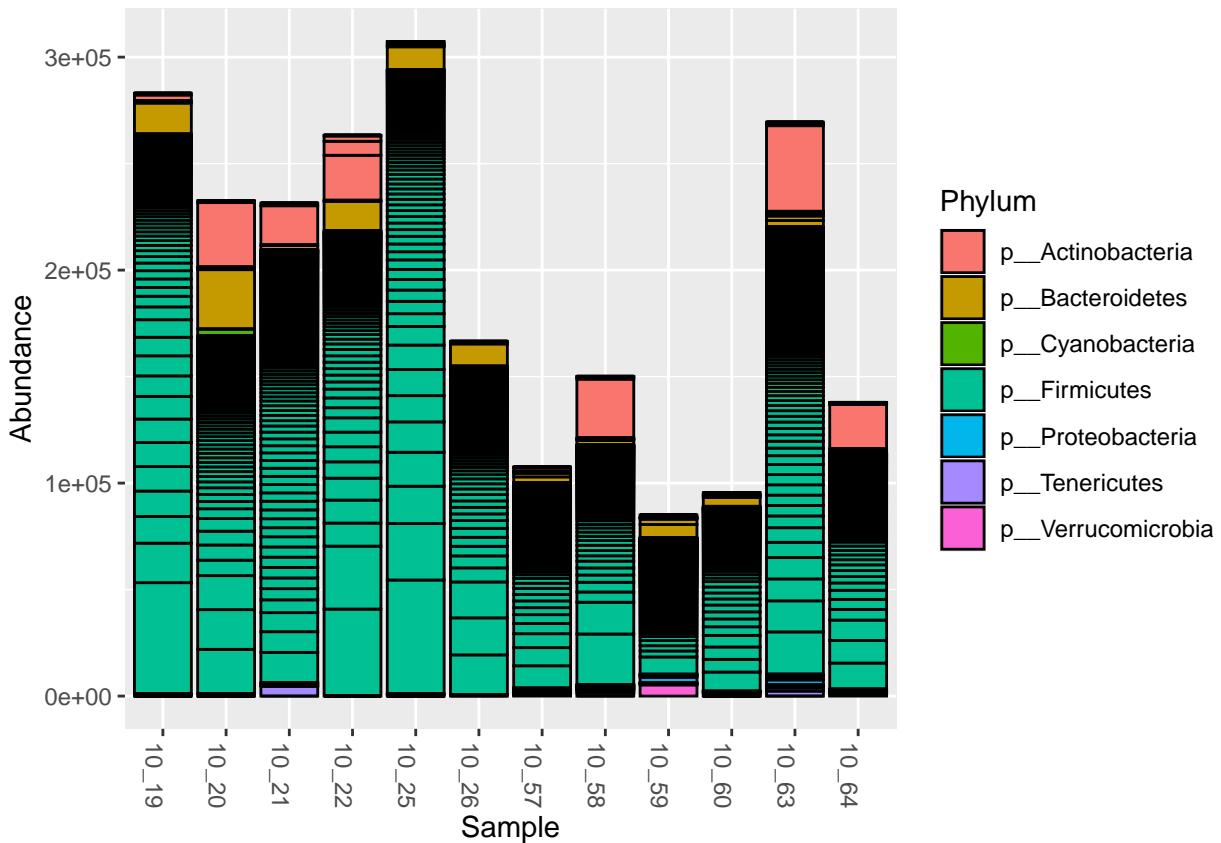
```

# absolute abundances - phylum
plot_bar(subset16S, fill="Phylum", title = "Absolute abundances per sample")

```

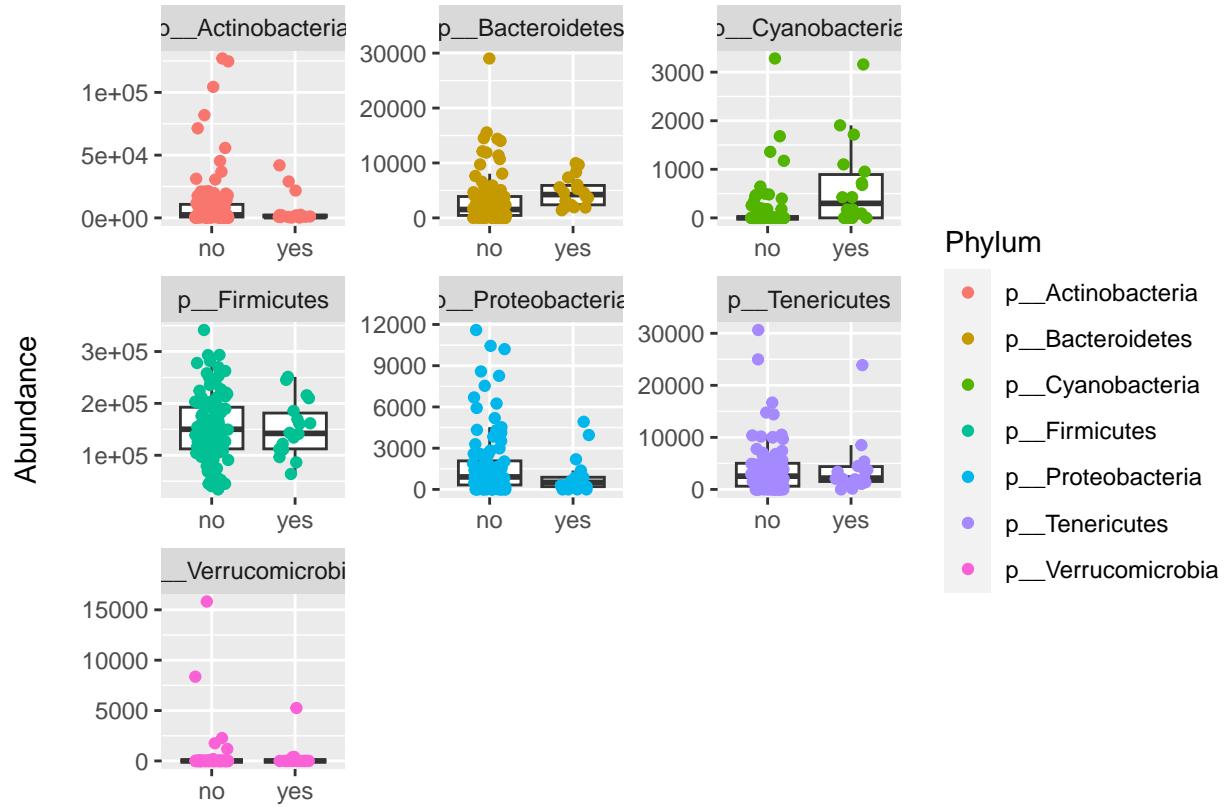


```
# for plotting abundances of specific stables
subset16S %>% ps_filter(Stables == c("Stable9")) %>% plot_bar(fill="Phylum")
```

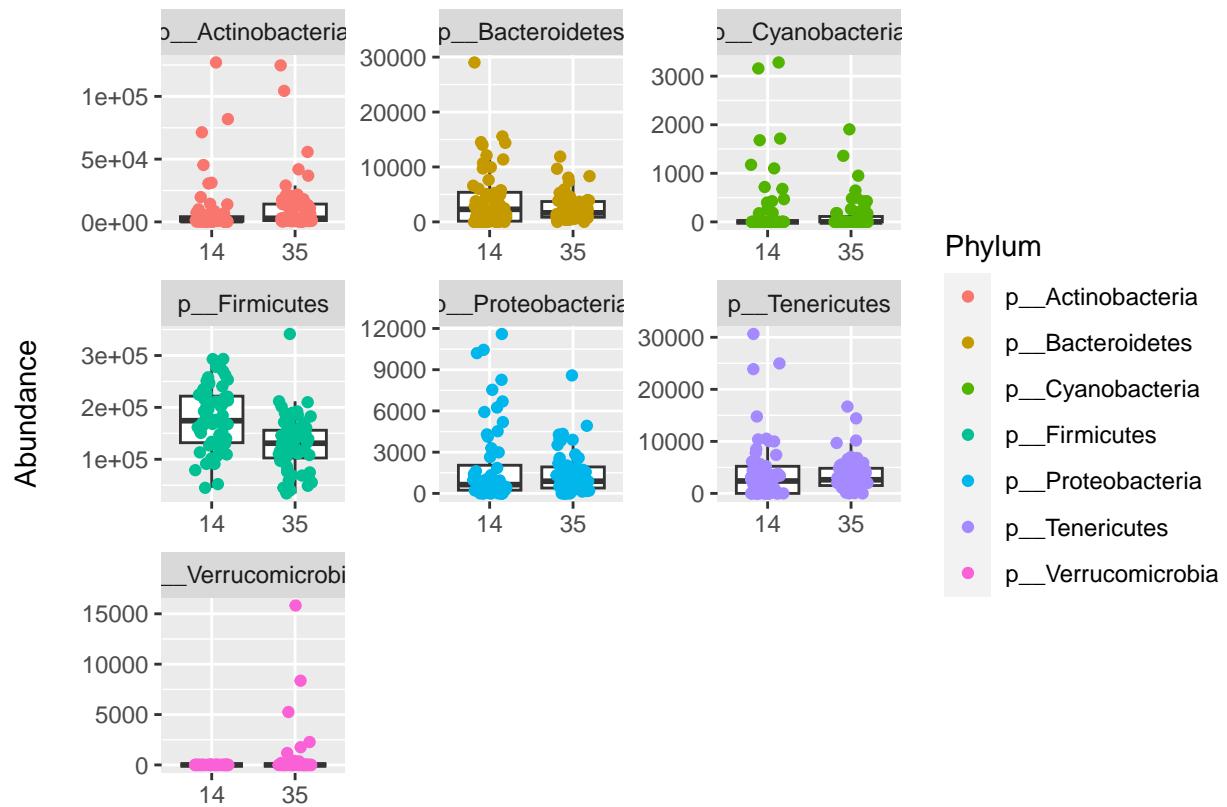


```
# visualisation on AB at Phylum level, more data for samples which have not been treated with AB, but a
ps_prim <- phyloseq::tax_glom(subset16S, "Phylum")
taxa_names(ps_prim) <- phyloseq::tax_table(ps_prim)[, "Phylum"]

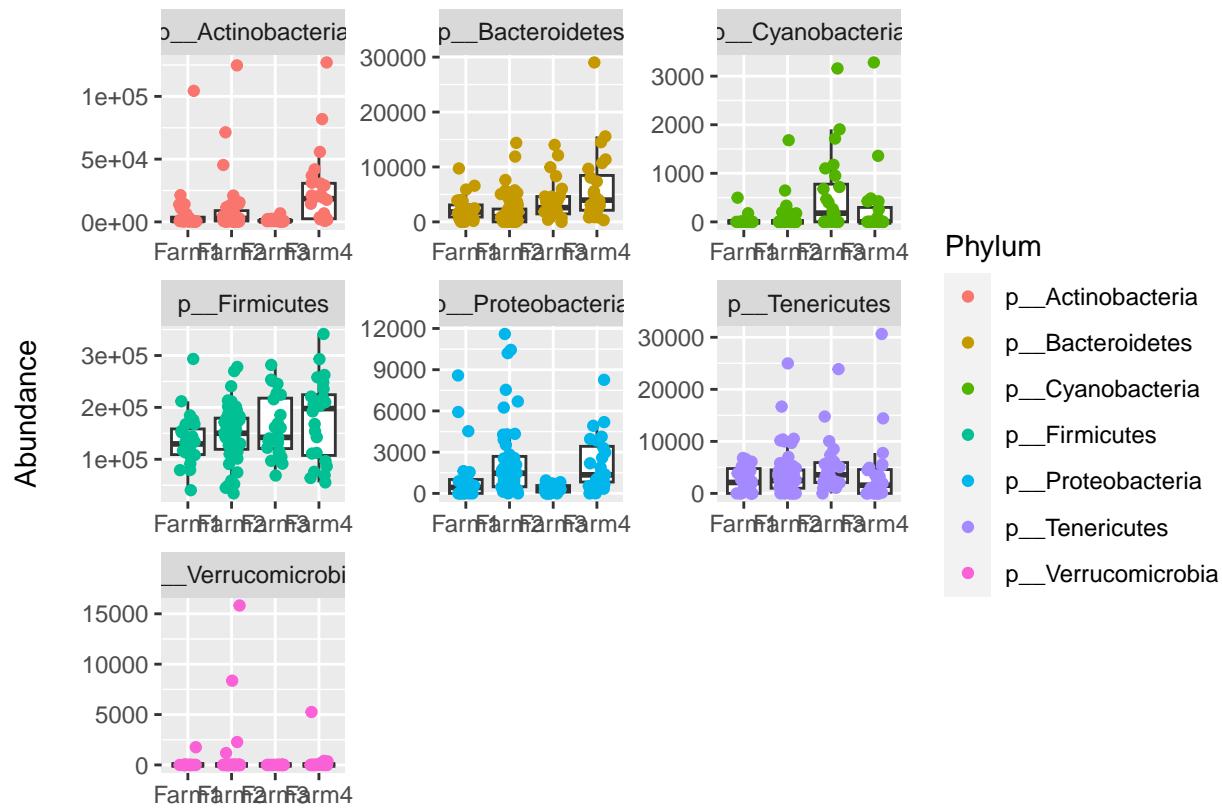
psmelt(ps_prim) %>% # AB
ggplot(data = ., aes(x = AB, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ Phylum, scales = "free")
```



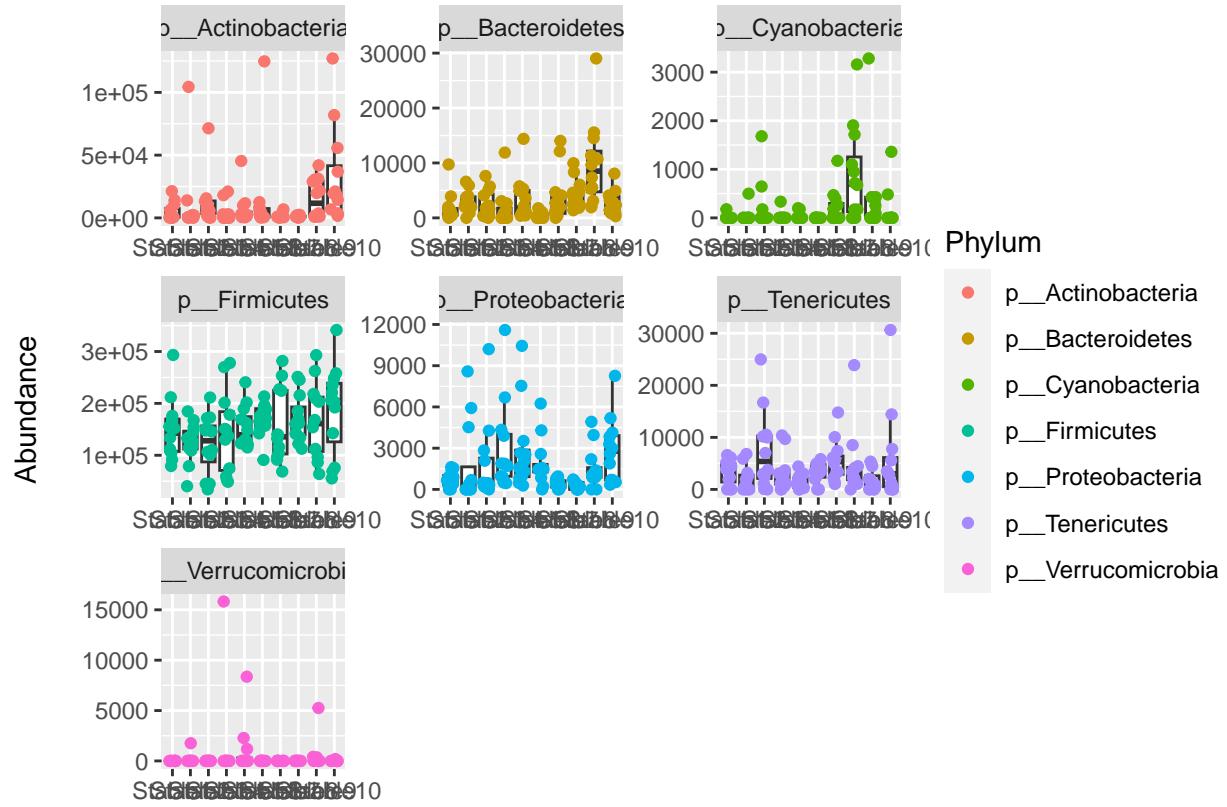
```
psmelt(ps_prim) %>% # Age
ggplot(data = ., aes(x = Age, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ Phylum, scales = "free")
```



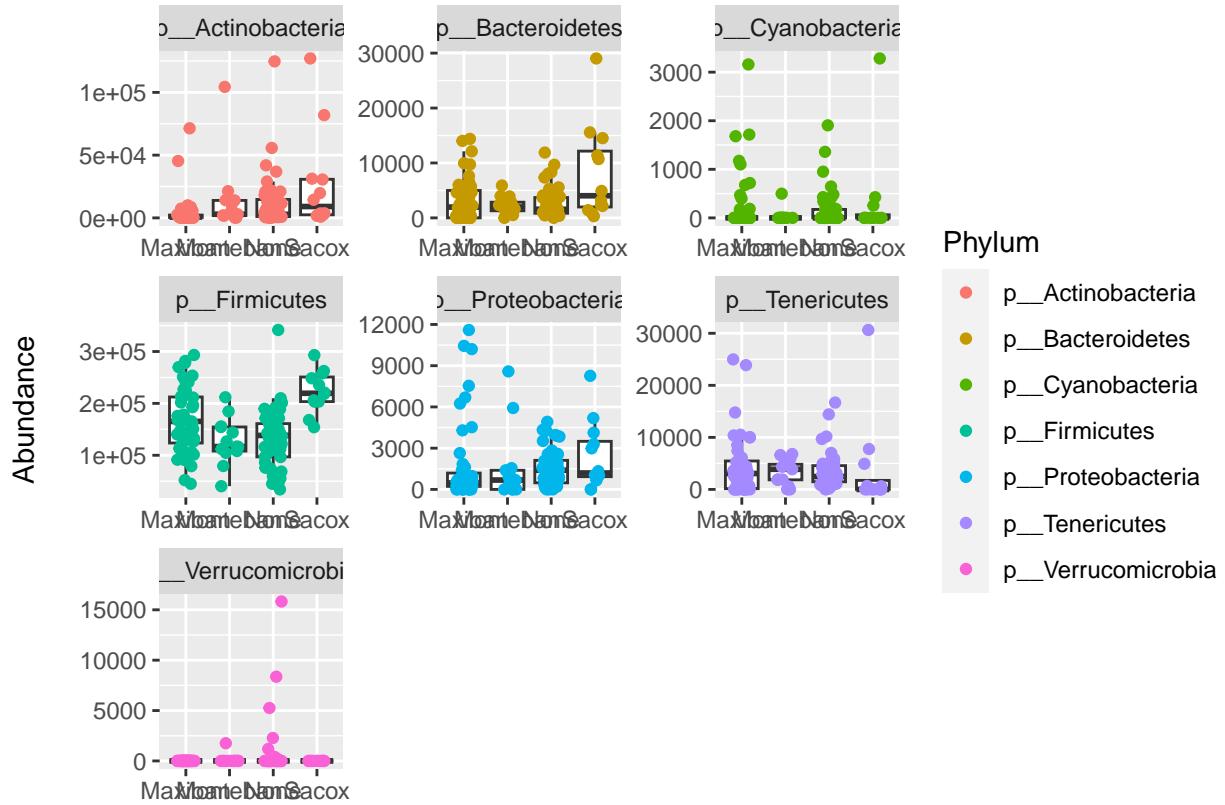
```
psmelt(ps_prim) %>% # Farm
ggplot(data = ., aes(x = Farm2, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ Phylum, scales = "free")
```



```
psmelt(ps_prim) %>% # Stable
ggplot(data = ., aes(x = Stables, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ Phylum, scales = "free")
```



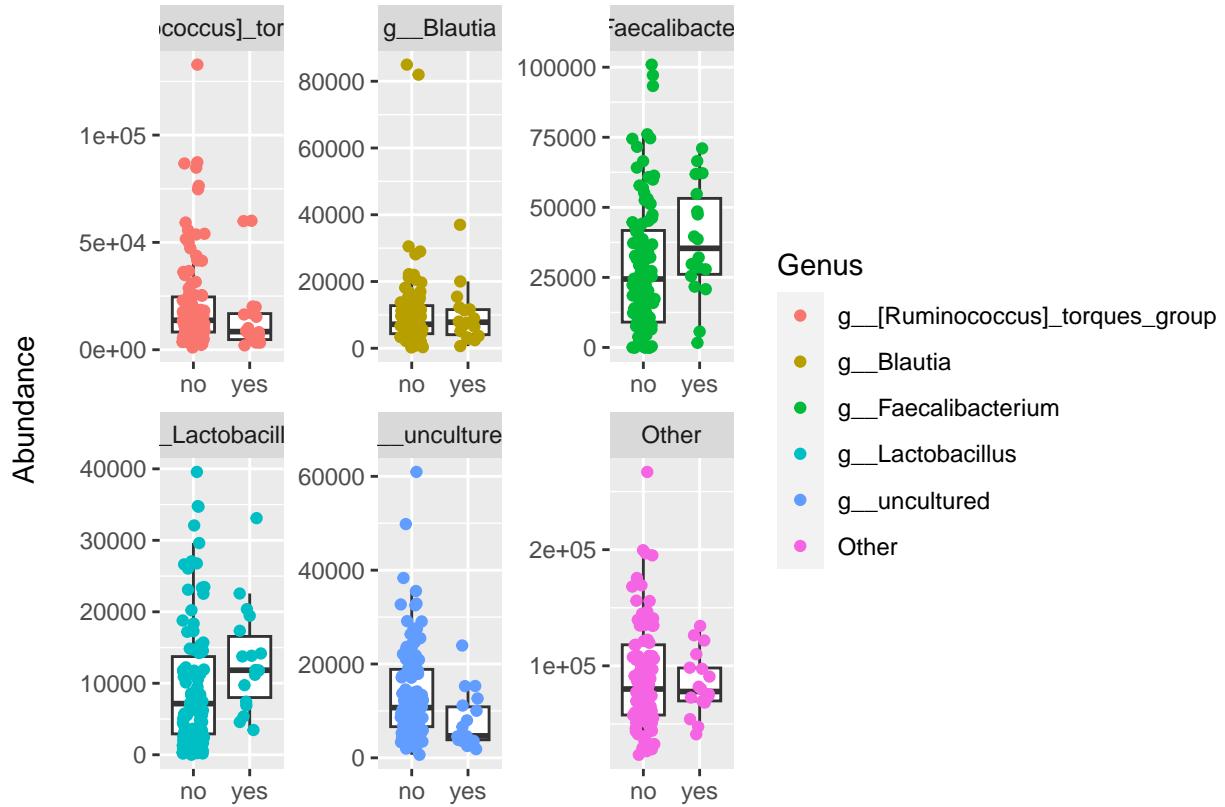
```
psmelt(ps_prim) %>% # Agent
ggplot(data = ., aes(x = Cox, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ Phylum, scales = "free")
```



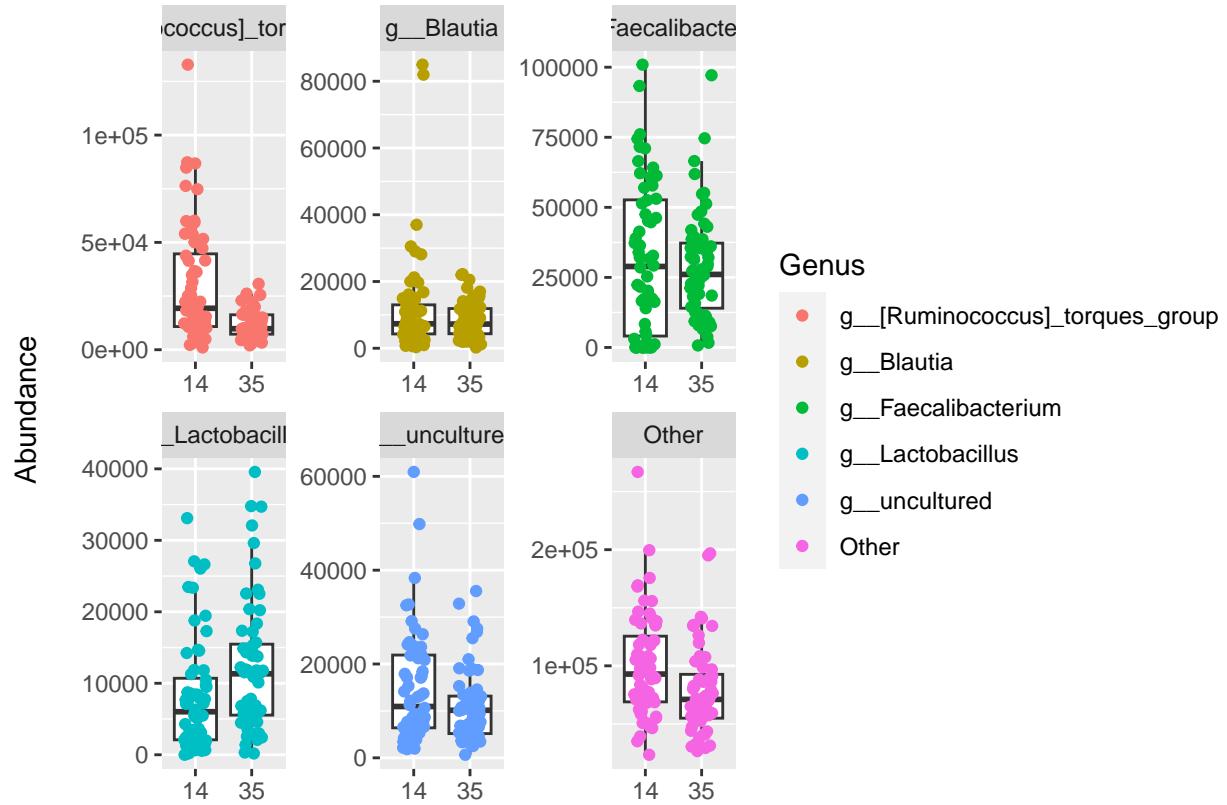
```
# visualisation on AB at Genus level, more data for samples which have not been treated with AB, but also

ps_prim <- subset16S %>% aggregate_top_taxa2("Genus", top = 5) %>% phyloseq::tax_glom("Genus")
taxa_names(ps_prim) <- phyloseq::tax_table(ps_prim)[, "Genus"]

psmelt(ps_prim) %>% # AB
  ggplot(data = ., aes(x = AB, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Genus), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ Genus, scales = "free")
```



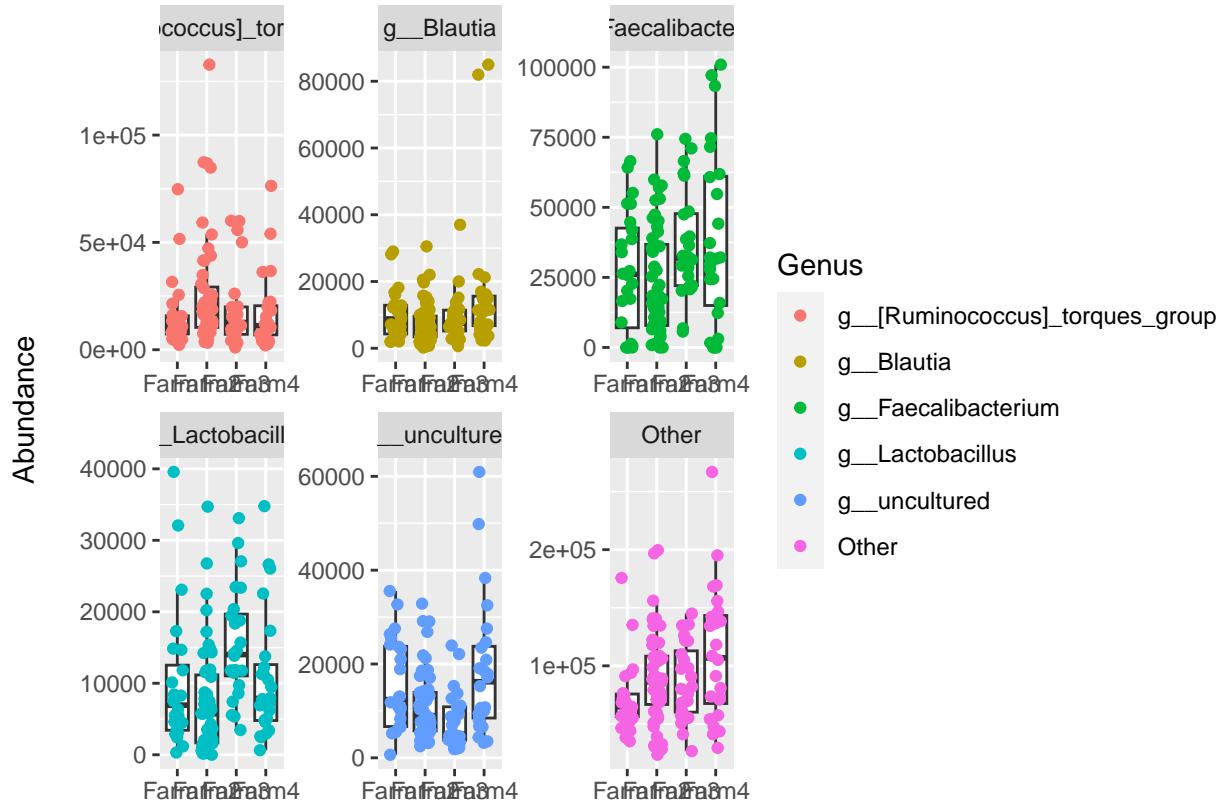
```
psmelt(ps_prim) %>% # Age
ggplot(data = ., aes(x = Age, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Genus), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ Genus, scales = "free")
```



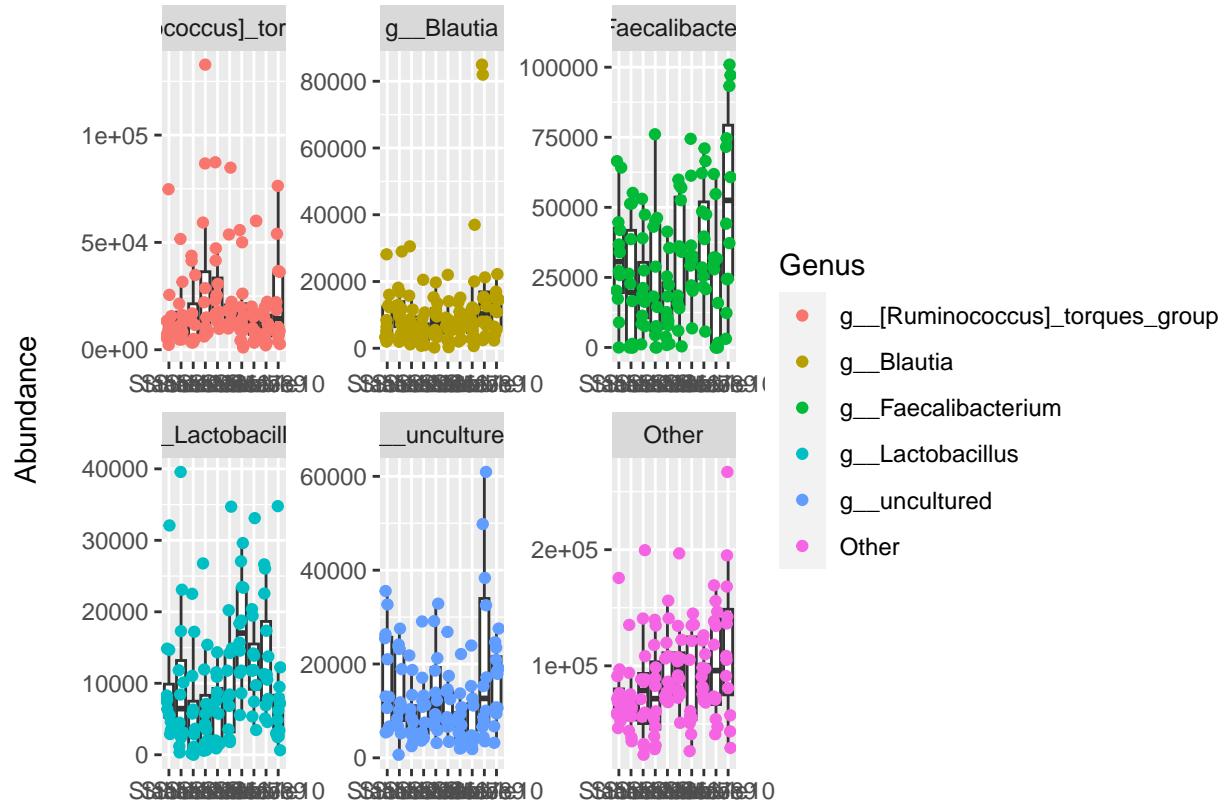
```

psmelt(ps_prim) %>% # Farm
ggplot(data = ., aes(x = Farm2, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Genus), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ Genus, scales = "free")

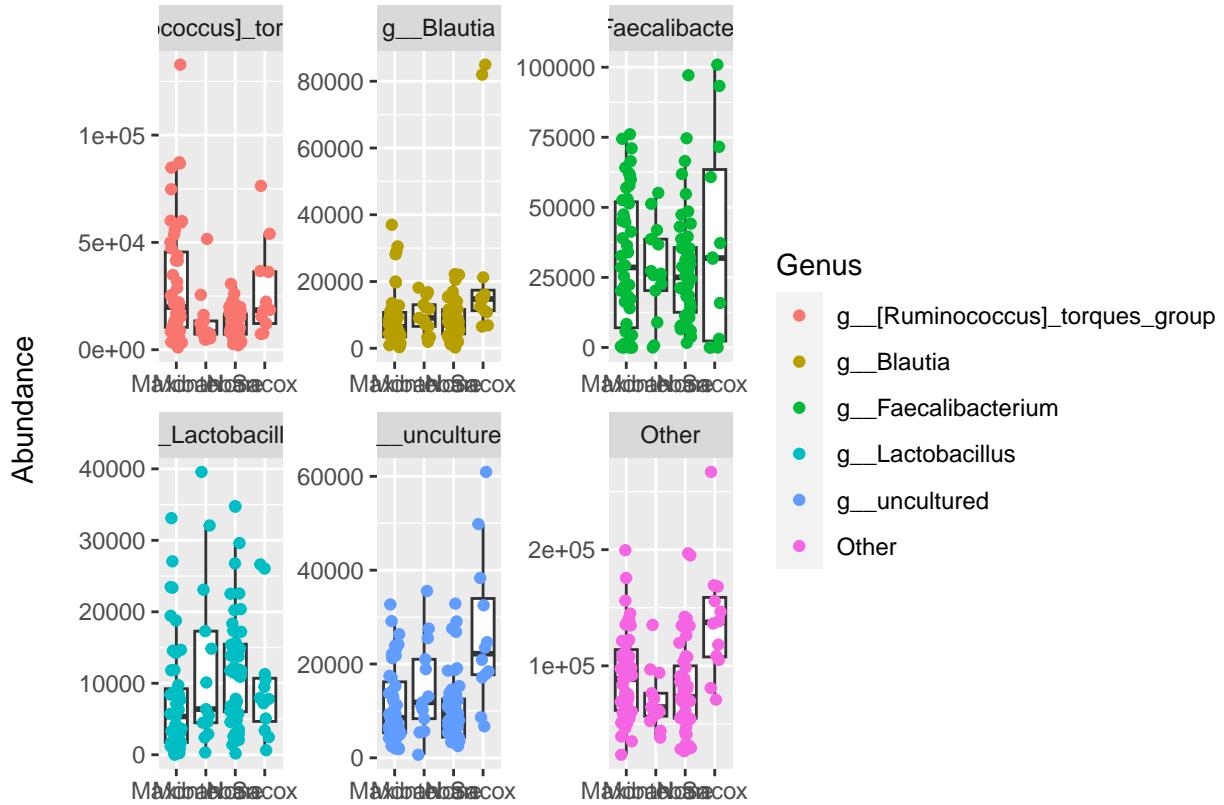
```



```
psmelt(ps_prim) %>% # Stable
ggplot(data = ., aes(x = Stables, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Genus), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ Genus, scales = "free")
```



```
psmelt(ps_prim) %>% # Agent
ggplot(data = ., aes(x = Cox, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Genus), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ Genus, scales = "free")
```



```
# Check the amount of unique genera in samples which have and have not been treated with antibiotics
subset16S %>% ps_filter(AB == "no") %>% get_taxa_unique("Genus") # 93 different genera for non AB treat
```

```
## [1] "g__"
## [2] "g__Dielma"
## [3] "g__uncultured"
## [4] "g__Anaeroplasma"
## [5] "g__Merdibacter"
## [6] "g__Erysipelatoclostridium"
## [7] "g__Blautia"
## [8] "g__Odoribacter"
## [9] "g__Parabacteroides"
## [10] "g__Coprobacter"
## [11] "g__Bacteroides"
## [12] "g__Alistipes"
## [13] "g__Bilophila"
## [14] "g__Parasutterella"
## [15] "g__Ralstonia"
## [16] "g__Escherichia-Shigella"
## [17] "g__Corynebacterium_1"
## [18] "g__Brachybacterium"
## [19] "g__Bifidobacterium"
## [20] "g__Ruminococcaceae_UCG-014"
## [21] "g__Lachnospiraceae_UCG-008"
## [22] "g__CHKC1002"
## [23] "g__Gordonibacter"
```

```

## [24] "g__Akkermansia"
## [25] "g__Paraeggerthella"
## [26] "g__Christensenellaceae_R-7_group"
## [27] "g__uncultured_bacterium"
## [28] "g__Ruminococcaceae_UCG-010"
## [29] "g__Tyzzerella_3"
## [30] "g__Family_XIII_AD3011_group"
## [31] "g__Romboutsia"
## [32] "g__Staphylococcus"
## [33] "g__Caldalkalibacillus"
## [34] "g__Lactobacillus"
## [35] "g__Weissella"
## [36] "g__Bacillus"
## [37] "g__Enterococcus"
## [38] "g__Eisenbergiella"
## [39] "g__Streptococcus"
## [40] "g__Phascolarctobacterium"
## [41] "g__uncultured_organism"
## [42] "g__Anaerofustis"
## [43] "g__Ruminococcaceae_UCG-013"
## [44] "g__Butyricicoccus"
## [45] "g__Faecalibacterium"
## [46] "g__Phocea"
## [47] "g__Subdoligranulum"
## [48] "g__GCA-900066225"
## [49] "g__[Eubacterium]_coprostanoligenes_group"
## [50] "g__Negativibacillus"
## [51] "g__Hydrogenoanaerobacterium"
## [52] "g__Angelakisella"
## [53] "g__Fournierella"
## [54] "g__Anaerofilum"
## [55] "g__Anaerotruncus"
## [56] "g__[Ruminococcus]_torques_group"
## [57] "g__Sellimonas"
## [58] "g__Lachnoclostridium"
## [59] "g__GCA-900066575"
## [60] "g__CHKC1001"
## [61] "g__UC5-1-2E3"
## [62] "g__Fusicatenibacter"
## [63] "g__Roseburia"
## [64] "g__Marvinbryantia"
## [65] "g__Coprococcus_3"
## [66] "g__Lachnospiraceae_UCG-006"
## [67] "g__[Ruminococcus]_gauvreauii_group"
## [68] "g__Lachnospiraceae_UCG-004"
## [69] "g__Lachnospiraceae_FCS020_group"
## [70] "g__Lachnospiraceae_FE2018_group"
## [71] "g__[Eubacterium]_ventriosum_group"
## [72] "g__Shuttleworthia"
## [73] "g__[Eubacterium]_hallii_group"
## [74] "g__Anaerostipes"
## [75] "g__UBA1819"
## [76] "g__Ruminococcus_1"
## [77] "g__Defluvitaleaceae_UCG-011"

```

```

## [78] "g__ASF356"
## [79] "g__Tyzzerella"
## [80] "g__[Eubacterium]_nodatum_group"
## [81] "g__Ruminiclostridium_9"
## [82] "g__Ruminiclostridium_5"
## [83] "g__Ruminiclostridium"
## [84] "g__Oscillospira"
## [85] "g__Oscillibacter"
## [86] "g__Ruminococcus_2"
## [87] "g__Ruminococcaceae_UCG-009"
## [88] "g__Ruminococcaceae_UCG-005"
## [89] "g__DTU089"
## [90] "g__Candidatus_Soleferrea"
## [91] "g__Ruminococcaceae_NK4A214_group"
## [92] "g__Flavonifractor"
## [93] "g__Intestinimonas"

subset16S %>% ps_filter(AB == "yes") %>% get_taxa_unique("Genus") # 74 different genera for AB treated

## [1] "g__"
## [2] "g__uncultured"
## [3] "g__Anaeroplasma"
## [4] "g__Merribacter"
## [5] "g__Erysipelatoclostridium"
## [6] "g__Odoribacter"
## [7] "g__Bacteroides"
## [8] "g__Alistipes"
## [9] "g__uncultured_bacterium"
## [10] "g__Bilophila"
## [11] "g__Parasutterella"
## [12] "g__Escherichia-Shigella"
## [13] "g__Bifidobacterium"
## [14] "g__Ruminococcaceae_UCG-014"
## [15] "g__CHKC1002"
## [16] "g__Gordonibacter"
## [17] "g__Akermansia"
## [18] "g__Christensenellaceae_R-7_group"
## [19] "g__Ruminococcaceae_UCG-010"
## [20] "g__Tyzzerella_3"
## [21] "g__Romboutsia"
## [22] "g__Lactobacillus"
## [23] "g__Bacillus"
## [24] "g__Enterococcus"
## [25] "g__Eisenbergiella"
## [26] "g__Streptococcus"
## [27] "g__Phascolarctobacterium"
## [28] "g__Anaerofustis"
## [29] "g__Ruminococcaceae_UCG-013"
## [30] "g__Butyricicoccus"
## [31] "g__Faecalibacterium"
## [32] "g__Phocea"
## [33] "g__Subdoligranulum"
## [34] "g__GCA-900066225"
## [35] "g__Negativibacillus"

```

```

## [36] "g__Hydrogenoanaerobacterium"
## [37] "g__Anaerofilum"
## [38] "g__Anaerotruncus"
## [39] "g__Fournierella"
## [40] "g__[Ruminococcus]_torques_group"
## [41] "g__Sellimonas"
## [42] "g__GCA-900066575"
## [43] "g__Blautia"
## [44] "g__CHKCI001"
## [45] "g__UC5-1-2E3"
## [46] "g__Fusicatenibacter"
## [47] "g__Marvinbryantia"
## [48] "g__Coprococcus_3"
## [49] "g__Lachnospiraceae_UCG-006"
## [50] "g__Lachnospiraceae_UCG-008"
## [51] "g__Lachnoclostridium"
## [52] "g__Lachnospiraceae_UCG-004"
## [53] "g__Lachnospiraceae_FCS020_group"
## [54] "g__Shuttleworthia"
## [55] "g__Ruminiclostridium_9"
## [56] "g__[Eubacterium]_hallii_group"
## [57] "g__Anaerostipes"
## [58] "g__Roseburia"
## [59] "g__Defluviateaceae_UCG-011"
## [60] "g__ASF356"
## [61] "g__Tyzzerella"
## [62] "g__[Eubacterium]_nodatum_group"
## [63] "g__Ruminococcus_1"
## [64] "g__Oscillospira"
## [65] "g__Oscillibacter"
## [66] "g__Ruminiclostridium"
## [67] "g__[Eubacterium]_coprostanoligenes_group"
## [68] "g__Ruminococcaceae_UCG-005"
## [69] "g__DTU089"
## [70] "g__Ruminiclostridium_5"
## [71] "g__Ruminococcus_2"
## [72] "g__Candidatus_Soleferrea"
## [73] "g__Ruminococcaceae_NK4A214_group"
## [74] "g__Flavonifractor"

```

```
subset16S %>% get_taxa_unique("Genus") # 93 different genes in total, which are all present in non-trea
```

```

## [1] "g__"
## [2] "g__Dielma"
## [3] "g__uncultured"
## [4] "g__Anaeroplasma"
## [5] "g__Merdibacter"
## [6] "g__Erysipelatoclostridium"
## [7] "g__Blautia"
## [8] "g__Odoribacter"
## [9] "g__Parabacteroides"
## [10] "g__Coprobacter"
## [11] "g__Bacteroides"
## [12] "g__Alistipes"

```

```

## [13] "g__uncultured_bacterium"
## [14] "g__Bilophila"
## [15] "g__Parasutterella"
## [16] "g__Ralstonia"
## [17] "g__Escherichia-Shigella"
## [18] "g__Corynebacterium_1"
## [19] "g__Brachybacterium"
## [20] "g__Bifidobacterium"
## [21] "g__Ruminococcaceae_UCG-014"
## [22] "g__Lachnospiraceae_UCG-008"
## [23] "g__CHKC1002"
## [24] "g__Gordonibacter"
## [25] "g__Akkermansia"
## [26] "g__Paraeggerthella"
## [27] "g__Christensenellaceae_R-7_group"
## [28] "g__Ruminococcaceae_UCG-010"
## [29] "g__Tyzzerella_3"
## [30] "g__Romboutsia"
## [31] "g__Family_XIII_AD3011_group"
## [32] "g__Staphylococcus"
## [33] "g__Caldalkalibacillus"
## [34] "g__Lactobacillus"
## [35] "g__Weissella"
## [36] "g__Bacillus"
## [37] "g__Enterococcus"
## [38] "g__Eisenbergiella"
## [39] "g__Streptococcus"
## [40] "g__Phascolarctobacterium"
## [41] "g__uncultured_organism"
## [42] "g__Anaerofustis"
## [43] "g__Ruminococcaceae_UCG-013"
## [44] "g__Butyricicoccus"
## [45] "g__Faecalibacterium"
## [46] "g__Phocea"
## [47] "g__Subdoligranulum"
## [48] "g__GCA-900066225"
## [49] "g__[Eubacterium]_coprostanoligenes_group"
## [50] "g__Negativibacillus"
## [51] "g__Hydrogenoanaerobacterium"
## [52] "g__Angelakisella"
## [53] "g__Fournierella"
## [54] "g__Anaerofilum"
## [55] "g__Anaerotruncus"
## [56] "g__[Ruminococcus]_torques_group"
## [57] "g__Sellimonas"
## [58] "g__Lachnoclostridium"
## [59] "g__GCA-900066575"
## [60] "g__CHKC1001"
## [61] "g__UC5-1-2E3"
## [62] "g__Fusicatenibacter"
## [63] "g__Roseburia"
## [64] "g__Marvinbryantia"
## [65] "g__Coprococcus_3"
## [66] "g__Lachnospiraceae_UCG-006"

```

```

## [67] "g__[Ruminococcus]_gauvreauii_group"
## [68] "g__[Lachnospiraceae_UCG-004"
## [69] "g__[Lachnospiraceae_FCS020_group"
## [70] "g__[Lachnospiraceae_FE2018_group"
## [71] "g__[Eubacterium]_ventriosum_group"
## [72] "g__[Shuttleworthia"
## [73] "g__[Ruminiclostridium_9"
## [74] "g__[Eubacterium]_hallii_group"
## [75] "g__[Anaerostipes"
## [76] "g__[UBA1819"
## [77] "g__[Ruminococcus_1"
## [78] "g__[Defluvititaleaceae_UCG-011"
## [79] "g__[ASF356"
## [80] "g__[Tyzzerella"
## [81] "g__[Eubacterium]_nodatum_group"
## [82] "g__[Ruminiclostridium_5"
## [83] "g__[Ruminiclostridium"
## [84] "g__[Oscillospira"
## [85] "g__[Oscillibacter"
## [86] "g__[Ruminococcus_2"
## [87] "g__[Ruminococcaceae_UCG-009"
## [88] "g__[Ruminococcaceae_UCG-005"
## [89] "g__[DTU089"
## [90] "g__[Candidatus_Soleaferrea"
## [91] "g__[Ruminococcaceae_NK4A214_group"
## [92] "g__[Flavonifractor"
## [93] "g__[Intestinimonas"

```

```

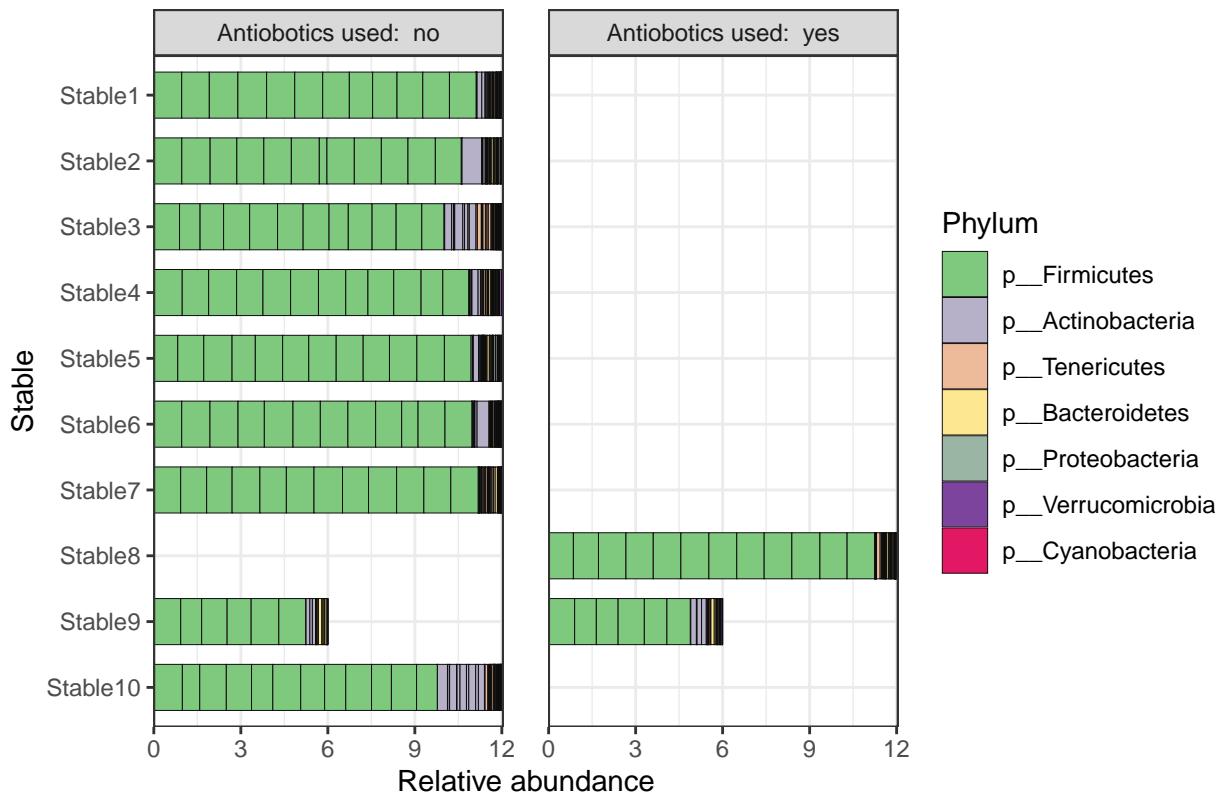
# Plots of relative abundances, fixing some genes that are clustered in the data twice, showing top 12

# Relative abundance for both stable and antibiotics used

subset16S %>%
  ps_arrange(Stables) %>%
  ps_mutate(
    Stables = factor(Stables, rev(unique(Stables)))
  ) %>%
  comp_barplot(
    tax_level = "Phylum", bar_width = 0.7, sample_order = "asis",
    palette = colorRampPalette(brewer.pal(8,"Accent"))(9),
    x = "Stables" +
    facet_wrap(
      facets = vars(AB), labeller = as_labeller(~ paste("Antibiotics used: ", .)),
      scales = "fixed"
    ) +
    coord_flip() +
    labs(x = "Stable", y = "Relative abundance") +
    scale_y_continuous(expand = expansion(add = c(0, 0.05))) +
    theme_bw() +
    theme(panel.spacing.x = unit(6, "mm")) +
    ggtitle("Relative abundance of Phyla by stable and antibiotics used")

```

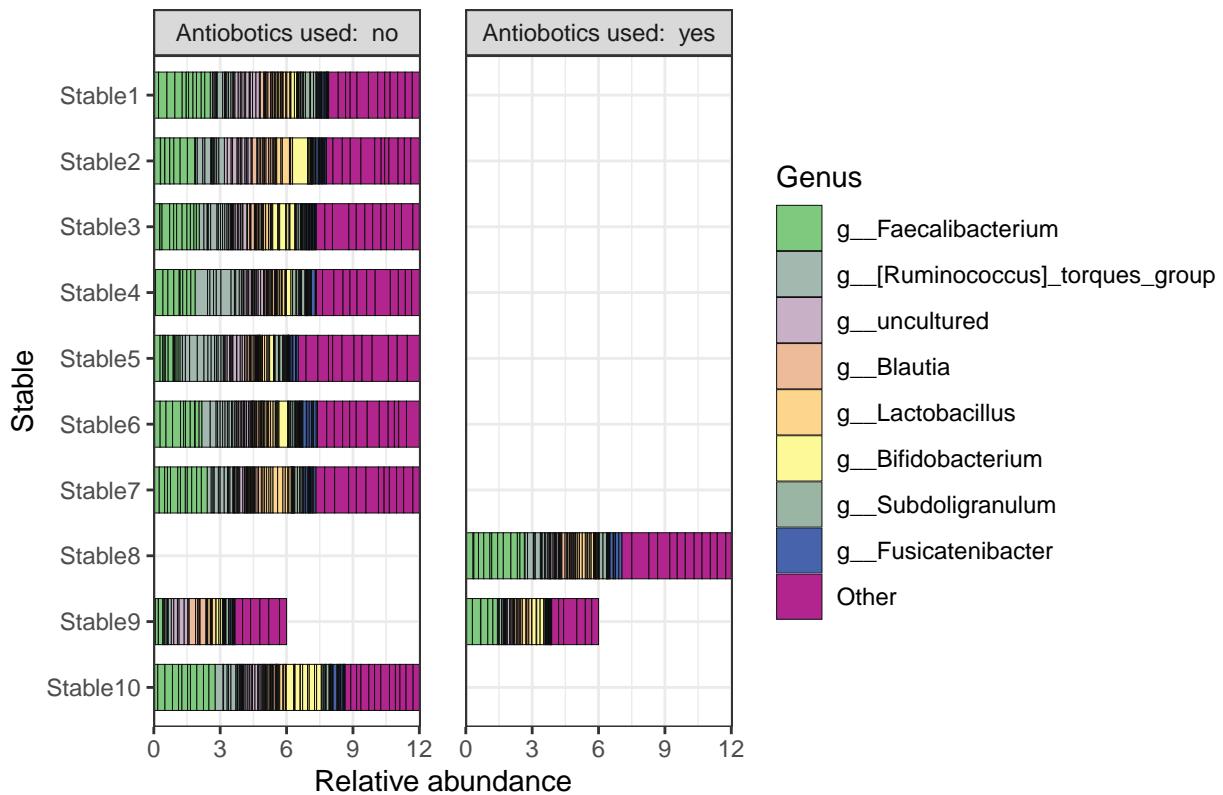
Relative abundance of Phyla by stable and antibiotics used



```
# Same plot but with Genus
```

```
subset16S %>% aggregate_top_taxa2("Genus", top = 8) %>% phyloseq::tax_glom("Genus") %>%
  ps_arrange(Stables) %>%
  ps_mutate(
    Stables = factor(Stables, rev(unique(Stables)))
  ) %>%
  comp_barplot(
    tax_level = "Genus", bar_width = 0.7, sample_order = "asis",
    palette = colorRampPalette(brewer.pal(8,"Accent"))(13),
    x = "Stables",
    n_taxa = 12, other_name = "Other ARG", merge_other = F) +
  facet_wrap(
    facets = vars(AB), labeller = as_labeller(~ paste("Antibiotics used: ", .)),
    scales = "fixed"
  ) +
  coord_flip() +
  labs(x = "Stable", y = "Relative abundance") +
  scale_y_continuous(expand = expansion(add = c(0, 0.05))) +
  theme_bw() +
  theme(panel.spacing.x = unit(6, "mm")) +
  ggtitle("Relative abundance of Genera by stable and antibiotics used")
```

Relative abundance of Genera by stable and antibiotics used



Rel abundance Phyla

```

ps1.com <- subset16S

taxa_names(ps1.com) <- paste0("ASV_", rownames(tax_table(ps1.com)))

# set Palette
taxic <- as.data.frame(ps1.com@tax_table) # this will help in setting large color options
taxic$OTU <- rownames(taxic) # Add the OTU ids from OTU table into the taxa table at the end.
colnames(taxic) # You can see that we now have extra taxonomy levels.

## [1] "Domain" "Phylum" "Class"  "Order"  "Family" "Genus"  "OTU"

taxmat <- as.matrix(taxic) # convert it into a matrix.
new.tax <- tax_table(taxmat) # convert into phyloseq compatible file.
tax_table(ps1.com) <- new.tax # incorporate into phyloseq Object

# now edit the unclassified taxa
tax_table(ps1.com)[tax_table(ps1.com)[, "Phylum"] == "", "Phylum"] <- "Unclassified phylum"

guide_italics <- guides(fill = guide_legend(label.theme = element_text(
  size = 15,

```

```

    face = "italic", colour = "Black", angle = 0
  )))

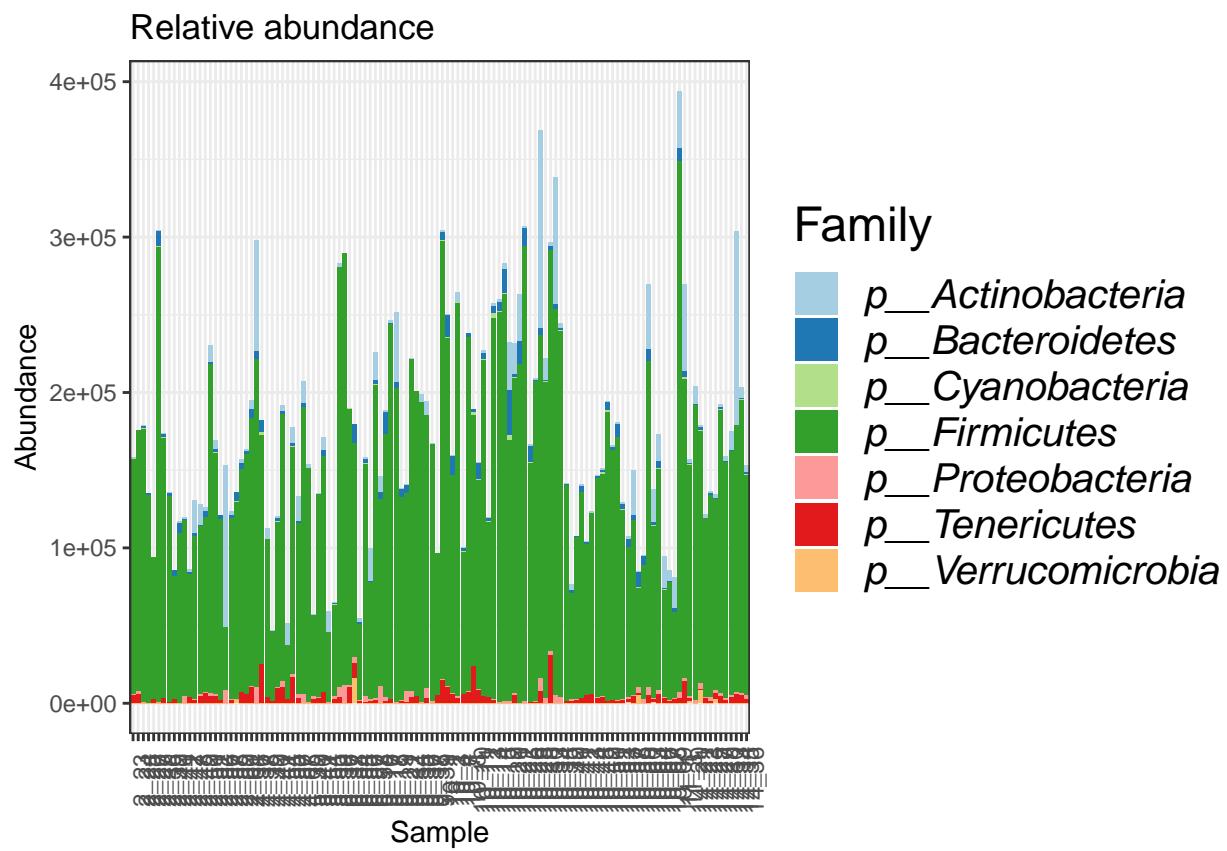
ps1.com@phy_tree <- NULL

# merge at phylum level

ps1.com.fam <- microbiomeutilities::aggregate_top_taxa2(ps1.com, "Phylum", top = 10)

plot_composition(ps1.com.fam) + theme(legend.position = "bottom") +
  scale_fill_brewer("Family", palette = "Paired") + theme_bw() +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Relative abundance") + guide_italics + theme(legend.title = element_text(size = 18))

```



Other Phyla plots

```

tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)

tse <- transformCounts(tse, method = "relabundance")

tse_phylum <- agglomerateByRank(tse,
                                   rank = "Phylum",
                                   onRankOnly = TRUE)
tse_phylum <- transformCounts(tse_phylum,
                               assay.type = "counts",

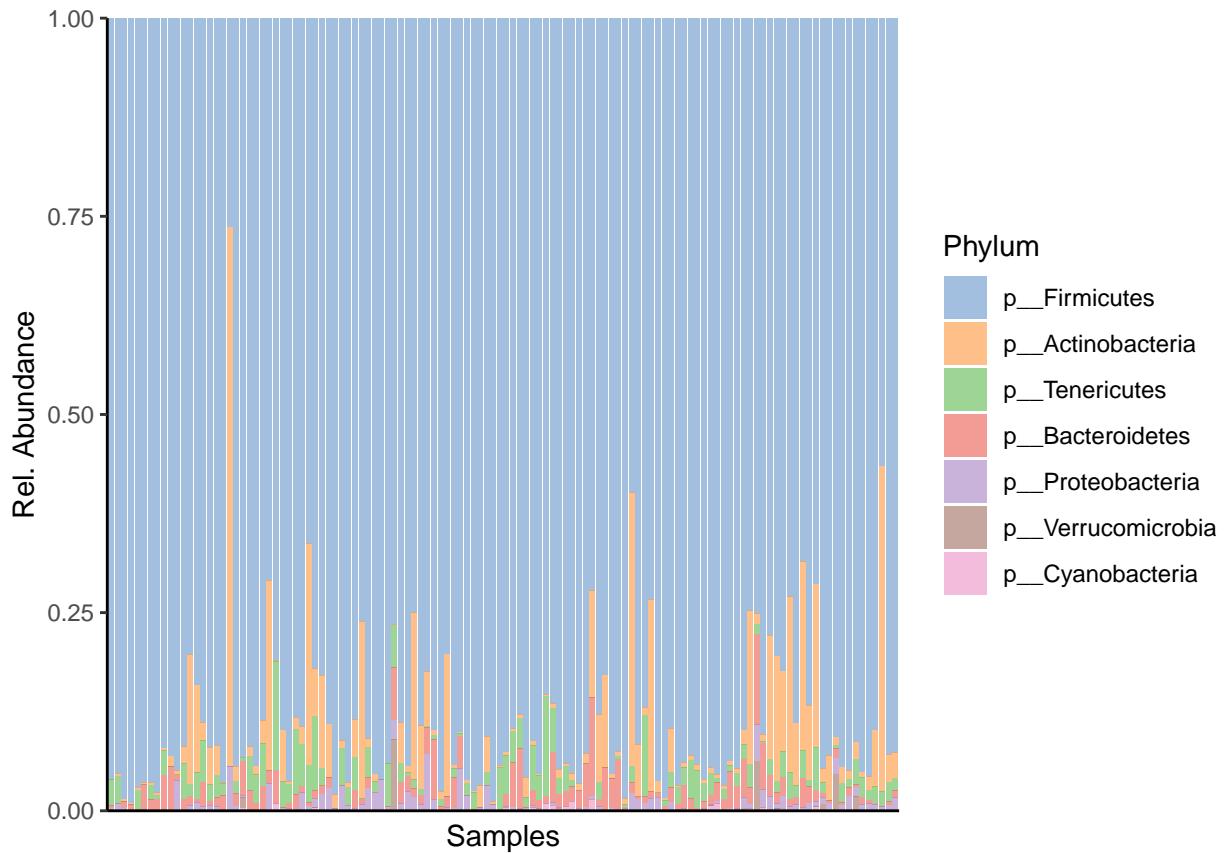
```

```

method = "relabundance")

miaViz::plotAbundance(tse_phylum,
                      assay.type = "relabundance",
                      rank = "Phylum",
                      order_rank_by = "abund")

```



Same plot but with antibiotic treatment added in

```

tse_phylum$Farm2 = as.factor(tse_phylum$Farm2)
tse_phylum$AB = as.factor(tse_phylum$AB)

plots <- miaViz::plotAbundance(tse_phylum,
                                 assay.type = "relabundance",
                                 rank = "Phylum",
                                 order_sample_by = "AB",
                                 features = "AB")

plots[[1]] <- plots[[1]] +
  theme(legend.key.size = unit(0.3, 'cm'),
        legend.text = element_text(size = 6),
        legend.title = element_text(size = 8))
plots[[2]] <- plots[[2]] +
  theme(legend.key.height = unit(0.3, 'cm'),
        legend.key.width = unit(0.3, 'cm'),
        legend.text = element_text(size = 6),
        legend.title = element_text(size = 8))

```

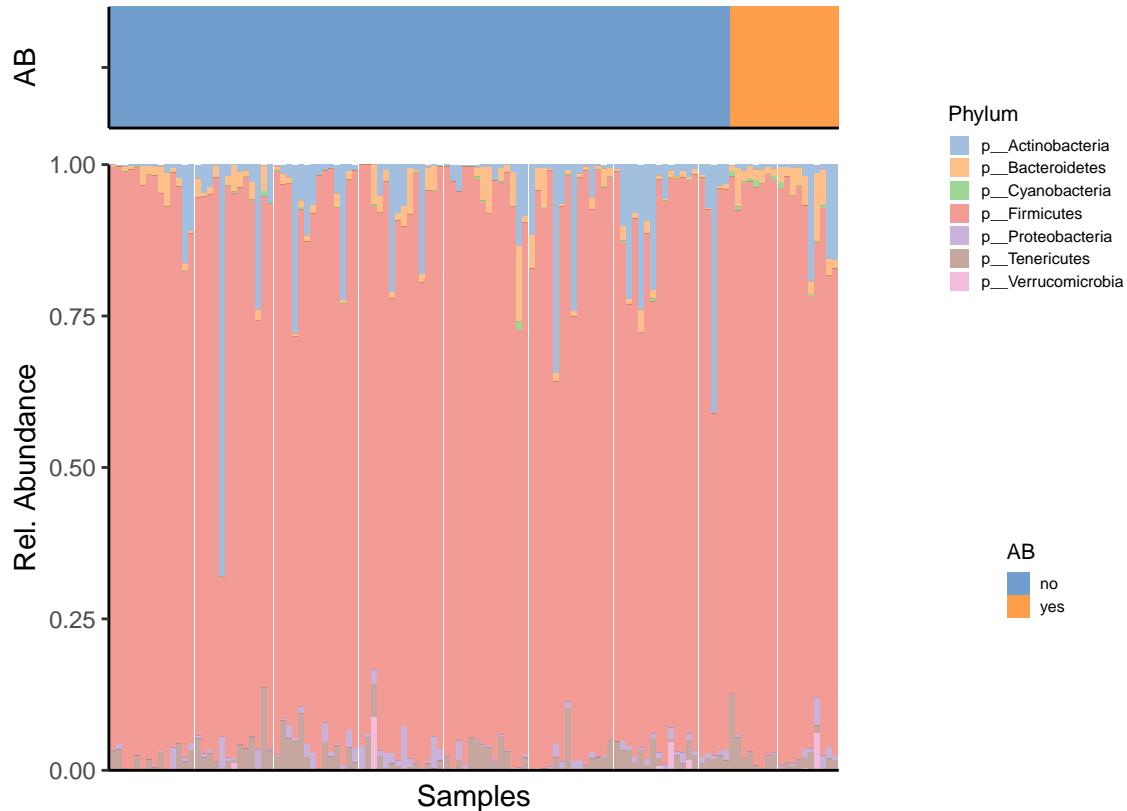
```

    legend.title = element_text(size = 8),
    legend.direction = "vertical")

legend <- wrap_plots(as_ggplot(get_legend(plots[[1]])), as_ggplot(get_legend(plots[[2]])), ncol = 1)
plots[[1]] <- plots[[1]] + theme(legend.position = "none")
plots[[2]] <- plots[[2]] + theme(legend.position = "none", axis.title.x=element_blank())

plot <- wrap_plots(plots[[2]], plots[[1]], ncol = 1, heights = c(2, 10))
wrap_plots(plot, legend, nrow = 1, widths = c(2, 1))

```



```

# heatmaps on phylum level

tse_phylum <- agglomerateByRank(tse,
                                    rank = "Phylum",
                                    onRankOnly = TRUE)

tse_phylum <- transformCounts(tse_phylum, MARGIN = "samples", method = "clr", assay.type = "counts", ps
tse_phylum <- transformCounts(tse_phylum, assay.type = "clr",
                               MARGIN = "features",
                               method = "z", name = "clr_z")

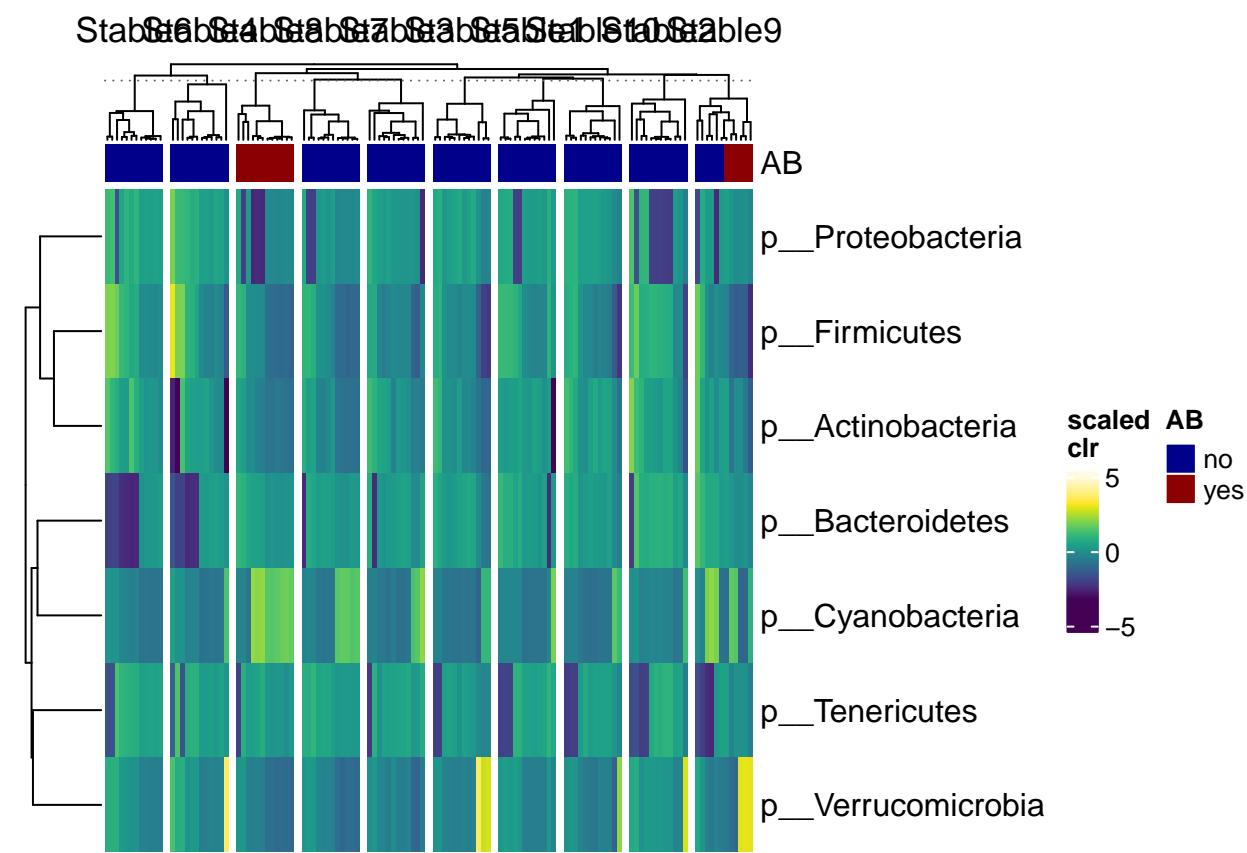
#top_taxa <- getTopTaxa(tse_phylum, top = 20) there are few phyla in this data so no need to exclude some
#tse_phylum <- tse_phylum[top_taxa, ]

# Phylum AB heatmap

```

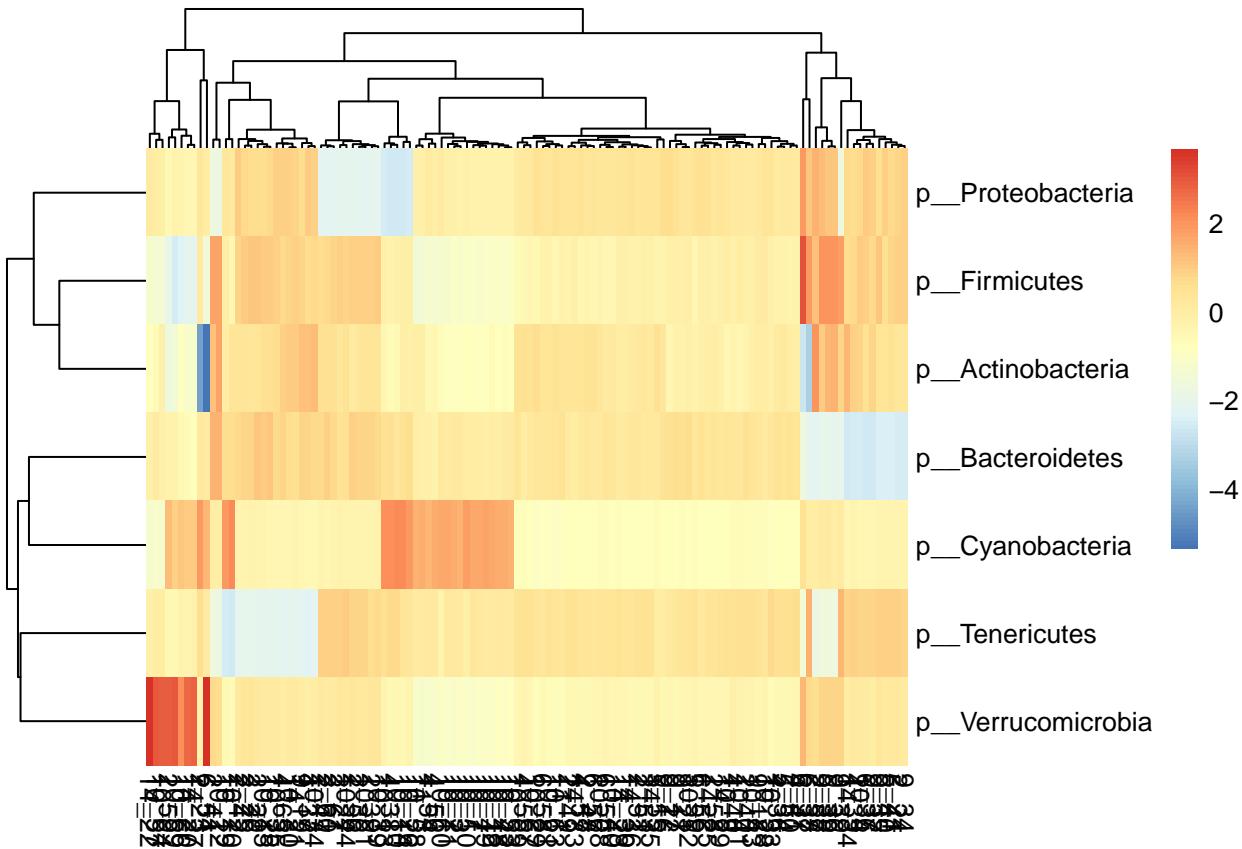
```
tse_phylum@metadata$anno_colors$AB = c(yes = "darkred", no ="darkblue")

sechm(tse_phylum,
      features = rownames(tse_phylum),
      assayName = "clr",
      do.scale = TRUE,
      top_annotation = "AB",
      gaps_at = "Stables",
      hmcols = viridis(256),
      cluster_cols = TRUE, cluster_rows = TRUE,
      sortRowsOn=NULL)
```



```
# Phylum heatmap
mat <- assay(tse_phylum, "clr_z")

pheatmap(mat)
```



Phylum heatmap hierachal clustering with AB Clustering both samples and features hierarchically

```

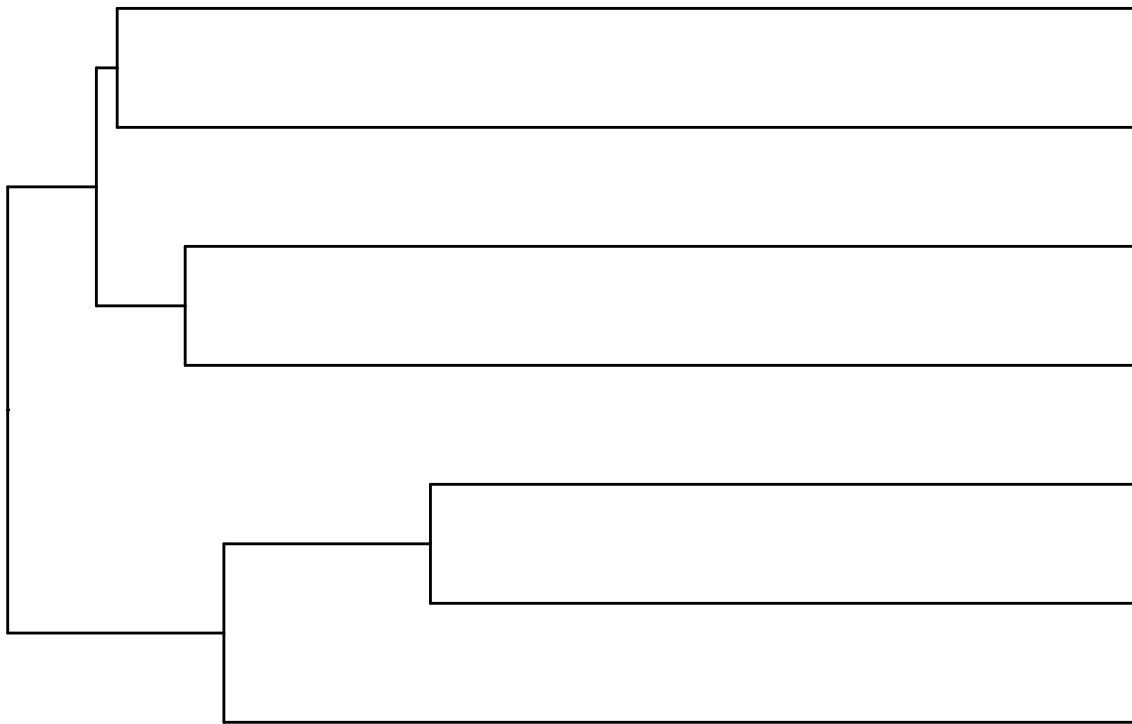
taxa_hclust <- hclust(dist(mat), method = "complete")

# Creates a phylogenetic tree
taxa_tree <- as.phylo(taxa_hclust)

# Plot taxa tree
taxa_tree <- ggtree(taxa_tree) +
  theme(plot.margin=margin(0,0,0,0)) # removes margins

taxa_tree # based on this three, we'll create two clusters

```



```

# Get order of taxa in plot
taxa_ordered <- get_taxa_name(taxa_tree)

taxa_clusters <- cutree(tree = taxa_hclust, k = 2) # 2 clusters based on tree figure

# Converts into data frame
taxa_clusters <- data.frame(clusters = taxa_clusters)
taxa_clusters$clusters <- factor(taxa_clusters$clusters)

# Order data so that it's same as in phylo tree
taxa_clusters <- taxa_clusters[taxa_ordered, , drop = FALSE]

# Prints taxa and their clusters
taxa_clusters

##           clusters
## p__Verrucomicrobia      1
## p__Tenericutes          1
## p__Cyanobacteria        1
## p__Bacteroidetes         1
## p__Actinobacteria        2
## p__Firmicutes            2
## p__Proteobacteria         2

```

```

rowData(tse_phylum)$clusters <- taxa_clusters[order(match(rownames(taxa_clusters), rownames(tse_phylum)))]  

# Prints taxa and their clusters  

rowData(tse_phylum)$clusters  
  

## [1] 1 2 1 2 2 1 1  

## Levels: 1 2  
  

sample_hclust <- hclust(dist(t(mat)), method = "complete")  

# Creates a phylogenetic tree  

sample_tree <- as.phylo(sample_hclust)  

# Plot sample tree  

sample_tree <- ggtree(sample_tree) + layout_dendrogram() +  

  theme(plot.margin=margin(0,0,0,0)) # removes margins  

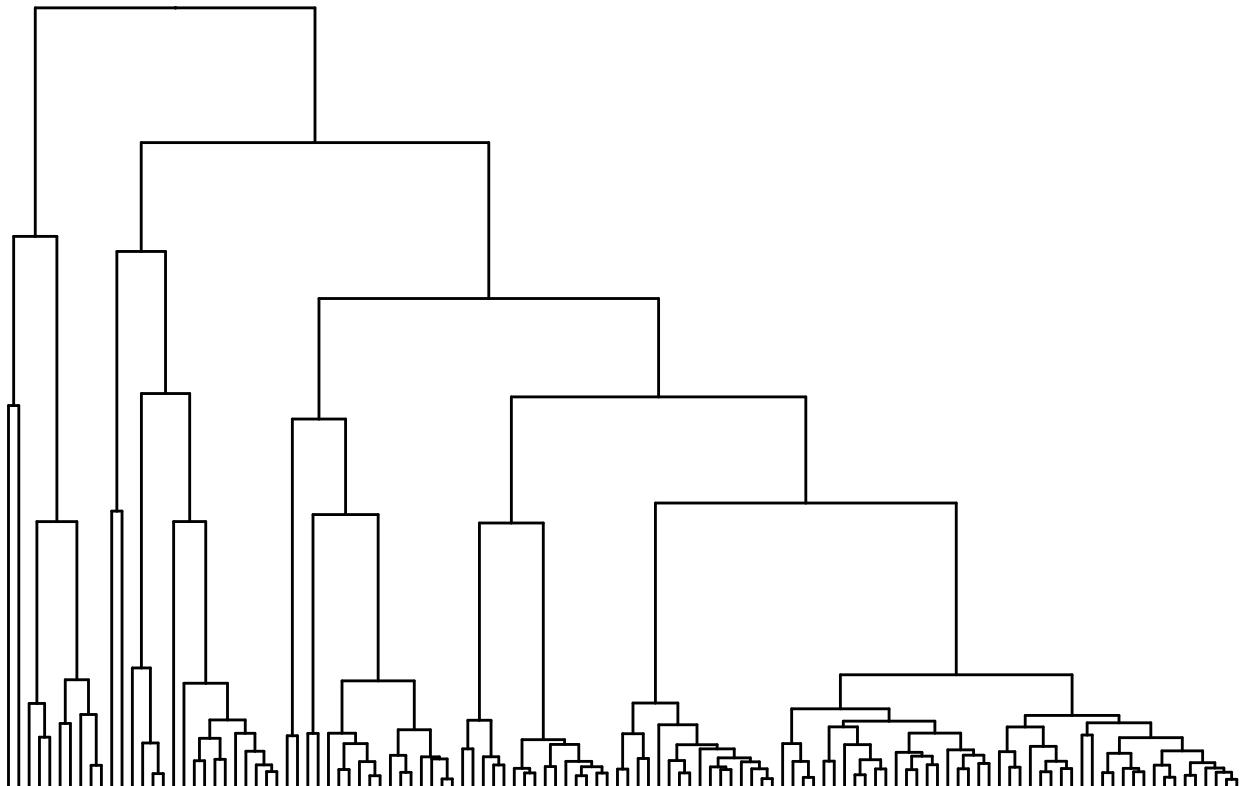
# Get order of samples in plot  

samples_ordered <- rev(get_taxa_name(sample_tree))  

# to view the tree, run  

sample_tree

```



```

# Creates clusters
sample_clusters <- factor(cutree(tree = sample_hclust, k = 2)) # 2 clusters based on methods in Cluster

# Converts into data frame
sample_data <- data.frame(clusters = sample_clusters)

# Order data so that it's same as in phylo tree
sample_data <- sample_data[samples_ordered, , drop = FALSE]

# Order data based on
tse_phylum <- tse_phylum[ , rownames(sample_data)]

# Add sample type data
sample_data$sample_types <- colData(tse_phylum)$AB

sample_data

```

	clusters	sample_types
## 2_23	1	no
## 6_54	1	no
## 14_22	1	no
## 10_57	1	yes
## 10_64	1	yes
## 2_59	1	no
## 10_59	1	yes
## 10_66	1	no
## 14_20	1	no
## 14_27	1	no
## 6_37	2	no
## 6_38	2	no
## 2_57	2	no
## 9_38	2	no
## 5_39	2	no
## 9_36	2	no
## 9_39	2	no
## 14_34	2	no
## 4_39	2	no
## 10_4	2	no
## 6_36	2	no
## 9_37	2	no
## 9_35	2	no
## 5_40	2	no
## 5_41	2	no
## 2_24	2	no
## 9_34	2	no
## 2_41	2	no
## 10_22	2	no
## 10_19	2	no
## 10_20	2	no
## 2_42	2	no
## 2_25	2	no
## 2_26	2	no
## 2_29	2	no

## 10_15	2	yes
## 10_25	2	no
## 9_19	2	no
## 4_37	2	no
## 10_34	2	no
## 10_35	2	no
## 4_65	2	no
## 10_30	2	no
## 14_21	2	no
## 10_14	2	yes
## 10_26	2	no
## 10_13	2	yes
## 4_38	2	no
## 10_10	2	yes
## 2_47	2	no
## 2_60	2	no
## 2_61	2	no
## 2_27	2	no
## 10_44	2	no
## 2_56	2	no
## 2_36	2	no
## 10_51	2	yes
## 2_39	2	no
## 10_39	2	no
## 10_58	2	yes
## 11_3	2	no
## 4_54	2	no
## 4_56	2	no
## 10_60	2	yes
## 10_11	2	yes
## 10_2	2	no
## 10_3	2	no
## 10_50	2	yes
## 10_1	2	no
## 10_7	2	no
## 10_49	2	yes
## 10_12	2	yes
## 10_48	2	yes
## 10_42	2	no
## 10_53	2	yes
## 9_22	2	no
## 9_17	2	no
## 9_21	2	no
## 10_52	2	yes
## 9_18	2	no
## 10_28	2	no
## 10_33	2	no
## 4_36	2	no
## 5_54	2	no
## 2_50	2	no
## 4_41	2	no
## 5_59	2	no
## 6_56	2	no
## 14_25	2	no

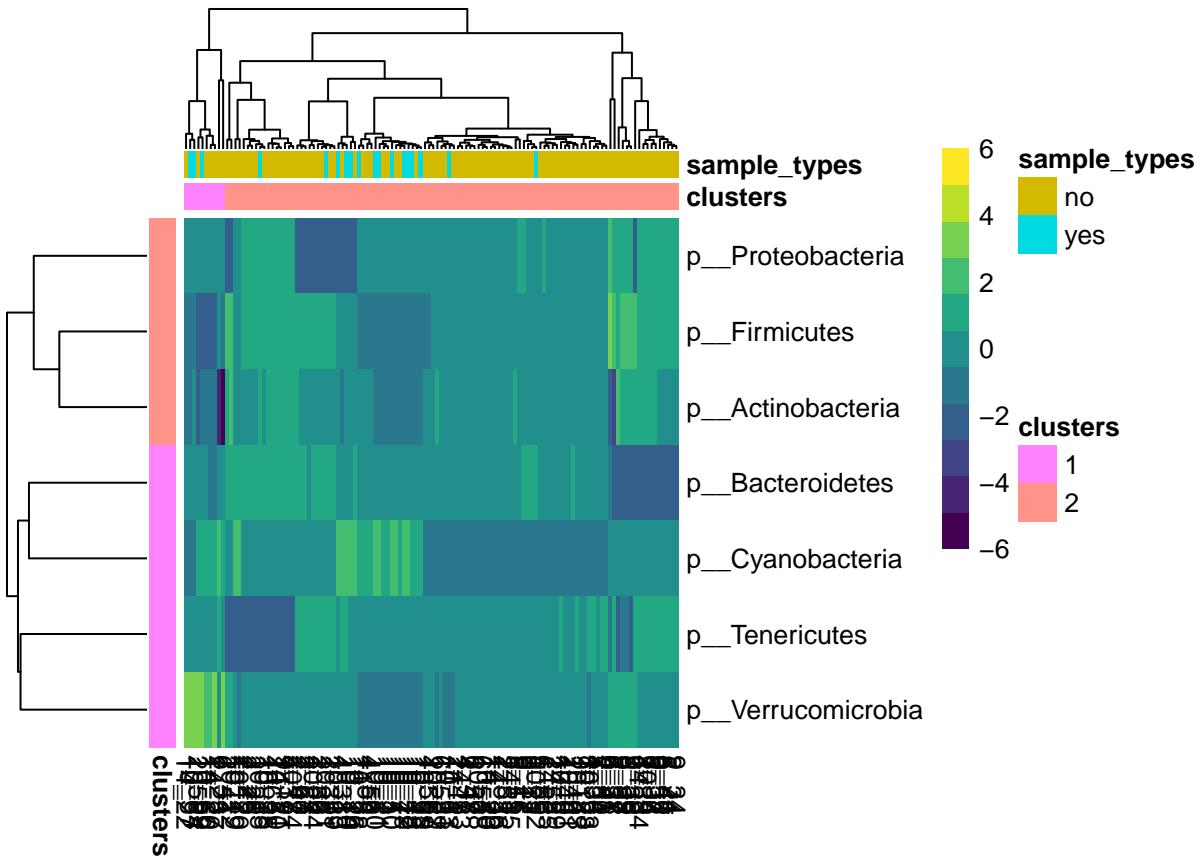
```

## 2_58      2      no
## 14_29     2      no
## 10_8      2      no
## 4_40      2      no
## 10_41     2      no
## 2_40      2      no
## 10_43     2      no
## 10_69     2      no
## 4_55      2      no
## 10_67     2      no
## 10_29     2      no
## 6_58      2      no
## 10_21     2      no
## 10_63     2      yes
## 11_1      2      no
## 5_55      2      no
## 9_16      2      no
## 2_49      2      no
## 14_33     2      no
## 4_57      2      no
## 2_48      2      no
## 6_57      2      no
## 10_68     2      no
## 6_55      2      no
## 10_40     2      no
## 14_23     2      no
## 14_30     2      no
## 14_36     2      no
## 2_52      2      no
## 2_51      2      no
## 14_35     2      no

breaks <- seq(-ceiling(max(abs(mat))), ceiling(max(abs(mat))),
               length.out = ifelse( max(abs(mat))>5, 2*ceiling(max(abs(mat))), 10 ) )
#colors <- colorRampPalette(c("darkblue", "blue", "white", "red", "darkred"))(length(breaks)-1) replace

pheatmap(mat, annotation_row = taxa_clusters,
         annotation_col = sample_data,
         breaks = breaks,
         color = colorRampPalette(viridis(256))(length(breaks)-1))

```

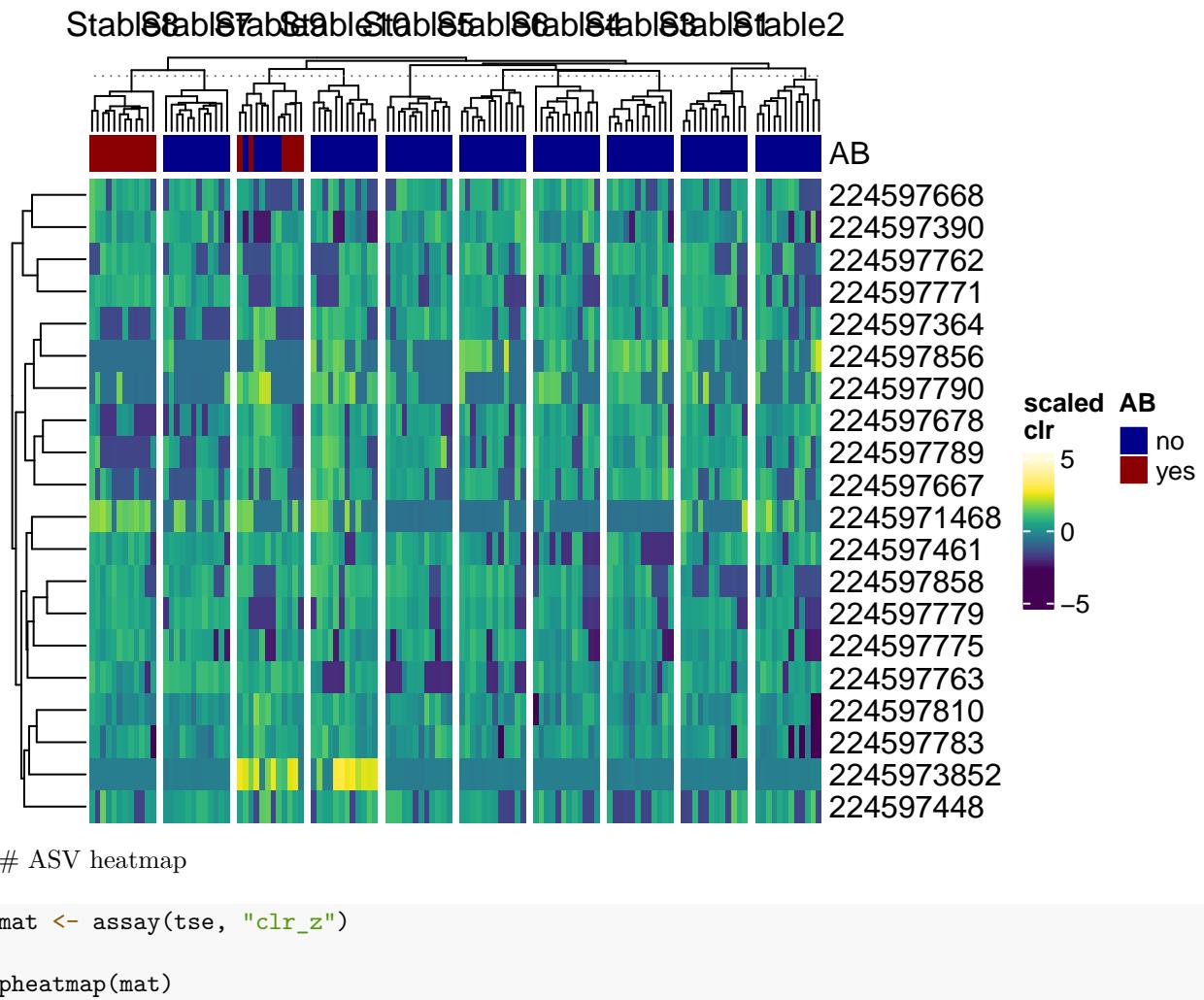


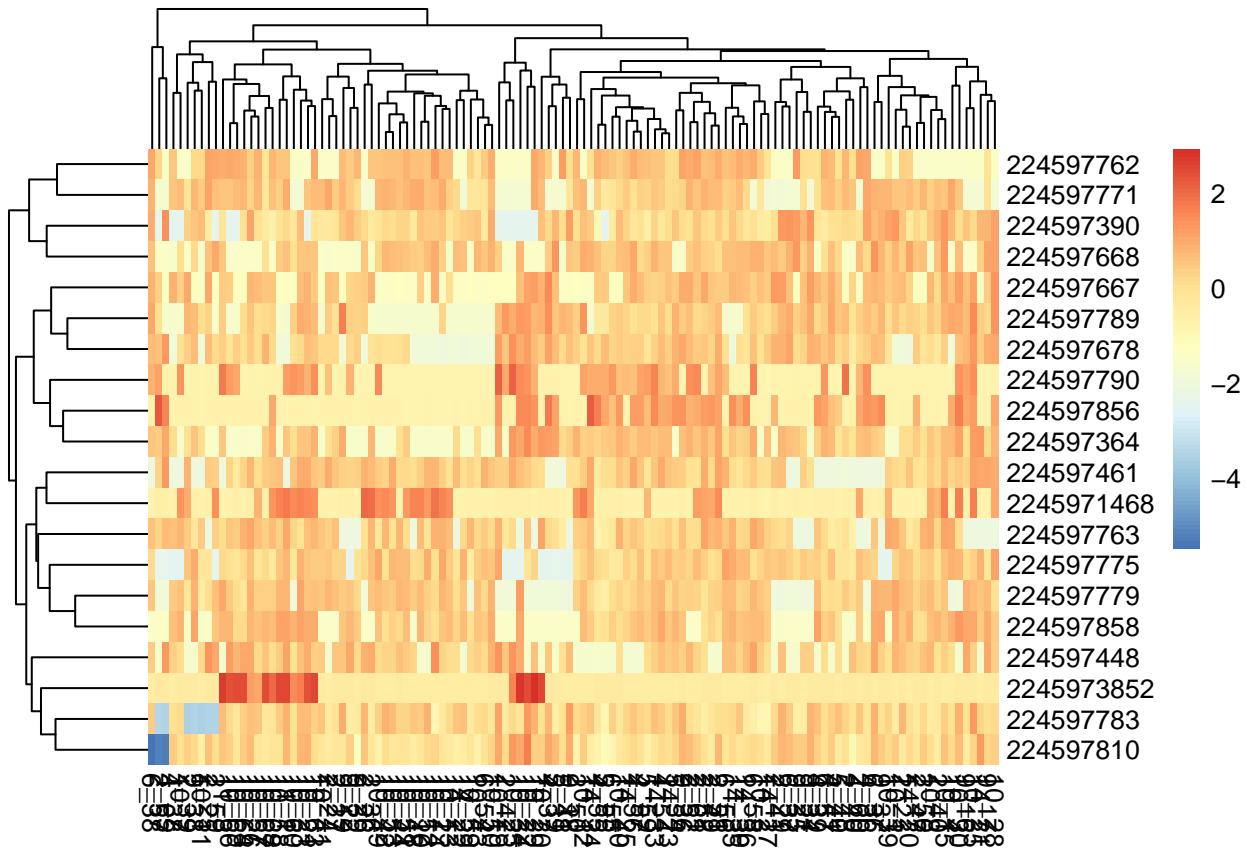
heatmaps on ASV level

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)
tse <- transformCounts(tse, method = "relabundance")
tse <- transformCounts(tse, MARGIN = "samples", method = "clr", assay.type = "counts", pseudocount=1) #
tse <- transformCounts(tse, assay.type = "clr",
                      MARGIN = "features",
                      method = "z", name = "clr_z")
top_taxa <- getTopTaxa(tse, top = 20)
tse <- tse[top_taxa, ]

# ASV heatmap AB
tse@metadata$anno_colors$AB = c(yes = "darkred", no = "darkblue")

sechm(tse,
       features = rownames(tse),
       assayName = "clr",
       do.scale = TRUE,
       top_annotation = c("AB"),
       gaps_at = "Stables",
       hmcols = viridis(256),
       cluster_cols = TRUE, cluster_rows = TRUE)
```





```

# ASV heatmap hierachal clustering with AB

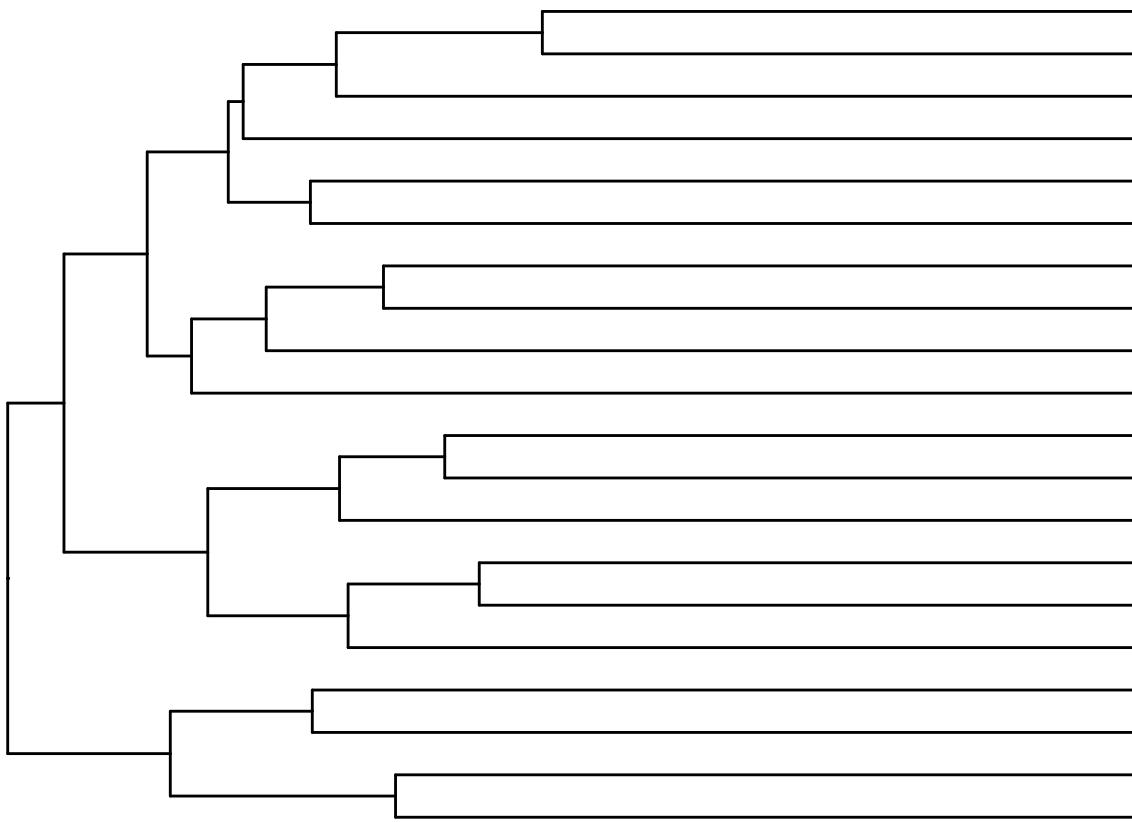
# Clustering both samples and features hierarchically

taxa_hclust <- hclust(dist(mat), method = "complete")

# Creates a phylogenetic tree
taxa_tree <- as.phylo(taxa_hclust)

# Plot taxa tree
taxa_tree <- ggtree(taxa_tree) +
  theme(plot.margin=margin(0,0,0,0)) # removes margins

taxa_tree # based on this three, we'll create two clusters
  
```



```

# Get order of taxa in plot
taxa_ordered <- get_taxa_name(taxa_tree)

taxa_clusters <- cutree(tree = taxa_hclust, k = 2) # 2 clusters based on methods in Clustering.R script

# Converts into data frame
taxa_clusters <- data.frame(clusters = taxa_clusters)
taxa_clusters$clusters <- factor(taxa_clusters$clusters)

# Order data so that it's same as in phylo tree
taxa_clusters <- taxa_clusters[taxa_ordered, , drop = FALSE]

# Prints taxa and their clusters
taxa_clusters

##           clusters
## 224597858      1
## 224597779      1
## 224597775      1
## 224597763      1
## 2245971468     1
## 224597461      1
## 224597810      1
## 224597783      1
## 2245973852     1
## 224597448      1

```

```

## 224597364      1
## 224597856      1
## 224597790      1
## 224597678      1
## 224597789      1
## 224597667      1
## 224597668      2
## 224597390      2
## 224597771      2
## 224597762      2

# Prints taxa and their clusters
rowData(tse_phylum)$clusters

## [1] 1 2 1 2 2 1 1
## Levels: 1 2

sample_hclust <- hclust(dist(t(mat)), method = "complete")

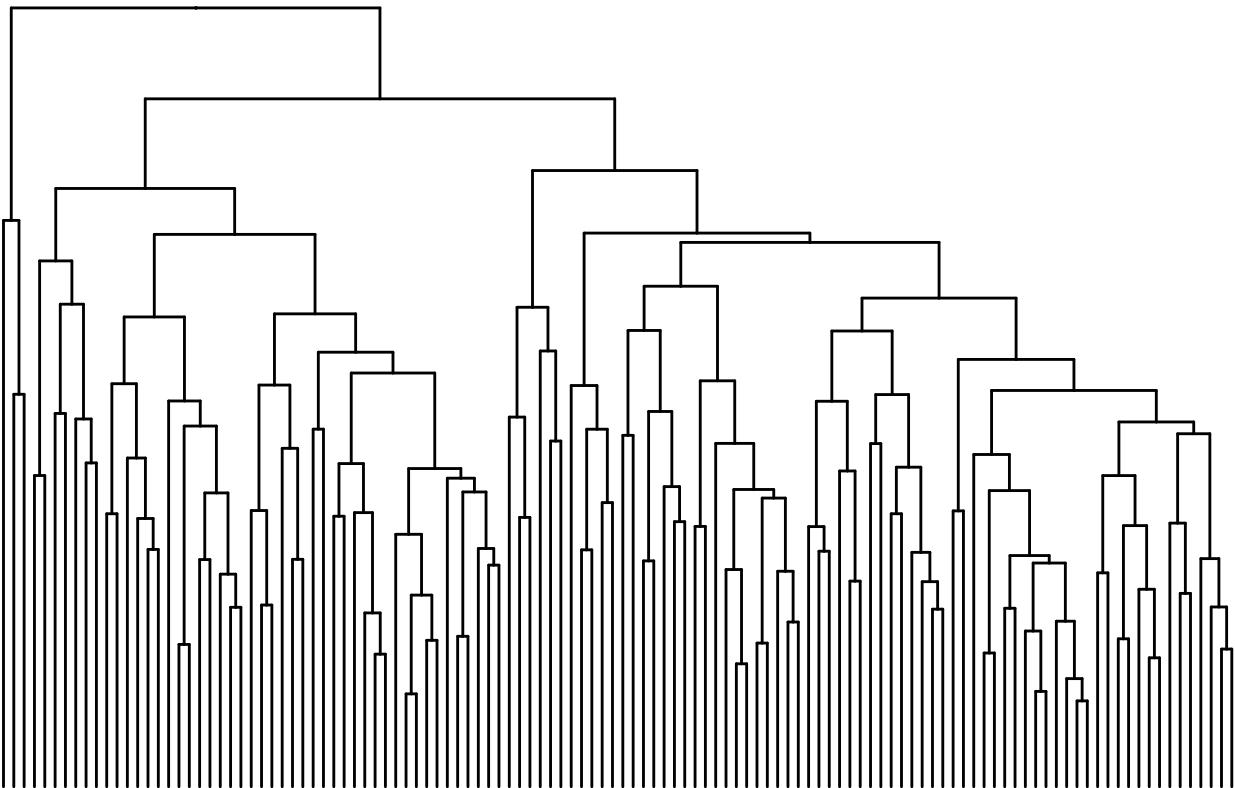
# Creates a phylogenetic tree
sample_tree <- as.phylo(sample_hclust)

# Plot sample tree
sample_tree <- ggtree(sample_tree) + layout_dendrogram() +
  theme(plot.margin=margin(0,0,0,0)) # removes margins

# Get order of samples in plot
samples_ordered <- rev(get_taxa_name(sample_tree))

# to view the tree, run
sample_tree

```



```

# Creates clusters
sample_clusters <- factor(cutree(tree = sample_hclust, k = 2))

# Converts into data frame
sample_data <- data.frame(clusters = sample_clusters)

# Order data so that it's same as in phylo tree
sample_data <- sample_data[samples_ordered, , drop = FALSE]

# Order data based on
tse_phylum <- tse_phylum[ , rownames(sample_data)]

# Add sample type data
sample_data$sample_types <- colData(tse_phylum)$AB

sample_data

##      clusters sample_types
## 6_38          2        no
## 2_57          2        no
## 4_65          2        no
## 4_38          1        no
## 10_7           1        no
## 2_25          1        no
## 2_59          1        no
## 2_39          1        no

```

## 9_38	1	no
## 10_11	1	yes
## 10_58	1	yes
## 11_1	1	no
## 10_60	1	yes
## 10_33	1	no
## 10_21	1	no
## 10_63	1	yes
## 11_3	1	no
## 10_66	1	no
## 10_69	1	no
## 10_67	1	no
## 10_68	1	no
## 10_64	1	yes
## 10_26	1	no
## 10_57	1	yes
## 10_41	1	no
## 2_24	1	no
## 2_27	1	no
## 9_19	1	no
## 9_22	1	no
## 9_39	1	no
## 2_26	1	no
## 2_36	1	no
## 14_22	1	no
## 14_29	1	no
## 10_59	1	yes
## 10_43	1	no
## 6_54	1	no
## 10_50	1	yes
## 10_49	1	yes
## 10_42	1	no
## 10_53	1	yes
## 10_44	1	no
## 10_51	1	yes
## 10_48	1	yes
## 10_10	1	yes
## 10_52	1	yes
## 10_14	1	yes
## 10_2	1	no
## 10_13	1	yes
## 10_19	1	no
## 2_42	1	no
## 10_25	1	no
## 10_34	1	no
## 10_22	1	no
## 10_29	1	no
## 10_20	1	no
## 4_37	1	no
## 5_39	1	no
## 5_41	1	no
## 9_18	1	no
## 10_4	1	no
## 10_30	1	no

## 9_16	1	no
## 10_35	1	no
## 14_21	1	no
## 9_17	1	no
## 10_28	1	no
## 6_36	1	no
## 9_37	1	no
## 10_39	1	no
## 10_1	1	no
## 2_23	1	no
## 14_20	1	no
## 2_47	1	no
## 2_56	1	no
## 10_8	1	no
## 2_40	1	no
## 10_15	1	yes
## 2_41	1	no
## 2_29	1	no
## 6_37	1	no
## 9_35	1	no
## 9_21	1	no
## 9_34	1	no
## 2_60	1	no
## 4_36	1	no
## 2_48	1	no
## 4_40	1	no
## 4_39	1	no
## 6_57	1	no
## 4_41	1	no
## 5_40	1	no
## 2_50	1	no
## 10_12	1	yes
## 14_34	1	no
## 4_55	1	no
## 5_55	1	no
## 6_55	1	no
## 10_40	1	no
## 14_25	1	no
## 4_56	1	no
## 4_57	1	no
## 5_59	1	no
## 14_33	1	no
## 4_54	1	no
## 14_23	1	no
## 5_54	1	no
## 9_36	1	no
## 2_52	1	no
## 2_61	1	no
## 2_58	1	no
## 2_49	1	no
## 2_51	1	no
## 6_58	1	no
## 10_3	1	no
## 14_27	1	no

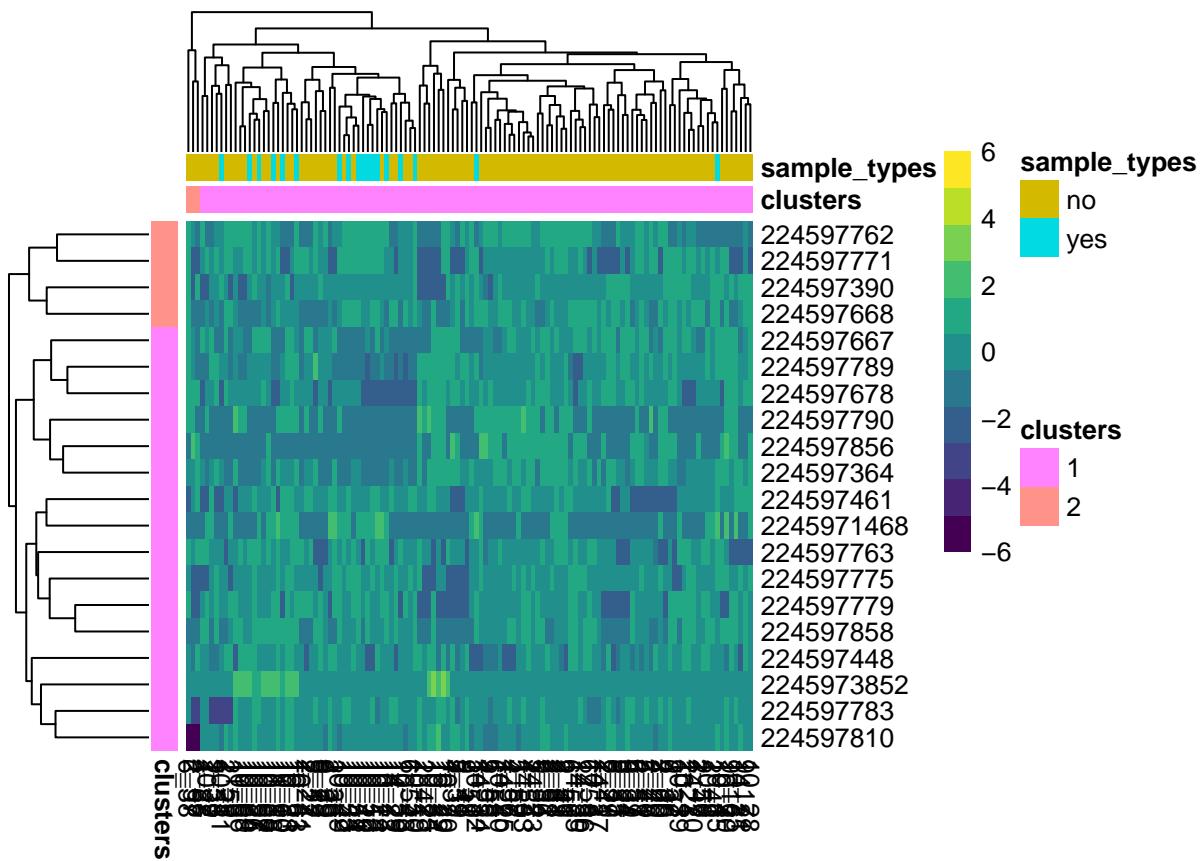
```

## 6_56      1      no
## 14_35     1      no
## 14_30     1      no
## 14_36     1      no

breaks <- seq(-ceiling(max(abs(mat))), ceiling(max(abs(mat))),
               length.out = ifelse( max(abs(mat))>5, 2*ceiling(max(abs(mat))), 10 ) )

pheatmap(mat, annotation_row = taxa_clusters,
          annotation_col = sample_data,
          breaks = breaks,
          color = colorRampPalette(viridis(256))(length(breaks)-1))

```

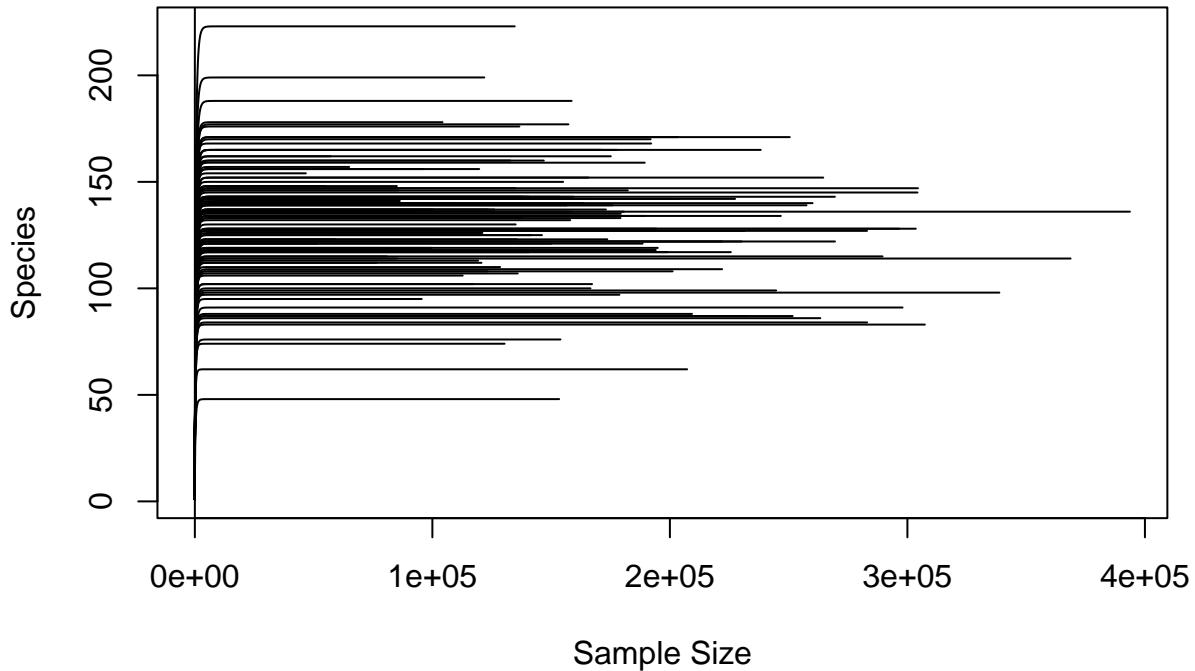


Alpha diversity

```

otu_tab <- t(abundances(subset16S))
# rarefaction curve
vegan::rarecurve(otu_tab,
                  step = 50, label = FALSE,
                  sample = min(rowSums(otu_tab),
                               col = "blue", cex = 0.6))

```



```
# samples plateau so sufficient sequencing depth

summary(goods(otu_tab)) # there are no singletons in this data, already filtered out, means that richness

##      no.sing      no.seqs         goods
##  Min.   :0   Min.   :46731   Min.   :100
##  1st Qu.:0   1st Qu.:125421  1st Qu.:100
##  Median :0   Median :161649  Median :100
##  Mean   :0   Mean   :173742  Mean   :100
##  3rd Qu.:0   3rd Qu.:212488  3rd Qu.:100
##  Max.   :0   Max.   :393697  Max.   :100

#rarefy to equal library size or not?
lib.div <- microbiome::alpha(subset16S, index = "all")
lib.div2 <- richness(subset16S)
lib.div$ReadsPerSample <- sample_sums(subset16S)
lib.div$Observed <- lib.div2$observed
colnames(lib.div)

## [1] "observed"                  "chao1"
## [3] "diversity_inverse_simpson" "diversity_gini_simpson"
## [5] "diversity_shannon"          "diversity_fisher"
## [7] "diversity_coverage"         "evenness_camargo"
## [9] "evenness_pielou"            "evenness_simpson"
## [11] "evenness_evar"              "evenness_bulla"
```

```

## [13] "dominance_dbp"
## [15] "dominance_absolute"
## [17] "dominance_simpson"
## [19] "dominance_gini"
## [21] "rarity_low_abundance"
## [23] "ReadsPerSample"           "Observed"

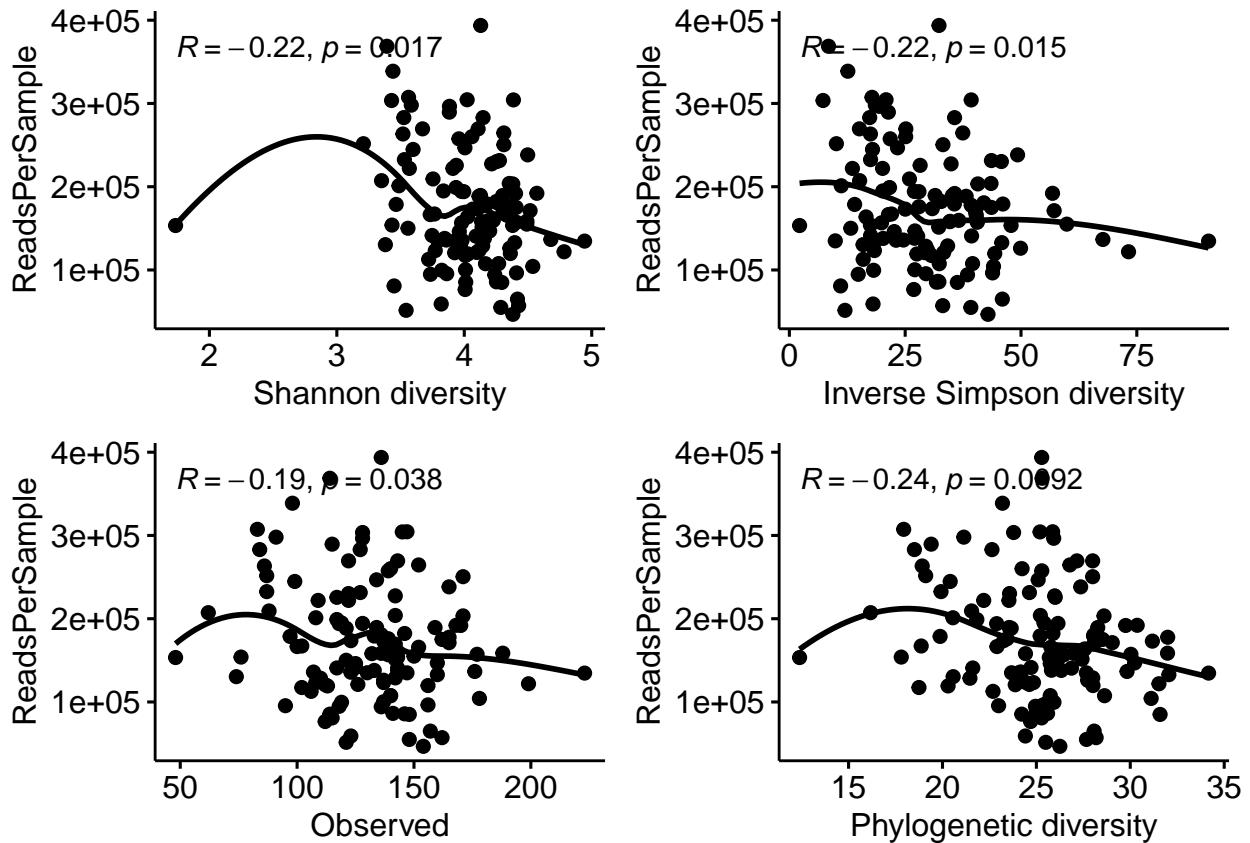
p1 = ggscatter(lib.div, "diversity_shannon", "ReadsPerSample", xlab = "Shannon diversity", add = "loess",
  stat_cor(method = "pearson")
p2 = ggscatter(lib.div, "diversity_inverse_simpson", "ReadsPerSample", xlab = "Inverse Simpson diversity",
  stat_cor(method = "pearson")
p3 = ggscatter(lib.div, "observed", "ReadsPerSample", xlab = "Observed", add = "loess") +
  stat_cor(method = "pearson")

df.pd <- pd(t(as.data.frame(subset16S@otu_table)), subset16S@phy_tree, include.root=T) # transposing for
lib.div$Phylogenetic_Diversity <- df.pd$PD

p4 = ggscatter(lib.div, "Phylogenetic_Diversity", "ReadsPerSample", xlab = "Phylogenetic diversity", add = "loess",
  stat_cor(method = "pearson"))

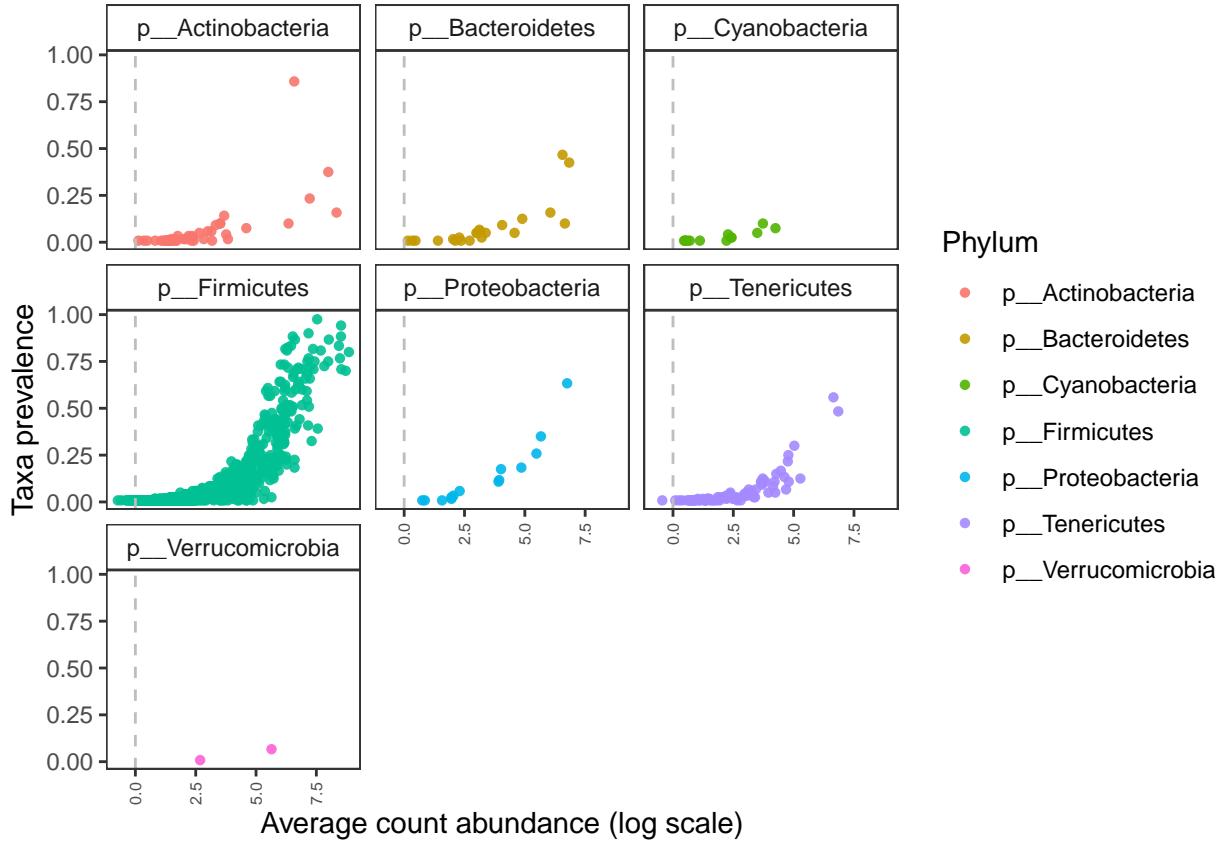
ggarrange(p1, p2, p3, p4, ncol = 2, nrow = 2)

```



removal of samples with lower sequencing depth not necessary for 16S dataset

```
plot_taxa_prevalence(subset16S, "Phylum") # taxa prevalence plot
```



Diversity metrics boxplots

```

hmp.div <- microbiome::alpha(subset16S, index = "all")
hmp.meta <- meta(subset16S)
hmp.meta$sam_name <- rownames(hmp.meta)
hmp.div$sam_name <- rownames(hmp.div)
div.df <- merge(hmp.div, hmp.meta, by = "sam_name")
colnames(div.df)

```

```

## [1] "sam_name"                      "observed"
## [3] "chao1"                          "diversity_inverse_simpson"
## [5] "diversity_gini_simpson"        "diversity_shannon"
## [7] "diversity_fisher"                "diversity_coverage"
## [9] "evenness_camargo"               "evenness_pielou"
## [11] "evenness_simpson"                "evenness_evar"
## [13] "evenness_bulla"                  "dominance_dbp"
## [15] "dominance_dmn"                  "dominance_absolute"
## [17] "dominance_relative"              "dominance_simpson"
## [19] "dominance_core_abundance"       "dominance_gini"
## [21] "rarity_log_modulo_skewness"    "rarity_low_abundance"
## [23] "rarity_rare_abundance"          "LibraryNumber"
## [25] "Sample_Unique"                  "LibraryName"
## [27] "Farm"                           "Farm2"
## [29] "Stable"                         "FarmRoundStable"
## [31] "Days"                            "Age"
## [33] "Sname"                           "WeightAnimal"
## [35] "Gender"                          "AgeParentStock"

```

```

## [37] "Hatchery"                      "Researcher"
## [39] "AB"                            "Abday"
## [41] "FlockSize"                     "FeedF"
## [43] "FeedType"                      "FeedProducent"
## [45] "Cox"                           "OPG"
## [47] "Cluster"                       "LitterType"
## [49] "Metagenomics"                  "Stables"

#based on microbial agent
div.df2 <- div.df[, c("Cox", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon")]
colnames(div.df2) <- c("Cox", "Inverse Simpson", "Gini-Simpson", "Shannon", "Observed", "Coverage", "Pi")

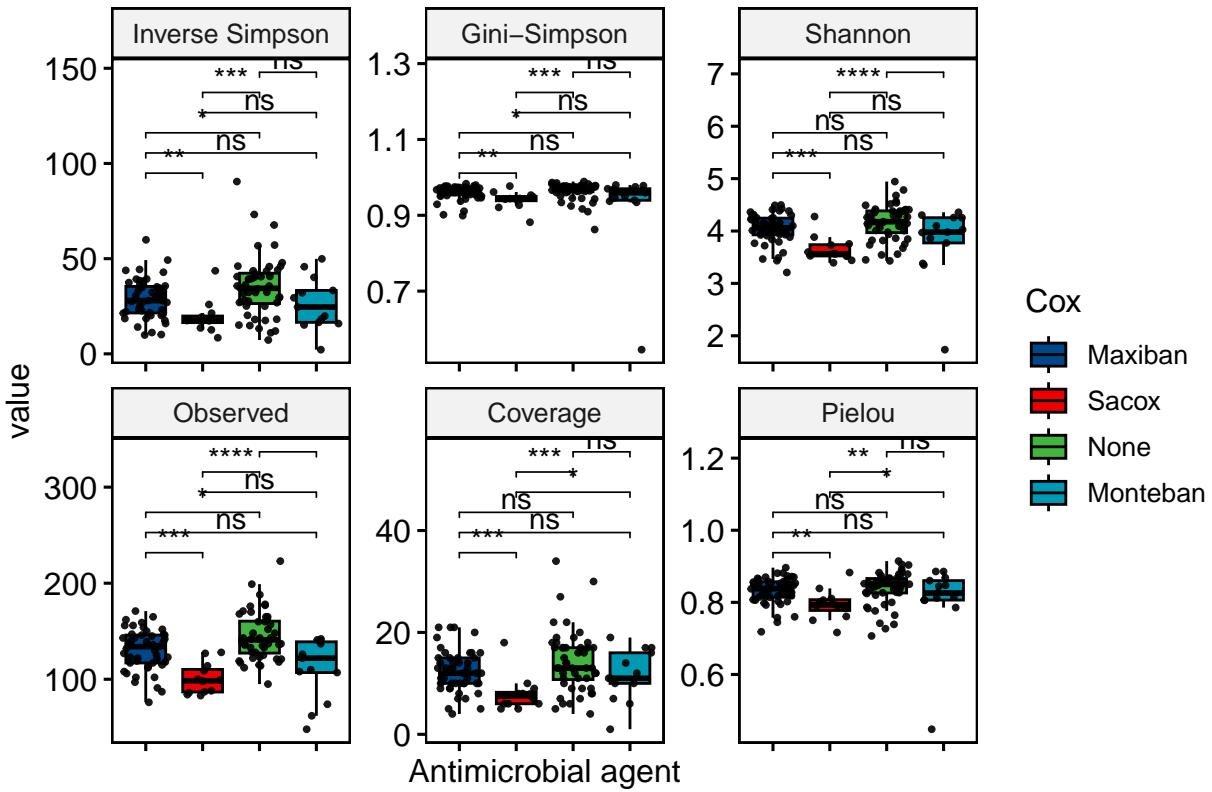
div_df_melt <- reshape2::melt(div.df2)

lev = c("Maxiban", "Sacox", "Monteban", "None")
L.pairs <- combn(seq_along(lev), 2, simplify = FALSE, FUN = function(i) lev[i])

ggboxplot(div_df_melt, x = "Cox", y = "value",
          fill = "Cox",
          palette = "lancet",
          legend= "right",
          facet.by = "variable",
          scales = "free",
          xlab = "Antimicrobial agent",
          title = "Alpha diversity metrics by microbial agent",
          outlier.shape = NA) +
  rremove("x.text") + stat_compare_means(
    comparisons = L.pairs,
    label = "p.signif"
  ) + geom_jitter(size = 0.7, alpha = 0.9)

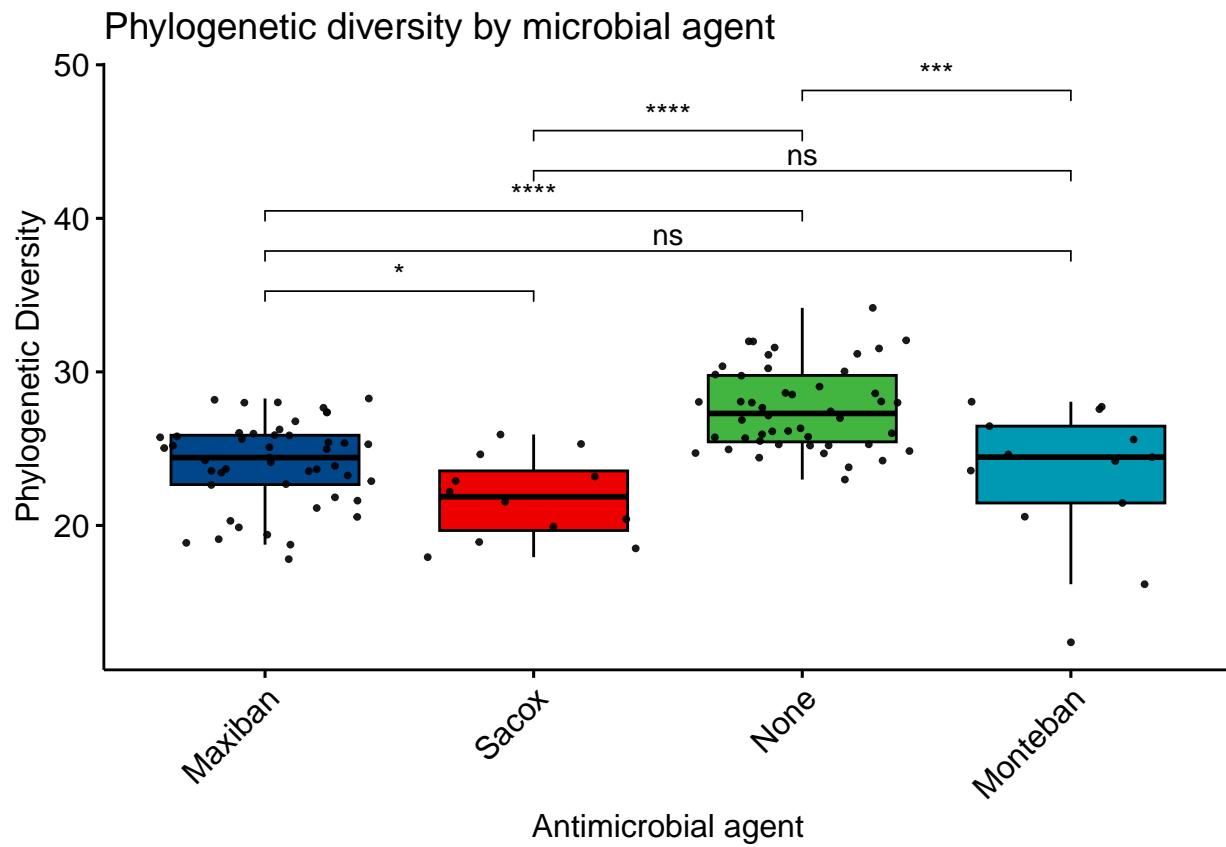
```

Alpha diversity metrics by microbial agent



```
df.pd <- pd(t(as.data.frame(subset16S@otu_table)), subset16S@phy_tree, include.root=T) # transposing for hmp.meta$Phylogenetic_Diversity <- df.pd$PD
```

```
ggboxplot(hmp.meta,
  x = "Cox",
  y = "Phylogenetic_Diversity",
  fill = "Cox",
  order = c("Maxiban", "Sacox", "None", "Monteban"),
  palette = "lancet",
  ylab = "Phylogenetic Diversity",
  xlab = "Antimicrobial agent",
  legend = "right",
  title = "Phylogenetic diversity by microbial agent",
  outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) +
  stat_compare_means(
    comparisons = L.pairs,
    label = "p.signif"
  ) + geom_jitter(size = 0.7, alpha = 0.9)
```



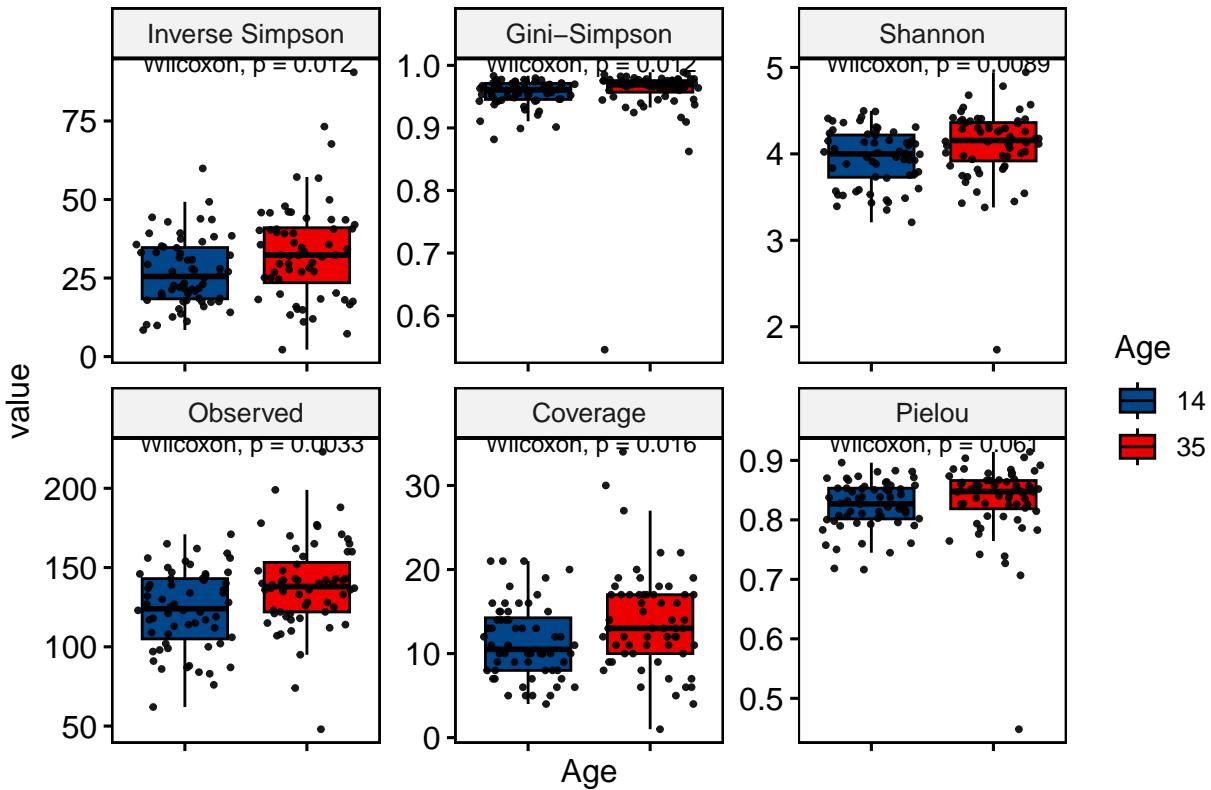
```
# age / days

div.df2 <- div.df[, c("Age", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon")]
colnames(div.df2) <- c("Age", "Inverse Simpson", "Gini-Simpson", "Shannon", "Observed", "Coverage", "Pi")

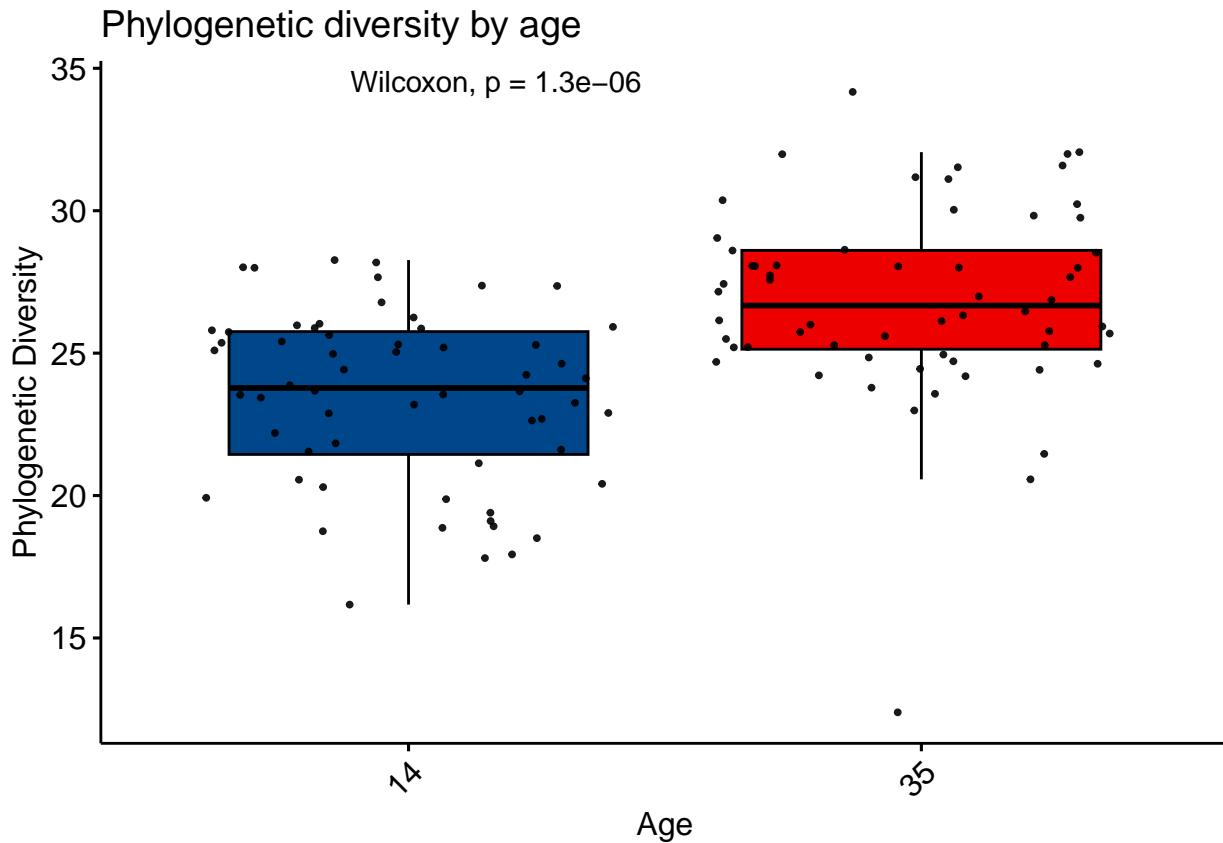
div.df2$Age = as.factor(div.df2$Age)
div_df_melt <- reshape2::melt(div.df2)

ggboxplot(div_df_melt, x = "Age", y = "value",
          fill = "Age",
          palette = "lancet",
          legend= "right",
          facet.by = "variable",
          scales = "free",
          title = "Alpha diversity metrics by age",
          outlier.shape = NA) +
  rremove("x.text") + stat_compare_means(method = "wilcox.test", size = 3.1) + geom_jitter(size = 0.7, a
```

Alpha diversity metrics by age



```
ggboxplot(hmp.meta,
  x = "Age",
  y = "Phylogenetic_Diversity",
  fill = "Age",
  palette = "lancet",
  ylab = "Phylogenetic Diversity",
  xlab = "Age",
  legend = "right",
  title = "Phylogenetic diversity by age",
  outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) +
  stat_compare_means(paired = TRUE) + geom_jitter(size = 0.7, alpha = 0.9)
```



```
# farms / company

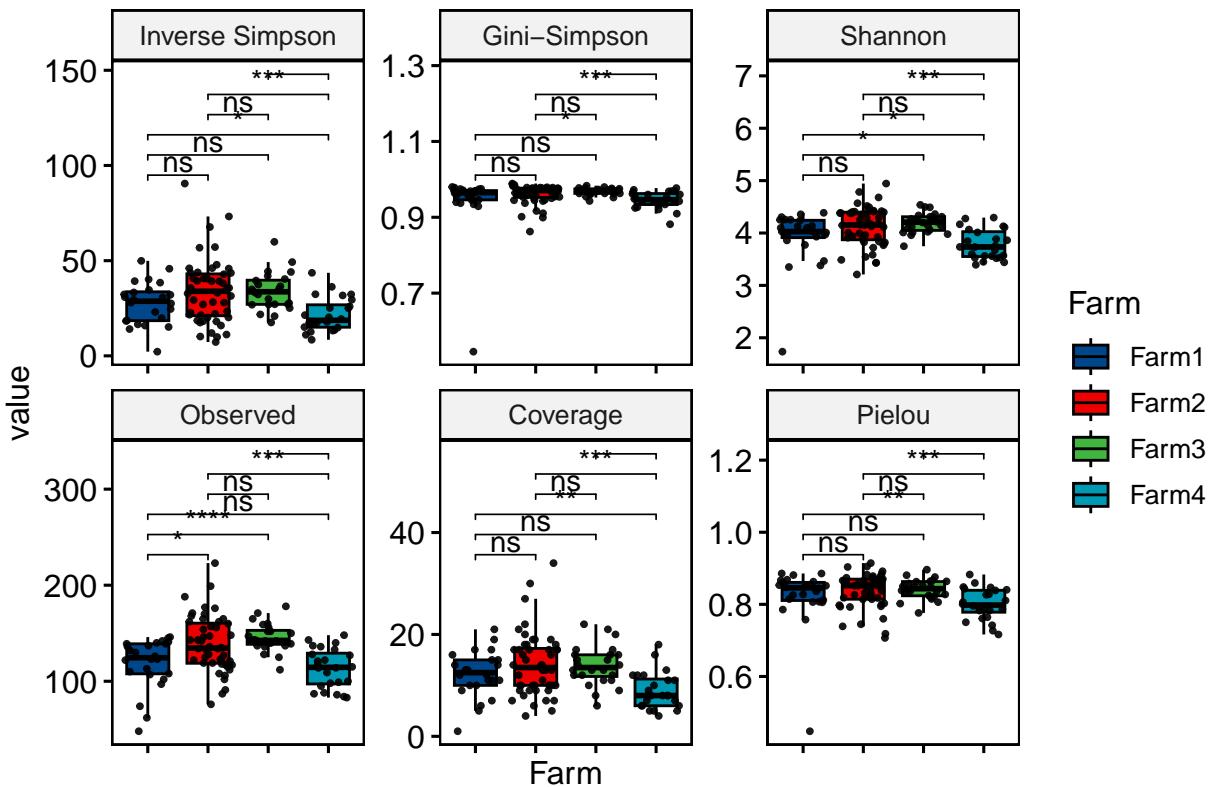
div.df2 <- div.df[, c("Farm2", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon")]
colnames(div.df2) <- c("Farm", "Inverse Simpson", "Gini-Simpson", "Shannon", "Observed", "Coverage", "P")

div_df_melt <- reshape2::melt(div.df2)

lev = c("Farm1", "Farm2", "Farm3", "Farm4")
L.pairs <- combn(seq_along(lev), 2, simplify = FALSE, FUN = function(i) lev[i])

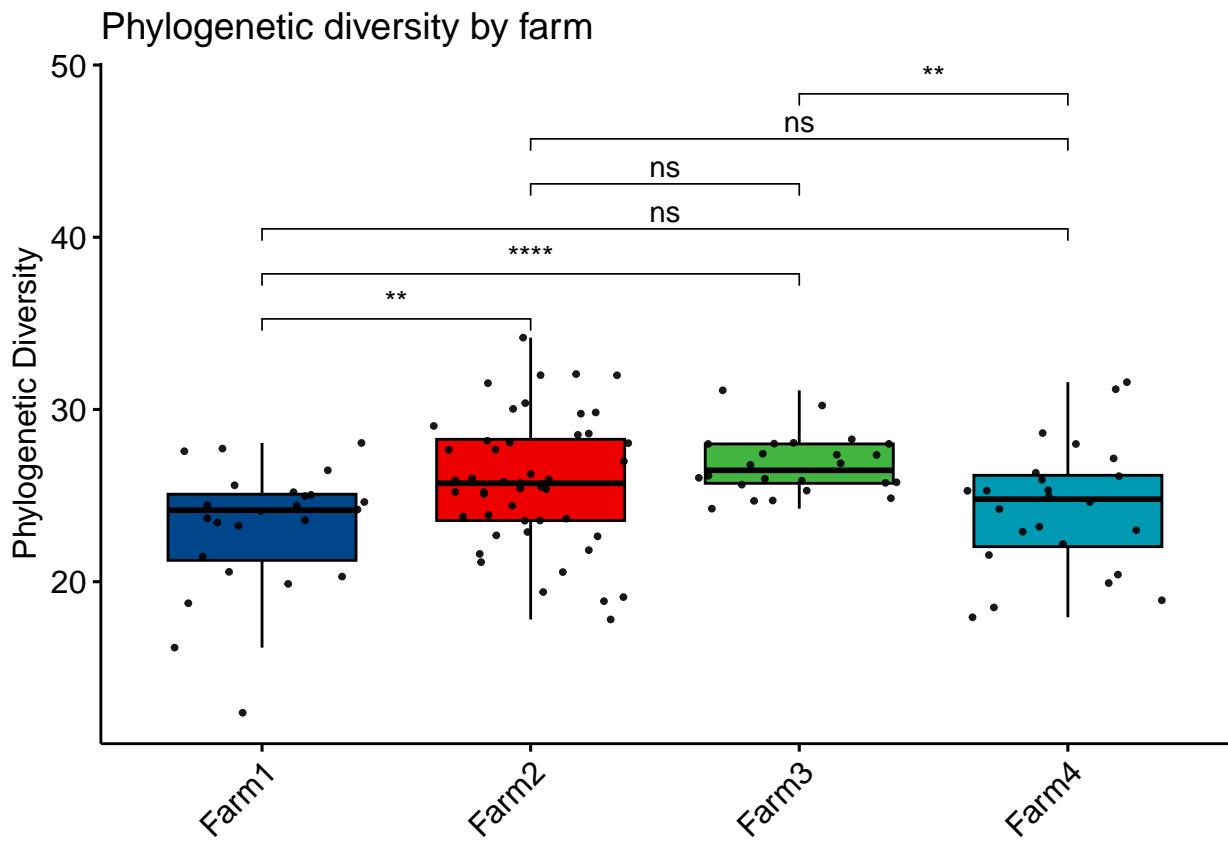
ggboxplot(div_df_melt, x = "Farm", y = "value",
          fill = "Farm",
          palette = "lancet",
          legend= "right",
          facet.by = "variable",
          scales = "free",
          order = lev,
          title = "Alpha diversity metrics by farm",
          outlier.shape = NA) + rotate_x_text() + rremove("x.text") + stat_compare_means(method = "wilcoxon",
                                                                 comparisons = 1,
                                                                 label = "p.signif")
    ) + geom_jitter(size = 0.7, alpha = 0.9)
```

Alpha diversity metrics by farm



```
ggboxplot(hmp.meta,
  x = "Farm2",
  y = "Phylogenetic_Diversity",
  fill = "Farm2",
  palette = "lancet",
  ylab = "Phylogenetic Diversity",
  xlab = "Farm",
  legend = "right",
  title = "Phylogenetic diversity by farm",
  outlier.shape = NA) + rotate_x_text() +
theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12), axis.title
```

stat_compare_means(
 comparisons = L.pairs,
 label = "p.signif"
) + geom_jitter(size = 0.7, alpha = 0.9)



```
# stable

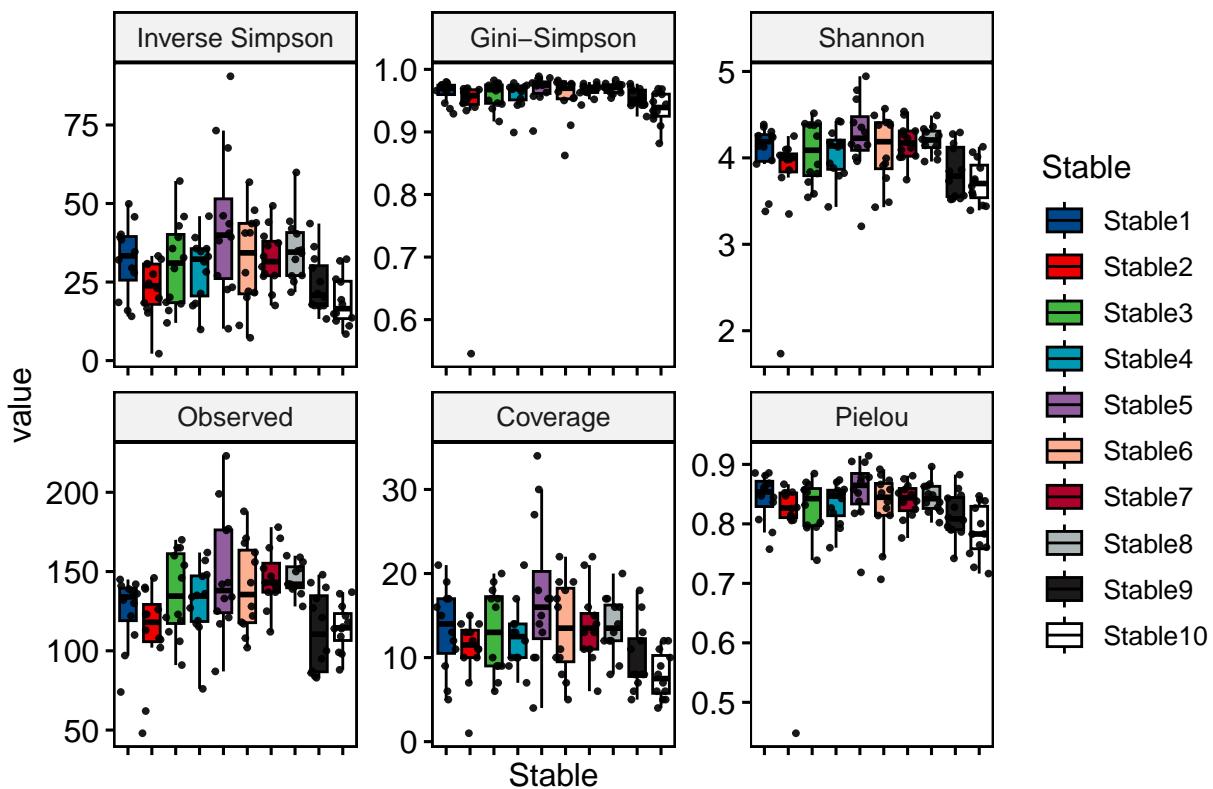
div.df2 <- div.df[, c("Stables", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shan-
colnames(div.df2) <- c("Stable", "Inverse Simpson", "Gini-Simpson", "Shannon", "Observed", "Coverage", "Cov

div_df_melt <- reshape2::melt(div.df2)

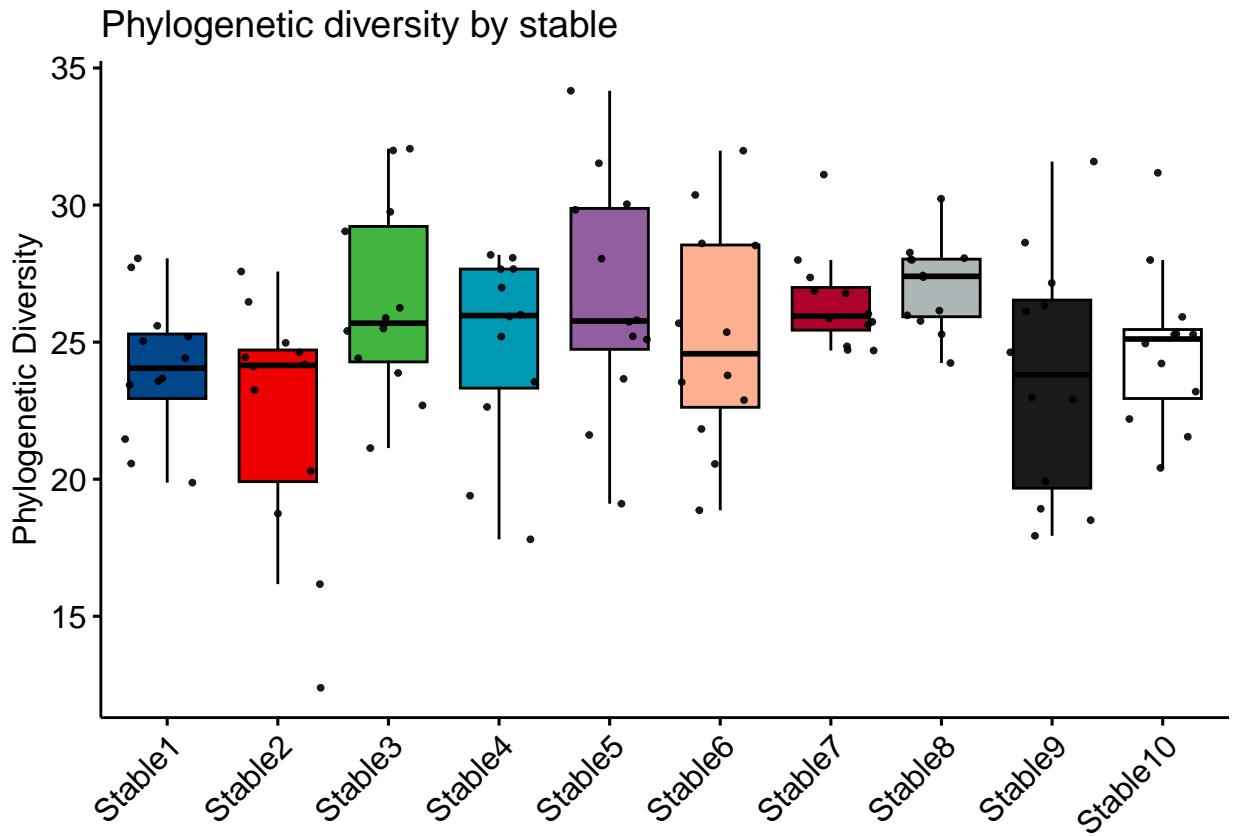
lev = c("Stable1", "Stable2", "Stable3", "Stable4", "Stable5", "Stable6", "Stable7", "Stable8", "Stable9", "Stab
L.pairs <- combn(seq_along(lev), 2, simplify = FALSE, FUN = function(i) lev[i])

ggboxplot(div_df_melt, x = "Stable", y = "value",
          fill = "Stable",
          palette = "lancet",
          legend= "right",
          facet.by = "variable",
          scales = "free",
          order = lev,
          title = "Alpha diversity metrics by stable",
          outlier.shape = NA) + rotate_x_text() + rremove("x.text") + geom_jitter(size = 0.7, alpha = 0.5)
```

Alpha diversity metrics by stable



```
ggboxplot(hmp.meta,
  x = "Stables",
  y = "Phylogenetic_Diversity",
  fill = "Stables",
  palette = "lancet",
  ylab = "Phylogenetic Diversity",
  xlab = "Farm",
  legend = "right",
  title = "Phylogenetic diversity by stable",
  outlier.shape = NA) + rotate_x_text() +
theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12),
axis.title.x = element_blank()) + geom_jitter(size = 0.7, alpha = 0.9)
```



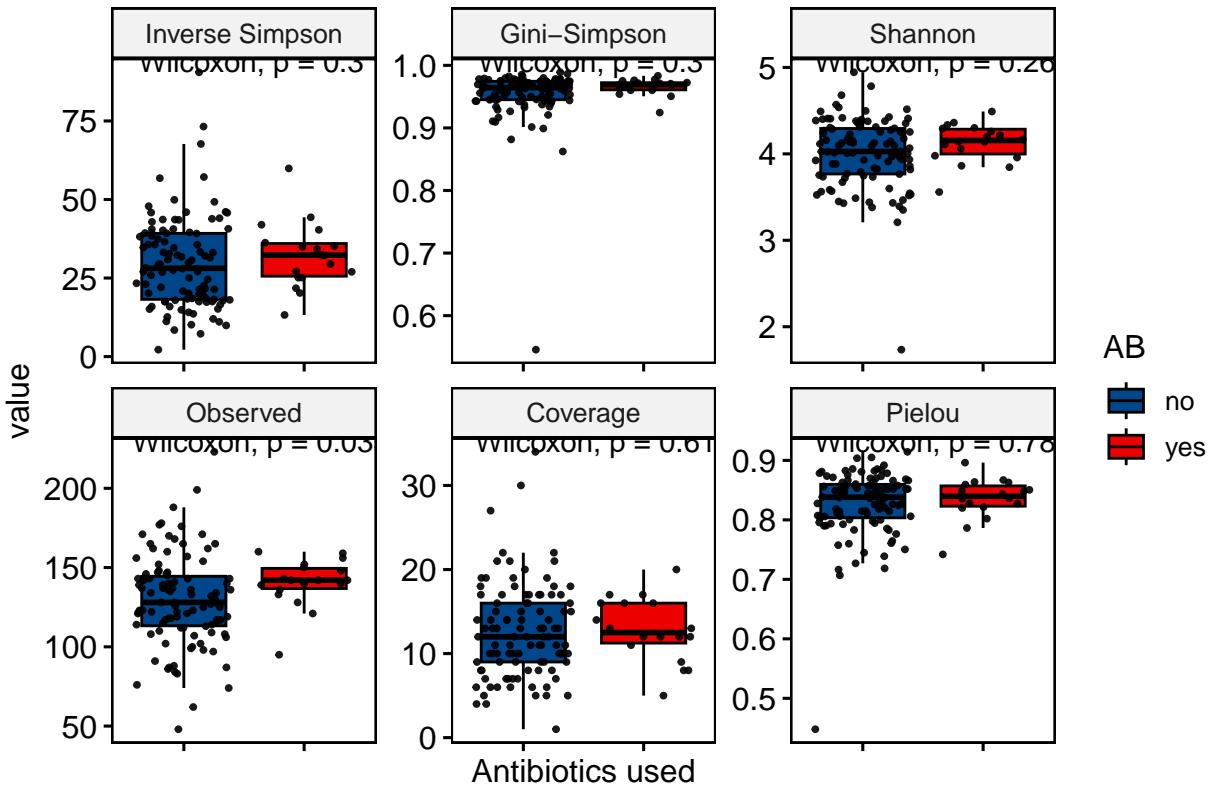
```
# based on AB

div.df2 <- div.df[, c("AB", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon",
colnames(div.df2) <- c("AB", "Inverse Simpson", "Gini-Simpson", "Shannon", "Observed", "Coverage", "Pie")

div_df_melt <- reshape2::melt(div.df2)

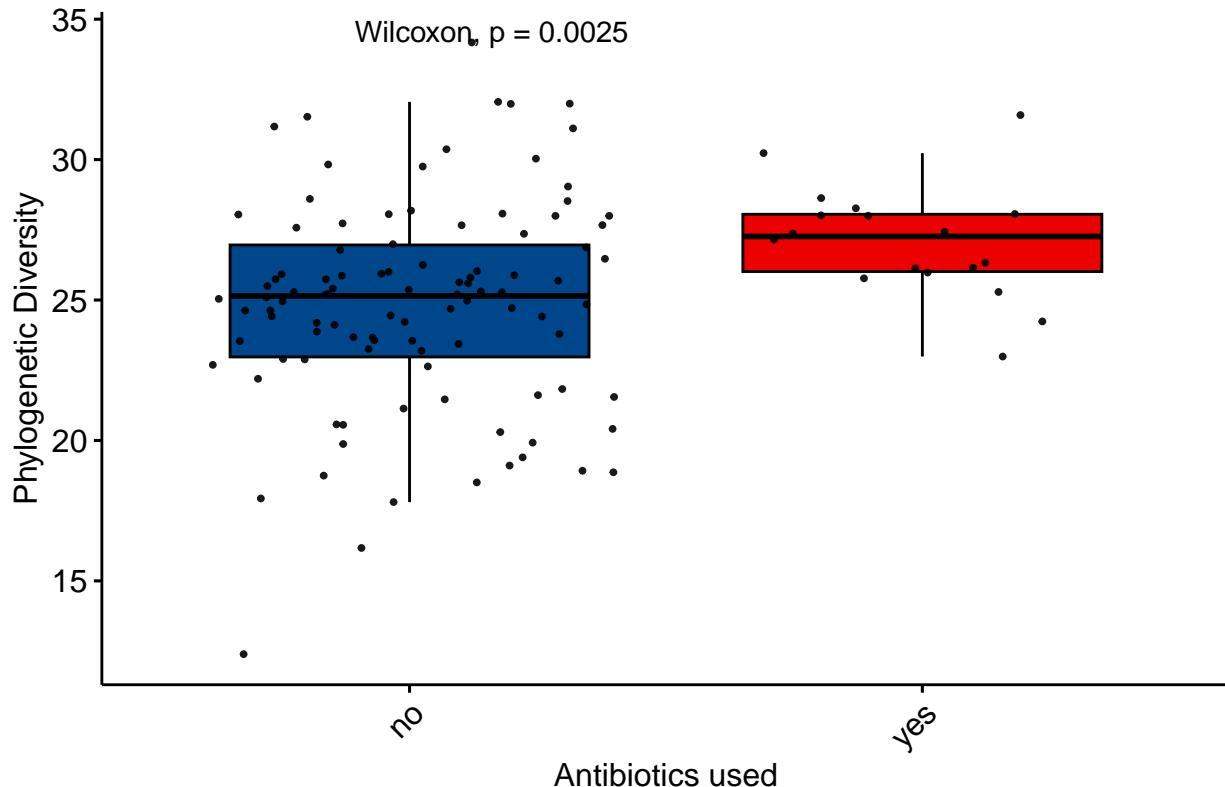
ggboxplot(div_df_melt, x = "AB", y = "value",
      fill = "AB",
      palette = "lancet",
      legend= "right",
      facet.by = "variable",
      scales = "free",
      xlab = "Antibiotics used",
      title = "Alpha diversity metrics by antibiotic usage",
      outlier.shape = NA) +
rremove("x.text") + stat_compare_means() + geom_jitter(size = 0.7, alpha = 0.9)
```

Alpha diversity metrics by antibiotic usage



```
ggboxplot(hmp.meta,
  x = "AB",
  y = "Phylogenetic_Diversity",
  fill = "AB",
  palette = "lancet",
  ylab = "Phylogenetic Diversity",
  xlab = "Antibiotics used",
  legend = "right",
  title = "Phylogenetic diversity by antibiotic usage",
  outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) +
  stat_compare_means() + geom_jitter(size = 0.7, alpha = 0.9)
```

Phylogenetic diversity by antibiotic usage



```
# based on stable and age

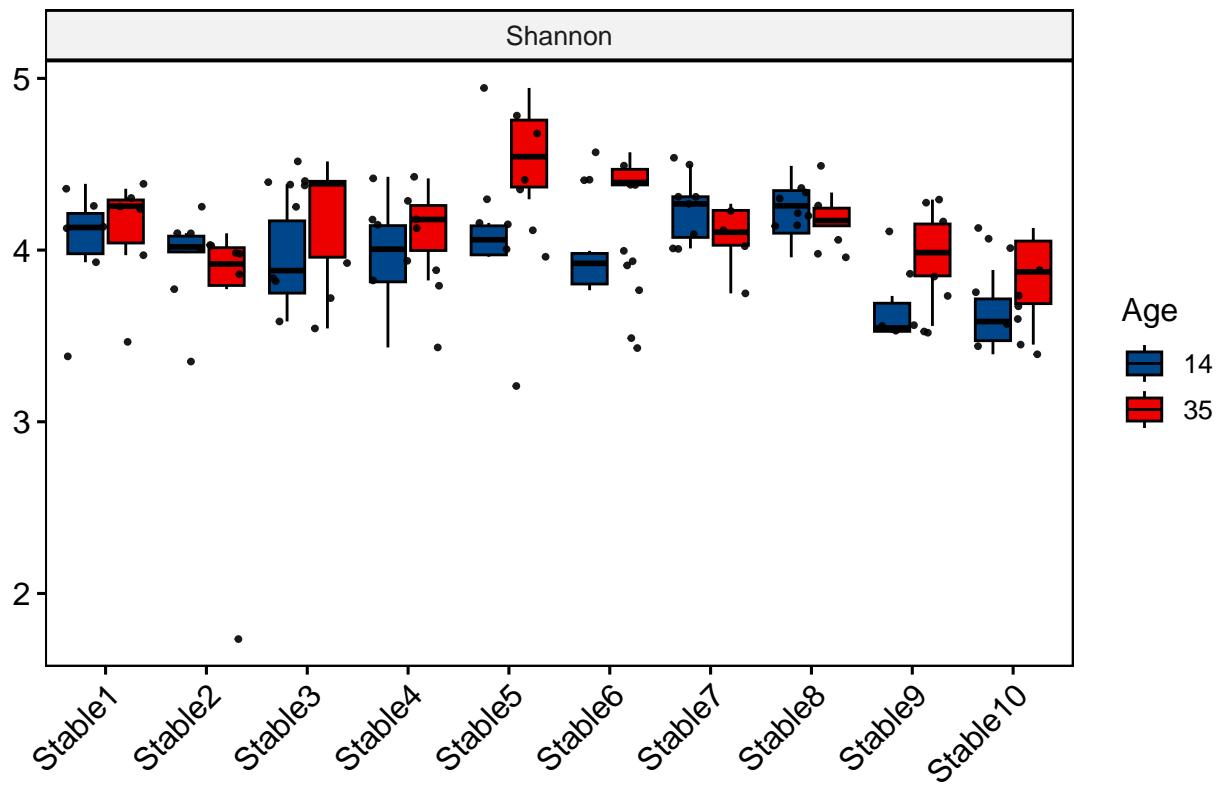
div.df2 <- div.df[, c("Stables", "Age", "diversity_shannon")]
colnames(div.df2) <- c("Stable", "Age", "Shannon")

div_df_melt <- reshape2::melt(div.df2)

lev = c("Stable1", "Stable2", "Stable3", "Stable4", "Stable5", "Stable6", "Stable7", "Stable8", "Stable9", "Stab

ggboxplot(div_df_melt, x = "Stable", y = "value",
  fill = "Age",
  palette = "lancet",
  legend= "right",
  facet.by = "variable",
  scales = "free",
  order = lev,
  title = "Shannon diversity by stable and age",
  xlab = FALSE,
  ylab = FALSE,
  outlier.shape = NA) + rotate_x_text() +
  theme(axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) + geom_jitter(size = 0.7, alpha = 0.5)
```

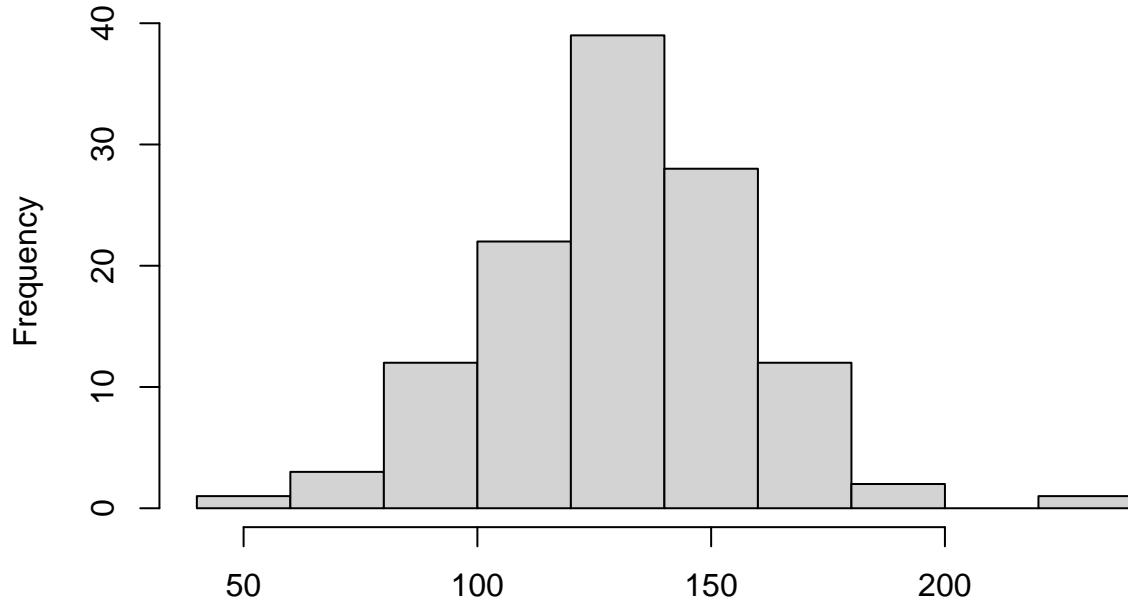
Shannon diversity by stable and age



```
## Looking at significance
```

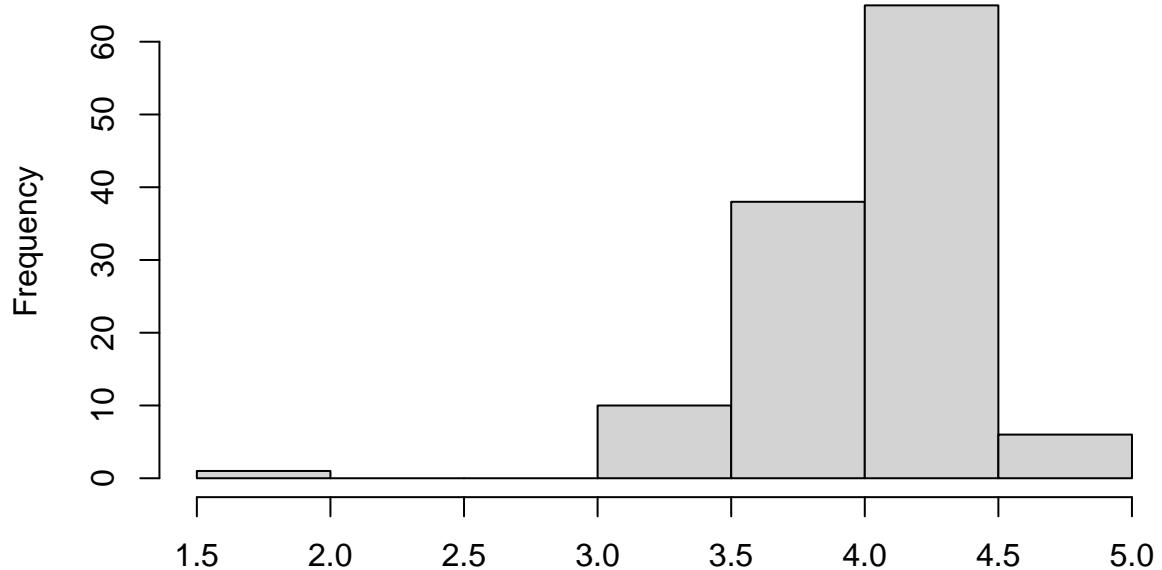
```
# Checking for normality  
hist(lib.div$observed, main="Observed richness", xlab="")
```

Observed richness



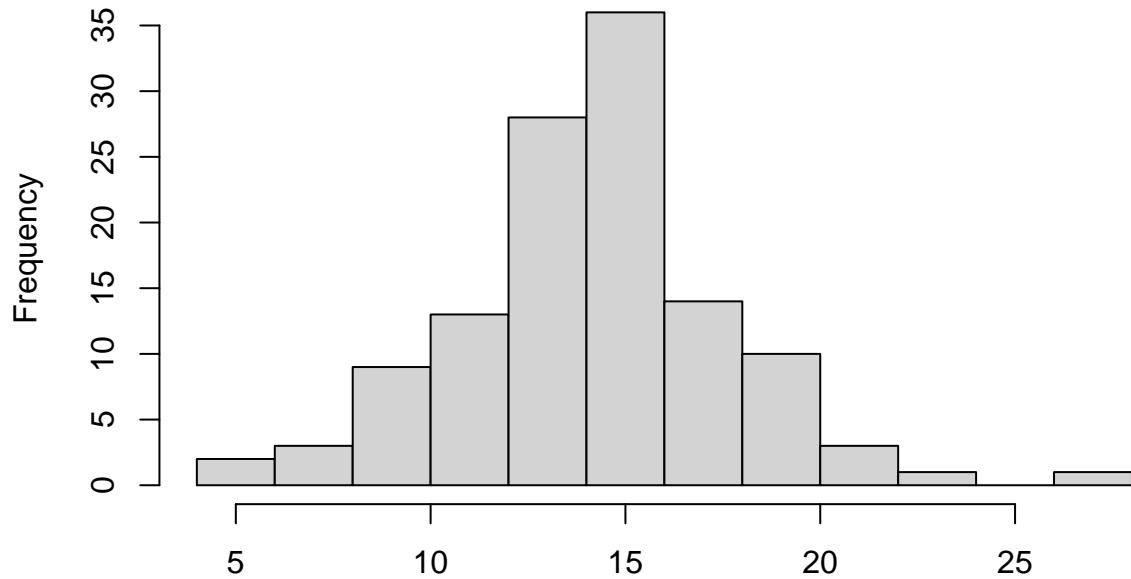
```
hist(lib.div$diversity_shannon, main="Shannon diversity", xlab="")
```

Shannon diversity



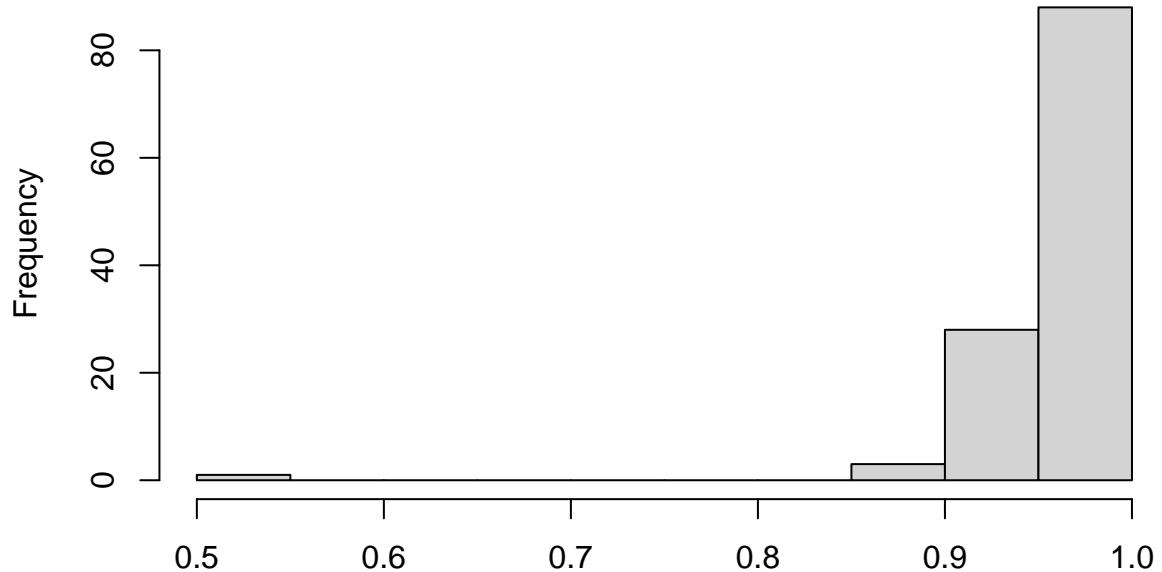
```
hist(lib.div$diversity_fisher, main="Fisher diversity", xlab="")
```

Fisher diversity



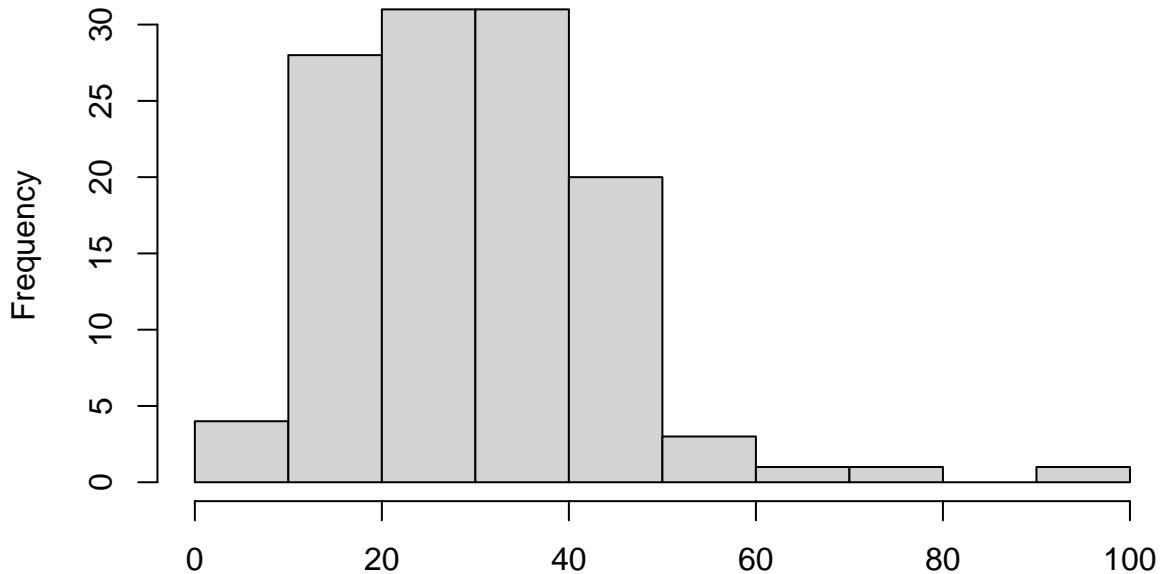
```
hist(lib.div$diversity_gini_simpson, main="Gini-Simpson diversity", xlab="")
```

Gini–Simpson diversity



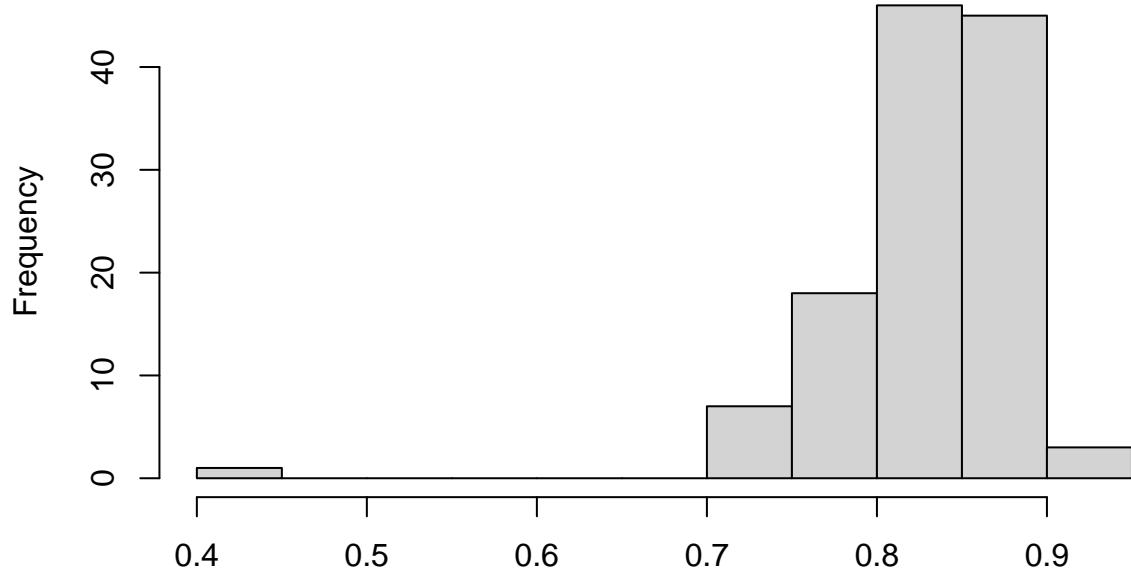
```
hist(lib.div$diversity_inverse_simpson, main="Inverse Simpson evenness", xlab="")
```

Inverse Simpson evenness



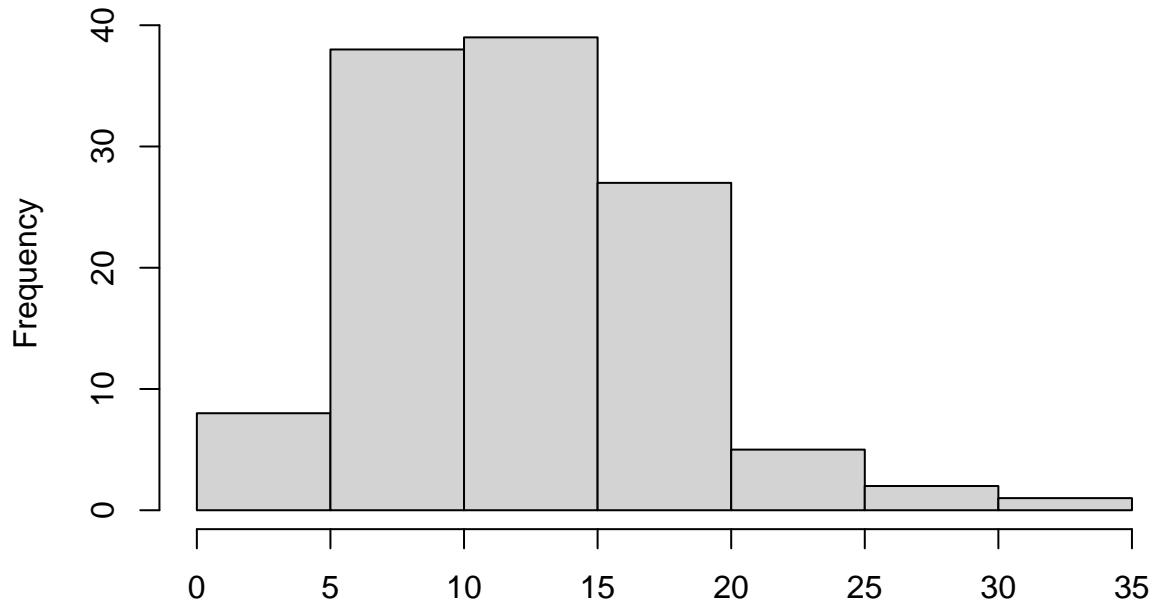
```
hist(lib.div$evenness_pielou, main="Pielou evenness", xlab="")
```

Pielou evenness



```
hist(lib.div$diversity_coverage, main="Coverage diversity", xlab="")
```

Coverage diversity



If data is normally distributed we can use ANOVA / t-tests, if not we will use Kruskal-Wallis tests. In this case, the data seems roughly normally distributed for some metrics, we can use Shapiro-Wilk tests to test for normality for individual measures

```
shapiro.test(lib.div$observed) # test deems it normally distributed p>0,05

##
##  Shapiro-Wilk normality test
##
## data: lib.div$observed
## W = 0.98871, p-value = 0.4266

shapiro.test(lib.div$diversity_shannon) # test deems this measure not normally distributed p<0,05

##
##  Shapiro-Wilk normality test
##
## data: lib.div$diversity_shannon
## W = 0.89676, p-value = 1.344e-07

shapiro.test(lib.div$diversity_fisher) # test deems this measure normally distributed p>0,05

##
##  Shapiro-Wilk normality test
##
```

```

## data: lib.div$diversity_fisher
## W = 0.98911, p-value = 0.4585

shapiro.test(lib.div$diversity_gini_simpson) # test deems this measure not normally distributed p<0,05

##
## Shapiro-Wilk normality test
##
## data: lib.div$diversity_gini_simpson
## W = 0.4334, p-value < 2.2e-16

shapiro.test(lib.div$diversity_inverse_simpson) # test deems this measure not normally distributed p<0,05

##
## Shapiro-Wilk normality test
##
## data: lib.div$diversity_inverse_simpson
## W = 0.9455, p-value = 0.0001025

shapiro.test(lib.div$evenness_pielou) # test deems this measure not normally distributed p<0,05

##
## Shapiro-Wilk normality test
##
## data: lib.div$evenness_pielou
## W = 0.79225, p-value = 9.689e-12

shapiro.test(lib.div$diversity_coverage) # test deems this measure not normally distributed p<0,05

##
## Shapiro-Wilk normality test
##
## data: lib.div$diversity_coverage
## W = 0.95673, p-value = 0.000697

shapiro.test(lib.div$Phylogenetic_Diversity) # test deems this measure normally distributed p>0,05

##
## Shapiro-Wilk normality test
##
## data: lib.div$Phylogenetic_Diversity
## W = 0.9791, p-value = 0.0587

# Based on shapiro-wilk tests we will assume normality for some measures
# The variables that we are interested in are the Age, which Farm the samples are from, and whether ant

# We will run ANOVAs for the normally distributed variables

# Age

# Normally distributed with only 2 levels, so we can use t-tests :

t.test(lib.div$observed ~ sample_data(subset16S)$Age) # significant

```

```

##  

## Welch Two Sample t-test  

##  

## data: lib.div$observed by sample_data(subset16S)$Age  

## t = -3.2512, df = 116.43, p-value = 0.001503  

## alternative hypothesis: true difference in means between group 14 and group 35 is not equal to 0  

## 95 percent confidence interval:  

## -25.371419 -6.161914  

## sample estimates:  

## mean in group 14 mean in group 35  

## 123.1833 138.9500

t.test(lib.div$diversity_fisher ~ sample_data(subset16S)$Age) # significant
```



```

##  

## Welch Two Sample t-test  

##  

## data: lib.div$diversity_fisher by sample_data(subset16S)$Age  

## t = -3.7527, df = 117.44, p-value = 0.0002734  

## alternative hypothesis: true difference in means between group 14 and group 35 is not equal to 0  

## 95 percent confidence interval:  

## -3.491963 -1.079515  

## sample estimates:  

## mean in group 14 mean in group 35  

## 13.01596 15.30170

t.test(lib.div$Phylogenetic_Diversity ~ sample_data(subset16S)$Age) # significant
```



```

##  

## Welch Two Sample t-test  

##  

## data: lib.div$Phylogenetic_Diversity by sample_data(subset16S)$Age  

## t = -5.8275, df = 116.62, p-value = 5.097e-08  

## alternative hypothesis: true difference in means between group 14 and group 35 is not equal to 0  

## 95 percent confidence interval:  

## -4.565813 -2.249550  

## sample estimates:  

## mean in group 14 mean in group 35  

## 23.46607 26.87375
```



```
# Non-normally distributed
```



```
wilcox.test(lib.div$diversity_shannon ~ sample_data(subset16S)$Age) # shannon diversity seems to signif
```



```

##  

## Wilcoxon rank sum test with continuity correction  

##  

## data: lib.div$diversity_shannon by sample_data(subset16S)$Age  

## W = 1301, p-value = 0.008885  

## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(lib.div$diversity_gini_simpson ~ sample_data(subset16S)$Age) # significant

## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$diversity_gini_simpson by sample_data(subset16S)$Age
## W = 1323, p-value = 0.01239
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(lib.div$diversity_inverse_simpson ~ sample_data(subset16S)$Age) # significant
```

```
## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$diversity_inverse_simpson by sample_data(subset16S)$Age
## W = 1323, p-value = 0.01239
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(lib.div$evenness_pielou ~ sample_data(subset16S)$Age) # not significant
```

```
## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$evenness_pielou by sample_data(subset16S)$Age
## W = 1443, p-value = 0.06133
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(lib.div$diversity_coverage ~ sample_data(subset16S)$Age) # significant
```

```
## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$diversity_coverage by sample_data(subset16S)$Age
## W = 1340.5, p-value = 0.01577
## alternative hypothesis: true location shift is not equal to 0
```

For age, the groups seems significantly different in all metrics except pielou evenness.

Antibiotics

```
t.test(lib.div$observed ~ sample_data(subset16S)$AB) # significant
```

```
## 
## Welch Two Sample t-test
## 
## data: lib.div$observed by sample_data(subset16S)$AB
## t = -2.3721, df = 42.794, p-value = 0.02225
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0
## 95 percent confidence interval:
## -20.17172 -1.63220
```

```

## sample estimates:
##   mean in group no mean in group yes
##             129.4314          140.3333

t.test(lib.div$diversity_fisher ~ sample_data(subset16S)$AB) # significant

## 
## Welch Two Sample t-test
##
## data: lib.div$diversity_fisher by sample_data(subset16S)$AB
## t = -2.0857, df = 44.774, p-value = 0.04273
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0
## 95 percent confidence interval:
## -2.3593692 -0.0410508
## sample estimates:
##   mean in group no mean in group yes
##             13.97879          15.17900

t.test(lib.div$Phylogenetic_Diversity ~ sample_data(subset16S)$AB) # significant

## 
## Welch Two Sample t-test
##
## data: lib.div$Phylogenetic_Diversity by sample_data(subset16S)$AB
## t = -3.7313, df = 40.877, p-value = 0.0005798
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0
## 95 percent confidence interval:
## -3.481642 -1.036191
## sample estimates:
##   mean in group no mean in group yes
##             24.83107          27.08999

# Non-normally distributed

wilcox.test(lib.div$diversity_shannon ~ sample_data(subset16S)$AB) # shannon diversity does not seem to

## 
## Wilcoxon rank sum test with continuity correction
##
## data: lib.div$diversity_shannon by sample_data(subset16S)$AB
## W = 765, p-value = 0.2624
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(lib.div$diversity_gini_simpson ~ sample_data(subset16S)$AB) # not significant

## 
## Wilcoxon rank sum test with continuity correction
##
## data: lib.div$diversity_gini_simpson by sample_data(subset16S)$AB
## W = 776, p-value = 0.2984
## alternative hypothesis: true location shift is not equal to 0

```

```
wilcox.test(lib.div$diversity_inverse_simpson ~ sample_data(subset16S)$AB) # not significant

## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$diversity_inverse_simpson by sample_data(subset16S)$AB
## W = 776, p-value = 0.2984
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(lib.div$evenness_pielou ~ sample_data(subset16S)$AB) # not significant

## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$evenness_pielou by sample_data(subset16S)$AB
## W = 879, p-value = 0.7772
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(lib.div$diversity_coverage ~ sample_data(subset16S)$AB) # not significant

## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$diversity_coverage by sample_data(subset16S)$AB
## W = 848, p-value = 0.6088
## alternative hypothesis: true location shift is not equal to 0

# used the following functions to get means and sd per variable and alpha diversity metric
#lib.div.ab = lib.div
#lib.div.ab$AB = sample_data(subset16S)$AB
#aggregate(lib.div.ab$observed, list(lib.div.ab$AB), FUN=mean)
#aggregate(lib.div.ab$observed, list(lib.div.ab$AB), FUN=sd)

# AB does not seem to significantly differ in their alpha diversities except for observed, PD and fisher

# Farm has more than 2 levels, so we will use ANOVAs for normally distributed metrics

aov.observed.farm = aov(lib.div$observed ~ sample_data(subset16S)$Farm2)
summary(aov.observed.farm)

##                               Df Sum Sq Mean Sq F value    Pr(>F)
## sample_data(subset16S)$Farm2     3   19238    6413   10.41 4.07e-06 ***
## Residuals                      116   71473     616
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(aov.observed.farm) # only not significant between 1 and 4 and 3 and 2
```

```

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lib.div$observed ~ sample_data(subset16S)$Farm2)
##
## $`sample_data(subset16S)$Farm2`
##          diff      lwr      upr     p adj
## Farm2-Farm1 20.4375  4.261634 36.613366 0.0070916
## Farm3-Farm1 27.2500  8.571718 45.928282 0.0012984
## Farm4-Farm1 -4.8750 -23.553282 13.803282 0.9043806
## Farm3-Farm2  6.8125 -9.363366 22.988366 0.6915866
## Farm4-Farm2 -25.3125 -41.488366 -9.136634 0.0004785
## Farm4-Farm3 -32.1250 -50.803282 -13.446718 0.0001014

aov.fisher.farm = aov(lib.div$diversity_fisher ~ sample_data(subset16S)$Farm2)
summary(aov.fisher.farm)

```

```

##                               Df Sum Sq Mean Sq F value    Pr(>F)
## sample_data(subset16S)$Farm2   3 294.8  98.27   9.7 9.2e-06 ***
## Residuals                     116 1175.2   10.13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
TukeyHSD(aov.fisher.farm) # only not significant between 1 and 4 and 3 and 2
```

```

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lib.div$diversity_fisher ~ sample_data(subset16S)$Farm2)
##
## $`sample_data(subset16S)$Farm2`
##          diff      lwr      upr     p adj
## Farm2-Farm1 2.4636185  0.3893902 4.537847 0.0129624
## Farm3-Farm1 3.0034387  0.6083262 5.398551 0.0076605
## Farm4-Farm1 -0.9432080 -3.3383205 1.451905 0.7342758
## Farm3-Farm2  0.5398202 -1.5344081 2.614048 0.9051069
## Farm4-Farm2 -3.4068265 -5.4810547 -1.332598 0.0002229
## Farm4-Farm3 -3.9466467 -6.3417592 -1.551534 0.0002113

```

```
# Non-normally distributed
```

```
kruskal.test(lib.div$diversity_shannon ~ sample_data(subset16S)$Farm2) # shannon diversity seems to sig
```

```

## 
## Kruskal-Wallis rank sum test
##
## data: lib.div$diversity_shannon by sample_data(subset16S)$Farm2
## Kruskal-Wallis chi-squared = 21.53, df = 3, p-value = 8.171e-05

```

```
pairwise.wilcox.test(lib.div$diversity_shannon, sample_data(subset16S)$Farm2, p.adjust.method="fdr") #
```

```

## 
## Pairwise comparisons using Wilcoxon rank sum exact test
## 
## data: lib.div$diversity_shannon and sample_data(subset16S)$Farm2
## 
##      Farm1     Farm2     Farm3
## Farm2 0.14005 -       -
## Farm3 0.03874 0.57297 -
## Farm4 0.04790 0.00047 2e-05
## 
## P value adjustment method: fdr

kruskal.test(lib.div$diversity_gini_simpson ~ sample_data(subset16S)$Farm2) # significant

## 
## Kruskal-Wallis rank sum test
## 
## data: lib.div$diversity_gini_simpson by sample_data(subset16S)$Farm2
## Kruskal-Wallis chi-squared = 17.336, df = 3, p-value = 0.0006028

pairwise.wilcox.test(lib.div$diversity_gini_simpson, sample_data(subset16S)$Farm2, p.adjust.method="fdr")

## 
## Pairwise comparisons using Wilcoxon rank sum exact test
## 
## data: lib.div$diversity_gini_simpson and sample_data(subset16S)$Farm2
## 
##      Farm1     Farm2     Farm3
## Farm2 0.15047 -       -
## Farm3 0.09692 0.78047 -
## Farm4 0.08732 0.00256 0.00015
## 
## P value adjustment method: fdr

kruskal.test(lib.div$diversity_inverse_simpson ~ sample_data(subset16S)$Farm2) # not significant

## 
## Kruskal-Wallis rank sum test
## 
## data: lib.div$diversity_inverse_simpson by sample_data(subset16S)$Farm2
## Kruskal-Wallis chi-squared = 17.336, df = 3, p-value = 0.0006028

pairwise.wilcox.test(lib.div$diversity_inverse_simpson, sample_data(subset16S)$Farm2, p.adjust.method="fdr")

## 
## Pairwise comparisons using Wilcoxon rank sum exact test
## 
## data: lib.div$diversity_inverse_simpson and sample_data(subset16S)$Farm2
## 
##      Farm1     Farm2     Farm3
## Farm2 0.15047 -       -

```

```

## Farm3 0.09692 0.78047 -
## Farm4 0.08732 0.00256 0.00015
##
## P value adjustment method: fdr

kruskal.test(lib.div$evenness_pielou ~ sample_data(subset16S)$Farm2) # significant

##
## Kruskal-Wallis rank sum test
##
## data: lib.div$evenness_pielou by sample_data(subset16S)$Farm2
## Kruskal-Wallis chi-squared = 14.059, df = 3, p-value = 0.002826

pairwise.wilcox.test(lib.div$evenness_pielou, sample_data(subset16S)$Farm2, p.adjust.method="fdr") # di

##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$evenness_pielou and sample_data(subset16S)$Farm2
##
##      Farm1   Farm2   Farm3
## Farm2 0.8283 -     -
## Farm3 0.8283 0.9197 -
## Farm4 0.0103 0.0028 0.0028
##
## P value adjustment method: fdr

# agent also has more than 2 levels, so we will use ANOVAs for normally distributed metrics

aov.observed.agent = aov(lib.div$observed ~ sample_data(subset16S)$Cox)
summary(aov.observed.agent)

##
##                               Df Sum Sq Mean Sq F value    Pr(>F)
## sample_data(subset16S)$Cox  3 24998   8333   14.71 3.52e-08 ***
## Residuals                  116 65714     566
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(aov.observed.agent) # only not significant between sacox and monteban

##
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lib.div$observed ~ sample_data(subset16S)$Cox)
##
## $`sample_data(subset16S)$Cox`
##          diff      lwr      upr      p adj
## Monteban-Maxiban -18.34043 -37.782360  1.101509 0.0719535
## None-Maxiban     14.30541  1.573992  27.036824 0.0210505
## Sacox-Maxiban   -30.09043 -50.156892 -10.023959 0.0008908
## None-Monteban    32.64583 13.247828  52.043839 0.0001481
## Sacox-Monteban  -11.75000 -36.586601 13.086601 0.6071257
## Sacox-None       -44.39583 -64.419741 -24.371926 0.0000004

```

```
aov.fisher.agent = aov(lib.div$diversity_fisher ~ sample_data(subset16S)$Cox)
summary(aov.fisher.agent)
```

```
##                                Df Sum Sq Mean Sq F value    Pr(>F)
## sample_data(subset16S)$Cox     3  443.3  147.75   16.69 4.42e-09 ***
## Residuals                     116 1026.8    8.85
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aov.fisher.agent) # only not significant maxiban & monteban and sacox & monteban
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lib.div$diversity_fisher ~ sample_data(subset16S)$Cox)
##
## $`sample_data(subset16S)$Cox`
##      diff      lwr      upr      p adj
## Monteban-Maxiban -1.947259 -4.3775075 0.4829893 0.1629193
## None-Maxiban     2.025225  0.4337933 3.6166559 0.0065804
## Sacox-Maxiban    -4.096756 -6.6050714 -1.5884411 0.0002443
## None-Monteban    3.972484  1.5477264 6.3972410 0.0002324
## Sacox-Monteban   -2.149497 -5.2540807 0.9550864 0.2763516
## Sacox-None       -6.121981 -8.6249761 -3.6189855 0.0000000
```

```
# Non-normally distributed
```

```
kruskal.test(lib.div$diversity_shannon ~ sample_data(subset16S)$Cox) # shannon diversity seems to signifi
```

```
##
## Kruskal-Wallis rank sum test
##
## data: lib.div$diversity_shannon by sample_data(subset16S)$Cox
## Kruskal-Wallis chi-squared = 20.995, df = 3, p-value = 0.0001056
```

```
pairwise.wilcox.test(lib.div$diversity_shannon, sample_data(subset16S)$Cox, p.adjust.method="fdr") # di
```

```
##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$diversity_shannon and sample_data(subset16S)$Cox
##
##          Maxiban Monteban None
## Monteban 0.31038 -      -
## None      0.09625 0.06237 -
## Sacox     0.00031 0.10411 8.1e-05
##
## P value adjustment method: fdr
```

```

kruskal.test(lib.div$diversity_gini_simpson ~ sample_data(subset16S)$Cox) # significant

##
## Kruskal-Wallis rank sum test
##
## data: lib.div$diversity_gini_simpson by sample_data(subset16S)$Cox
## Kruskal-Wallis chi-squared = 15.296, df = 3, p-value = 0.00158

pairwise.wilcox.test(lib.div$diversity_gini_simpson, sample_data(subset16S)$Cox, p.adjust.method="fdr")

##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$diversity_gini_simpson and sample_data(subset16S)$Cox
##
##          Maxiban Monteban None
## Monteban 0.4557   -      -
## None     0.0906  0.1557   -
## Sacox    0.0040  0.2020  0.0035
##
## P value adjustment method: fdr

kruskal.test(lib.div$diversity_inverse_simpson ~ sample_data(subset16S)$Cox) # not significant

##
## Kruskal-Wallis rank sum test
##
## data: lib.div$diversity_inverse_simpson by sample_data(subset16S)$Cox
## Kruskal-Wallis chi-squared = 15.296, df = 3, p-value = 0.00158

pairwise.wilcox.test(lib.div$diversity_inverse_simpson, sample_data(subset16S)$Cox, p.adjust.method="fdr")

##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$diversity_inverse_simpson and sample_data(subset16S)$Cox
##
##          Maxiban Monteban None
## Monteban 0.4557   -      -
## None     0.0906  0.1557   -
## Sacox    0.0040  0.2020  0.0035
##
## P value adjustment method: fdr

kruskal.test(lib.div$evenness_pielou ~ sample_data(subset16S)$Cox) # significant

##
## Kruskal-Wallis rank sum test
##
## data: lib.div$evenness_pielou by sample_data(subset16S)$Cox
## Kruskal-Wallis chi-squared = 10.984, df = 3, p-value = 0.01181

```

```

pairwise.wilcox.test(lib.div$evenness_pielou, sample_data(subset16S)$Cox, p.adjust.method="fdr") # same

##
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$evenness_pielou and sample_data(subset16S)$Cox
##
##          Maxiban Monteban None
## Monteban 0.750   -      -
## None     0.377   0.454   -
## Sacox    0.010   0.051   0.010
##
## P value adjustment method: fdr

```

Beta diversity

```
estimate_richness(subset16S) # no singletons
```

	Observed	Chao1	se.chao1	ACE	se.ACE	Shannon	Simpson	InvSimpson
## X2_23	132	132	0	NaN	NaN	4.136003	0.9711196	34.625599
## X2_24	139	139	0	NaN	NaN	4.126936	0.9642492	27.971432
## X2_25	97	97	0	NaN	NaN	3.465461	0.9290117	14.086821
## X2_26	130	130	0	NaN	NaN	3.929347	0.9460238	18.526673
## X2_27	136	136	0	NaN	NaN	4.239237	0.9739942	38.452916
## X2_29	145	145	0	NaN	NaN	4.385673	0.9745243	39.253112
## X2_36	123	123	0	NaN	NaN	4.098432	0.9675785	30.843755
## X2_39	123	123	0	NaN	NaN	3.982521	0.9565922	23.037346
## X2_40	146	146	0	NaN	NaN	4.252383	0.9690197	32.278559
## X2_41	102	102	0	NaN	NaN	4.008183	0.9674078	30.682184
## X2_42	113	113	0	NaN	NaN	4.029328	0.9636649	27.521614
## X2_47	141	141	0	NaN	NaN	4.257202	0.9688112	32.062832
## X2_48	74	74	0	NaN	NaN	3.380875	0.9371476	15.910281
## X2_49	110	110	0	NaN	NaN	3.970268	0.9660476	29.453042
## X2_50	137	137	0	NaN	NaN	4.357221	0.9799604	49.901207
## X2_51	122	122	0	NaN	NaN	4.253943	0.9781493	45.765045
## X2_52	142	142	0	NaN	NaN	4.303497	0.9751186	40.190629
## X2_56	126	126	0	NaN	NaN	4.097594	0.9699867	33.318612
## X2_57	48	48	0	NaN	NaN	1.734529	0.5455967	2.200688
## X2_58	108	108	0	NaN	NaN	3.772379	0.9455148	18.353603
## X2_59	107	107	0	NaN	NaN	3.858977	0.9593703	24.612528
## X2_60	139	139	0	NaN	NaN	3.978746	0.9497115	19.885269
## X2_61	140	140	0	NaN	NaN	4.025627	0.9395879	16.552984
## X4_36	119	119	0	NaN	NaN	3.836459	0.9503957	20.159535
## X4_37	91	91	0	NaN	NaN	3.584282	0.9461311	18.563597
## X4_38	146	146	0	NaN	NaN	4.252073	0.9694887	32.774785
## X4_39	106	106	0	NaN	NaN	3.720320	0.9372538	15.937227
## X4_40	154	154	0	NaN	NaN	4.381358	0.9766794	42.880526
## X4_41	112	112	0	NaN	NaN	3.924462	0.9658913	29.318027
## X4_54	170	170	0	NaN	NaN	4.402081	0.9719656	35.670497
## X4_55	121	121	0	NaN	NaN	3.543604	0.9166889	12.003196
## X4_56	165	165	0	NaN	NaN	4.376416	0.9745189	39.244723

## X4_57	160	160	0	NaN	NaN	4.395525	0.9781892	45.848768
## X4_65	62	62	0	NaN	NaN	3.350815	0.9340956	15.173499
## X5_39	76	76	0	NaN	NaN	3.432905	0.9428191	17.488352
## X5_40	162	162	0	NaN	NaN	4.426712	0.9697948	33.106899
## X5_41	147	147	0	NaN	NaN	3.792368	0.8991109	9.911873
## X5_54	165	165	0	NaN	NaN	4.516410	0.9825044	57.157274
## X5_55	123	123	0	NaN	NaN	3.819140	0.9446352	18.062019
## X5_59	157	157	0	NaN	NaN	4.417525	0.9782484	45.973571
## X6_36	127	127	0	NaN	NaN	4.147429	0.9719594	35.662588
## X6_37	115	115	0	NaN	NaN	3.882798	0.9531995	21.367304
## X6_38	134	134	0	NaN	NaN	4.127256	0.9682224	31.468693
## X6_54	135	135	0	NaN	NaN	4.178638	0.9719210	35.613788
## X6_55	148	148	0	NaN	NaN	4.286540	0.9744719	39.172460
## X6_56	136	136	0	NaN	NaN	4.177493	0.9711918	34.712361
## X6_57	119	119	0	NaN	NaN	3.823249	0.9450037	18.183044
## X6_58	117	117	0	NaN	NaN	3.937373	0.9645764	28.229778
## X9_16	125	125	0	NaN	NaN	3.961053	0.9558198	22.634587
## X9_17	121	121	0	NaN	NaN	4.115315	0.9737945	38.159910
## X9_18	134	134	0	NaN	NaN	4.005118	0.9571588	23.342017
## X9_19	87	87	0	NaN	NaN	3.208366	0.9014346	10.145551
## X9_21	143	143	0	NaN	NaN	4.158081	0.9629915	27.020836
## X9_22	117	117	0	NaN	NaN	4.149825	0.9745724	39.327320
## X9_34	122	122	0	NaN	NaN	3.910564	0.9502966	20.119340
## X9_35	108	108	0	NaN	NaN	3.487252	0.9107067	11.199048
## X9_36	117	117	0	NaN	NaN	3.935009	0.9537531	21.623091
## X9_37	118	118	0	NaN	NaN	3.995557	0.9642583	27.978554
## X9_38	102	102	0	NaN	NaN	3.766274	0.9546683	22.059594
## X9_39	156	156	0	NaN	NaN	4.409694	0.9771994	43.858458
## X10_1	147	147	0	NaN	NaN	4.022611	0.9520885	20.871823
## X10_2	171	171	0	NaN	NaN	4.310664	0.9698373	33.153575
## X10_3	143	143	0	NaN	NaN	4.227802	0.9726454	36.556920
## X10_4	152	152	0	NaN	NaN	4.310282	0.9732866	37.434357
## X10_7	137	137	0	NaN	NaN	4.009593	0.9630614	27.071976
## X10_8	165	165	0	NaN	NaN	4.497594	0.9797021	49.266232
## X10_10	159	159	0	NaN	NaN	4.300777	0.9715780	35.183966
## X10_11	150	150	0	NaN	NaN	4.490556	0.9833046	59.896830
## X10_12	142	142	0	NaN	NaN	4.214847	0.9713536	34.908339
## X10_13	156	156	0	NaN	NaN	4.361647	0.9774440	44.334088
## X10_14	139	139	0	NaN	NaN	3.957786	0.9540649	21.769848
## X10_15	140	140	0	NaN	NaN	4.059547	0.9601771	25.111174
## X10_19	84	84	0	NaN	NaN	3.525545	0.9423290	17.339745
## X10_20	87	87	0	NaN	NaN	3.530217	0.9429301	17.522359
## X10_21	127	127	0	NaN	NaN	4.276047	0.9770737	43.618080
## X10_22	86	86	0	NaN	NaN	3.518669	0.9428566	17.499831
## X10_25	83	83	0	NaN	NaN	3.562820	0.9439357	17.836654
## X10_26	100	100	0	NaN	NaN	3.732990	0.9532478	21.389377
## X10_28	88	88	0	NaN	NaN	3.754545	0.9614331	25.928982
## X10_29	114	114	0	NaN	NaN	3.393108	0.8816650	8.450584
## X10_30	109	109	0	NaN	NaN	3.568981	0.9265013	13.605679
## X10_33	128	128	0	NaN	NaN	3.883777	0.9483538	19.362526
## X10_34	98	98	0	NaN	NaN	3.440144	0.9207547	12.619050
## X10_35	99	99	0	NaN	NaN	3.598482	0.9442832	17.947893
## X10_39	125	125	0	NaN	NaN	3.747857	0.9430010	17.544153
## X10_40	112	112	0	NaN	NaN	4.006892	0.9627609	26.853519

```

## X10_41      140    140      0 NaN      NaN 4.269641 0.9747131 39.546109
## X10_42      143    143      0 NaN      NaN 4.092277 0.9638905 27.693543
## X10_43      178    178      0 NaN      NaN 4.537584 0.9773062 44.064913
## X10_44      137    137      0 NaN      NaN 4.115558 0.9665253 29.873330
## X10_48      160    160      0 NaN      NaN 4.199373 0.9631082 27.106304
## X10_49      142    142      0 NaN      NaN 4.140257 0.9689761 32.233257
## X10_50      128    128      0 NaN      NaN 3.978572 0.9629375 26.981421
## X10_51      152    152      0 NaN      NaN 4.335136 0.9752193 40.354031
## X10_52      136    136      0 NaN      NaN 4.259292 0.9761764 41.975197
## X10_53      142    142      0 NaN      NaN 4.144321 0.9707317 34.166653
## X10_57      140    140      0 NaN      NaN 4.166506 0.9690959 32.358201
## X10_58      121    121      0 NaN      NaN 3.558282 0.9245442 13.252790
## X10_59      148    148      0 NaN      NaN 4.294089 0.9724177 36.255089
## X10_60      95     95      0 NaN      NaN 3.862023 0.9660824 29.483175
## X10_63      143    143      0 NaN      NaN 4.108655 0.9602230 25.140134
## X10_64      133    133      0 NaN      NaN 3.845720 0.9506031 20.244171
## X10_66      137    137      0 NaN      NaN 4.066091 0.9600511 25.031975
## X10_67      118    118      0 NaN      NaN 3.734614 0.9327431 14.868368
## X10_68      114    114      0 NaN      NaN 4.011428 0.9684984 31.744427
## X10_69      115    115      0 NaN      NaN 3.449319 0.9095590 11.056929
## X11_1       136    136      0 NaN      NaN 4.128541 0.9690378 32.297435
## X11_3       122    122      0 NaN      NaN 3.672527 0.9338568 15.118723
## X14_20      177    177      0 NaN      NaN 4.409254 0.9753444 40.558697
## X14_21      142    142      0 NaN      NaN 4.353094 0.9770271 43.529539
## X14_22      133    133      0 NaN      NaN 4.296036 0.9783014 46.086004
## X14_23      199    199      0 NaN      NaN 4.783165 0.9863423 73.218561
## X14_25      176    176      0 NaN      NaN 4.679120 0.9852256 67.684481
## X14_27      223    223      0 NaN      NaN 4.944154 0.9889503 90.500150
## X14_29      168    168      0 NaN      NaN 4.569564 0.9823959 56.804951
## X14_30      188    188      0 NaN      NaN 4.491616 0.9753695 40.600036
## X14_33      162    162      0 NaN      NaN 4.406904 0.9770711 43.613089
## X14_34      128    128      0 NaN      NaN 3.429396 0.8623682 7.265762
## X14_35      171    171      0 NaN      NaN 4.380340 0.9754211 40.685323
## X14_36      143    143      0 NaN      NaN 4.379227 0.9791060 47.860532
##           Fisher
## X2_23      14.163010
## X2_24      14.814971
## X2_25      9.896331
## X2_26      14.190333
## X2_27      15.623575
## X2_29      14.578601
## X2_36      12.941416
## X2_39      13.328383
## X2_40      17.140224
## X2_41      10.996498
## X2_42      12.311257
## X2_47      16.462924
## X2_48      7.588284
## X2_49      11.837343
## X2_50      15.179815
## X2_51      12.413289
## X2_52      15.242362
## X2_56      13.886854
## X2_57      4.609881

```

```
## X2_58 11.654290
## X2_59 11.399075
## X2_60 15.023335
## X2_61 15.063981
## X4_36 12.305789
## X4_37 8.716916
## X4_38 15.586719
## X4_39 11.537281
## X4_40 19.831817
## X4_41 12.171287
## X4_54 18.370424
## X4_55 14.841290
## X4_56 17.931372
## X4_57 17.960087
## X4_65 5.925936
## X5_39 7.671534
## X5_40 20.407865
## X5_41 16.292875
## X5_54 18.013456
## X5_55 14.835637
## X5_59 19.333468
## X6_36 12.683630
## X6_37 11.331777
## X6_38 14.095720
## X6_54 14.306480
## X6_55 18.506106
## X6_56 14.640803
## X6_57 13.343276
## X6_58 11.875237
## X9_16 13.450478
## X9_17 12.585239
## X9_18 13.672532
## X9_19 8.444623
## X9_21 15.750868
## X9_22 12.545120
## X9_34 12.464780
## X9_35 11.005212
## X9_36 12.046206
## X9_37 12.196534
## X9_38 10.546501
## X9_39 18.188937
## X10_1 14.801263
## X10_2 17.913932
## X10_3 15.474146
## X10_4 15.608820
## X10_7 15.627062
## X10_8 17.314410
## X10_10 17.070298
## X10_11 16.383901
## X10_12 14.722089
## X10_13 17.686567
## X10_14 14.172646
## X10_15 14.270153
## X10_19 8.022038
```

```

## X10_20 8.517057
## X10_21 12.973489
## X10_22 8.296877
## X10_25 7.848329
## X10_26 10.320954
## X10_28 8.725458
## X10_29 10.934345
## X10_30 10.995780
## X10_33 12.728324
## X10_34 9.333958
## X10_35 9.774689
## X10_39 13.503588
## X10_40 12.889365
## X10_41 15.867978
## X10_42 15.715324
## X10_43 20.904381
## X10_44 15.222496
## X10_48 17.734041
## X10_49 15.451132
## X10_50 13.353367
## X10_51 16.494365
## X10_52 14.412306
## X10_53 15.760869
## X10_57 15.870862
## X10_58 12.927447
## X10_59 17.425828
## X10_60 10.411951
## X10_63 14.552490
## X10_64 14.521762
## X10_66 14.605380
## X10_67 13.300472
## X10_68 12.965425
## X10_69 13.186276
## X11_1 13.199825
## X11_3 12.195960
## X14_20 19.698534
## X14_21 14.910568
## X14_22 14.070685
## X14_23 23.233005
## X14_25 19.924708
## X14_27 26.083193
## X14_29 18.125905
## X14_30 21.059405
## X14_33 17.598106
## X14_34 12.695835
## X14_35 18.362334
## X14_36 15.547467

dist = "bray"
ord_meths = c("DCA", "CCA", "RDA", "NMDS", "MDS", "PCoA", "DPCoA")
plist = llply(as.list(ord_meths), function(i, physeq, dist){
  ordi = ordinate(subset16S, method=i, distance=dist)
  plot_ordination(subset16S, ordi, "samples", color="Age", shape = "AB")
}, subset16S, dist)

```

```

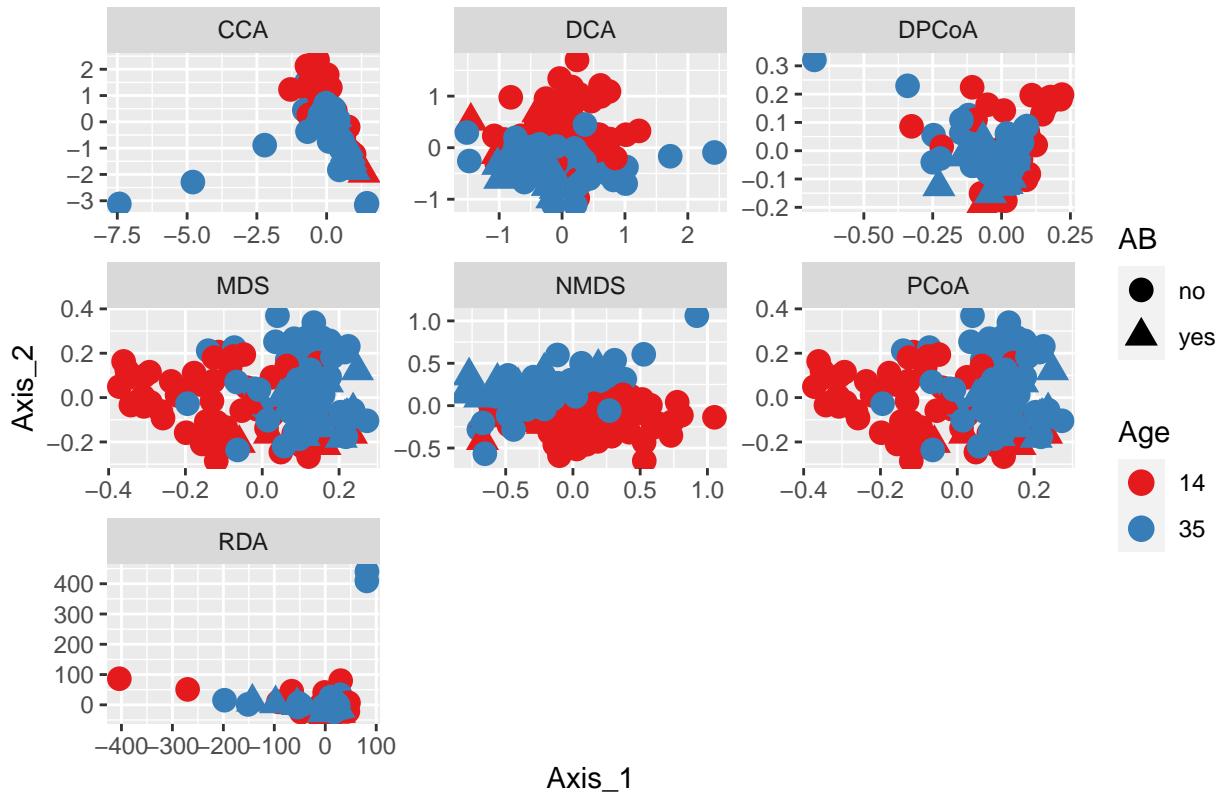
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2084961
## Run 1 stress 0.2159558
## Run 2 stress 0.2356788
## Run 3 stress 0.211886
## Run 4 stress 0.2141743
## Run 5 stress 0.2084768
## ... New best solution
## ... Procrustes: rmse 0.007183811 max resid 0.06358386
## Run 6 stress 0.2095338
## Run 7 stress 0.220354
## Run 8 stress 0.2174809
## Run 9 stress 0.2136929
## Run 10 stress 0.2204088
## Run 11 stress 0.2161384
## Run 12 stress 0.2159558
## Run 13 stress 0.2159558
## Run 14 stress 0.2203541
## Run 15 stress 0.2084813
## ... Procrustes: rmse 0.001045892 max resid 0.006152675
## ... Similar to previous best
## Run 16 stress 0.2216108
## Run 17 stress 0.212831
## Run 18 stress 0.2165475
## Run 19 stress 0.2159558
## Run 20 stress 0.2084962
## ... Procrustes: rmse 0.00721279 max resid 0.0639836
## *** Best solution repeated 1 times

names(plist) <- ord_meths

pdataframe = ldply(plist, function(x){
  df = x$data[, 1:2]
  colnames(df) = c("Axis_1", "Axis_2")
  return(cbind(df, x$data))
})
names(pdataframe)[1] = "method"
ggplot(pdataframe, aes(Axis_1, Axis_2, color=Age, shape=AB)) +
  geom_point(size=4) +
  facet_wrap(~method, scales="free") +
  scale_fill_brewer(type="qual", palette="Set1") +
  scale_colour_brewer(type="qual", palette="Set1") +
  ggtitle("Different ordination methods for 16S data (Bray-Curtis)")

```

Different ordination methods for 16S data (Bray–Curtis)



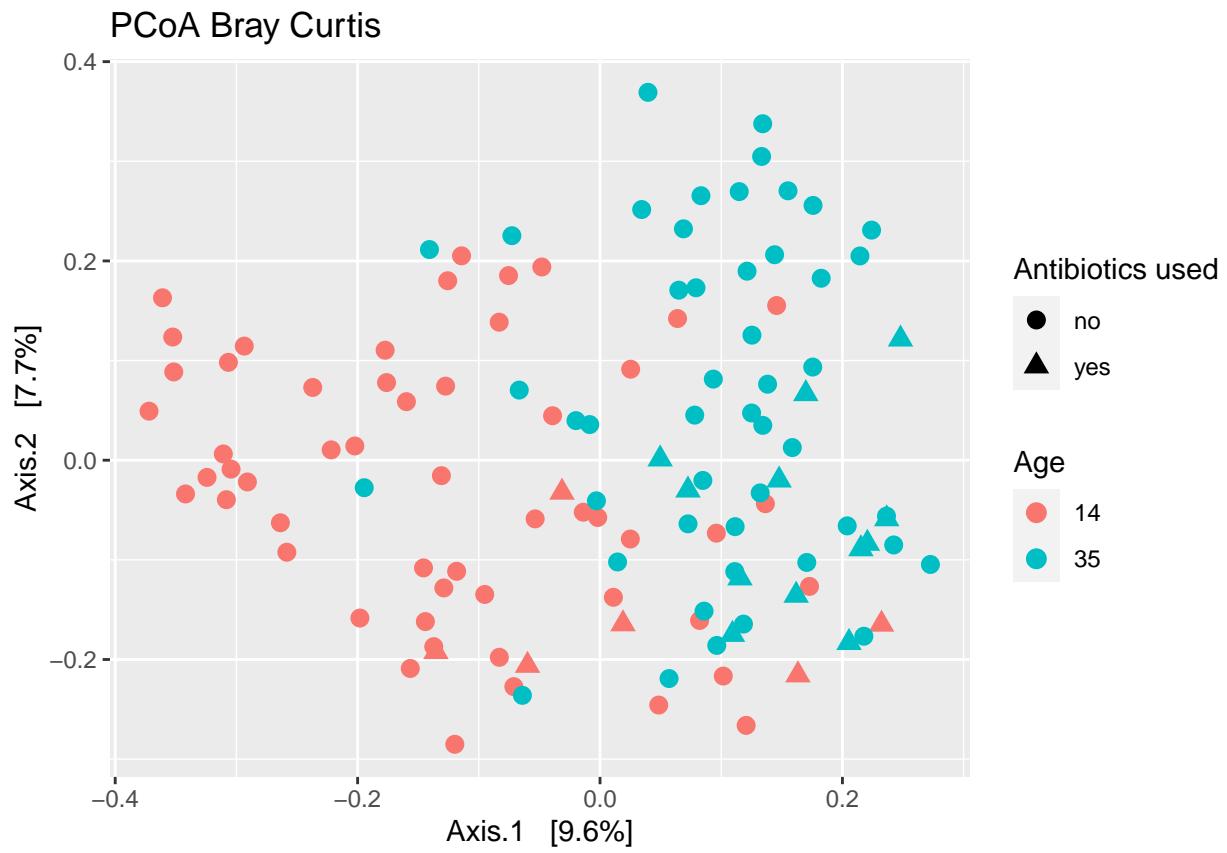
```
# PCoAs for different methods

# functionize plotting pcoa
plot_pcoa_ordination <- function(data, pcoa, var, title) {
  p <- plot_ordination(data, pcoa, color = var, shape = "AB") +
    geom_point(size = 3) +
    labs(title = title, color = var, shape = "Antibiotics used")

  return(p)
}

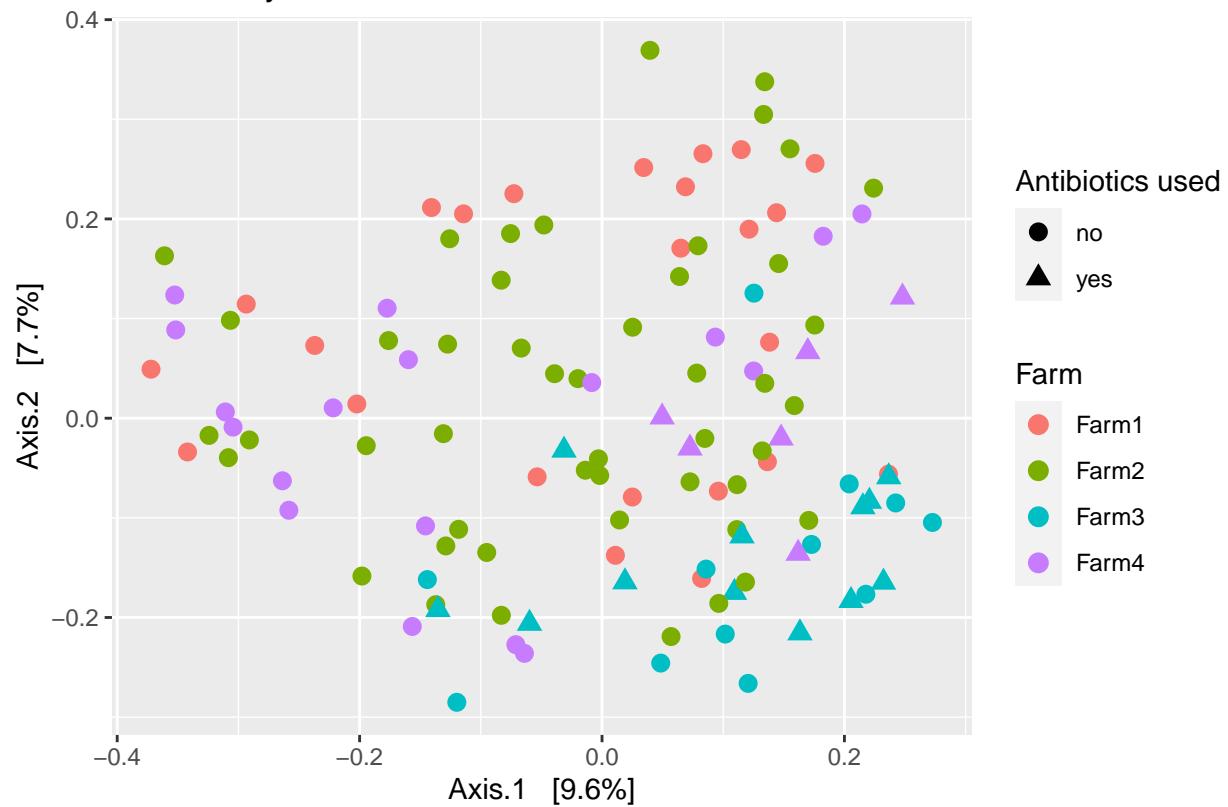
pcoa_bc = ordinate(subset16S, "PCoA", "bray")
pcoa_unifrac = ordinate(subset16S, "PCoA", "unifrac")
pcoa_wunifrac = ordinate(subset16S, "PCoA", "wunifrac")
pcoa_jsd = ordinate(subset16S, "PCoA", "jsd")
pcoa_jaccard = ordinate(subset16S, "PCoA", "jaccard", binary=TRUE)

plot_pcoa_ordination(subset16S, pcoa_bc, "Age", "PCoA Bray Curtis")
```



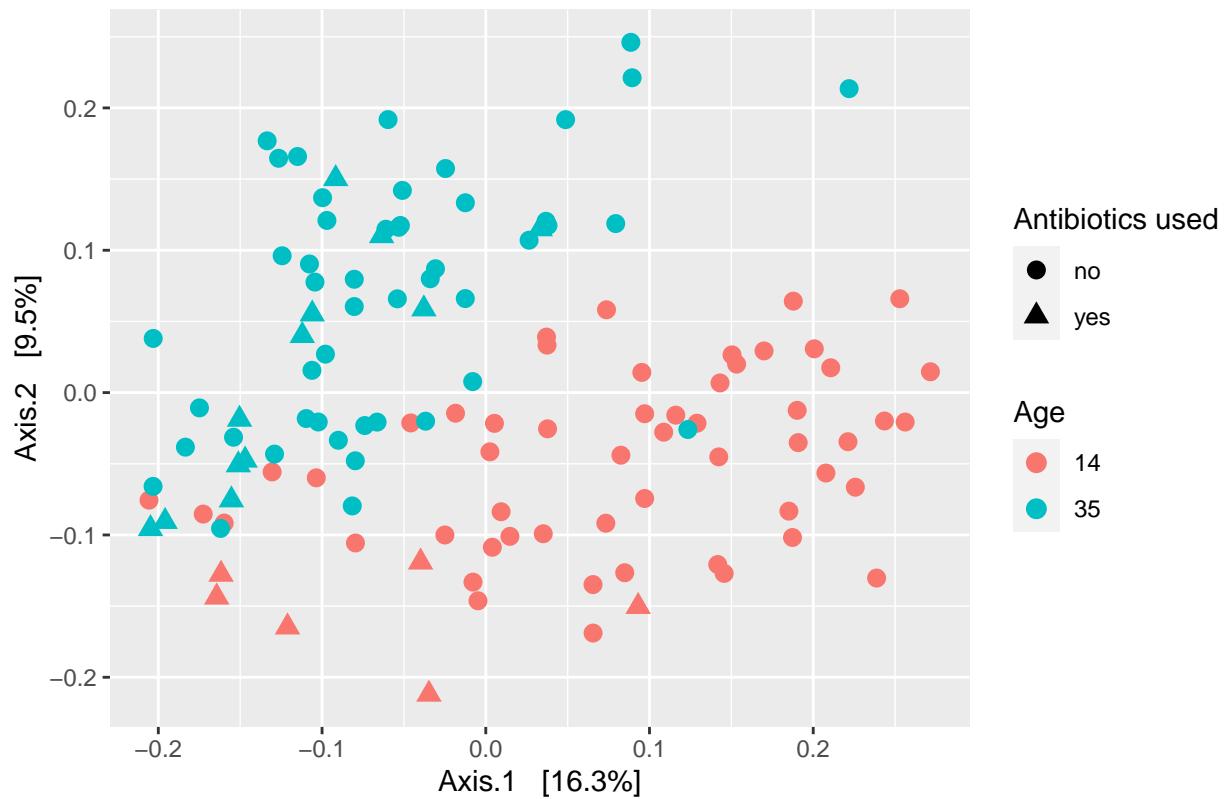
```
# proper order of legend:
plot_ordination(subset16S, pcoa_bc, color = "Farm2", shape = "AB") +
  geom_point(size = 3) +
  labs(title = "PCoA Bray Curtis", color = "Farm", shape = "Antibiotics used")
```

PCoA Bray Curtis



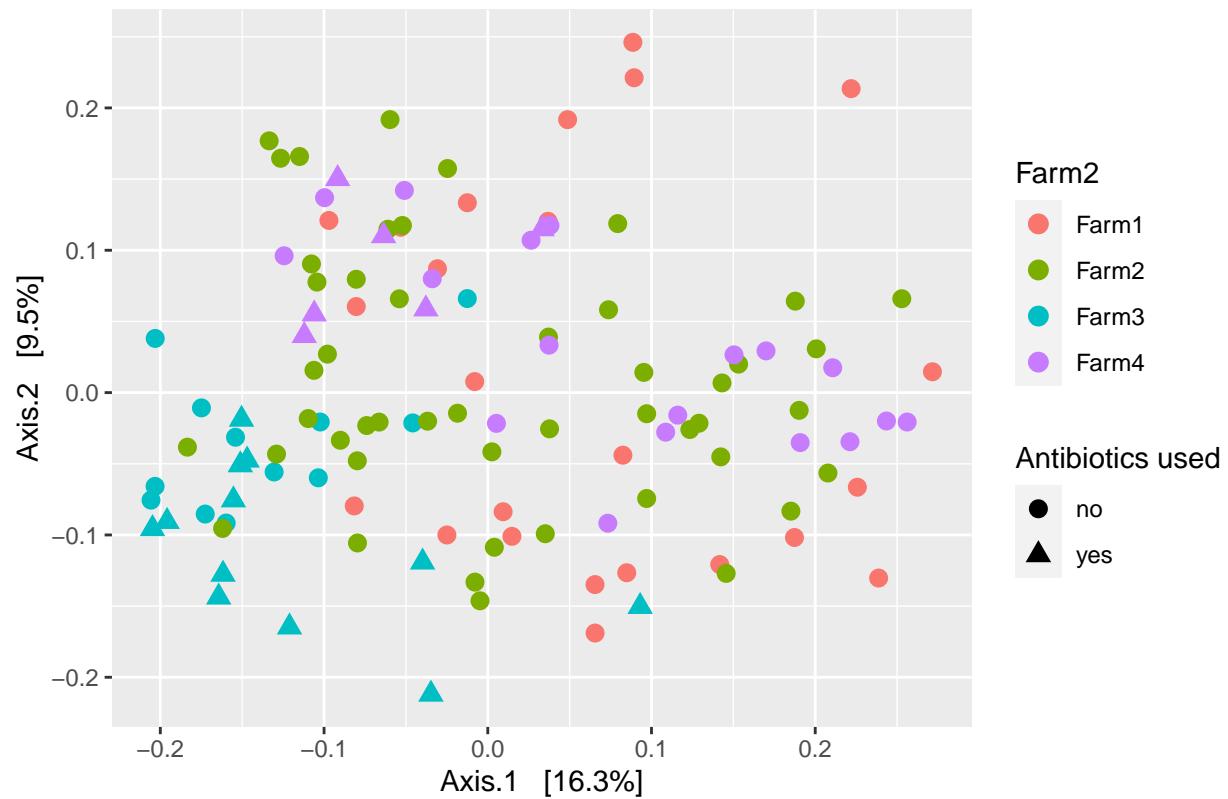
```
plot_pcoa_ordination(subset16S, pcoa_unifrac, "Age", "PCoA Unifrac")
```

PCoA Unifrac



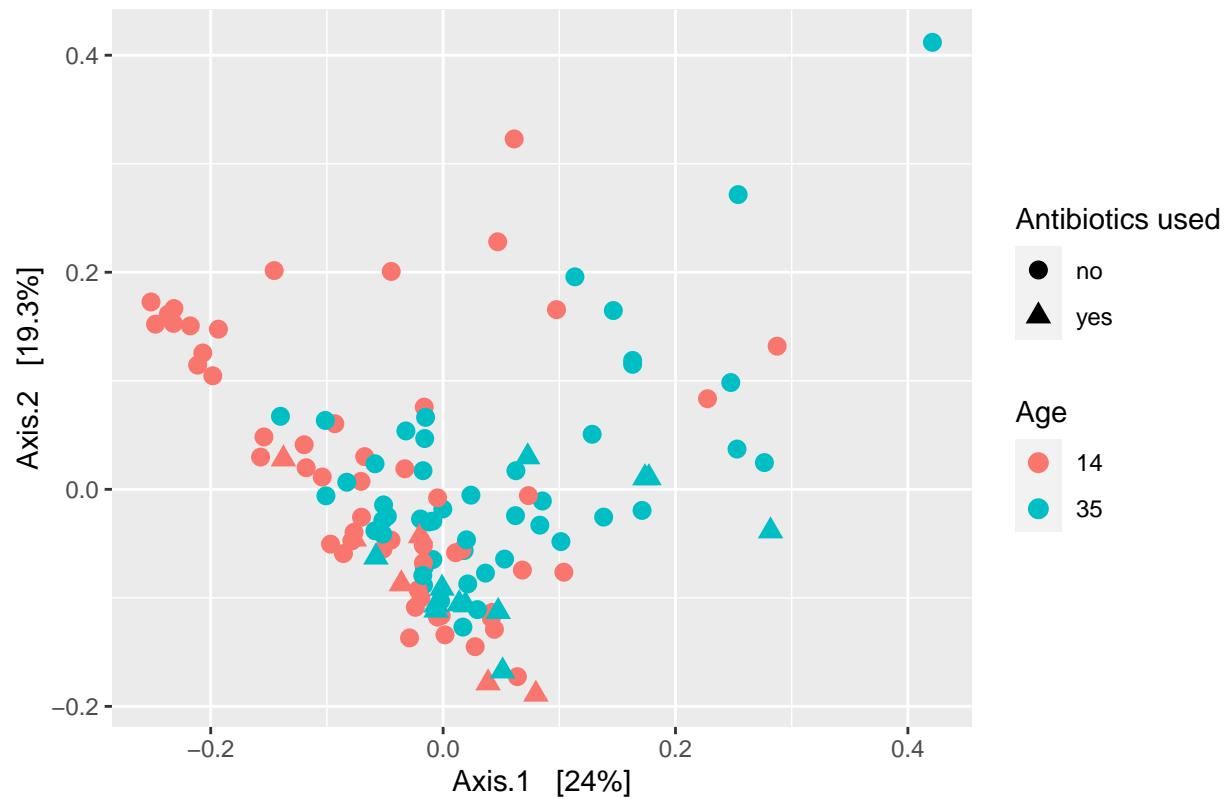
```
plot_pcoa_ordination(subset16S, pcoa_unifrac, "Farm2", "PCoA Unifrac")
```

PCoA Unifrac



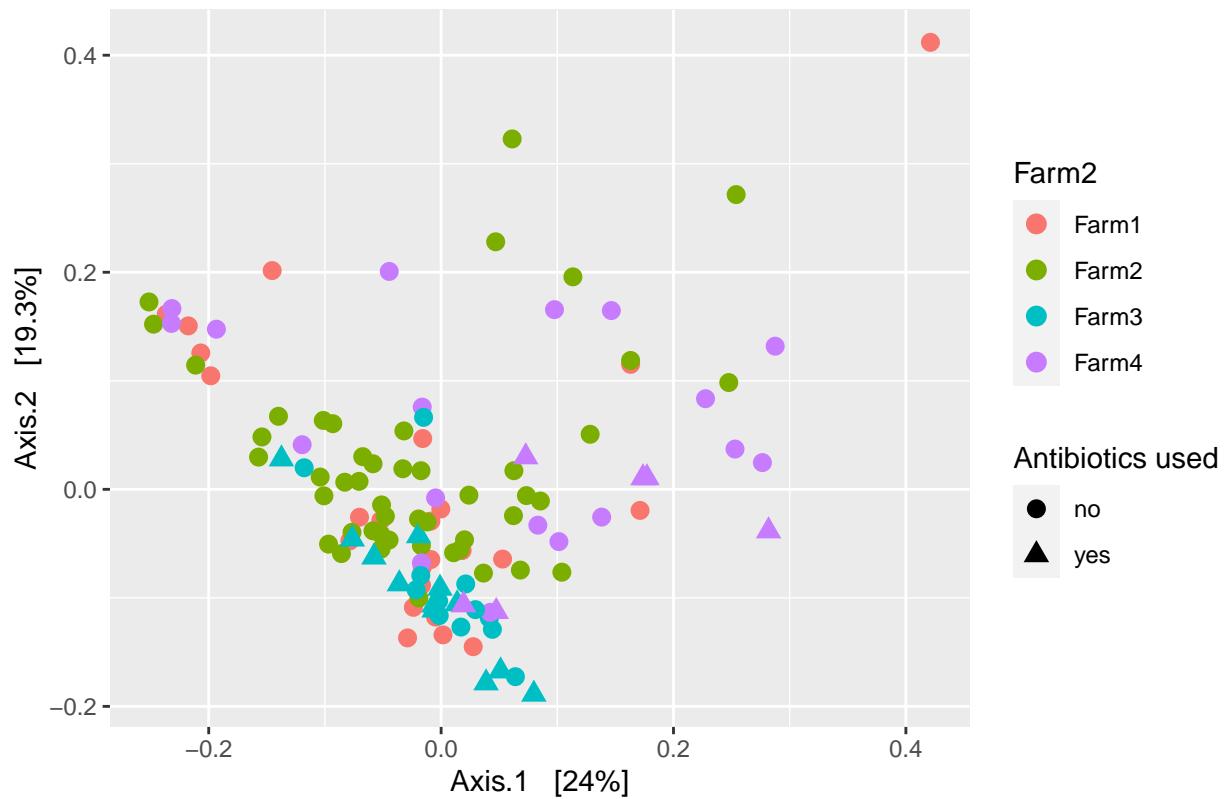
```
plot_pcoa_ordination(subset16S, pcoa_wunifrac, "Age", "PCoA Weighted Unifrac")
```

PCoA Weighted Unifrac



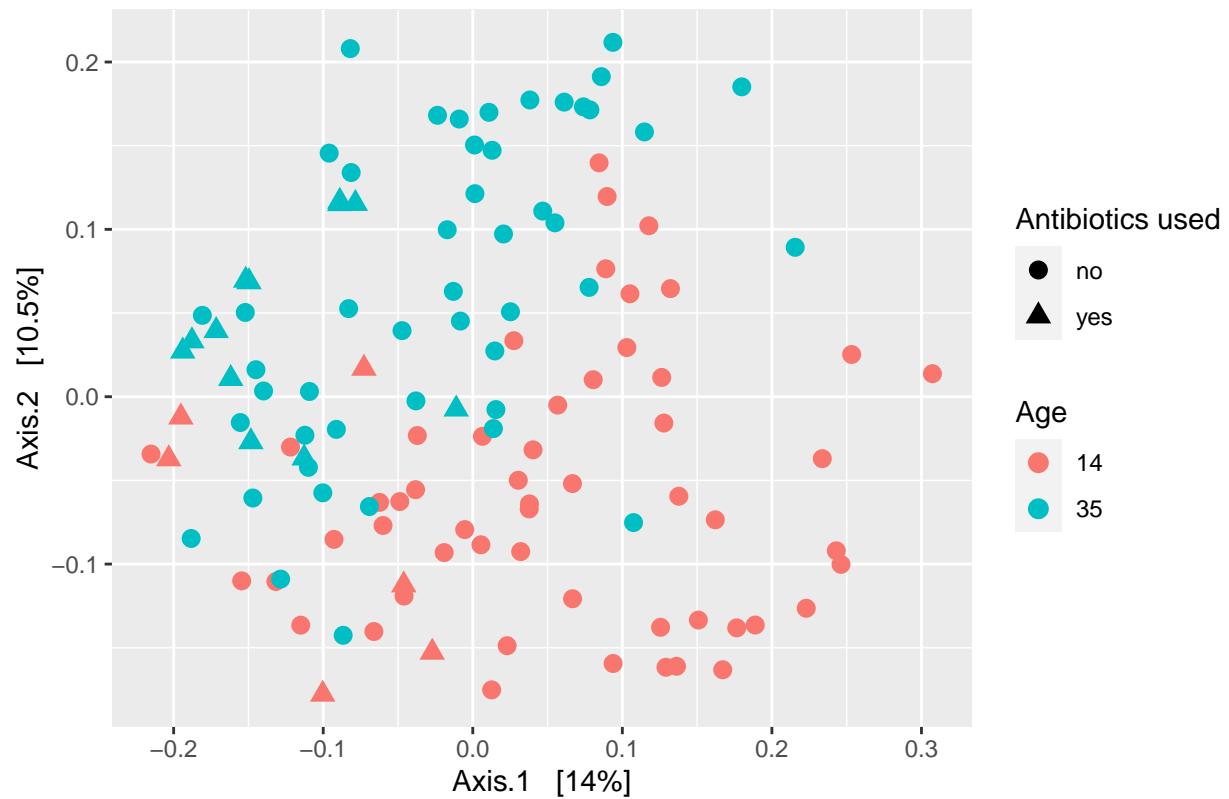
```
plot_pcoa_ordination(subset16S, pcoa_wunifrac, "Farm2", "PCoA Weighted Unifrac")
```

PCoA Weighted Unifrac



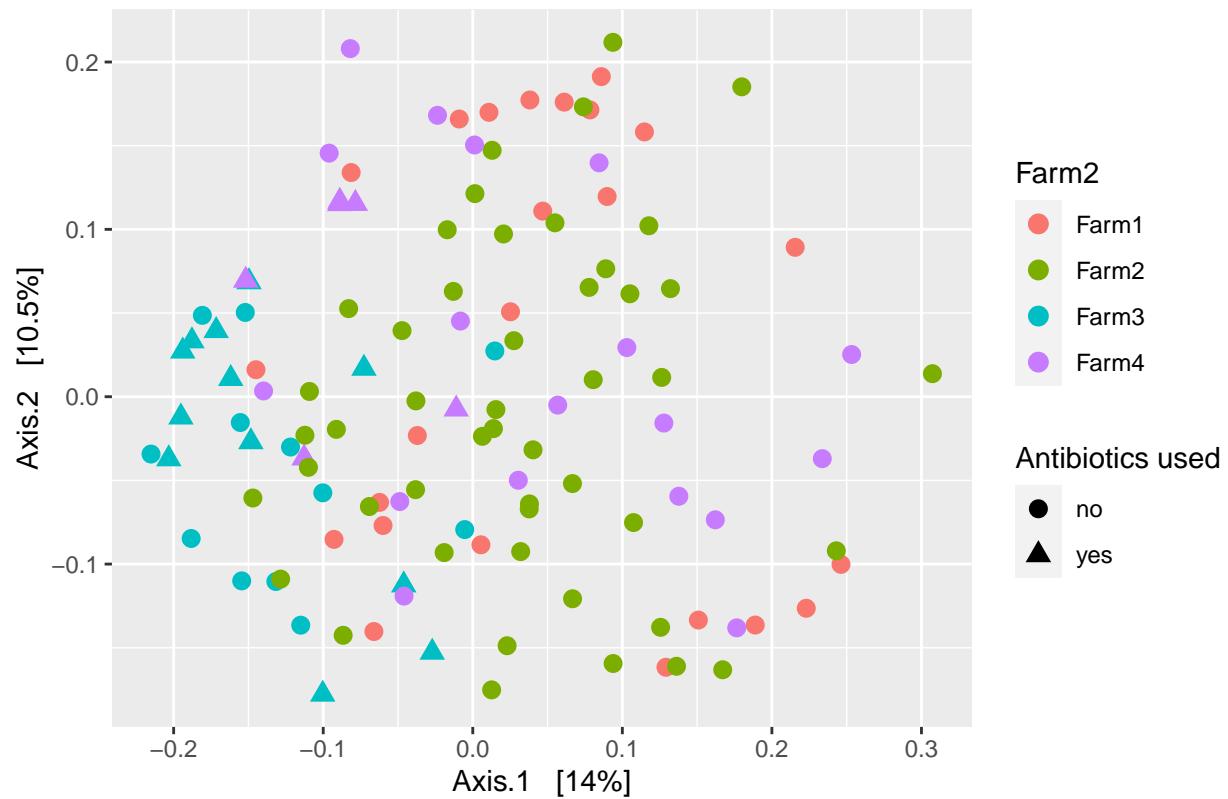
```
plot_pcoa_ordination(subset16S, pcoa_jsd, "Age", "PCoA Jensen-Shannon Divergence")
```

PCoA Jensen–Shannon Divergence



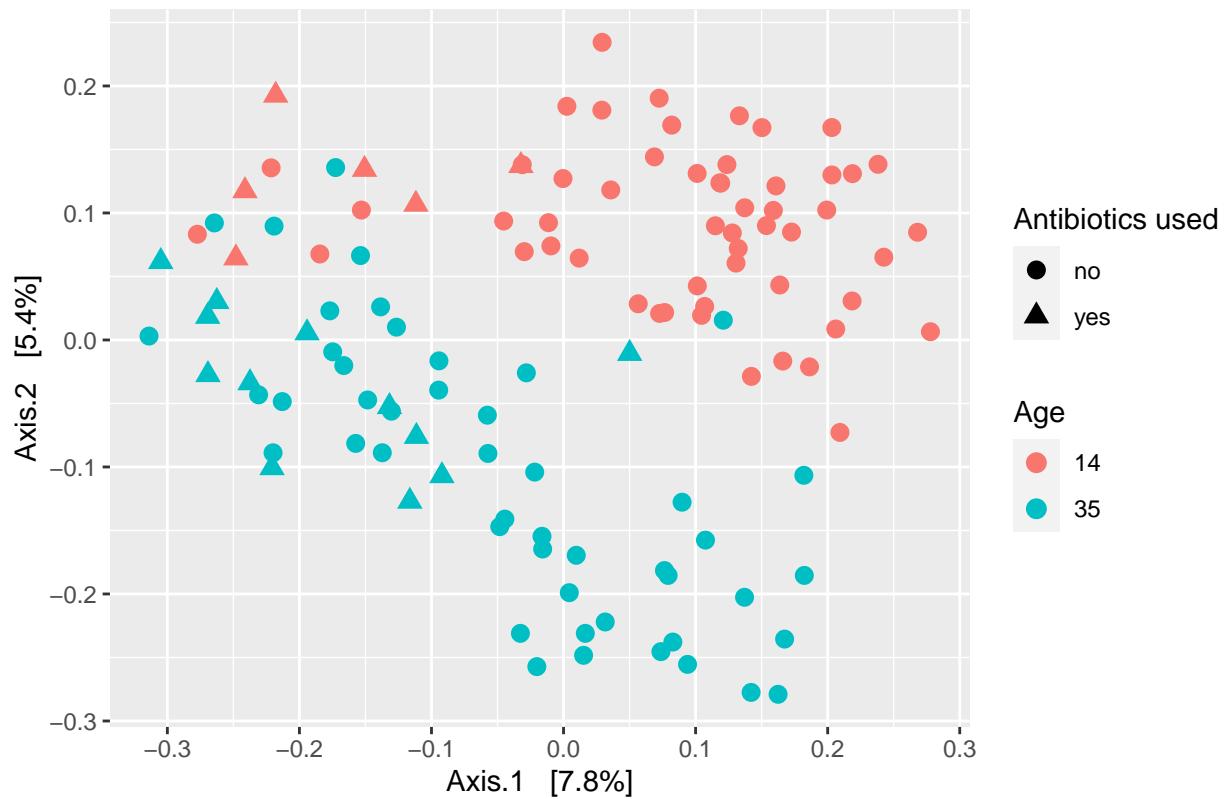
```
plot_pcoa_ordination(subset16S, pcoa_jsd, "Farm2", "PCoA Jensen–Shannon Divergence")
```

PCoA Jensen–Shannon Divergence



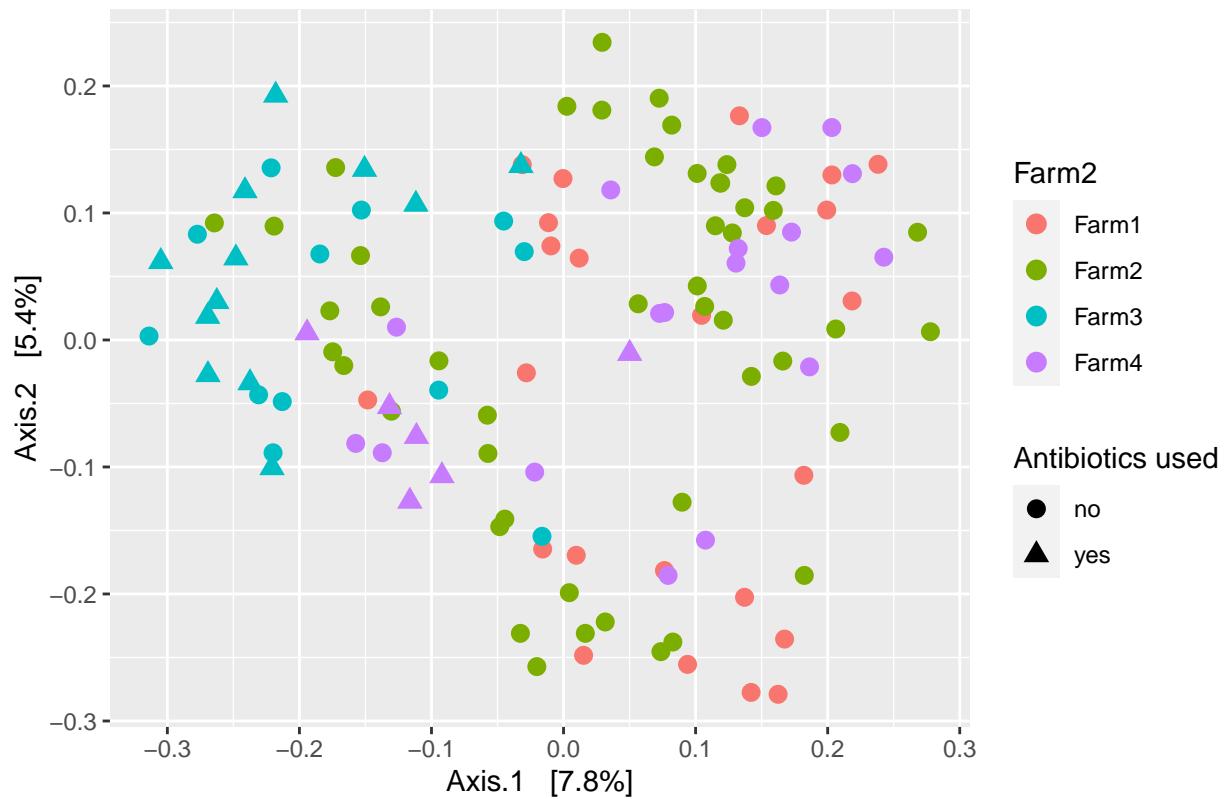
```
plot_pcoa_ordination(subset16S, pcoa_jaccard, "Age", "PCoA Jaccard")
```

PCoA Jaccard

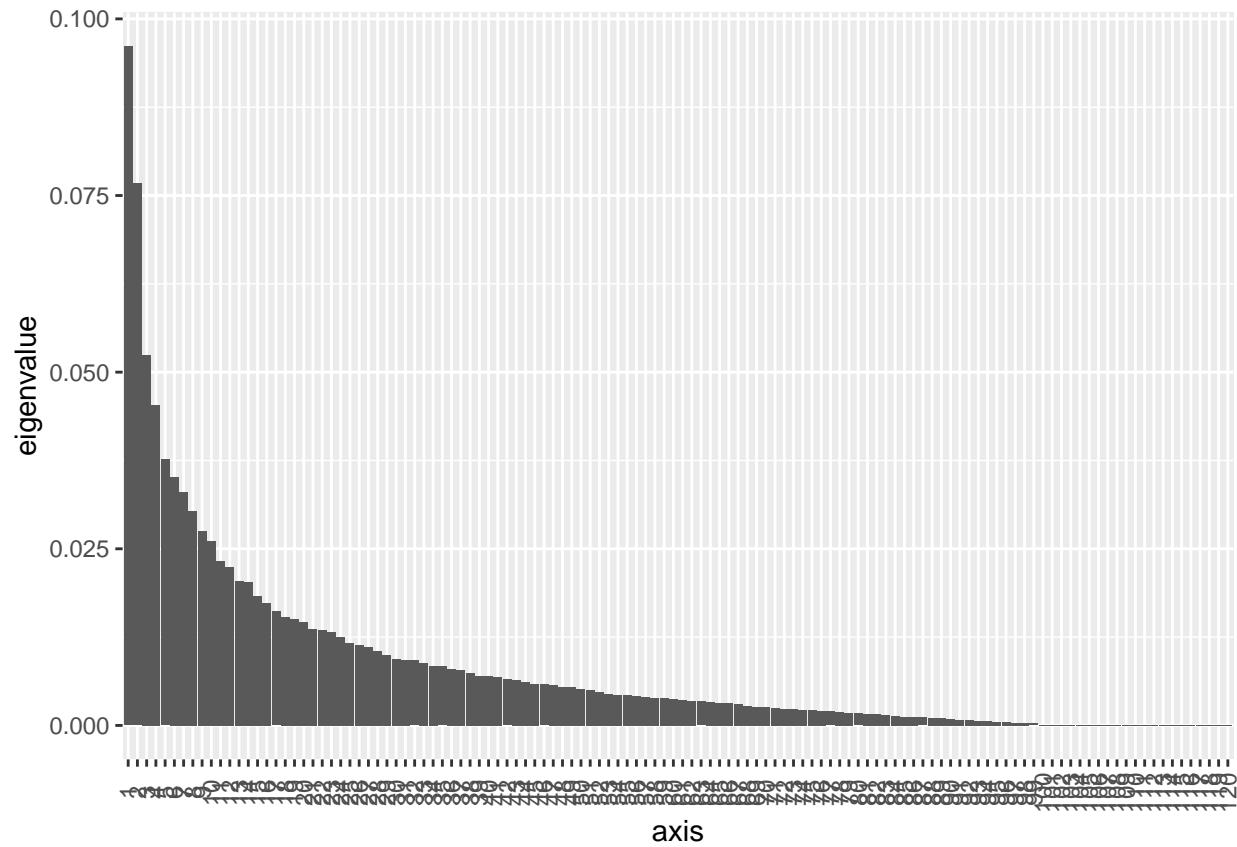


```
plot_pcoa_ordination(subset16S, pcoa_jaccard, "Farm2", "PCoA Jaccard")
```

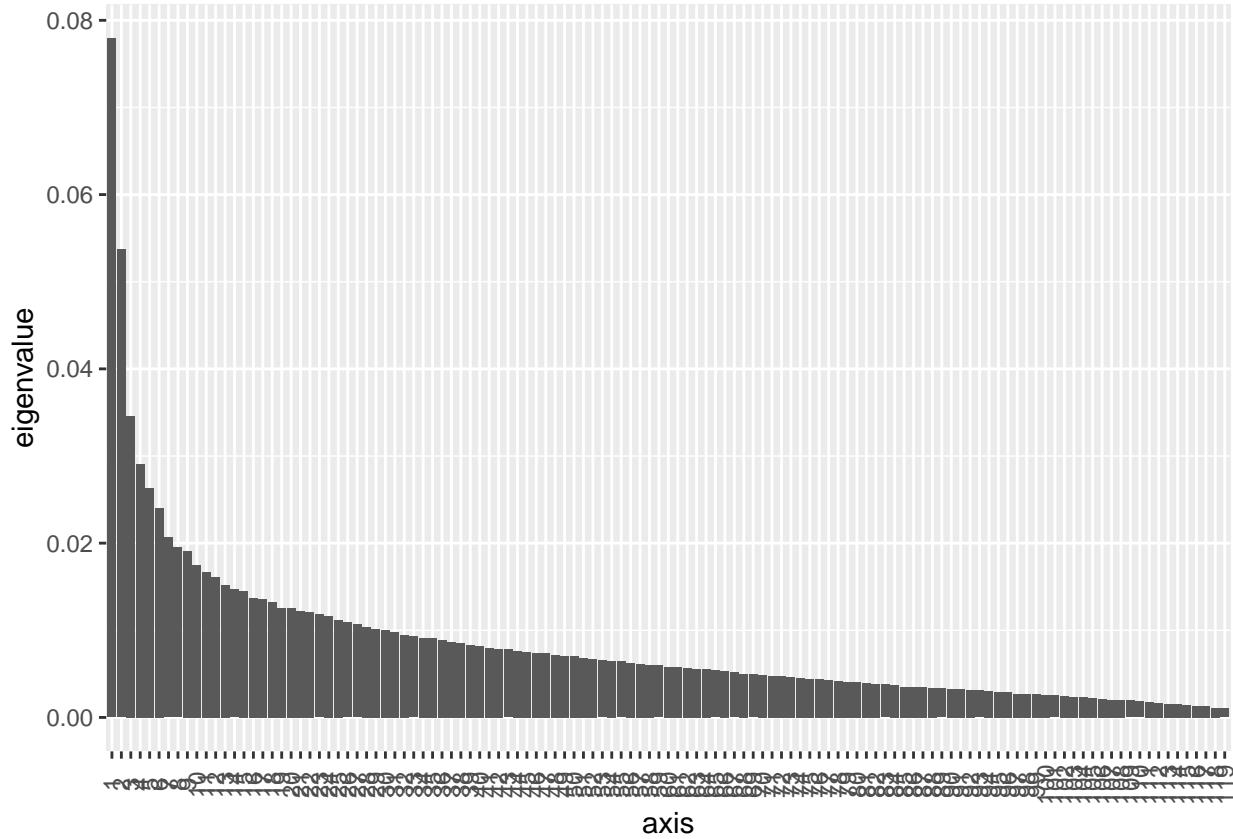
PCoA Jaccard



```
#scree plots can be made for any of the PCoAs, those that explain less than 10% of variance on first axis
plot_scree(pcoa_bc)
```



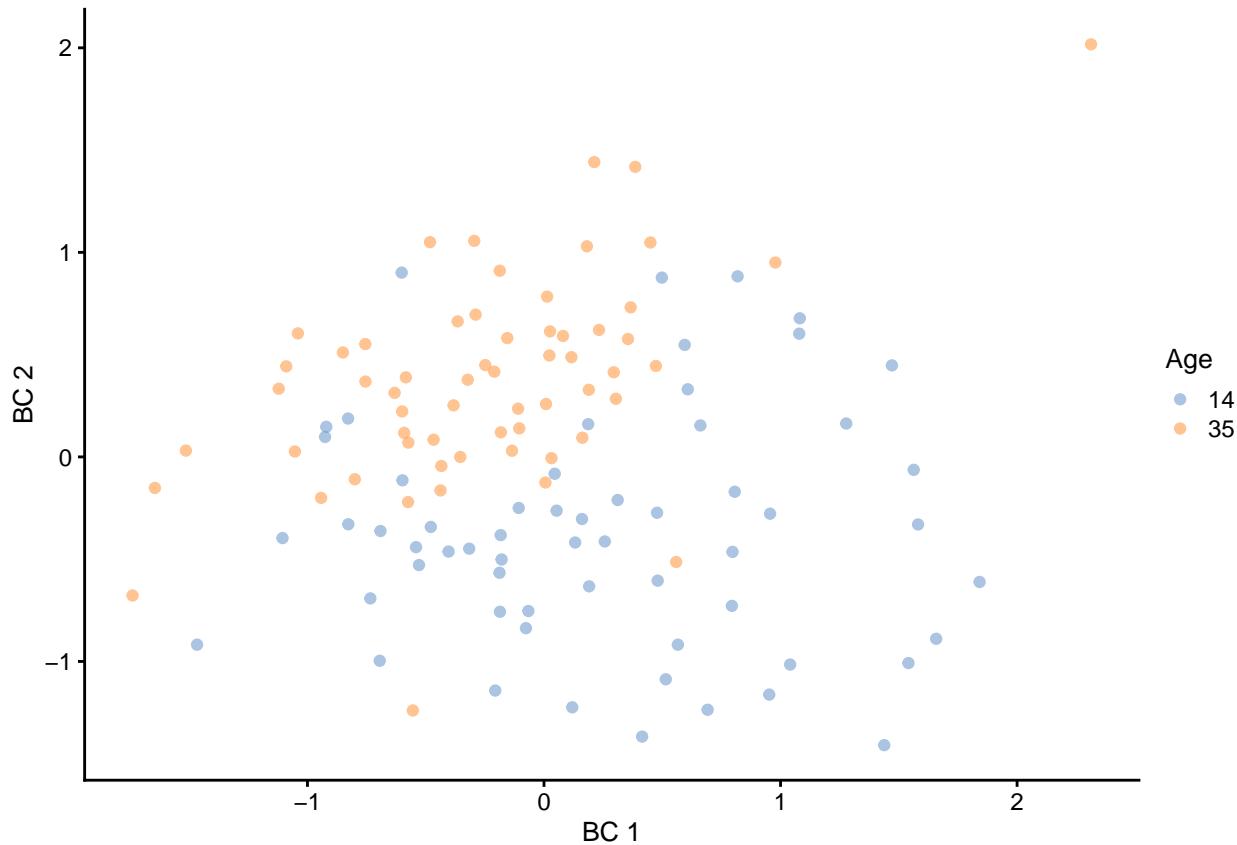
```
plot_scree(pcoa_jaccard)
```



```
# NMDS

tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)
tse %<% transformCounts( method = "relabundance")
tse %<% runNMDS(FUN = vegan::vegdist, name = "BC", nmdsFUN = "monoMDS",
                     exprs_values = "relabundance",
                     keep_dist = TRUE)

tse %>% plotReducedDim("BC", colour_by = "Age")
```



PERMANOVAs

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)
tse <- transformCounts(tse, method = "relabundance")

adonis2(t(assay(tse, "relabundance")) ~ AB, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ AB, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## AB        1   0.867 0.02591 3.139  1e-04 ***
## Residual 118   32.594 0.97409
## Total    119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Cox, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
```

```

## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Cox, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox       3     3.424 0.10233 4.4078  1e-04 ***
## Residual 116   30.037 0.89767
## Total    119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Researcher, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Researcher, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher 4     1.942 0.05803 1.7711  1e-04 ***
## Residual 115   31.520 0.94197
## Total    119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ FeedProducent, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ FeedProducent, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedProducent 2     2.262 0.0676 4.241  1e-04 ***
## Residual 117   31.200 0.9324
## Total    119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ LitterType, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ LitterType, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType 2     2.054 0.06137 3.825  1e-04 ***
## Residual 117   31.408 0.93863
## Total    119   33.462 1.00000

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ FeedType, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ FeedType, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedType    1     1.828 0.05464 6.8206  1e-04 ***
## Residual 118    31.633 0.94536
## Total     119    33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Gender, data = colData(tse), permutations = 9999) # NOT significant

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Gender, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender     1     0.247 0.00739 0.8784 0.6775
## Residual 118    33.214 0.99261
## Total     119    33.462 1.00000

adonis2(t(assay(tse, "relabundance")) ~ Stables, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Stables, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables    9     5.606 0.16753 2.4596  1e-04 ***
## Residual 110    27.856 0.83247
## Total     119    33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ FlockSize, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free

```

```

## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ FlockSize, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FlockSize    5     4.167 0.12453 3.2432  1e-04 ***
## Residual   114    29.295 0.87547
## Total      119    33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Farm2, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Farm2, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2       3     3.452 0.10315 4.4474  1e-04 ***
## Residual   116    30.010 0.89685
## Total      119    33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ AgeParentStock, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ AgeParentStock, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## AgeParentStock  4     3.980 0.11895 3.8817  1e-04 ***
## Residual      115    29.481 0.88105
## Total         119    33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Age, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Age, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Age          1     1.828 0.05464 6.8206  1e-04 ***
## Residual    118    31.633 0.94536
## Total       119    33.462 1.00000

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# variances: AB: 0.026, Cox: 0.102, Researcher: 0.06, FP : 0.067, LitterType: 0.061, FT :0.055, Gender:
# Stable: 0.167, FS: 0.1245, Farm 0.103, APS : 0.118, Age: 0.054
# Order: Stable>FS>APS>Farm>Cox>FP>LT>Researcher>FT>Age>AB>Gender

# Mixed models ( out of scope)
#adonis2(t(assay(tse, "relabundance")) ~ Stables * AB, data = colData(tse), permutations = 9999)

# basically, composition seems to be different over every single variable, except for gender

# on genus level
tse_genus <- agglomerateByRank(tse, "Genus")
tse_genus <- transformCounts(tse_genus, method = "relabundance")

adonis2(t(assay(tse_genus, "relabundance")) ~ AB, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ AB, data = colData(tse_genus), permutations =
##          Df SumOfSqs      R2      F Pr(>F)
##          AB        1  0.4471 0.03209 3.9117  5e-04 ***
##          Residual 118 13.4862 0.96791
##          Total    119 13.9333 1.00000
##          ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ Cox, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ Cox, data = colData(tse_genus), permutations =
##          Df SumOfSqs      R2      F Pr(>F)
##          Cox       3  1.505 0.10801 4.6823  1e-04 ***
##          Residual 116 12.428 0.89199
##          Total    119 13.933 1.00000
##          ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ Researcher, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free

```

```

## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ Researcher, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher    4   1.0471 0.07515 2.3362  1e-04 ***
## Residual    115  12.8862 0.92485
## Total       119  13.9333 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ FeedProducent, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ FeedProducent, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedProducent  2   0.9176 0.06586 4.1241  1e-04 ***
## Residual     117  13.0157 0.93414
## Total        119  13.9333 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ LitterType, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ LitterType, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType    2   0.6996 0.05021 3.0926  2e-04 ***
## Residual     117  13.2337 0.94979
## Total        119  13.9333 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ FeedType, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ FeedType, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedType     1   0.9844 0.07065 8.9705  1e-04 ***
## Residual    118  12.9489 0.92935
## Total       119  13.9333 1.00000

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ Gender, data = colData(tse_genus), permutations = 9999) #>

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ Gender, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender     1   0.1175 0.00843 1.0033 0.4312
## Residual  118  13.8158 0.99157
## Total     119  13.9333 1.00000

adonis2(t(assay(tse_genus, "relabundance")) ~ Stables, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ Stables, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables    9   2.3359 0.16765 2.4618  1e-04 ***
## Residual  110  11.5974 0.83235
## Total     119  13.9333 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ FlockSize, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ FlockSize, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FlockSize   5   1.7048 0.12235 3.1785  1e-04 ***
## Residual  114  12.2285 0.87765
## Total     119  13.9333 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ Farm2, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free

```

```

## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ Farm2, data = colData(tse_genus), permutations =
##           Df SumOfSqs      R2      F Pr(>F)
## Farm2      3   1.3884 0.09965 4.2795  1e-04 ***
## Residual  116  12.5449 0.90035
## Total     119  13.9333 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ AgeParentStock, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ AgeParentStock, data = colData(tse_genus), permutations = 9999)
##           Df SumOfSqs      R2      F Pr(>F)
## AgeParentStock  4   1.5725 0.11286 3.6576  1e-04 ***
## Residual       115  12.3608 0.88714
## Total          119  13.9333 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# same trends on genus level (and on phylum level, though p values become higher)

```

for different ordination methods

```

ps1.rel <- microbiome::transform(subset16S, "compositional")
metadf <- data.frame(sample_data(ps1.rel))

# alternative calculations
#otu <- abundances(ps1.rel)
#meta <- meta(ps1.rel)
#adonis2(t(otu) ~ Age, data = meta, permutations=9999, method = "bray")

#permanova = adonis(t(otu) ~ Age, data = meta, permutations=9999, method = "bray")
#permanova$aov.tab

unifrac.dist <- UniFrac(ps1.rel)

adonis2(unifrac.dist ~ Age, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Age, data = metadf)

```

```

##          Df SumOfSqs      R2      F Pr(>F)
## Age       1  1.0901 0.09194 11.947  0.001 ***
## Residual 118 10.7664 0.90806
## Total    119 11.8565 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

adonis2(unifrac.dist ~ AB, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB        1  0.4532 0.03822 4.6893  0.001 ***
## Residual 118 11.4033 0.96178
## Total    119 11.8565 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

adonis2(unifrac.dist ~ Farm2, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2     3  1.5651 0.13201 5.8806  0.001 ***
## Residual 116 10.2914 0.86799
## Total    119 11.8565 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

adonis2(unifrac.dist ~ Cox, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox       3  1.8178 0.15331 7.0015  0.001 ***
## Residual 116 10.0388 0.84669
## Total    119 11.8565 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```

adonis2(unifrac.dist ~ Researcher, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher    4   0.8538 0.07201 2.231  0.001 ***
## Residual    115   11.0027 0.92799
## Total       119   11.8565 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(unifrac.dist ~ LitterType, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ LitterType, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType    2   0.7815 0.06591 4.1281  0.001 ***
## Residual    117   11.0750 0.93409
## Total       119   11.8565 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(unifrac.dist ~ Gender, data = metadf) # not sign
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Gender, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender       1   0.0947 0.00799 0.9502  0.476
## Residual   118   11.7618 0.99201
## Total       119   11.8565 1.00000

```

```
adonis2(unifrac.dist ~ Stables, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Stables, data = metadf)

```

```

##          Df SumOfSqs      R2      F Pr(>F)
## Stables     9   2.5652 0.21636 3.3744  0.001 ***
## Residual 110   9.2913 0.78364
## Total     119  11.8565 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

same patterns arise

```
wunifrac.dist <- UniFrac(ps1.rel,
                           weighted = TRUE)
```

```
adonis2(wunifrac.dist ~ Age, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Age, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Age       1   0.4772 0.06566 8.2918  0.001 ***
## Residual 118   6.7911 0.93434
## Total     119   7.2683 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
adonis2(wunifrac.dist ~ AB, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB        1   0.2231 0.03069 3.7362  0.002 **
## Residual 118   7.0452 0.96931
## Total     119   7.2683 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
adonis2(wunifrac.dist ~ Farm2, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2     3   0.7339 0.10098 4.343  0.001 ***

```

```

## Residual 116  6.5344 0.89902
## Total     119  7.2683 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(wunifrac.dist ~ Cox, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox       3   0.7493 0.1031 4.4446  0.001 ***
## Residual 116   6.5190 0.8969
## Total     119   7.2683 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(wunifrac.dist ~ Researcher, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher 4   0.5650 0.07773 2.4232  0.002 **
## Residual 115   6.7033 0.92227
## Total     119   7.2683 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(wunifrac.dist ~ LitterType, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ LitterType, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType 2   0.3604 0.04959 3.0524  0.001 ***
## Residual 117   6.9079 0.95041
## Total     119   7.2683 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(wunifrac.dist ~ Gender, data = metadf) # not sign
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Gender, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender     1   0.0359 0.00493 0.5849  0.848
## Residual 118   7.2325 0.99507
## Total     119   7.2683 1.00000
```

```
adonis2(wunifrac.dist ~ Stables, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Stables, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables    9   1.2663 0.17422 2.5785  0.001 ***
## Residual 110   6.0021 0.82578
## Total     119   7.2683 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# same patterns
```

```
jsd.dist <- phyloseq::distance(ps1.rel, "jsd")
```

```
adonis2(jsd.dist ~ Age, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Age, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Age        1   0.8836 0.07447 9.4942  0.001 ***
## Residual 118   10.9818 0.92553
## Total     119   11.8654 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(jsd.dist ~ AB, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
```

```

## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB          1    0.4364 0.03678 4.5059  0.001 ***
## Residual  118   11.4290 0.96322
## Total     119   11.8654 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jsd.dist ~ Farm2, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2       3    1.6671 0.1405 6.3206  0.001 ***
## Residual  116   10.1983 0.8595
## Total     119   11.8654 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jsd.dist ~ Cox, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox         3    1.6355 0.13783 6.1817  0.001 ***
## Residual  116   10.2299 0.86217
## Total     119   11.8654 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jsd.dist ~ Researcher, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher  4    0.8256 0.06958 2.1499  0.001 ***
## Residual   115   11.0398 0.93042

```

```
## Total      119 11.8654 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(jsd.dist ~ LitterType, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ LitterType, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType   2    0.970 0.08175 5.208  0.001 ***
## Residual   117   10.895 0.91825
## Total     119   11.865 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(jsd.dist ~ Gender, data = metadf) # not sign
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Gender, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender      1    0.0978 0.00824 0.9809  0.455
## Residual  118   11.7676 0.99176
## Total     119   11.8654 1.00000
```

```
adonis2(jsd.dist ~ Stables, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Stables, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables     9    2.5398 0.21405 3.3287  0.001 ***
## Residual  110    9.3256 0.78595
## Total     119   11.8654 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# same is true for JSD
```

```
bray.dist <- phyloseq::distance(ps1.rel, "bray")
```

```
adonis2(bray.dist ~ Age, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Age, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Age       1     1.828 0.05464 6.8206  0.001 ***
## Residual 118   31.633 0.94536
## Total    119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
adonis2(bray.dist ~ AB, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB        1     0.867 0.02591 3.139  0.001 ***
## Residual 118   32.594 0.97409
## Total    119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
adonis2(bray.dist ~ Farm2, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2     3     3.452 0.10315 4.4474  0.001 ***
## Residual 116   30.010 0.89685
## Total    119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
adonis2(bray.dist ~ Cox, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)

```

```

## Cox      3    3.424 0.10233 4.4078  0.001 ***
## Residual 116   30.037 0.89767
## Total     119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(bray.dist ~ Researcher, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Researcher, data = metadf)
##           Df SumOfSqs      R2      F Pr(>F)
## Researcher  4    1.942 0.05803 1.7711  0.001 ***
## Residual   115   31.520 0.94197
## Total      119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(bray.dist ~ LitterType, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ LitterType, data = metadf)
##           Df SumOfSqs      R2      F Pr(>F)
## LitterType  2    2.054 0.06137 3.825  0.001 ***
## Residual   117   31.408 0.93863
## Total      119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(bray.dist ~ Gender, data = metadf) # not sign
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Gender, data = metadf)
##           Df SumOfSqs      R2      F Pr(>F)
## Gender     1    0.247 0.00739 0.8784  0.702
## Residual  118   33.214 0.99261
## Total     119   33.462 1.00000

```

```
adonis2(bray.dist ~ Stables, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Stables, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables     9    5.606 0.16753 2.4596  0.001 ***
## Residual 110   27.856 0.83247
## Total     119   33.462 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# and BC

jaccard.dist <- phyloseq::distance(ps1.rel, "jaccard")

adonis2(jaccard.dist ~ Age, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Age, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Age        1    1.594 0.03685 4.5146  0.001 ***
## Residual 118   41.670 0.96315
## Total     119   43.264 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(jaccard.dist ~ AB, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB         1    0.813 0.01879 2.2592  0.001 ***
## Residual 118   42.451 0.98121
## Total     119   43.264 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(jaccard.dist ~ Farm2, data = metadf)

```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free

```

```

## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2      3    3.226 0.07456 3.1151  0.001 ***
## Residual  116   40.038 0.92544
## Total     119   43.264 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jaccard.dist ~ Cox, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox       3    3.172 0.07331 3.0589  0.001 ***
## Residual  116   40.092 0.92669
## Total     119   43.264 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jaccard.dist ~ Researcher, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher 4    2.091 0.04833 1.4601  0.001 ***
## Residual   115   41.173 0.95167
## Total      119   43.264 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jaccard.dist ~ LitterType, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ LitterType, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType 2    1.950 0.04508 2.7616  0.001 ***
## Residual   117   41.314 0.95492
## Total      119   43.264 1.00000

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(jaccard.dist ~ Gender, data = metadf) # not sign

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Gender, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender     1    0.332 0.00766 0.9111  0.726
## Residual 118   42.932 0.99234
## Total     119   43.264 1.00000

adonis2(jaccard.dist ~ Stables, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Stables, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables    9    5.818 0.13447 1.8989  0.001 ***
## Residual 110   37.446 0.86553
## Total     119   43.264 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# as well as jaccard

```

PERMANOVA plots - Age

```

permanova_age <- adonis(t(assay(tse, "relabundance")) ~ Age, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_age)[["Age1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                             unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

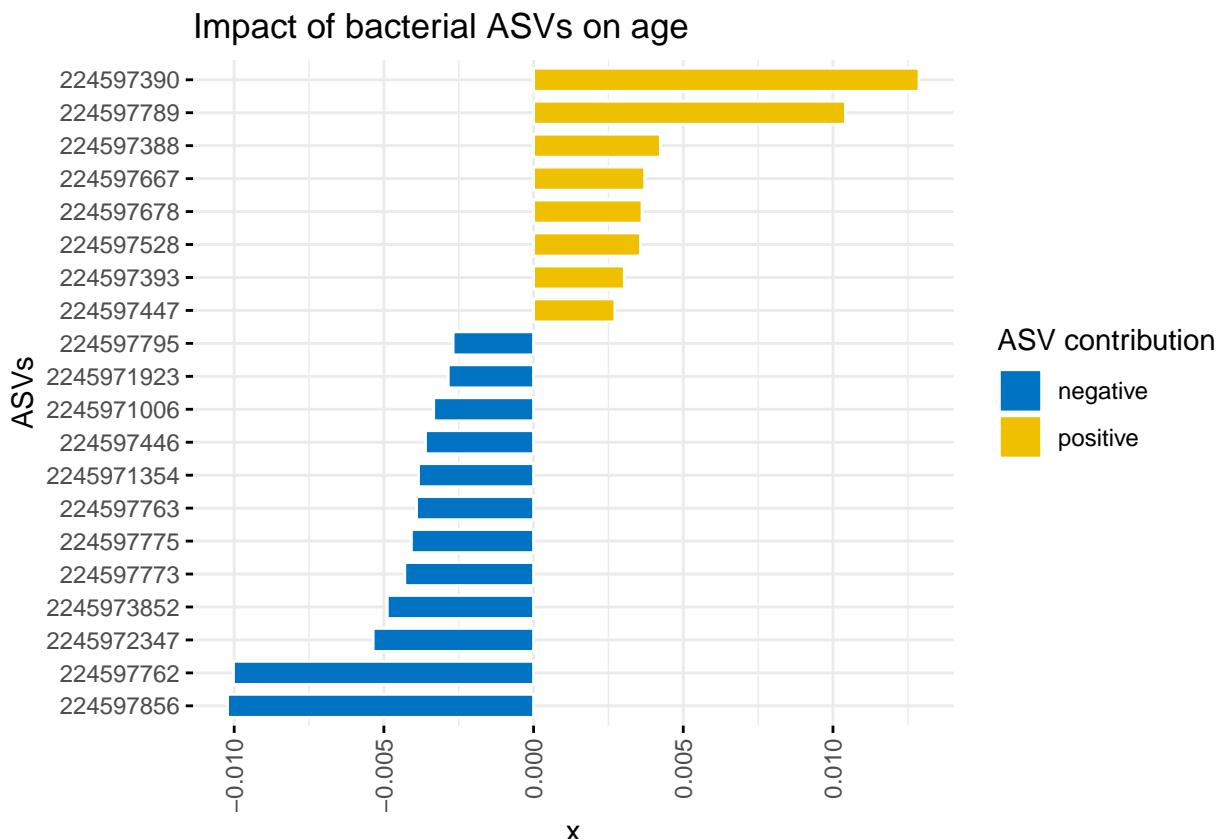
ggbarplot(df, x = "y", y = "x",
           fill = "contr",           # change fill color by mpg_level
           color = "white",          # Set bar border colors to white
           palette = "jco",           # jco journal color palett. see ?ggpar
           sort.val = "asc",          # Sort the value in ascending order

```

```

sort.by.groups = FALSE,      # Don't sort inside each group
x.text.angle = 90,          # Rotate vertically x axis texts
xlab = "ASVs",
legend.title = "ASV contribution",
title = "Impact of bacterial ASVs on age",
rotate = TRUE,
ggtheme = theme_minimal())

```



Antibiotic treatment

```

permanova_AB <- adonis(t(assay(tse, "relabundance")) ~ AB, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_AB)[["AB1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                             unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

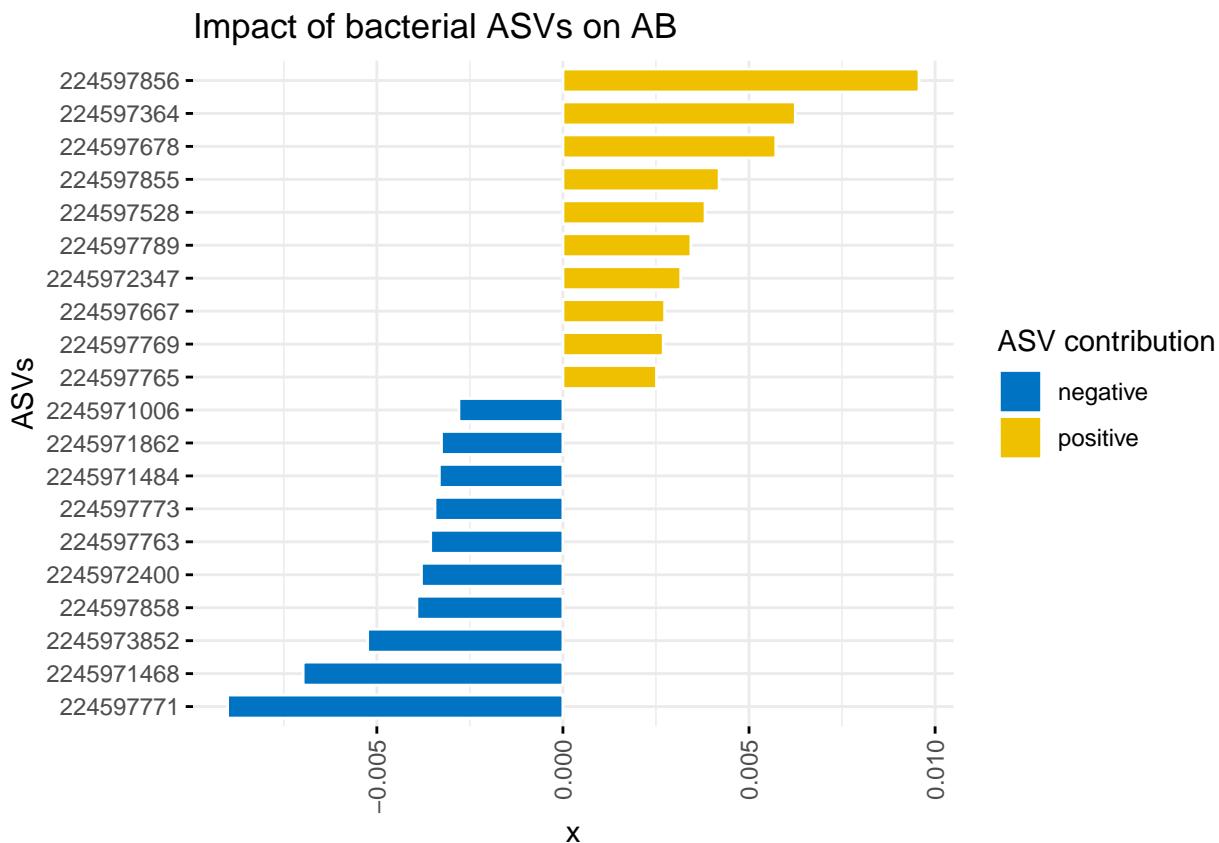
ggbarplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",          # jco journal color palett. see ?ggpar

```

```

sort.val = "asc",           # Sort the value in ascending order
sort.by.groups = FALSE,    # Don't sort inside each group
x.text.angle = 90,         # Rotate vertically x axis texts
xlab = "ASVs",
legend.title = "ASV contribution",
title = "Impact of bacterial ASVs on AB",
rotate = TRUE,
ggtheme = theme_minimal())

```



Stable

```

permanova_stable <- adonis(t(assay(tse, "relabundance")) ~ Stables, data = colData(tse), permutations = 1000)

coef <- coefficients(permanova_stable)[["Stables1"],]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                           unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

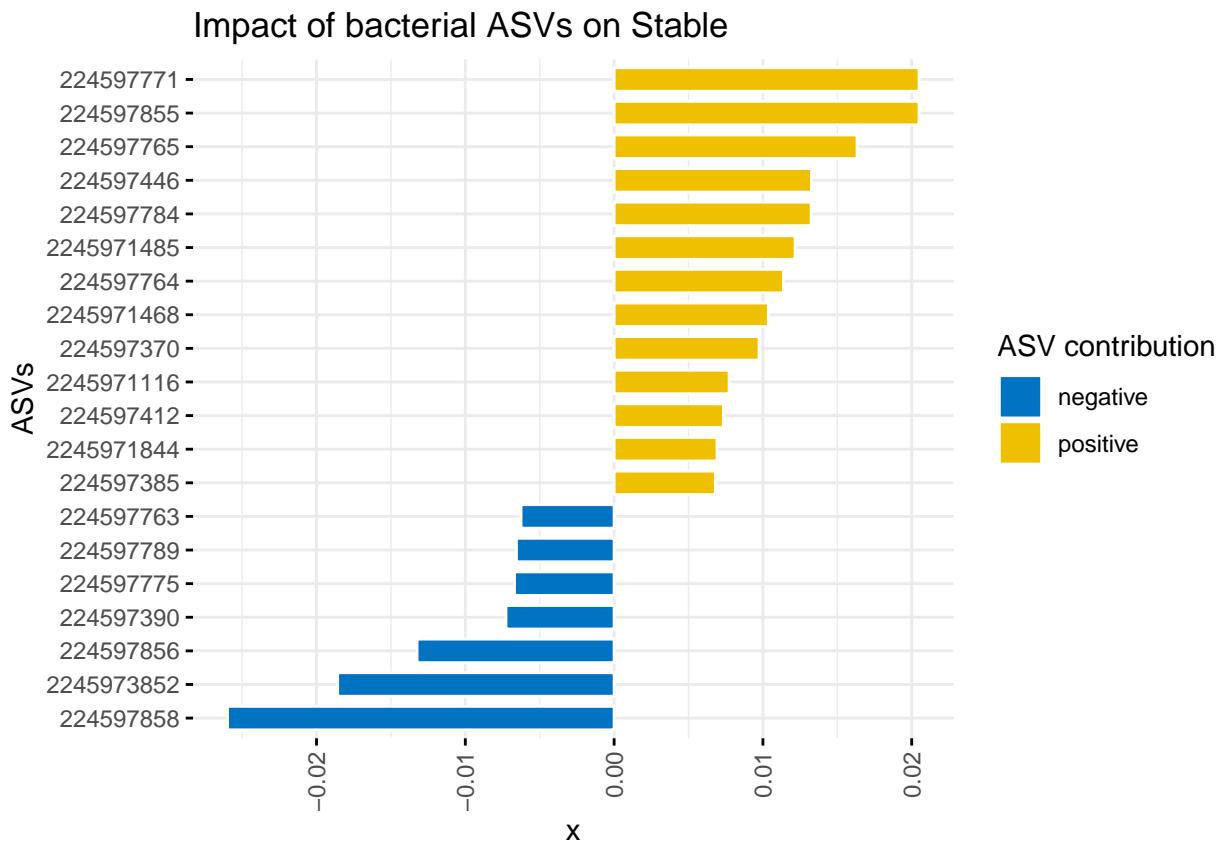
ggbarplot(df, x = "y", y = "x",
          fill = "contr",           # change fill color by mpg_level
          color = "white",          # Set bar border colors to white

```

```

palette = "jco",           # jco journal color palett. see ?ggpar
sort.val = "asc",          # Sort the value in ascending order
sort.by.groups = FALSE,    # Don't sort inside each group
x.text.angle = 90,         # Rotate vertically x axis texts
xlab = "ASVs",
legend.title = "ASV contribution",
title = "Impact of bacterial ASVs on Stable",
rotate = TRUE,
ggtheme = theme_minimal())

```



Farm

```

permanova_farm <- adonis(t(assay(tse, "relabundance")) ~ Farm2, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_farm)[["Farm21",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                   levels = c("negative", "positive"))

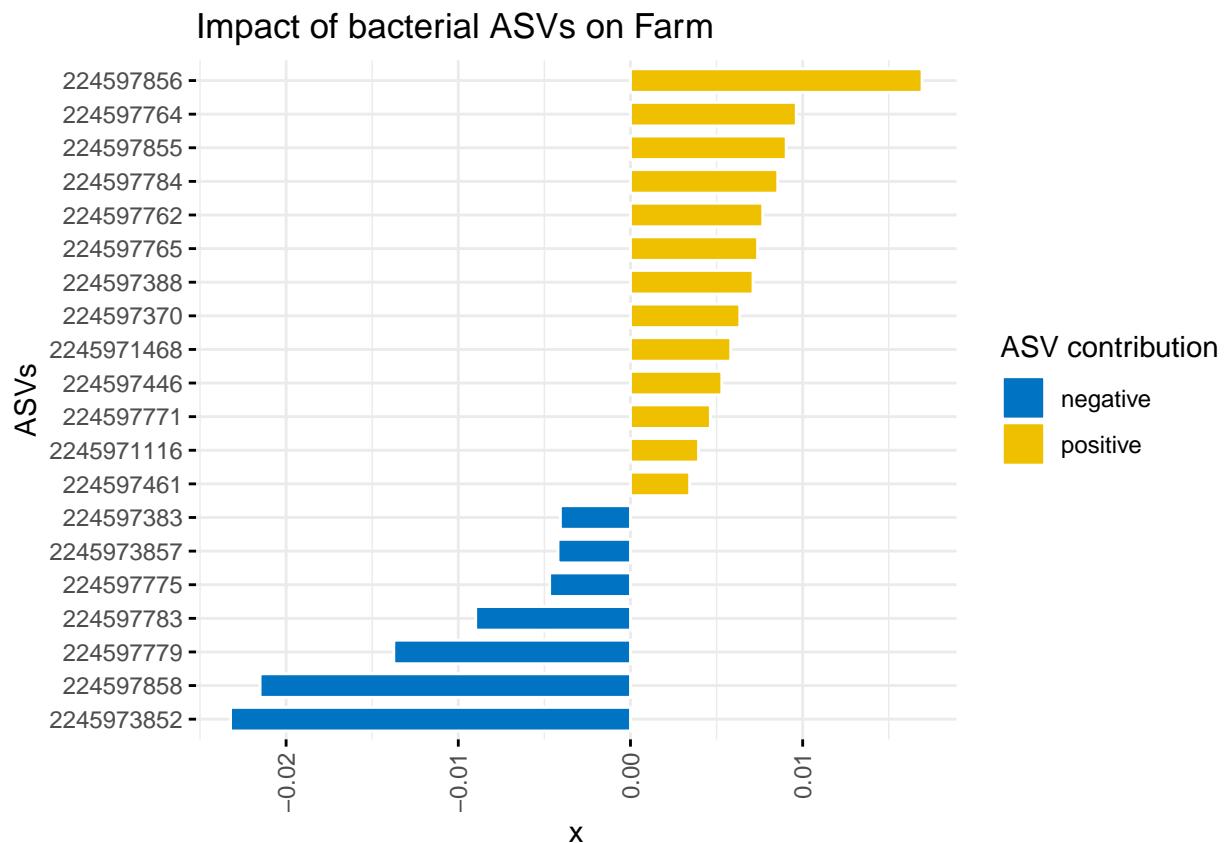
ggbarplot(df, x = "y", y = "x",
          fill = "contr",           # change fill color by mpg_level

```

```

color = "white",           # Set bar border colors to white
palette = "jco",          # jco journal color palett. see ?ggpar
sort.val = "asc",          # Sort the value in ascending order
sort.by.groups = FALSE,    # Don't sort inside each group
x.text.angle = 90,         # Rotate vertically x axis texts
xlab = "ASVs",
legend.title = "ASV contribution",
title = "Impact of bacterial ASVs on Farm",
rotate = TRUE,
ggtheme = theme_minimal())

```



Agent

```

permanova_agent <- adonis(t(assay(tse, "relabundance")) ~ Cox, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_agent)[["Cox1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

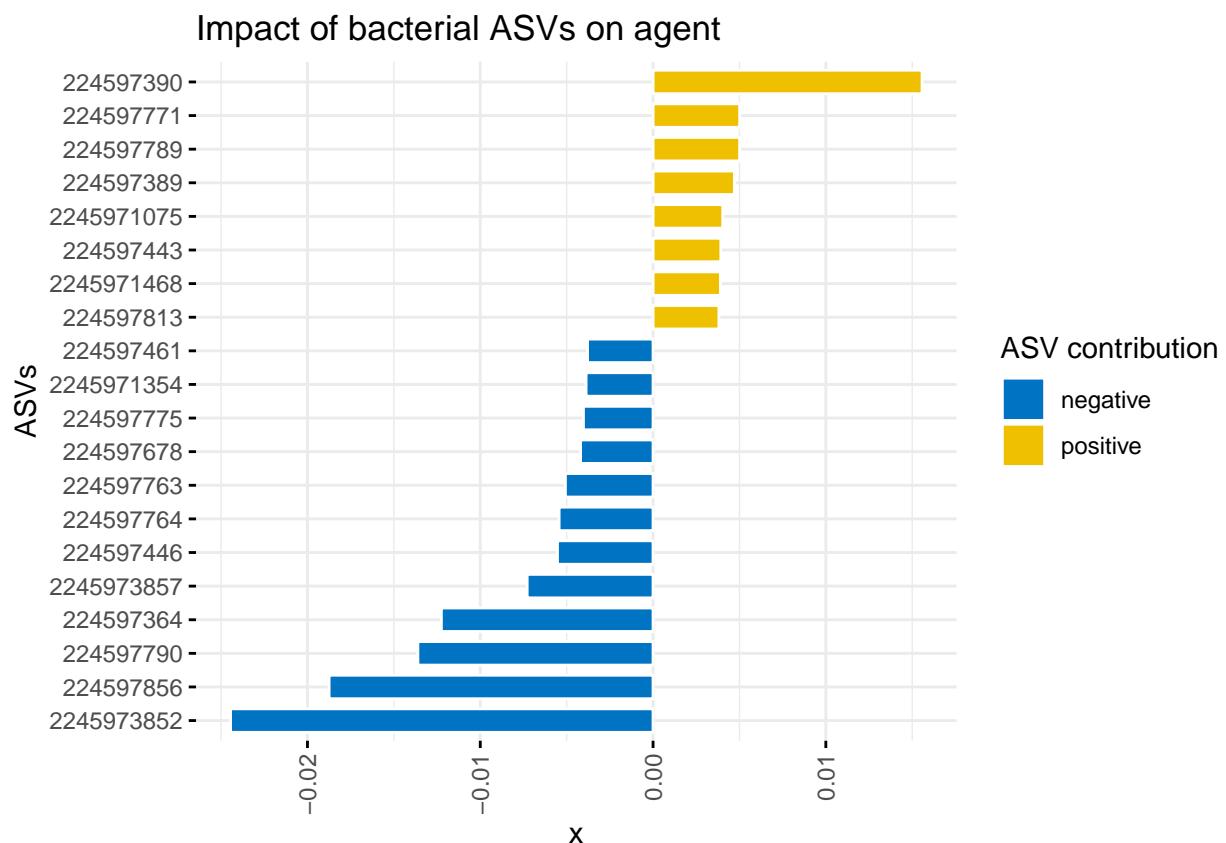
ggbarplot(df, x = "y", y = "x",

```

```

fill = "contr",           # change fill color by mpg_level
color = "white",          # Set bar border colors to white
palette = "jco",          # jco journal color palett. see ?ggpar
sort.val = "asc",          # Sort the value in ascending order
sort.by.groups = FALSE,    # Don't sort inside each group
x.text.angle = 90,         # Rotate vertically x axis texts
xlab = "ASVs",
legend.title = "ASV contribution",
title = "Impact of bacterial ASVs on agent",
rotate = TRUE,
ggtheme = theme_minimal())

```



Same plots but for genera - Age

```

permanova_age <- adonis(t(assay(tse_genus, "relabundance")) ~ Age, data = colData(tse_genus), permutations = 999)

coef <- coefficients(permanova_age)[["Age1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

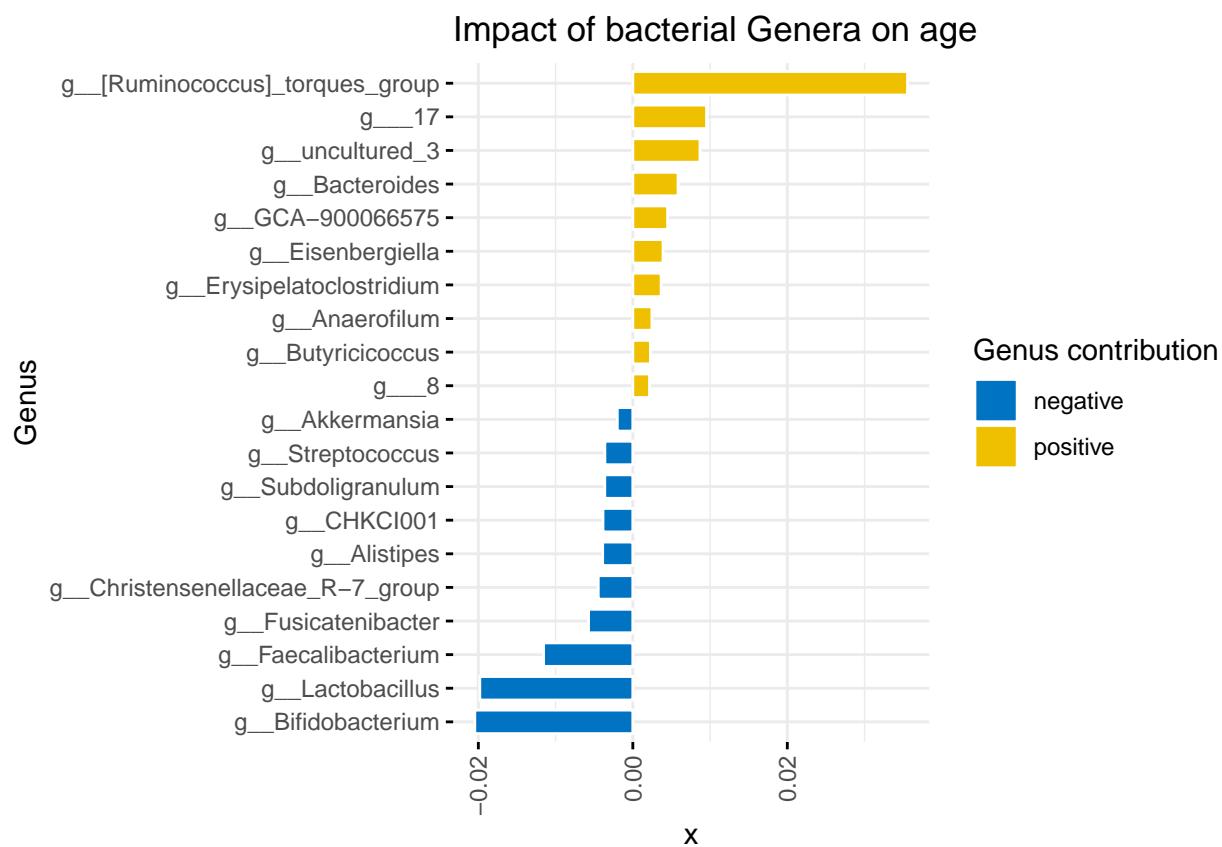
df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                   levels = c("negative", "positive"))

```

```

ggbarplot(df, x = "y", y = "x",
          fill = "contr",           # change fill color by mpg_level
          color = "white",          # Set bar border colors to white
          palette = "jco",          # jco journal color palett. see ?ggpar
          sort.val = "asc",          # Sort the value in ascending order
          sort.by.groups = FALSE,    # Don't sort inside each group
          x.text.angle = 90,         # Rotate vertically x axis texts
          xlab = "Genus",
          legend.title = "Genus contribution",
          title = "Impact of bacterial Genera on age",
          rotate = TRUE,
          ggtheme = theme_minimal())

```



Antibiotic treatment

```

permanova_AB <- adonis(t(assay(tse_genus, "relabundance")) ~ AB, data = colData(tse_genus), permutations = 999)

coef <- coefficients(permanova_AB)[["AB1", ]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

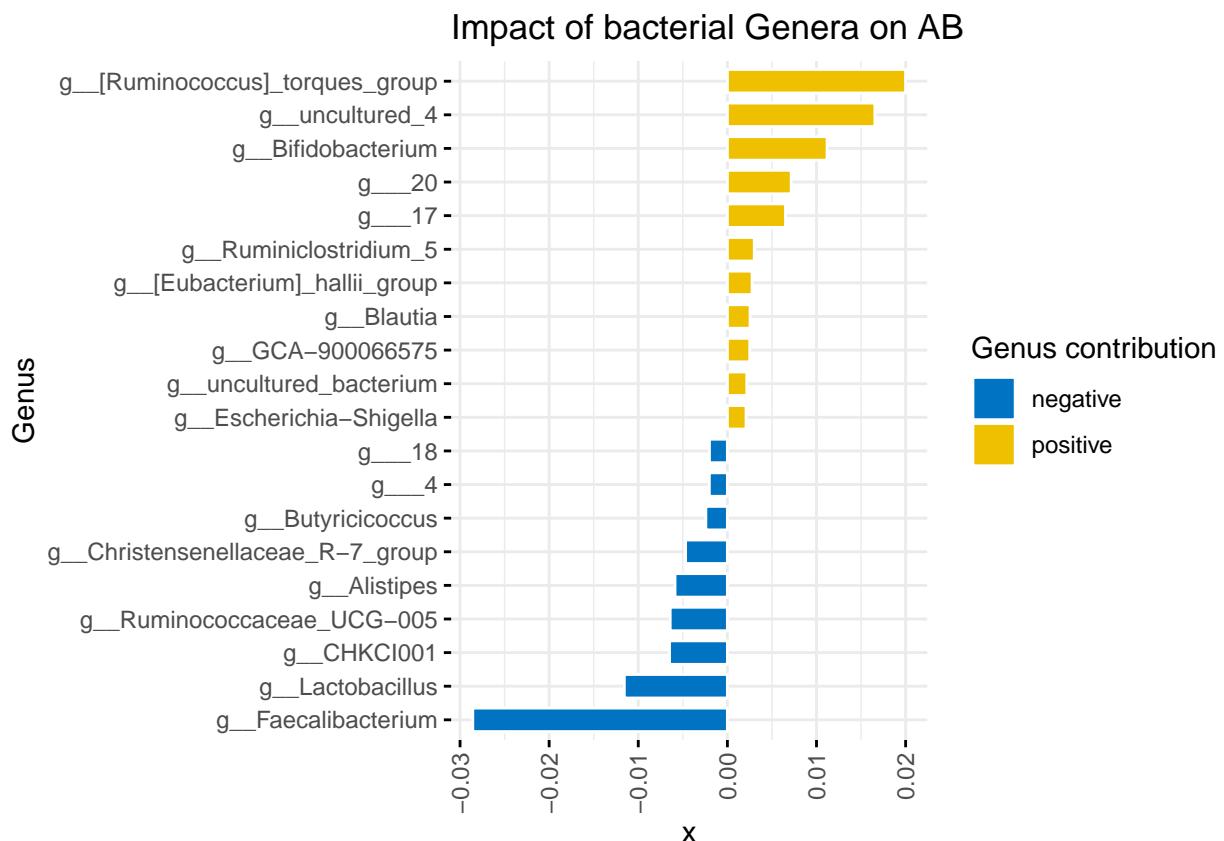
df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

```

```

ggbarrplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",         # jco journal color palett. see ?ggpar
           sort.val = "asc",         # Sort the value in ascending order
           sort.by.groups = FALSE,   # Don't sort inside each group
           x.text.angle = 90,        # Rotate vertically x axis texts
           xlab = "Genus",
           legend.title = "Genus contribution",
           title = "Impact of bacterial Genera on AB",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



Stable

```

permanova_stable <- adonis(t(assay(tse_genus, "relabundance")) ~ Stables, data = colData(tse_genus), per...
coef <- coefficients(permanova_stable)[["Stables1",],]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

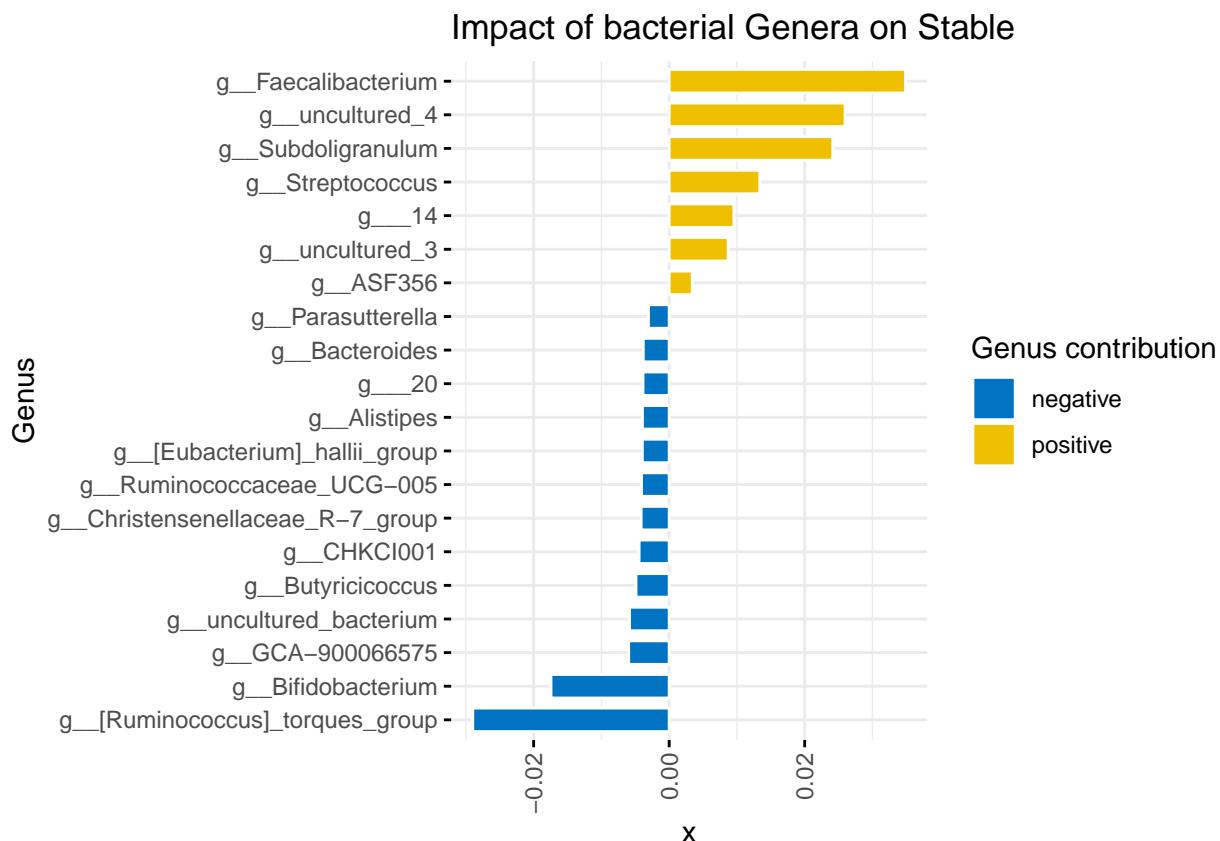
df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

```

```

ggbbarplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",         # jco journal color palett. see ?ggpar
           sort.val = "asc",         # Sort the value in ascending order
           sort.by.groups = FALSE,   # Don't sort inside each group
           x.text.angle = 90,        # Rotate vertically x axis texts
           xlab = "Genus",
           legend.title = "Genus contribution",
           title = "Impact of bacterial Genera on Stable",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



Farm

```

permanova_farm <- adonis(t(assay(tse_genus, "relabundance")) ~ Farm2, data = colData(tse_genus), permutations = 999)

coef <- coefficients(permanova_farm)[["Farm21",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

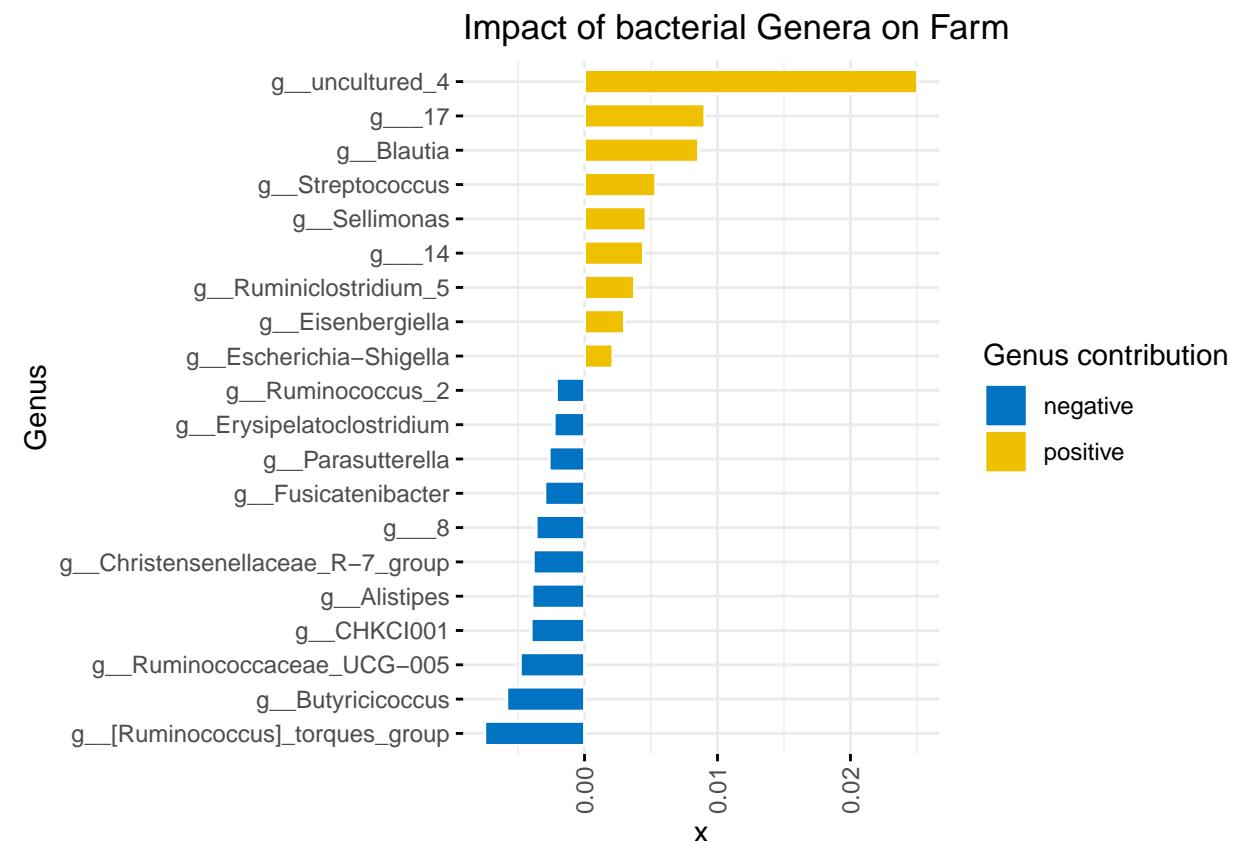
df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

```

```

ggbbarplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",         # jco journal color palett. see ?ggpar
           sort.val = "asc",        # Sort the value in ascending order
           sort.by.groups = FALSE,  # Don't sort inside each group
           x.text.angle = 90,        # Rotate vertically x axis texts
           xlab = "Genus",
           legend.title = "Genus contribution",
           title = "Impact of bacterial Genera on Farm",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



Agent

```

permanova_agent <- adonis(t(assay(tse_genus, "relabundance")) ~ Cox, data = colData(tse_genus), permutations = 999)

coef <- coefficients(permanova_agent)[["Cox1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

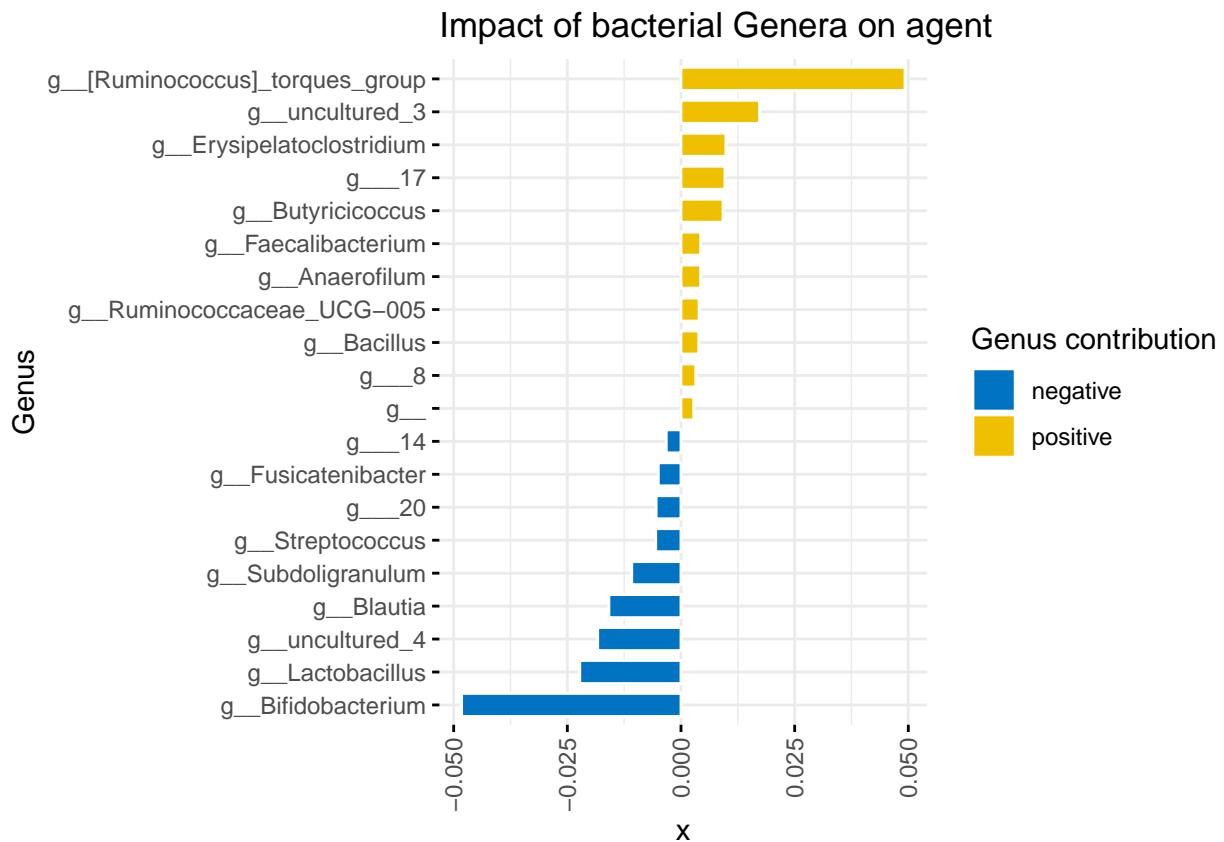
df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

```

```

ggbarrplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",         # jco journal color palett. see ?ggpar
           sort.val = "asc",         # Sort the value in ascending order
           sort.by.groups = FALSE,   # Don't sort inside each group
           x.text.angle = 90,        # Rotate vertically x axis texts
           xlab = "Genus",
           legend.title = "Genus contribution",
           title = "Impact of bacterial Genera on agent",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



checking homogeneity condition - bray-curtis ANOVAs are performed on betadispers of our rel abund data to test whether groups are more variable than others

```

# Bray
ps.rel = microbiome::transform(subset16S, "compositional")
meta = meta(ps.rel)
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Age))

```

```

## Analysis of Variance Table
##
## Response: Distances
##              Df  Sum Sq  Mean Sq F value    Pr(>F)

```

```

## Groups      1 0.04882 0.048824  12.036 0.0007295 ***
## Residuals 118 0.47868 0.004057
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), meta$AB))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      1 0.03423 0.034229  8.4594 0.00434 **
## Residuals 118 0.47746 0.004046
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Farm2))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      3 0.05386 0.0179546   3.488 0.01805 *
## Residuals 116 0.59712 0.0051476
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Stables))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      9 0.10294 0.0114374   2.2578 0.02333 *
## Residuals 110 0.55723 0.0050658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Cox))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      3 0.05437 0.0181226   3.5909 0.01585 *
## Residuals 116 0.58544 0.0050469
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Researcher))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      4 0.11864 0.0296606  6.4411 0.0001031 ***
## Residuals 115 0.52956 0.0046049
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$LitterType)) # homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      2 0.00394 0.0019691  0.3935 0.6756
## Residuals 117 0.58548 0.0050041
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Gender)) # homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      1 0.00107 0.0010715  0.2599 0.6111
## Residuals 118 0.48643 0.0041223
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$FlockSize))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      5 0.10963 0.0219269  4.6166 0.0007136 ***
## Residuals 114 0.54145 0.0047496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$AgeParentStock)) # homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      4 0.04865 0.0121630   2.213 0.07188 .
## Residuals 115 0.63206 0.0054962
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

anova(betadisper(vegdist(t(abundances(ps.rel))), meta$FeedProducent))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      2 0.06978 0.034891  7.7777 0.0006739 ***
## Residuals 117 0.52487 0.004486
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), meta$FeedType))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      1 0.04882 0.048824 12.036 0.0007295 ***
## Residuals 118 0.47868 0.004057
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Jaccard
anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Age))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      1 0.023087 0.0230874 12.572 0.0005626 ***
## Residuals 118 0.216706 0.0018365
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$AB))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      1 0.022032 0.0220319 12.296 0.000643 ***
## Residuals 118 0.211439 0.0017919
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Farm2))

## Analysis of Variance Table
##
## Response: Distances

```

```

##          Df   Sum Sq   Mean Sq F value    Pr(>F)
## Groups      3 0.027311 0.0091035  3.7252 0.01337 *
## Residuals 116 0.283478 0.0024438
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Stables)

## Analysis of Variance Table
##
## Response: Distances
##          Df   Sum Sq   Mean Sq F value    Pr(>F)
## Groups      9 0.046992 0.0052213  2.1071 0.03471 *
## Residuals 110 0.272580 0.0024780
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Researcher)

## Analysis of Variance Table
##
## Response: Distances
##          Df   Sum Sq   Mean Sq F value    Pr(>F)
## Groups      4 0.10745 0.0268626  12.64 1.474e-08 ***
## Residuals 115 0.24439 0.0021251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$LitterType) # homogeneous

## Analysis of Variance Table
##
## Response: Distances
##          Df   Sum Sq   Mean Sq F value    Pr(>F)
## Groups      2 0.000853 0.00042627  0.1859 0.8306
## Residuals 117 0.268245 0.00229269

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Gender) # homogeneous

## Analysis of Variance Table
##
## Response: Distances
##          Df   Sum Sq   Mean Sq F value    Pr(>F)
## Groups      1 0.000528 0.0005283  0.2968 0.5869
## Residuals 118 0.210023 0.0017799

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$FlockSize))

## Analysis of Variance Table
##

```

```

## Response: Distances
##          Df   Sum Sq Mean Sq F value    Pr(>F)
## Groups      5 0.067558 0.013512  6.0427 5.345e-05 ***
## Residuals 114 0.254905 0.002236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$AgeParentStock)) # homogeneou

## Analysis of Variance Table
##
## Response: Distances
##          Df   Sum Sq Mean Sq F value    Pr(>F)
## Groups      4 0.022427 0.0056068  2.1203 0.08273 .
## Residuals 115 0.304107 0.0026444
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$FeedProducent))

## Analysis of Variance Table
##
## Response: Distances
##          Df   Sum Sq Mean Sq F value    Pr(>F)
## Groups      2 0.037045 0.0185225  8.975 0.0002365 ***
## Residuals 117 0.241463 0.0020638
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$FeedType))

## Analysis of Variance Table
##
## Response: Distances
##          Df   Sum Sq Mean Sq F value    Pr(>F)
## Groups      1 0.023087 0.0230874 12.572 0.0005626 ***
## Residuals 118 0.216706 0.0018365
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# group variances are not homogenous in most cases, so there are differences in variances between groups

# Tukey tests can be performed to see if and which groups differ in relation to variance

TukeyHSD(betadisper(vegdist(t(abundances(ps.rel))), meta$Farm2))

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
```

```

## $group
##          diff      lwr      upr     p adj
## Farm2-Farm1 -0.01283746 -0.05959222 0.033917288 0.8906792
## Farm3-Farm1 -0.06128653 -0.11527427 -0.007298795 0.0193219
## Farm4-Farm1 -0.02958711 -0.08357485 0.024400627 0.4842422
## Farm3-Farm2 -0.04844907 -0.09520382 -0.001694317 0.0391860
## Farm4-Farm2 -0.01674965 -0.06350440 0.030005106 0.7867504
## Farm4-Farm3  0.03169942 -0.02228831 0.085687160 0.4227048

# different way of calculating homogeneity, permutation tests, null = no difference in dispersion between
permute(vegdist(t(abundances(ps.rel))), meta$Age), pairwise = TRUE)

## 
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##          Df   Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups      1 0.04882 0.048824 12.036    999  0.001 ***
## Residuals 118 0.47868 0.004057
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##          14     35
## 14        0.001
## 35  0.00072948

permute(betadisper(unifrac.dist, metadf$Age), pairwise = TRUE) # looks like unifrac distances are homogeneous

## 
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##          Df   Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups      1 0.002189 0.0021888 0.8904    999  0.358
## Residuals 118 0.290075 0.0024583
## 
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##          14     35
## 14        0.36
## 35  0.3473

permute(betadisper(unifrac.dist, metadf$AB), pairwise = TRUE) # not for AB though

## 
## Permutation test for homogeneity of multivariate dispersions

```

```

## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##          Df   Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups      1 0.016272 0.0162721 7.5077    999  0.003 **
## Residuals 118 0.255751 0.0021674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##       no     yes
## no      0.003
## yes  0.0070982

permutest(betadisper(bray.dist, metadf$Age), pairwise = TRUE) # there are differences in P value with o

##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##          Df   Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups      1 0.04882 0.048824 12.036    999  0.002 **
## Residuals 118 0.47868 0.004057
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##       14     35
## 14      0.002
## 35  0.00072948

```

SIMPER - to save time, analyses are not rerun for Rmarkdown, but earlier results are loaded we'll use MT as abbreviation for metataxonomics instead of 16s since R does not like its objects starting with numbers

```

source("../Results/Scripts/Steinberger_scripts/simper_pretty.r")
source("../Results/Scripts/Steinberger_scripts/R_krusk.r")

#Age

#simper.pretty(otu_table(subset16S), metrics = sample_data(subset16S), interesting = c("Age"), perc_cut = 0.05)

#MT_age = data.frame(read.csv("MT_age_clean_simper.csv"))

#kruskal.pretty(otu_table(subset16S), metrics = sample_data(subset16S), csv = MT_age, interesting = c('Age'))

KW_MT_age = data.frame(read.csv("MT_Age_krusk_simper.csv"))
KW_MT_age = KW_MT_age[KW_MT_age$fdr_krusk_p.val < 0.05,] # filter out non-significant results, based on p-values
KW_MT_age = KW_MT_age[with(KW_MT_age, order(SIMPER, decreasing = TRUE)),]

```

```

KW_MT_age$OTU = as.factor(KW_MT_age$OTU)

KW_MT_age %>% mutate_if(is.numeric, format.pval, 2) %>% dplyr::select("SIMPER", "OTU", "fdr_krusk_p.val"
  rowwise() %>% mutate(Combined = paste("ASV =", OTU, ", SIMPER =", SIMPER, ", p-value =", fdr_krusk_p.
  dplyr::select(Combined)

## # A tibble: 3 x 1
## # Rowwise:
##   Combined
##   <chr>
## 1 ASV = 224597762 , SIMPER = 0.035 , p-value = 0.00068
## 2 ASV = 224597390 , SIMPER = 0.028 , p-value = 0.00906
## 3 ASV = 224597789 , SIMPER = 0.018 , p-value = 1.8e-07

#AB
#simper.pretty(otu_table(subset16S), metrics = sample_data(subset16S), interesting = c("AB"), perc_cuto

#MT_AB = data.frame(read.csv("MT_AB_clean_simper.csv"))

#kruskal.pretty(otu_table(subset16S), metrics = sample_data(subset16S), csv = MT_AB, interesting = c('A

KW_MT_AB = data.frame(read.csv("MT_AB_krusk_simper.csv"))
KW_MT_AB = KW_MT_AB[KW_MT_AB$fdr_krusk_p.val < 0.05,] # filter out non-significant results, based on fd
KW_MT_AB = KW_MT_AB[with(KW_MT_AB, order(SIMPER, decreasing = TRUE)),]
KW_MT_AB$OTU = as.factor(KW_MT_AB$OTU)

KW_MT_AB %>% mutate_if(is.numeric, format.pval, 2) %>% dplyr::select("SIMPER", "OTU", "fdr_krusk_p.val")
  rowwise() %>% mutate(Combined = paste("ASV =", OTU, ", SIMPER =", SIMPER, ", p-value =", fdr_krusk_p.
  dplyr::select(Combined)

## # A tibble: 5 x 1
## # Rowwise:
##   Combined
##   <chr>
## 1 ASV = 224597771 , SIMPER = 0.027 , p-value = 0.0103
## 2 ASV = 2245971468 , SIMPER = 0.015 , p-value = 2.8e-05
## 3 ASV = 224597789 , SIMPER = 0.015 , p-value = 0.0066
## 4 ASV = 224597856 , SIMPER = 0.013 , p-value = 0.0027
## 5 ASV = 224597364 , SIMPER = 0.010 , p-value = 0.0026

#Farms - too many comparisons so maybe too extensive for report

#simper.pretty(otu_table(subset16S), metrics = sample_data(subset16S), interesting = c("Farm2"), perc_c

#MT_Farm = data.frame(read.csv("MT_Farm_clean_simper.csv"))

#kruskal.pretty(otu_table(subset16S), metrics = sample_data(subset16S), csv = MT_Farm, interesting = c()

KW_MT_Farm = data.frame(read.csv("MT_Farm_krusk_simper.csv"))
KW_MT_Farm = KW_MT_Farm[KW_MT_Farm$fdr_krusk_p.val < 0.05,] # filter out non-significant results, based
KW_MT_Farm = KW_MT_Farm[with(KW_MT_Farm, order(SIMPER, decreasing = TRUE)),]
KW_MT_Farm$OTU = as.factor(KW_MT_Farm$OTU)

```

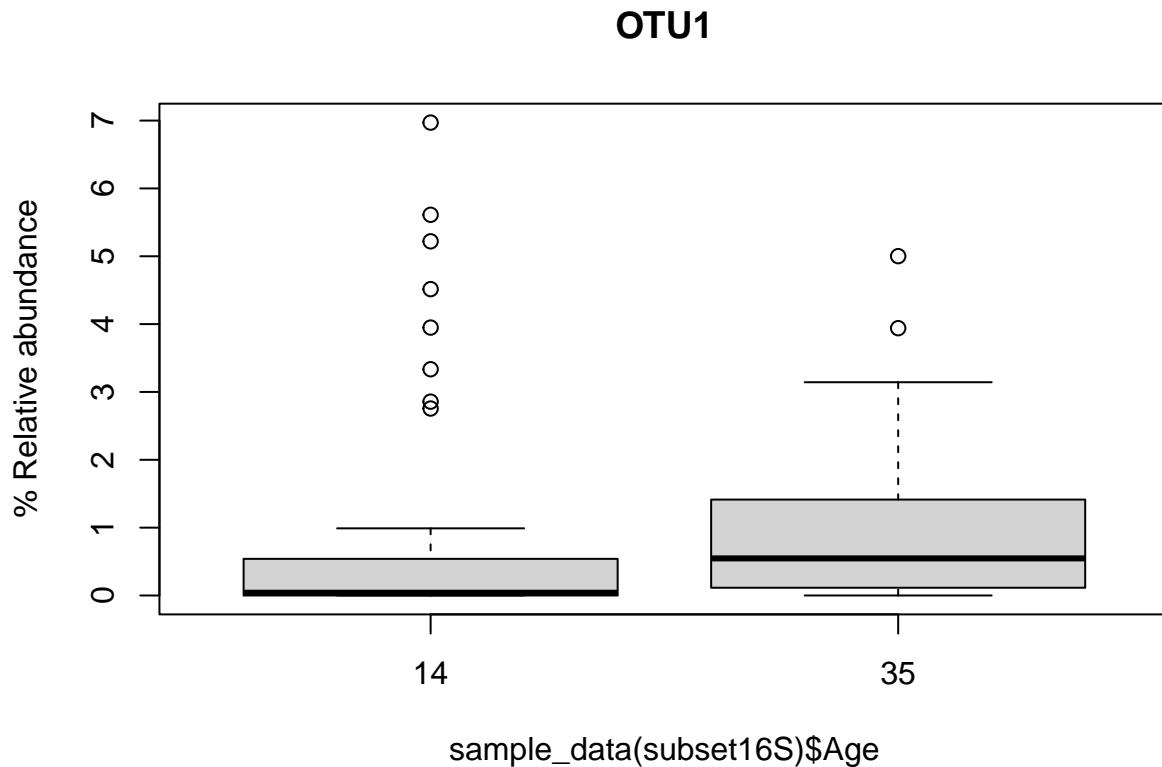
```

KW_MT_Farm %>% mutate_if(is.numeric, format.pval, 2) %>% dplyr::select("Comparison", "SIMPER", "OTU", "ASV")
  rowwise() %>% mutate(Combined = paste(Comparison, "ASV =", OTU, ", SIMPER =", SIMPER, ", p-value =", p_value))
  dplyr::select(Combined)

## # A tibble: 28 x 1
## # Rowwise:
##   Combined
##   <chr>
## 1 Farm1_Farm4 ASV = 2245973852 , SIMPER = 0.067 , p-value = 2.0e-06
## 2 Farm2_Farm4 ASV = 2245973852 , SIMPER = 0.066 , p-value = 1.9e-10
## 3 Farm3_Farm4 ASV = 2245973852 , SIMPER = 0.065 , p-value = 2.0e-06
## 4 Farm1_Farm4 ASV = 224597858 , SIMPER = 0.034 , p-value = 0.00081
## 5 Farm2_Farm3 ASV = 224597779 , SIMPER = 0.033 , p-value = 0.02551
## 6 Farm1_Farm3 ASV = 224597779 , SIMPER = 0.032 , p-value = 0.00518
## 7 Farm2_Farm3 ASV = 224597763 , SIMPER = 0.031 , p-value = 0.00038
## 8 Farm2_Farm3 ASV = 224597771 , SIMPER = 0.029 , p-value = 0.00073
## 9 Farm3_Farm4 ASV = 224597771 , SIMPER = 0.025 , p-value = 0.01604
## 10 Farm1_Farm4 ASV = 224597783 , SIMPER = 0.023 , p-value = 0.01605
## # i 18 more rows

# plots to look at specific ASVs (age)
abund = otu_table(subset16S)/rowSums(otu_table(subset16S))*100
boxplot(unlist(data.frame(abund["224597762"])) ~ sample_data(subset16S)$Age, ylab="% Relative abundance")

```



```

# specific test
kruskal.test(unlist(data.frame(otu_table(subset16S)[ "224597762" ]), use.names = FALSE) ~ sample_data(subset16S))

## 
## Kruskal-Wallis rank sum test
##
## data: unlist(data.frame(otu_table(subset16S)[ "224597762" ]), use.names = FALSE) by sample_data(subset16S)
## Kruskal-Wallis chi-squared = 15.191, df = 1, p-value = 9.716e-05

```

Clustering

```

# Trying out different distances, aggregation methods and indices for finding optimal number of clusters

tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)

tse <- transformCounts(tse, method = "relabundance")

assay <- t(assay(tse, "relabundance"))

diss_jaccard <- vegdist(assay, method = "jaccard")

# different aggregation methods and indices will grant different amount of clusters
NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "mcclain")$Best.nc # two clusters

## 
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be in binary form

## Number_clusters      Value_Index
##          2.0000        0.0313

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "frey")$Best.nc # two clusters

## 
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be in binary form

## Number_clusters      Value_Index
##          2.0000        2.1845

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "cindex")$Best.nc # 15 clusters

## 
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be in binary form

## Number_clusters      Value_Index
##          15.0000       0.638

```

```

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "silhouette")$Best.nc # two clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.000          0.086

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "dunn")$Best.nc # four clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 4.000          0.763

NbClust(diss = diss_jaccard, distance = NULL, method = "ward.D2", index = "silhouette")$Best.nc # 15 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                15.0000         0.0453

NbClust(diss = diss_jaccard, distance = NULL, method = "ward.D", index = "silhouette")$Best.nc # 11 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                11.0000         0.0448

NbClust(diss = diss_jaccard, distance = NULL, method = "single", index = "silhouette")$Best.nc # 15 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                15.0000         0.0827

NbClust(diss = diss_jaccard, distance = NULL, method = "average", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.000          0.086

```

```

NbClust(diss = diss_jaccard, distance = NULL, method = "mcquitty", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##              2.000          0.086

NbClust(diss = diss_jaccard, distance = NULL, method = "median", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##              2.0000         0.1194

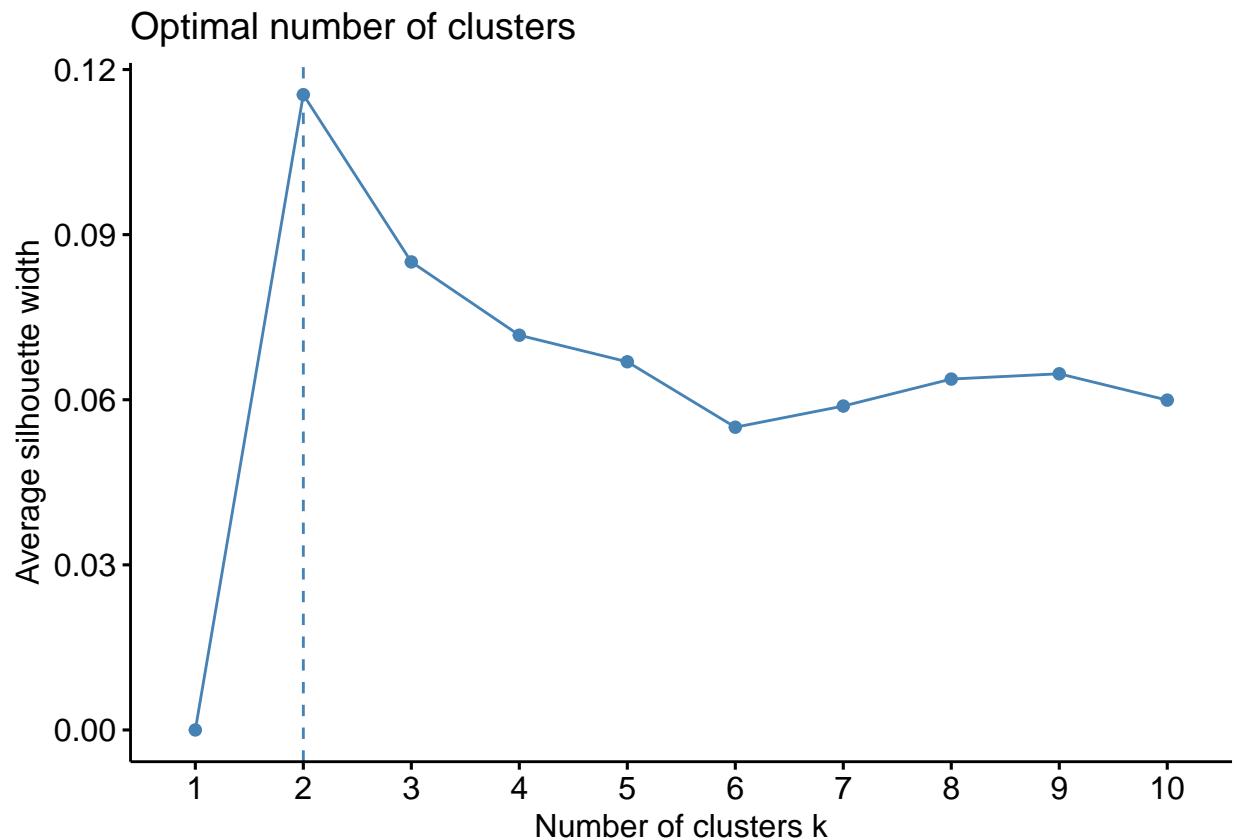
NbClust(diss = diss_jaccard, distance = NULL, method = "centroid", index = "silhouette")$Best.nc # 15 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

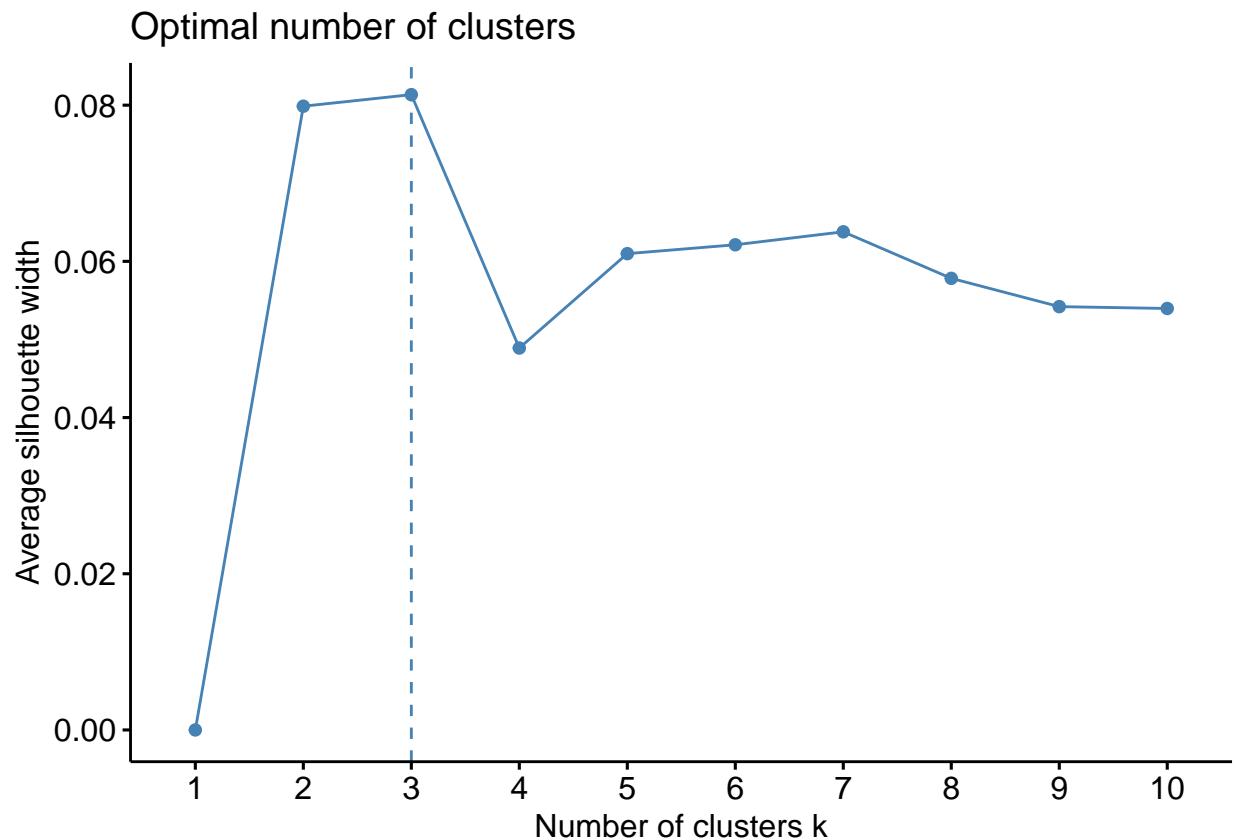
## Number_clusters      Value_Index
##             15.0000        0.1223

# silhouette (ASW), different clustering methods
diss_jaccard <- as.matrix(diss_jaccard)
fviz_nbclust(diss_jaccard, kmeans, method = "silhouette") # 2 seems optimal for k-means

```

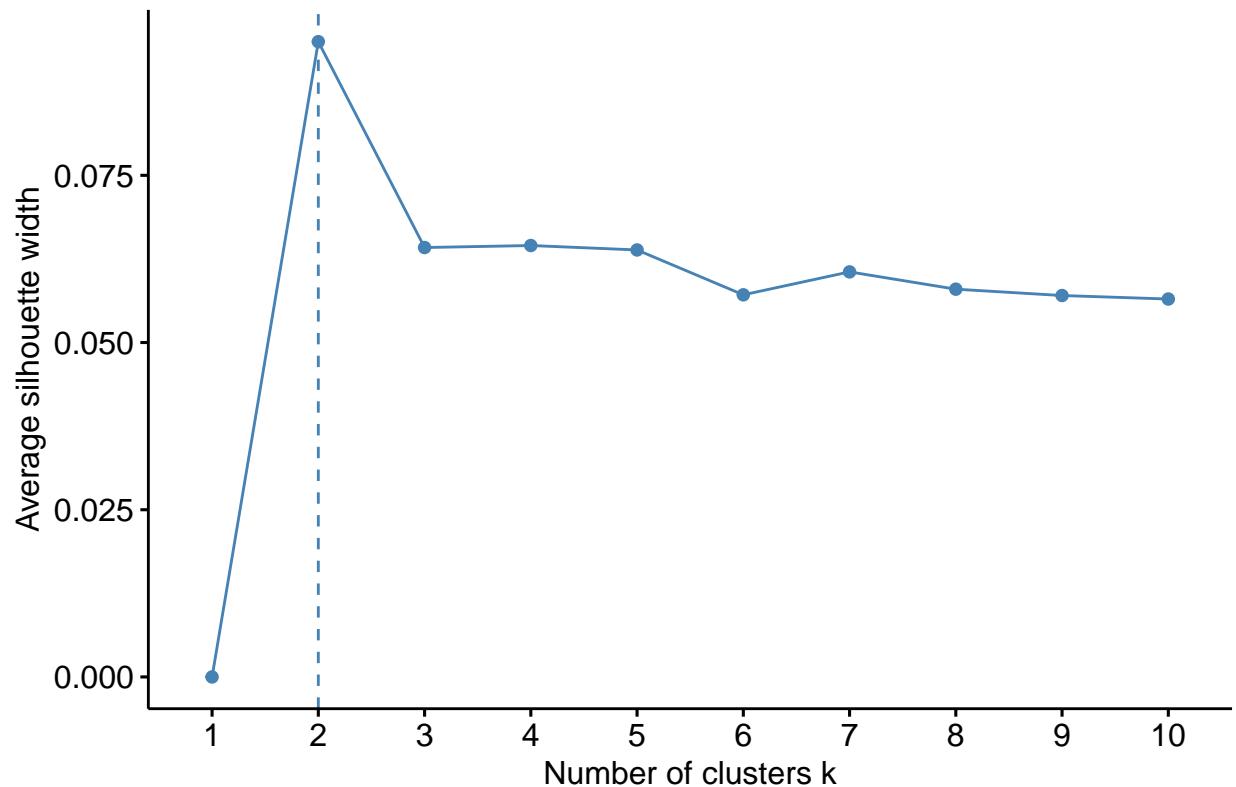


```
fviz_nbclust(diss_jaccard, cluster::pam, method = "silhouette") # 3 seems optimal for PAM
```

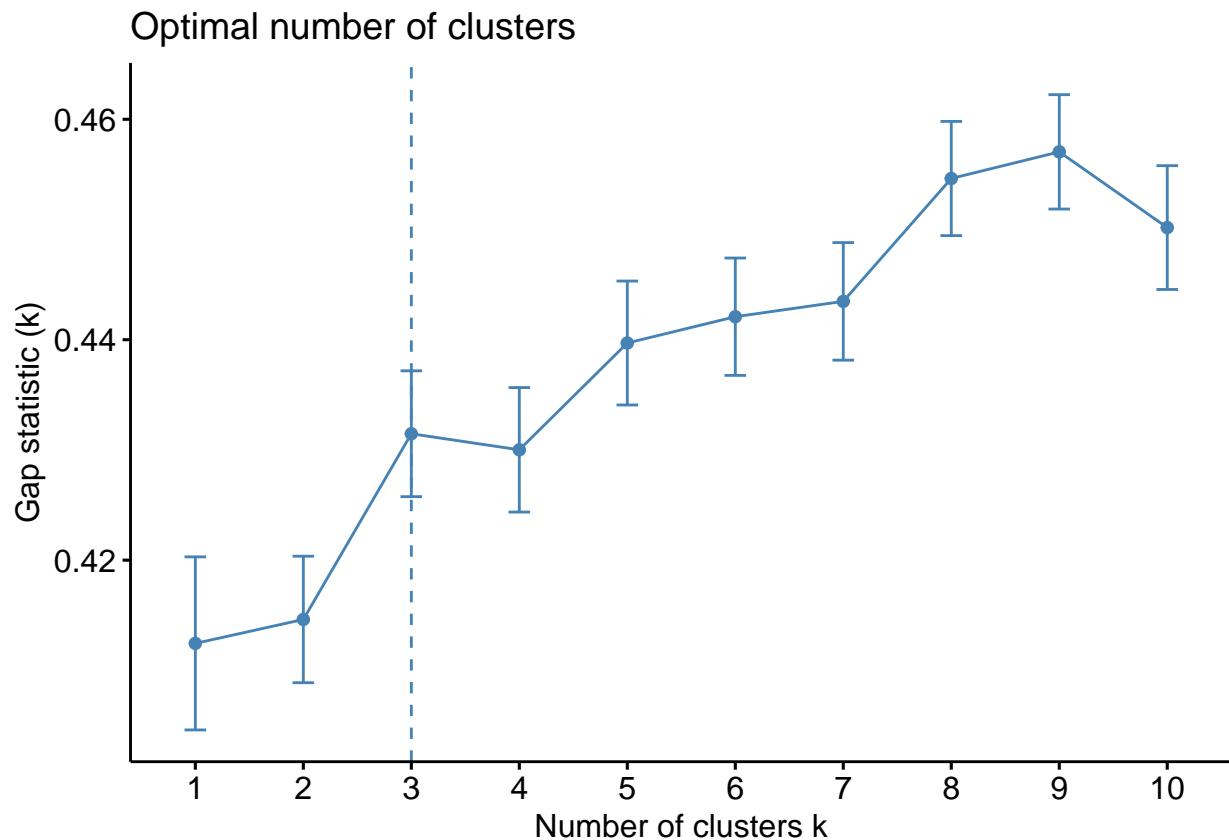


```
fviz_nbclust(diss_jaccard, hc, method = "silhouette") # 2 seems optimal for hc
```

Optimal number of clusters

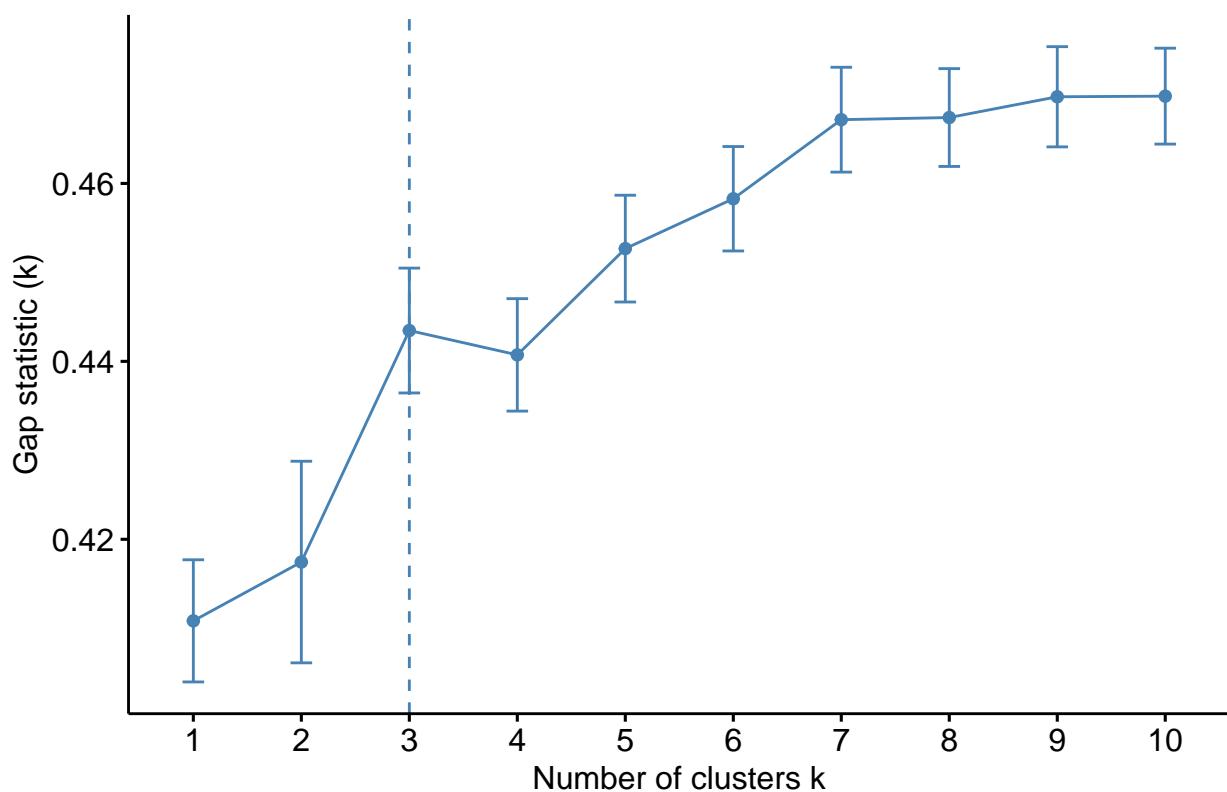


```
fviz_nbclust(diss_jaccard, kmeans, method = "gap_stat") # 3 seems optimal for k-means gap stat
```

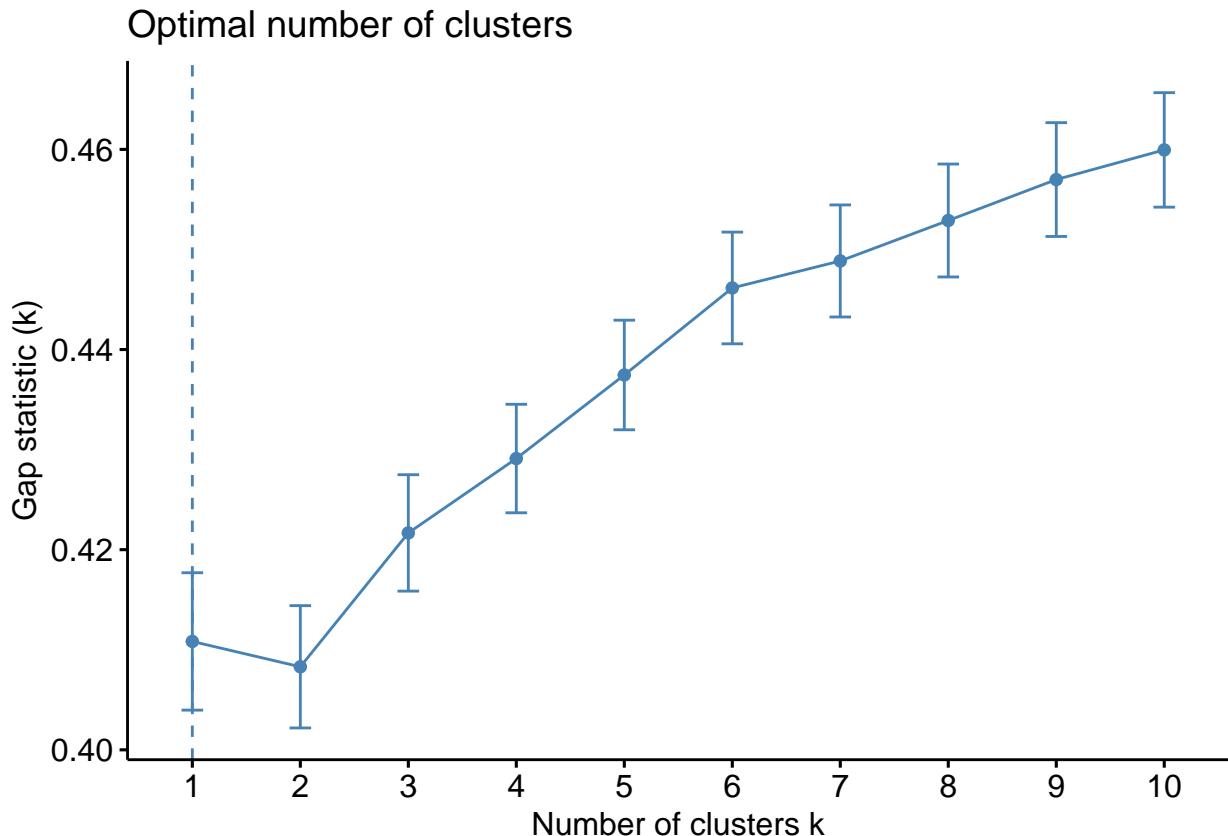


```
fviz_nbclust(diss_jaccard, cluster::pam, method = "gap_stat") # 3 seems optimal for PAM gap stat
```

Optimal number of clusters



```
fviz_nbclust(diss_jaccard, hc, method = "gap_stat") # 1 seems optimal for hc cut gap stat
```



```
# now, let's repeat this for BC

diss_bray <- vegdist(assay, method = "bray")

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "mcclain")$Best.nc # two clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##              2.0000        0.0293

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "frey")$Best.nc # two clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##              2.0000        2.4419

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "cindex")$Best.nc # 5 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m
```

```

## Number_clusters      Value_Index
##                 5.0000      0.5838

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "silhouette")$Best.nc # two clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.1461

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "dunn")$Best.nc # four clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 4.0000      0.6277

NbClust(diss = diss_bray, distance = NULL, method = "ward.D2", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.0955

NbClust(diss = diss_bray, distance = NULL, method = "ward.D", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.0955

NbClust(diss = diss_bray, distance = NULL, method = "single", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.1158

NbClust(diss = diss_bray, distance = NULL, method = "average", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.1461

```

```

NbClust(diss = diss_bray, distance = NULL, method = "mcquitty", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##              2.0000      0.1461

NbClust(diss = diss_bray, distance = NULL, method = "median", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##              2.0000      0.1158

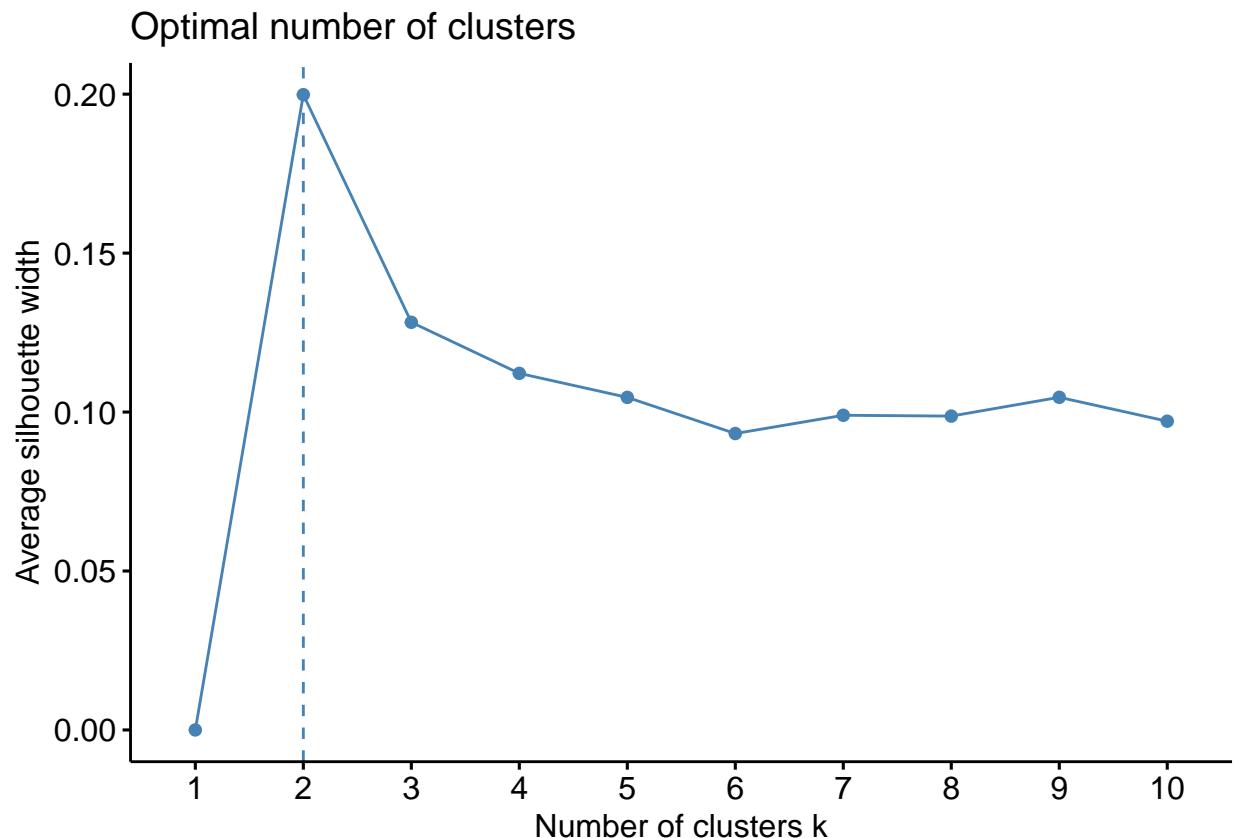
NbClust(diss = diss_bray, distance = NULL, method = "centroid", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##              2.0000      0.1972

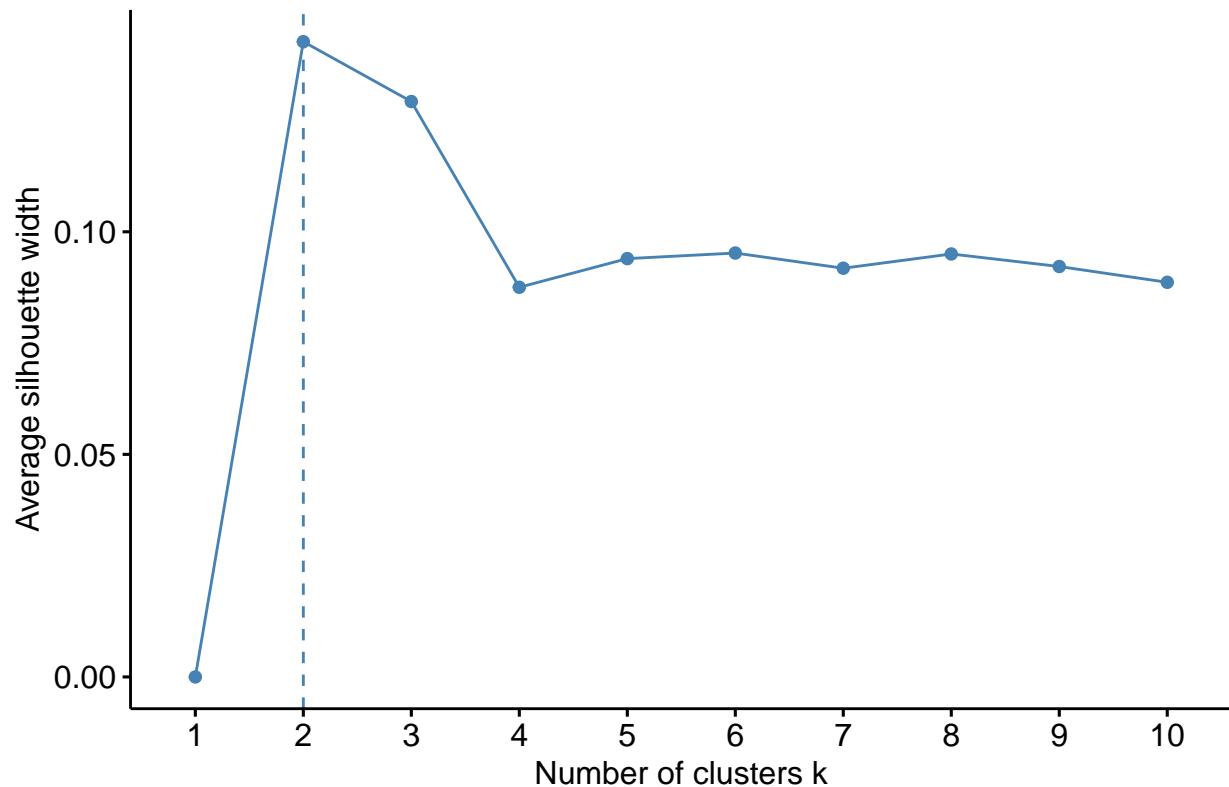
# silhouette (ASW)
diss_bray <- as.matrix(diss_bray)
fviz_nbclust(diss_bray, kmeans, method = "silhouette") # 2 seems optimal

```

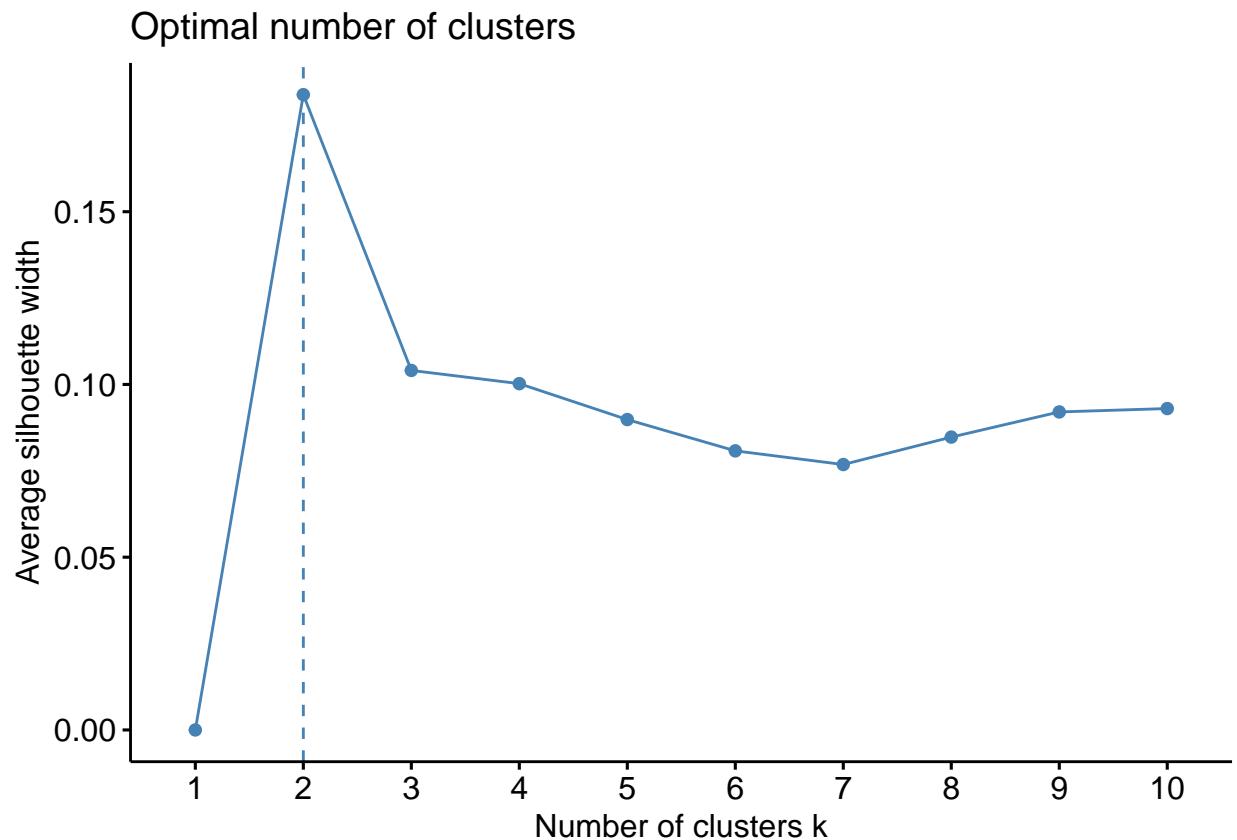


```
fviz_nbclust(diss_bray, cluster::pam, method = "silhouette") # 2 seems optimal for PAM
```

Optimal number of clusters

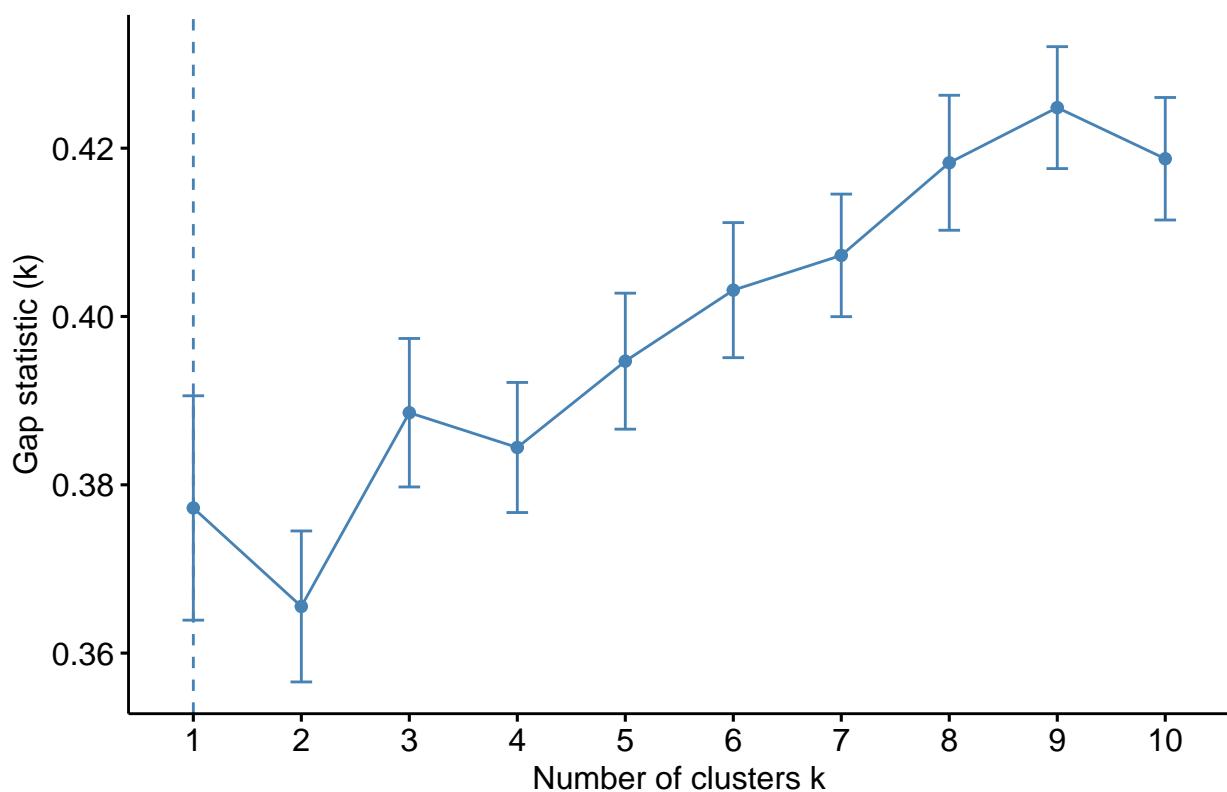


```
fviz_nbclust(diss_bray, hcut, method = "silhouette") # 2 seems optimal for hcut
```

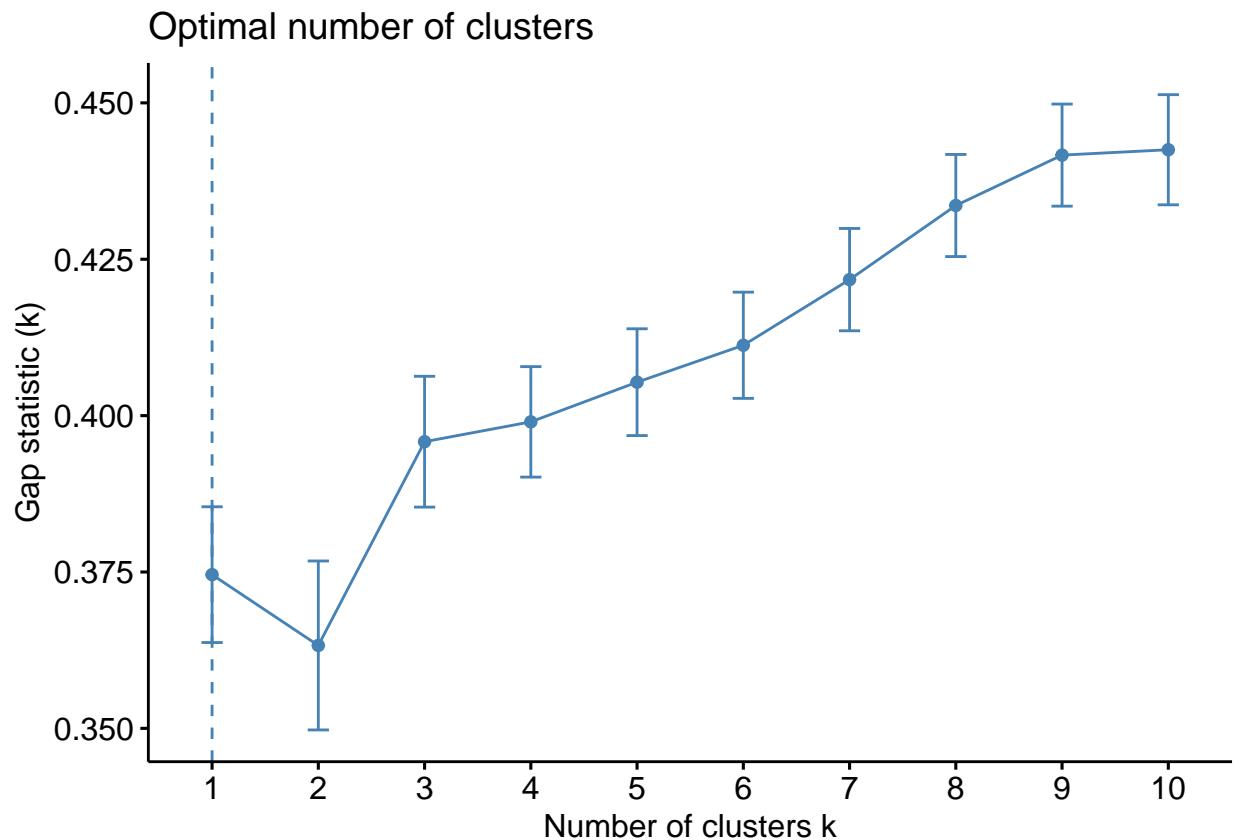


```
fviz_nbclust(diss_bray, kmeans, method = "gap_stat") # 1 seems optimal for k-means gap stat
```

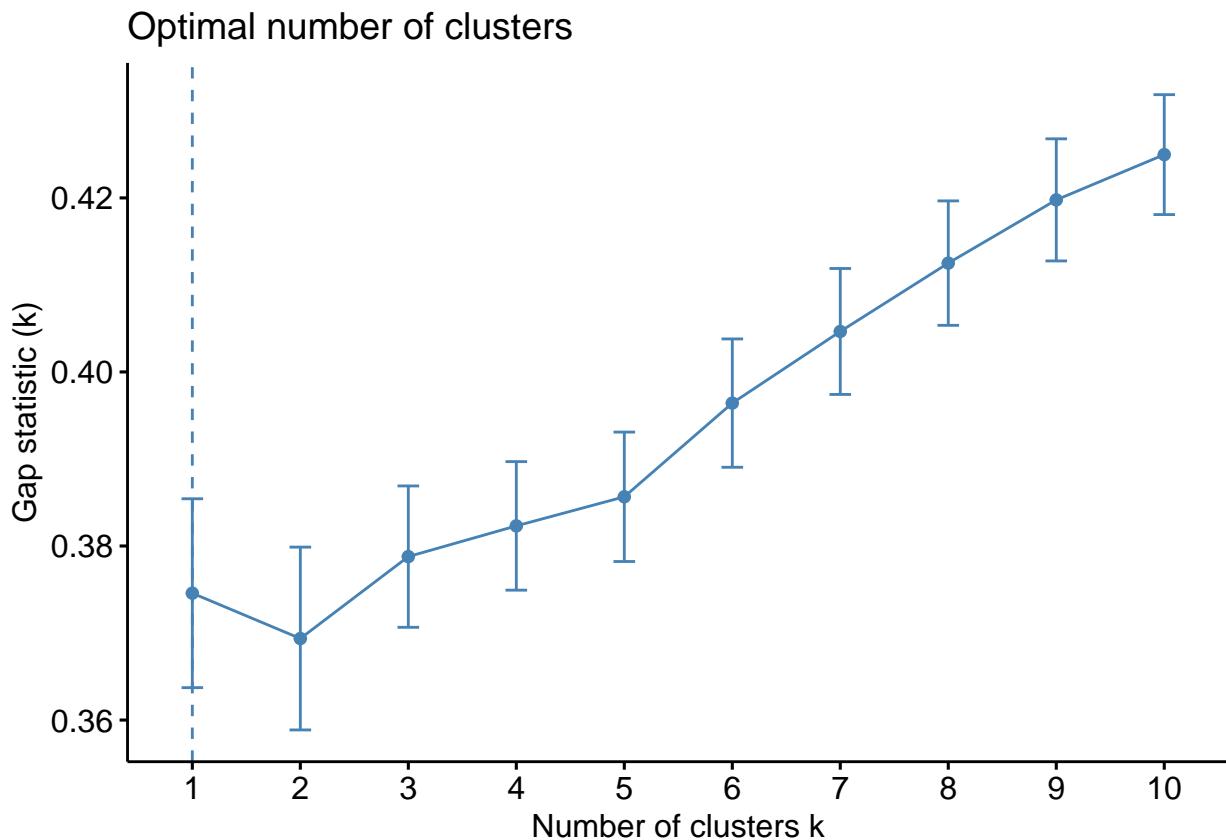
Optimal number of clusters



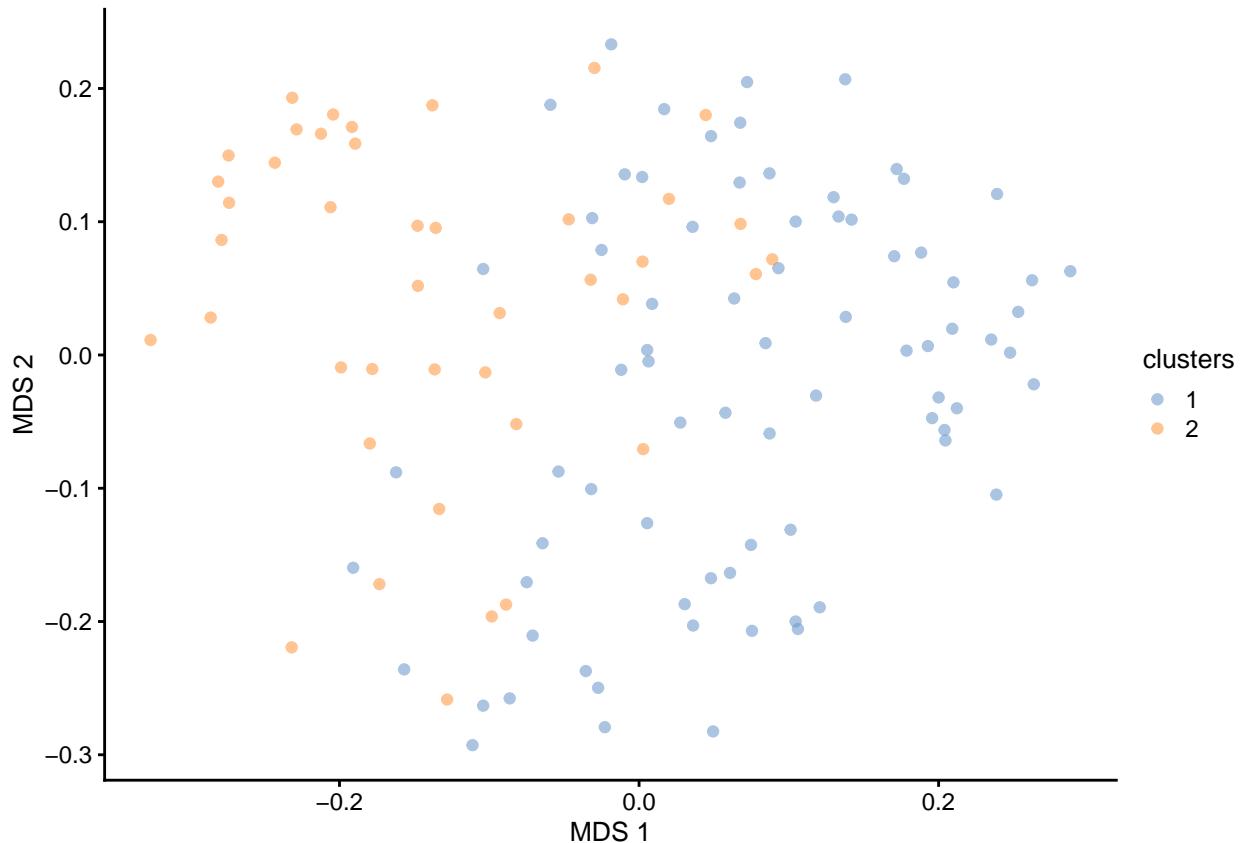
```
fviz_nbclust(diss_bray, cluster::pam, method = "gap_stat") # 1 seems optimal for PAM gap stat
```



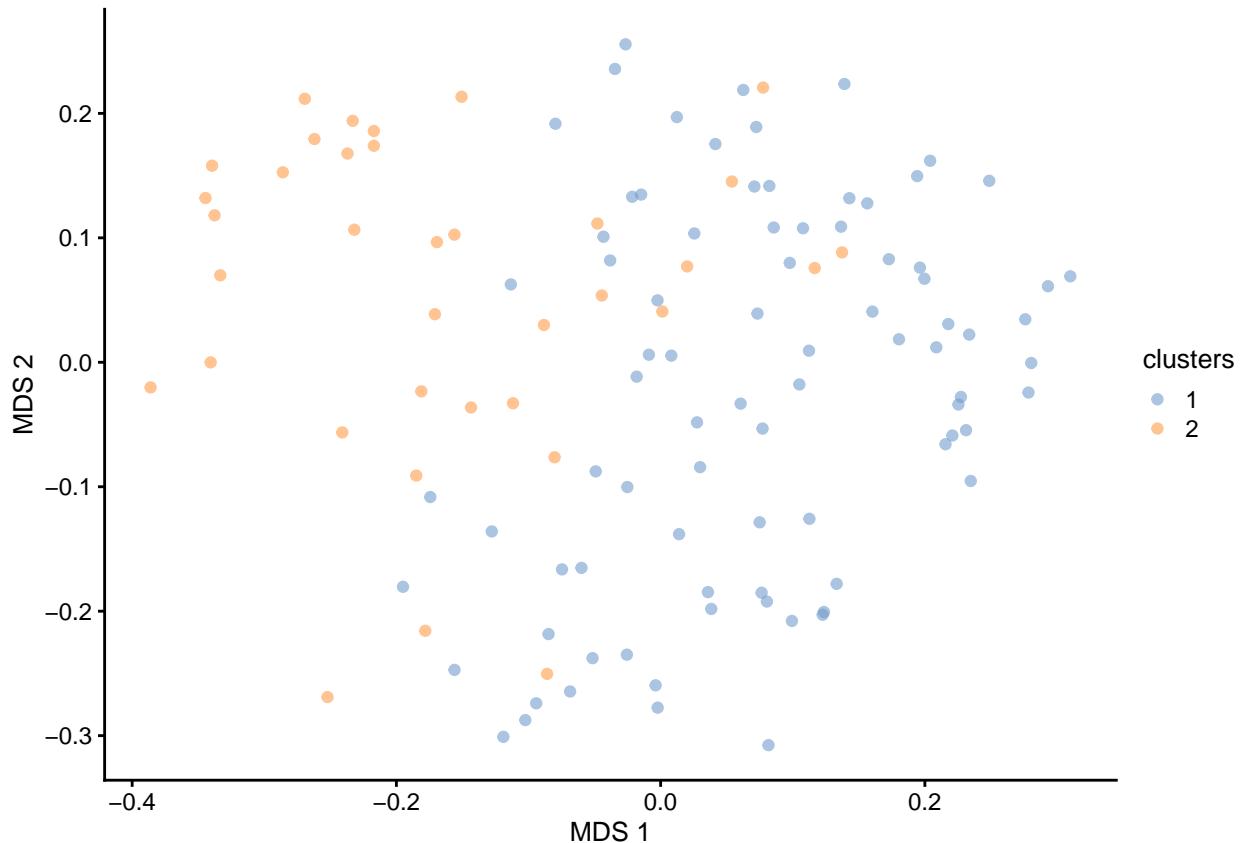
```
fviz_nbclust(diss_bray, hcut, method = "gap_stat") # 1 seems optimal for hcut gap stat
```



```
# k-means jaccard clusters
set.seed(1337)
km <- kmeans(diss_jaccard, 2, nstart = 25)
colData(tse)$clusters <- as.factor(km$cluster)
tse <- runMDS(tse, assay.type = "relabundance", FUN = vegan::vegdist, method = "jaccard")
plotReducedDim(tse, "MDS", colour_by = "clusters")
```



```
# k-means bray clusters MDS
set.seed(1337)
km <- kmeans(diss_bray, 2, nstart = 25)
colData(tse)$clusters <- as.factor(km$cluster)
tse <- runMDS(tse, assay.type = "relabundance", FUN = vegan::vegdist, method = "bray")
plotReducedDim(tse, "MDS", colour_by = "clusters")
```



```
# DMM (Laplace approximation) - ASV level
tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)
tse_dmn <- mia::runDMN(tse, name = "DMN", k = 1:7) # calculate most likely number of clusters from 1 to 7
tse_dmn

## class: TreeSummarizedExperiment
## dim: 1536 120
## metadata(1): DMN
## assays(1): counts
## rownames(1536): 2245974182 2245974034 ... 2245971310 224597986
## rowData names(6): Domain Phylum ... Family Genus
## colnames(120): 2_23 2_24 ... 14_35 14_36
## colData names(27): LibraryNumber Sample_Unique ... Metagenomics Stables
## reducedDimNames(0):
## mainExpName: NULL
## altExpNames(0):
## rowLinks: a LinkDataFrame (1536 rows)
## rowTree: 1 phylo tree(s) (1536 leaves)
## collLinks: NULL
## colTree: NULL

getDMN(tse_dmn)

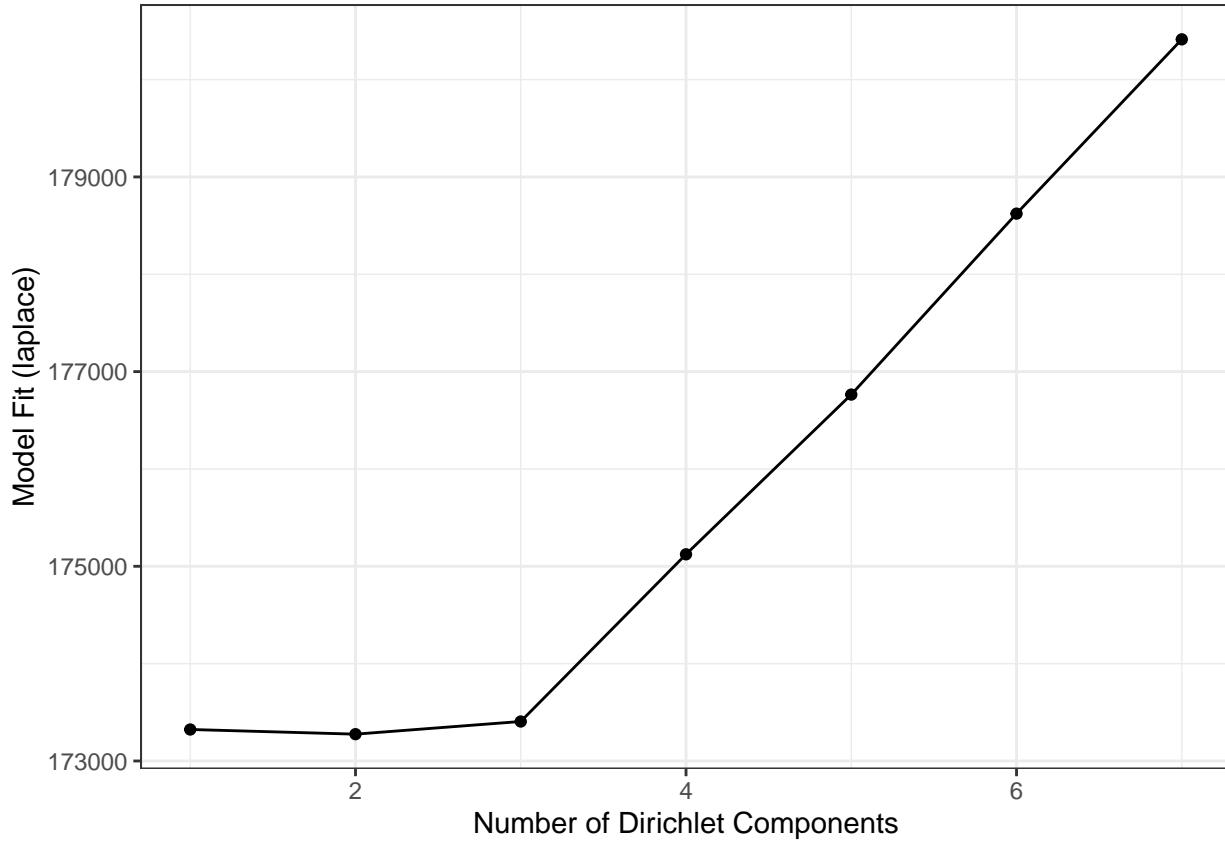
## [[1]]
## class: DMN
```

```

## k: 1
## samples x taxa: 120 x 1536
## Laplace: 173323.4 BIC: 177378.4 AIC: 175237.6
##
## [[2]]
## class: DMN
## k: 2
## samples x taxa: 120 x 1536
## Laplace: 173275.5 BIC: 183190 AIC: 178907.1
##
## [[3]]
## class: DMN
## k: 3
## samples x taxa: 120 x 1536
## Laplace: 173405.6 BIC: 189443.5 AIC: 183018.3
##
## [[4]]
## class: DMN
## k: 4
## samples x taxa: 120 x 1536
## Laplace: 175123.3 BIC: 197172 AIC: 188604.7
##
## [[5]]
## class: DMN
## k: 5
## samples x taxa: 120 x 1536
## Laplace: 176764.2 BIC: 204755.2 AIC: 194045.7
##
## [[6]]
## class: DMN
## k: 6
## samples x taxa: 120 x 1536
## Laplace: 178623.1 BIC: 212663.4 AIC: 199811.7
##
## [[7]]
## class: DMN
## k: 7
## samples x taxa: 120 x 1536
## Laplace: 180413.4 BIC: 220036.1 AIC: 205042.2

miaViz::plotDMNFit(tse_dmn, type = "laplace")

```



```

getBestDMNFit(tse_dmn, type = "laplace") # 2 again

## class: DMN
## k: 2
## samples x taxa: 120 x 1536
## Laplace: 173275.5 BIC: 183190 AIC: 178907.1

# genus level

tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)
tse_genus <- agglomerateByRank(tse, rank = "Genus", agglomerateTree = TRUE)
tse_dmn <- mia::runDMN(tse_genus, name = "DMN", k = 1:7) # calculate most likely number of clusters from tse_dmn

## class: TreeSummarizedExperiment
## dim: 119 120
## metadata(2): agglomerated_by_rank DMN
## assays(1): counts
## rownames(119): Genus:g__7_2 Genus:g__5_1 ... Genus:g__Flavonifractor
##   Genus:g__Intestinimonas
## rowData names(6): Domain Phylum ... Family Genus
## colnames(120): 2_23 2_24 ... 14_35 14_36
## colData names(27): LibraryNumber Sample_Unique ... Metagenomics Stables
## reducedDimNames(0):
## mainExpName: NULL

```

```

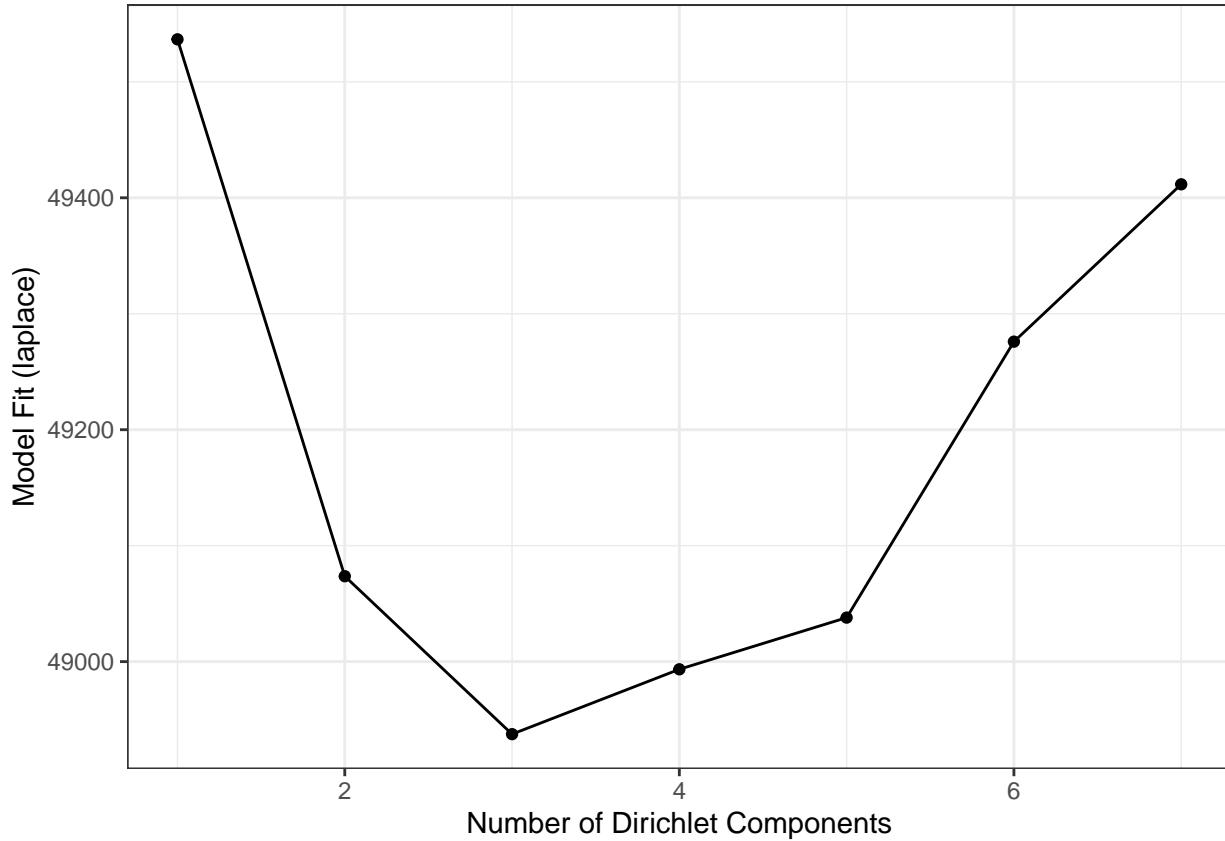
## altExpNames(0):
## rowLinks: a LinkDataFrame (119 rows)
## rowTree: 1 phylo tree(s) (119 leaves)
## collLinks: NULL
## colTree: NULL

getDMN(tse_dmn)

## [[1]]
## class: DMN
## k: 1
## samples x taxa: 120 x 119
## Laplace: 49536.64 BIC: 49751.56 AIC: 49585.7
##
## [[2]]
## class: DMN
## k: 2
## samples x taxa: 120 x 119
## Laplace: 49073.65 BIC: 49634.91 AIC: 49301.81
##
## [[3]]
## class: DMN
## k: 3
## samples x taxa: 120 x 119
## Laplace: 48937.44 BIC: 49800.88 AIC: 49300.53
##
## [[4]]
## class: DMN
## k: 4
## samples x taxa: 120 x 119
## Laplace: 48993.38 BIC: 50167.48 AIC: 49499.88
##
## [[5]]
## class: DMN
## k: 5
## samples x taxa: 120 x 119
## Laplace: 49037.94 BIC: 50571.21 AIC: 49736.36
##
## [[6]]
## class: DMN
## k: 6
## samples x taxa: 120 x 119
## Laplace: 49275.92 BIC: 51124.69 AIC: 50122.58
##
## [[7]]
## class: DMN
## k: 7
## samples x taxa: 120 x 119
## Laplace: 49411.63 BIC: 51635.61 AIC: 50466.26

miaViz::plotDMNFit(tse_dmn, type = "laplace")

```



```
getBestDMNFit(tse_dmn, type = "laplace") # Gives 3! as best fit for genus level data
```

```
## class: DMN
## k: 3
## samples x taxa: 120 x 119
## Laplace: 48937.44 BIC: 49800.88 AIC: 49300.53
```

phylum level

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)
tse_phylum <- agglomerateByRank(tse, rank = "Phylum", agglomerateTree = TRUE)
tse_dmn <- mia::runDMN(tse_phylum, name = "DMN", k = 1:7) # calculate most likely number of clusters fr
tse_dmn
```

```
## class: TreeSummarizedExperiment
## dim: 7 120
## metadata(2): agglomerated_by_rank DMN
## assays(1): counts
## rownames(7): Phylum:p__Tenericutes Phylum:p__Firmicutes ...
##   Phylum:p__Verrucomicrobia Phylum:p__Cyanobacteria
## rowData names(6): Domain Phylum ... Family Genus
## colnames(120): 2_23 2_24 ... 14_35 14_36
## colData names(27): LibraryNumber Sample_Uncode ... Metagenomics Stables
## reducedDimNames(0):
## mainExpName: NULL
```

```

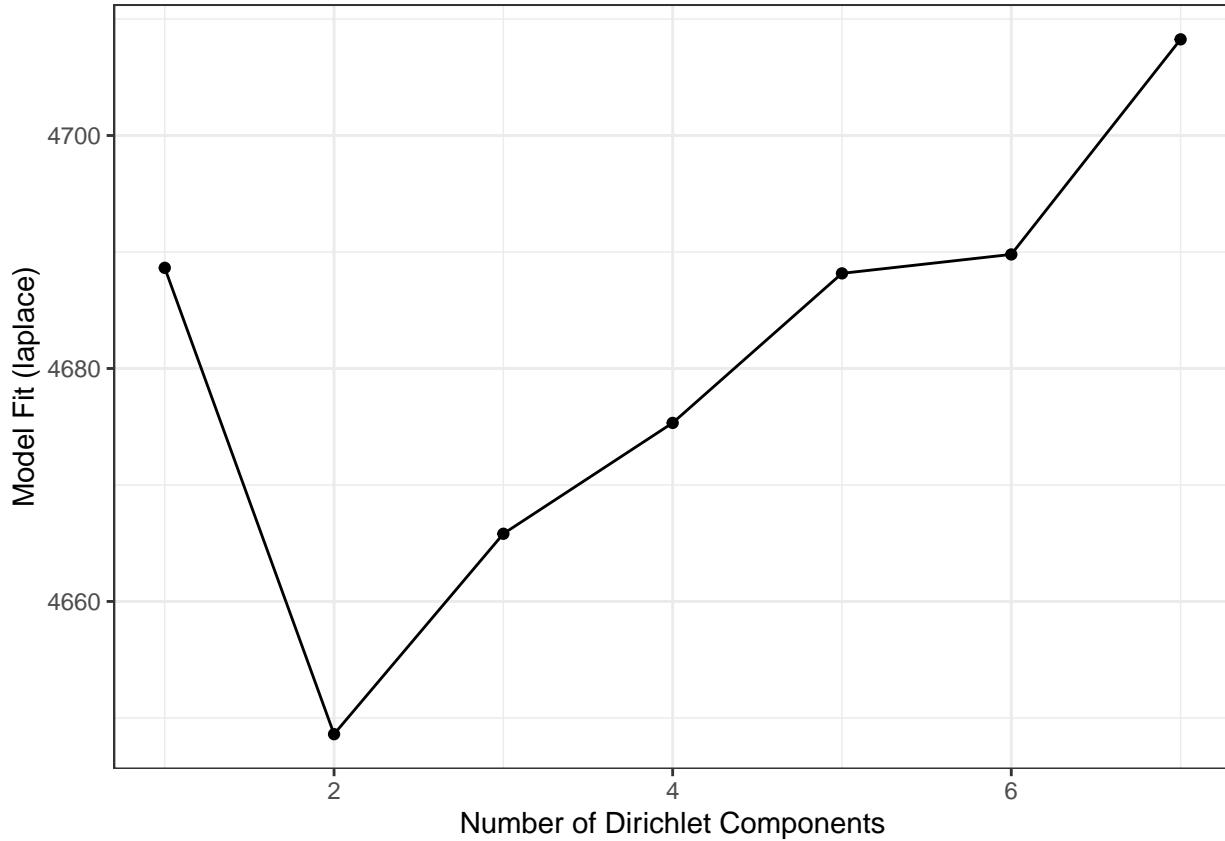
## altExpNames(0):
## rowLinks: a LinkDataFrame (7 rows)
## rowTree: 1 phylo tree(s) (7 leaves)
## collLinks: NULL
## colTree: NULL

getDMN(tse_dmn)

## [[1]]
## class: DMN
## k: 1
## samples x taxa: 120 x 7
## Laplace: 4688.641 BIC: 4697.146 AIC: 4687.39
##
## [[2]]
## class: DMN
## k: 2
## samples x taxa: 120 x 7
## Laplace: 4648.607 BIC: 4667.756 AIC: 4646.85
##
## [[3]]
## class: DMN
## k: 3
## samples x taxa: 120 x 7
## Laplace: 4665.811 BIC: 4689.917 AIC: 4657.861
##
## [[4]]
## class: DMN
## k: 4
## samples x taxa: 120 x 7
## Laplace: 4675.328 BIC: 4705.498 AIC: 4662.292
##
## [[5]]
## class: DMN
## k: 5
## samples x taxa: 120 x 7
## Laplace: 4688.166 BIC: 4721.189 AIC: 4666.833
##
## [[6]]
## class: DMN
## k: 6
## samples x taxa: 120 x 7
## Laplace: 4689.791 BIC: 4746.113 AIC: 4680.607
##
## [[7]]
## class: DMN
## k: 7
## samples x taxa: 120 x 7
## Laplace: 4708.256 BIC: 4768.619 AIC: 4691.963

miaViz::plotDMNFit(tse_dmn, type = "laplace")

```



```
getBestDMNFit(tse_dmn, type = "laplace") # Gives 2 as best fit for phylum level data
```

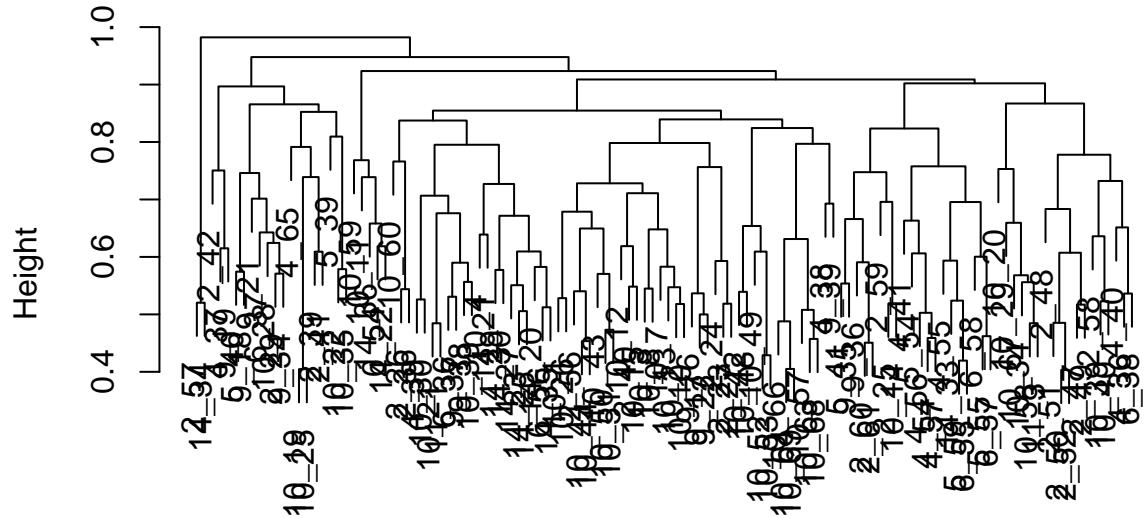
```
## class: DMN
## k: 2
## samples x taxa: 120 x 7
## Laplace: 4648.607 BIC: 4667.756 AIC: 4646.85
```

Hierarchal clustering BC on ASV level

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)
tse <- transformCounts(tse, method = "relabundance")
tse <- runMDS(tse,
               assay.type = "relabundance",
               FUN = vegan::vegdist,
               method = "bray")

hc_bray <- hclust(vegdist(t(assay(tse, "relabundance"))), method = "bray"), method = "complete")
plot(hc_bray)
```

Cluster Dendrogram



```
vegdist(t(assay(tse, "relabundance")), method = "bray")
hclust (*, "complete")
```

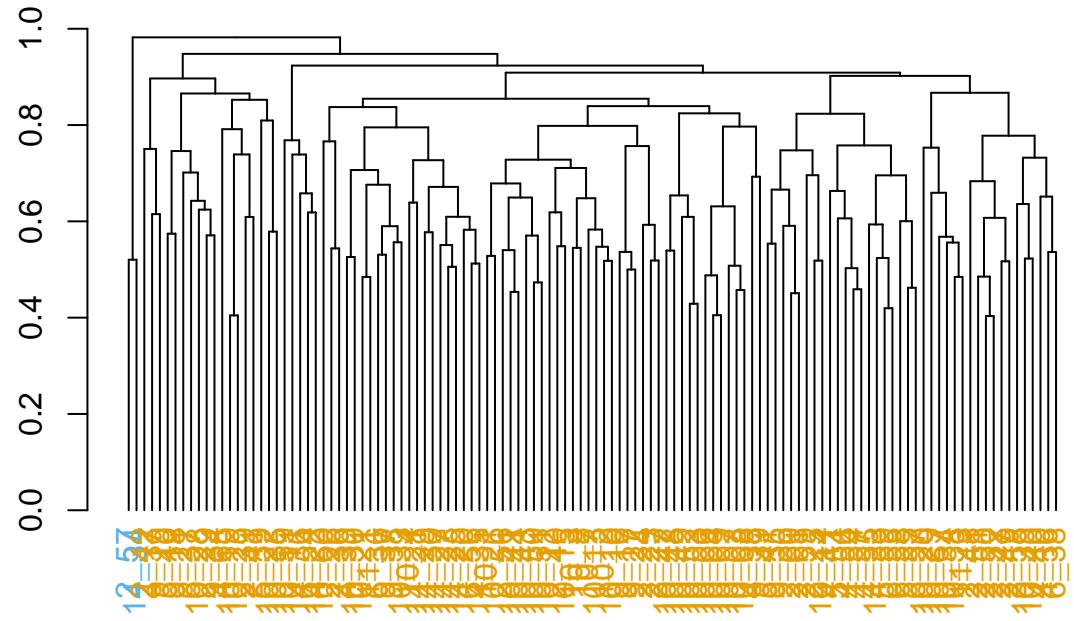
```
hcd = as.dendrogram(hc_bray)

cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442",
               "#0072B2", "#D55E00", "#CC79A7")

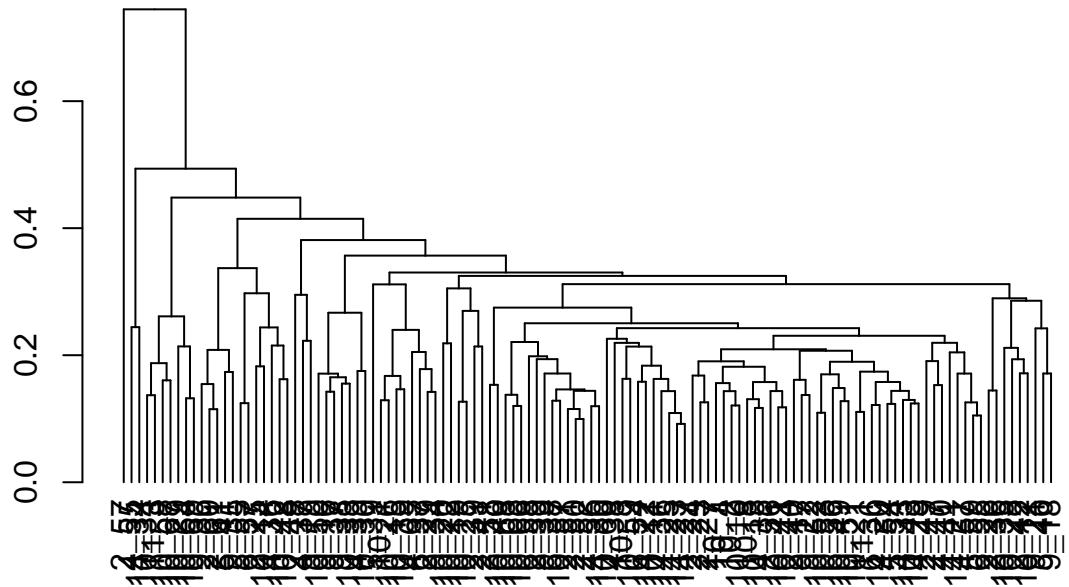
colorCode <- c(Control=cbPalette[2], CRC = cbPalette[3])

grouping = cutree(hc_bray, k = 2) # most methods gave 2 clusters, based on cuttree

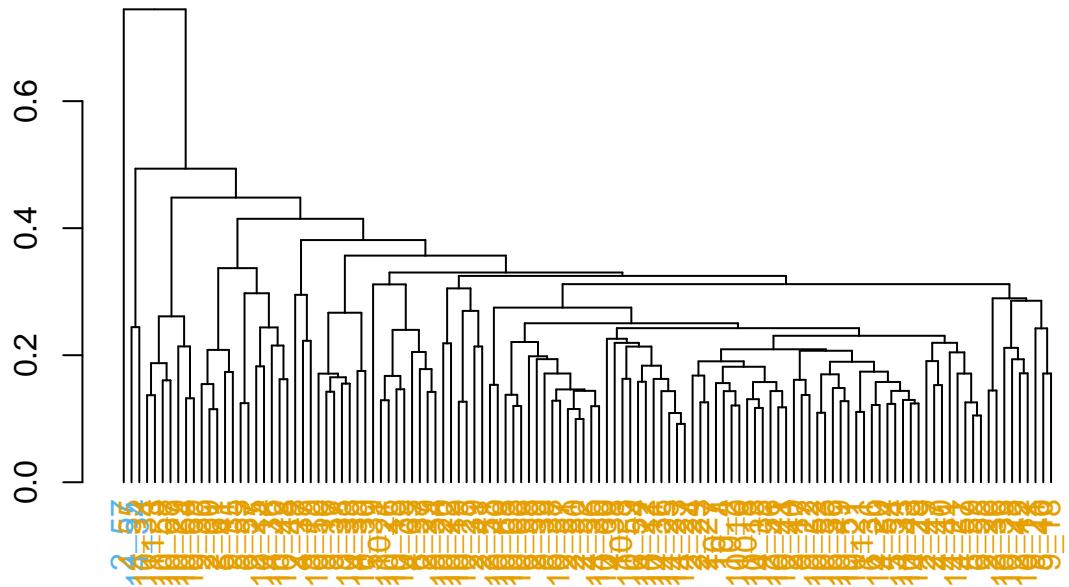
labels_colors(hcd) <- colorCode[grouping][order.dendrogram(hcd)]
plot(hcd)
```



```
hclust.out <- clusterRows(assay, HclustParam(method = "complete"), full = TRUE) # cutting based on complete linkage
colData(tse)$clusters <- hclust.out$clusters
dendro <- as.dendrogram(hclust.out$objects$hclust)
plot(dendro)
```



```
labels_colors(dendro) <- colorCode[grouping][order.dendrogram(dendro)]
plot(dendro)
```

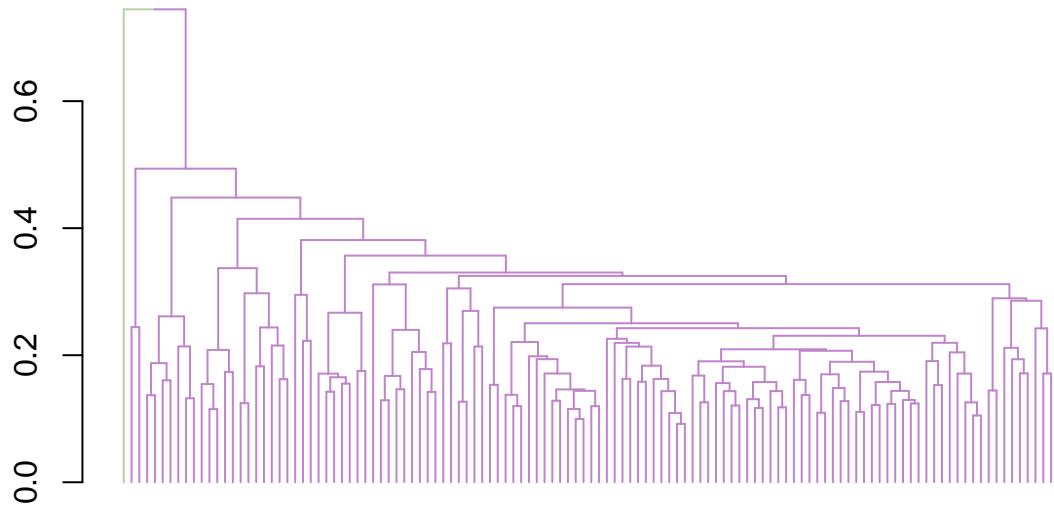


```

col_val_map <- randomcoloR::distinctColorPalette("2") %>%
  as.list() %>%
  setNames(paste0("clust_", seq("2")))

dend <- color_branches(dendro, k = 2, col = unlist(col_val_map))
labels(dend) <- NULL
plot(dend) # based on all three visualisations, only a few samples are clustered distinctly, based on s

```



```
## PAM clustering
```

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)

tse <- transformCounts(tse, method = "relabundance")

pam.out <- clusterCells(tse,
                         assay.type = "relabundance",
                         BLUSPARAM = PamParam(centers = 2))

pam.out
```

```
## 2_23 2_24 2_25 2_26 2_27 2_29 2_36 2_39 2_40 2_41 2_42 2_47 2_48
## 1     1     2     1     1     1     1     1     1     1     1     1     1     1
## 2_49 2_50 2_51 2_52 2_56 2_57 2_58 2_59 2_60 2_61 4_36 4_37 4_38
## 1     1     1     1     1     1     1     2     2     2     2     1     1
## 4_39 4_40 4_41 4_54 4_55 4_56 4_57 4_65 5_39 5_40 5_41 5_54 5_55
## 1     1     1     1     2     1     1     1     1     1     2     1     1
## 5_59 6_36 6_37 6_38 6_54 6_55 6_56 6_57 6_58 9_16 9_17 9_18 9_19
## 1     1     1     1     1     1     1     1     1     1     1     1     1
## 9_21 9_22 9_34 9_35 9_36 9_37 9_38 9_39 10_1 10_2 10_3 10_4 10_7
## 1     1     1     1     2     2     1     1     1     1     1     1     1
## 10_8 10_10 10_11 10_12 10_13 10_14 10_15 10_19 10_20 10_21 10_22 10_25 10_26
## 1     1     1     1     1     1     2     1     1     2     1     1     1
## 10_28 10_29 10_30 10_33 10_34 10_35 10_39 10_40 10_41 10_42 10_43 10_44 10_48
## 1     2     1     1     2     1     1     1     1     1     1     1     1
```

```

## 10_49 10_50 10_51 10_52 10_53 10_57 10_58 10_59 10_60 10_63 10_64 10_66 10_67
##   1     1     1     1     1     2     2     1     1     2     2     2     2
## 10_68 10_69 11_1 11_3 14_20 14_21 14_22 14_23 14_25 14_27 14_29 14_30 14_33
##   2     2     1     2     1     1     1     1     1     1     1     1     1
## 14_34 14_35 14_36
##   1     1     1
## Levels: 1 2

n_iterations <- 1000
previous_cluster_assignment <- NULL
cluster_assignments <- list()

# loop that runs PAM clusterings X times and stores the results in a list, additionally checks if any changes occur
for (i in 1:n_iterations) {
  result <- clusterCells(tse, assay.type = "relabundance", BLUSPARAM = PamParam(centers = 2))
  cluster_assignments[[i]] <- result

  # Check if cluster assignments have changed
  if (!is.null(previous_cluster_assignment)) {
    samples_changed <- which(result != previous_cluster_assignment)
    if (length(samples_changed) > 0) {
      cat(sprintf("In iteration %d, the following samples changed clusters: %s\n", i, paste(samples_changed)))
    }
  }
  previous_cluster_assignment <- result
}

# To see if all of the clusters are the same or not
if (all(sapply(cluster_assignments, identical, cluster_assignments[[1]]))) {
  cat("All cluster assignments are the same across iterations.\n")
} else {
  cat("Cluster assignments vary across iterations.\n")
}

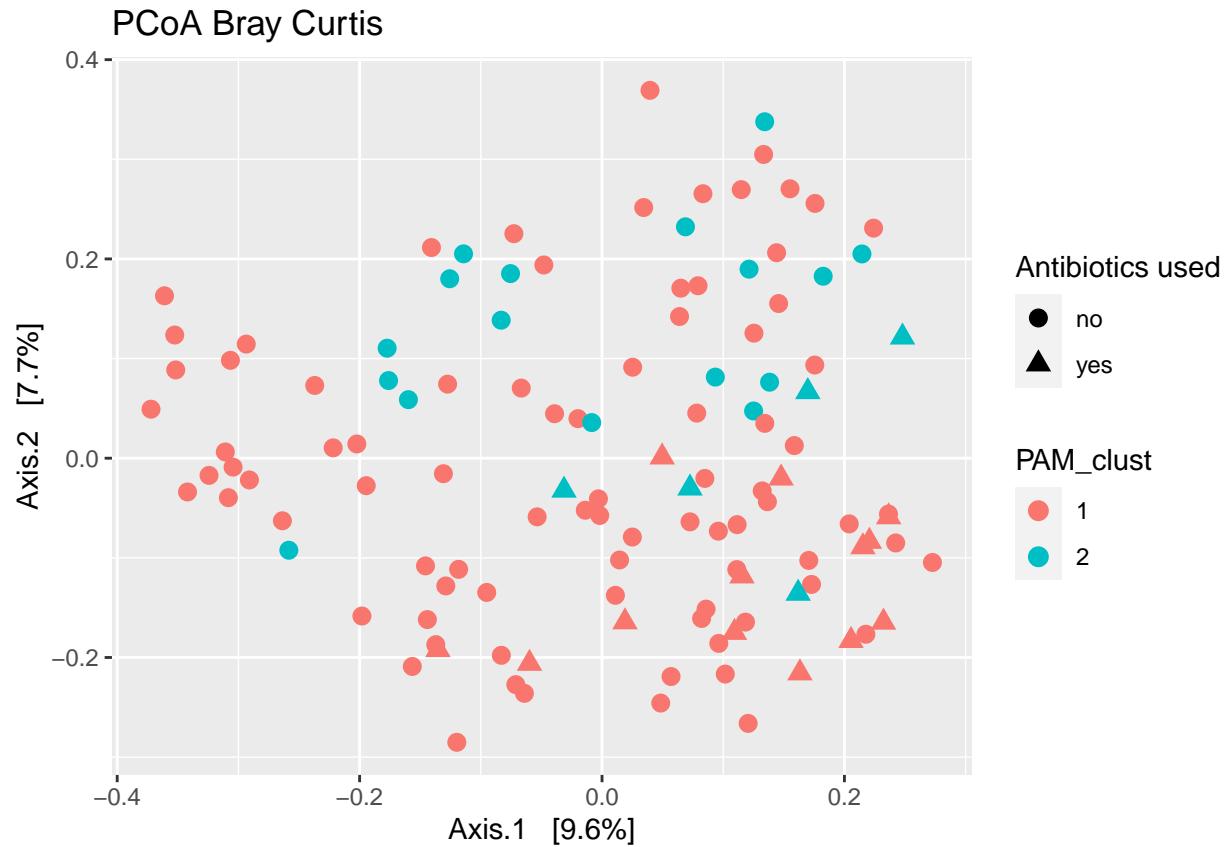
## All cluster assignments are the same across iterations.

# There are no differences in clusters when run 1000 times

# save to metadata and make original PCoA plot
subset16S@sam_data$PAM_clust = pam.out
sample_data(subset16S)$PAM_clust = as.factor(sample_data(subset16S)$PAM_clust)
pcoa_bc = ordinate(subset16S, "PCoA", "bray")

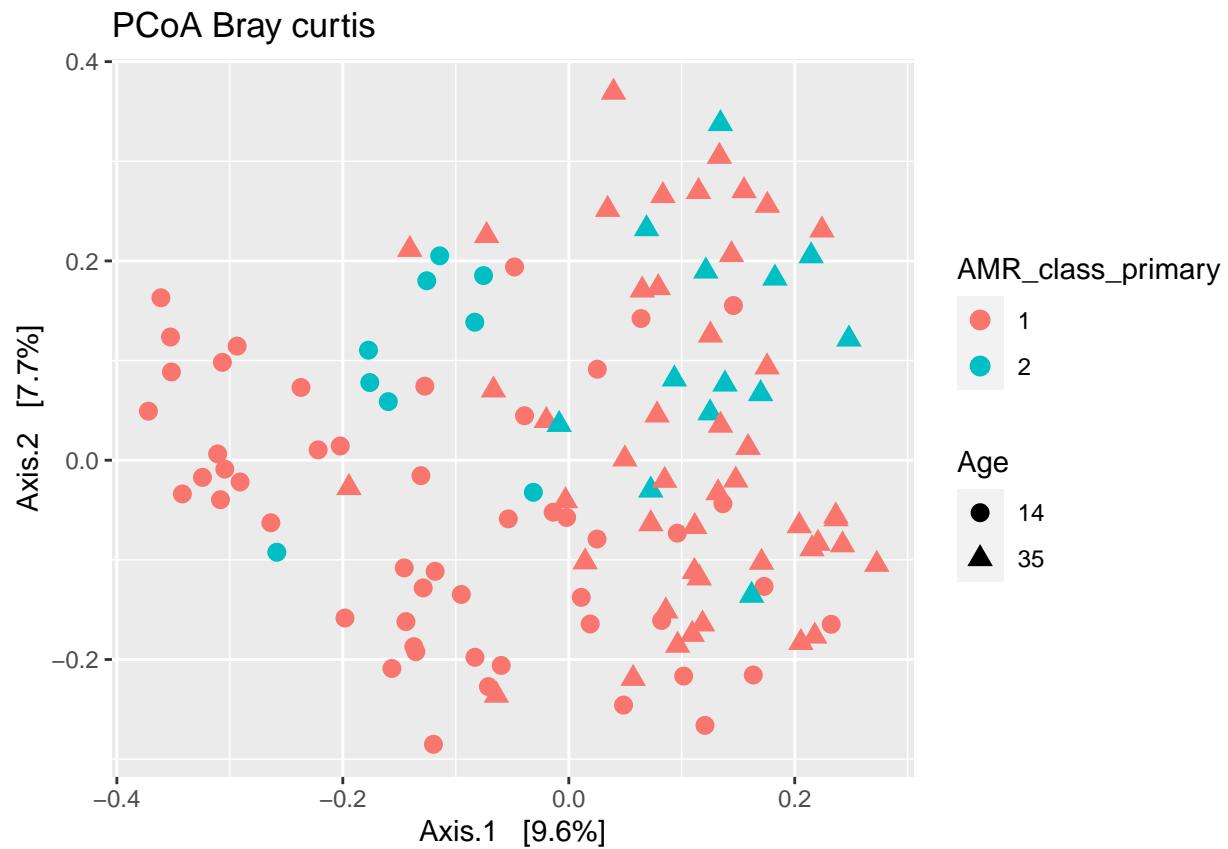
plot_pcoa_ordination(subset16S, pcoa_bc, "PAM_clust", "PCoA Bray Curtis")

```

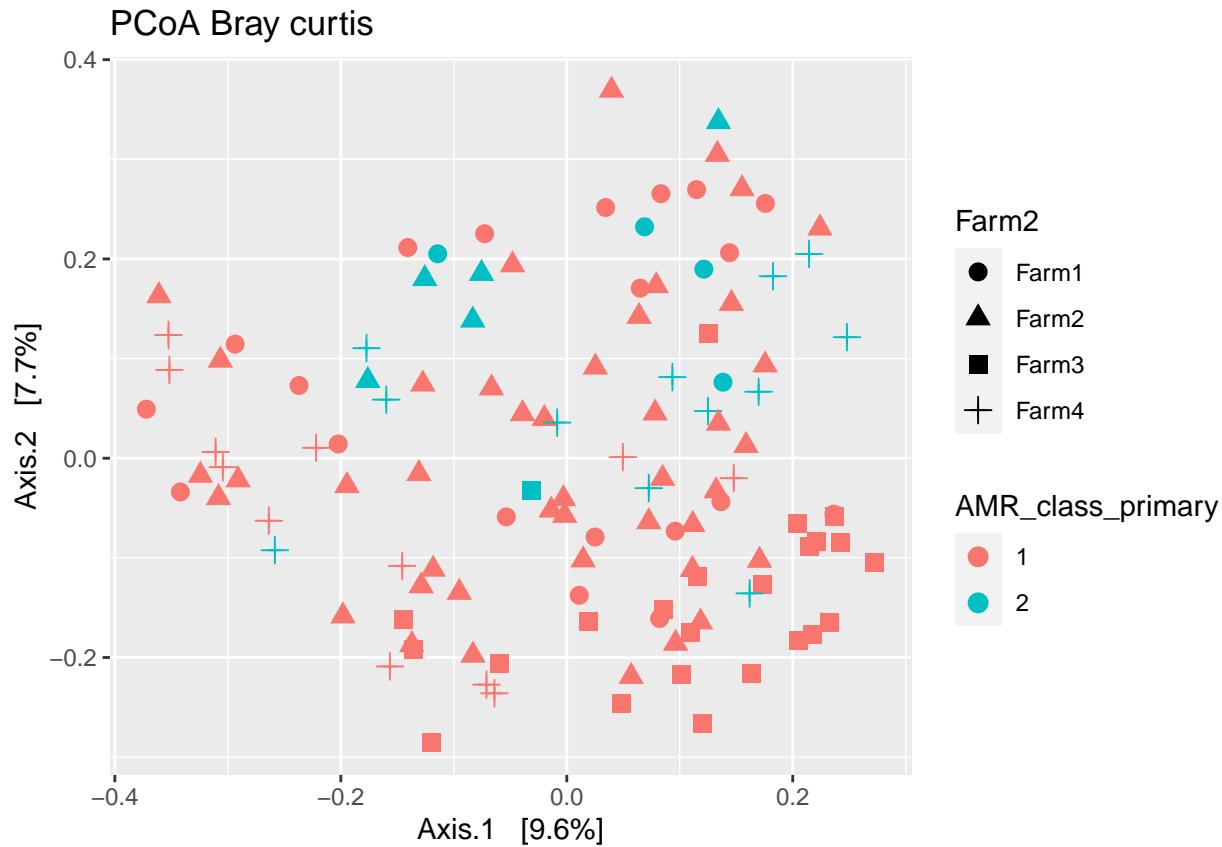


```
#plot_pcoa_ordination(subset16S, pcoa_bc, "Cluster", "PCoA Bray Curtis")

# change shape to different variables, age
plot_ordination(subset16S, pcoa_bc, color = "PAM_clust", shape = "Age") +
  geom_point(size = 3) + labs(title = "PCoA Bray curtis", color = "AMR_class_primary")
```



```
# change shape to different variables, farm
plot_ordination(subset16S, pcoa_bc, color = "PAM_clust", shape = "Farm2") +
  geom_point(size = 3) + labs(title = "PCoA Bray curtis", color = "AMR_class_primary")
```



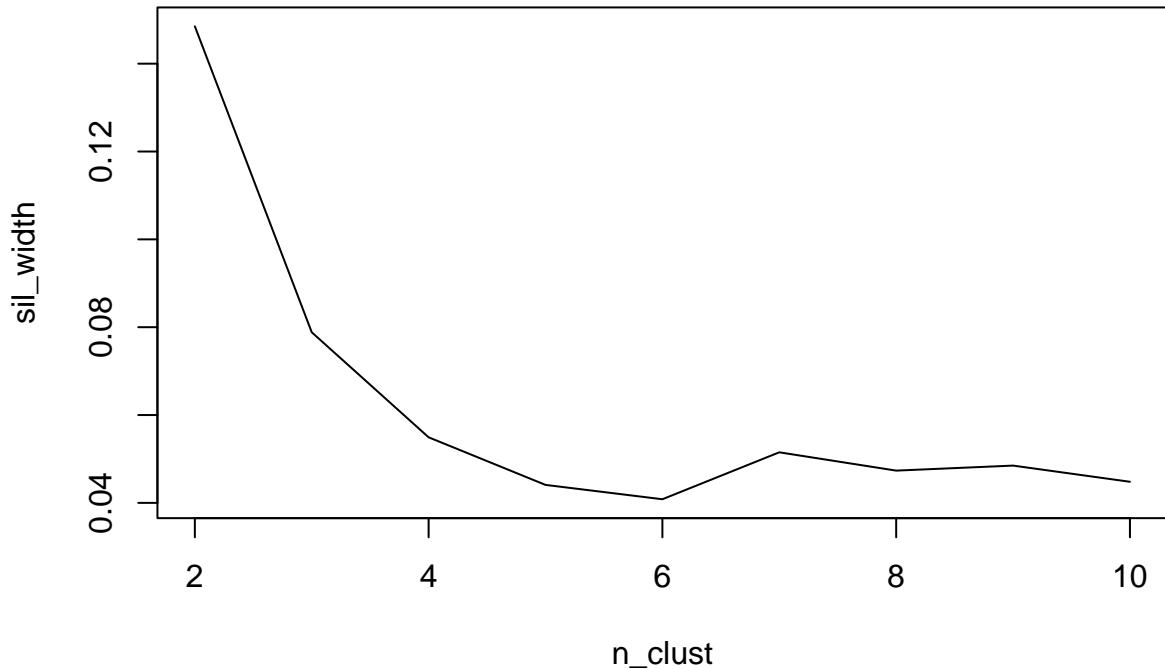
Create PAM UF PCoA - from 2 to 10 clusters

```

phy_rel <- transform_sample_counts(subset16S, function(x) log10(x+1/sum(x+1)))
UF <- UniFrac(phy_rel, weighted = TRUE)
n_clust <- 2:10
pam_list <- lapply(n_clust, function(x) pam(UF, k = x))

sil_width <- lapply(pam_list, function(x) mean(x$silinfo$widths[, "sil_width"]))
plot(n_clust, sil_width, type="l")

```



```

pcoa_data <- cmdscale(UF, eig = TRUE)
pcoa_df <- data.frame(PC1 = c(pcoa_data$points[,1]),
                       PC2 = c(pcoa_data$points[,2]),
                       Sample = rownames(pcoa_data$points))

# Add sample data
Samp <- data.frame(sample_data(subset16S))
Samp$Sample <- sample_names(subset16S)

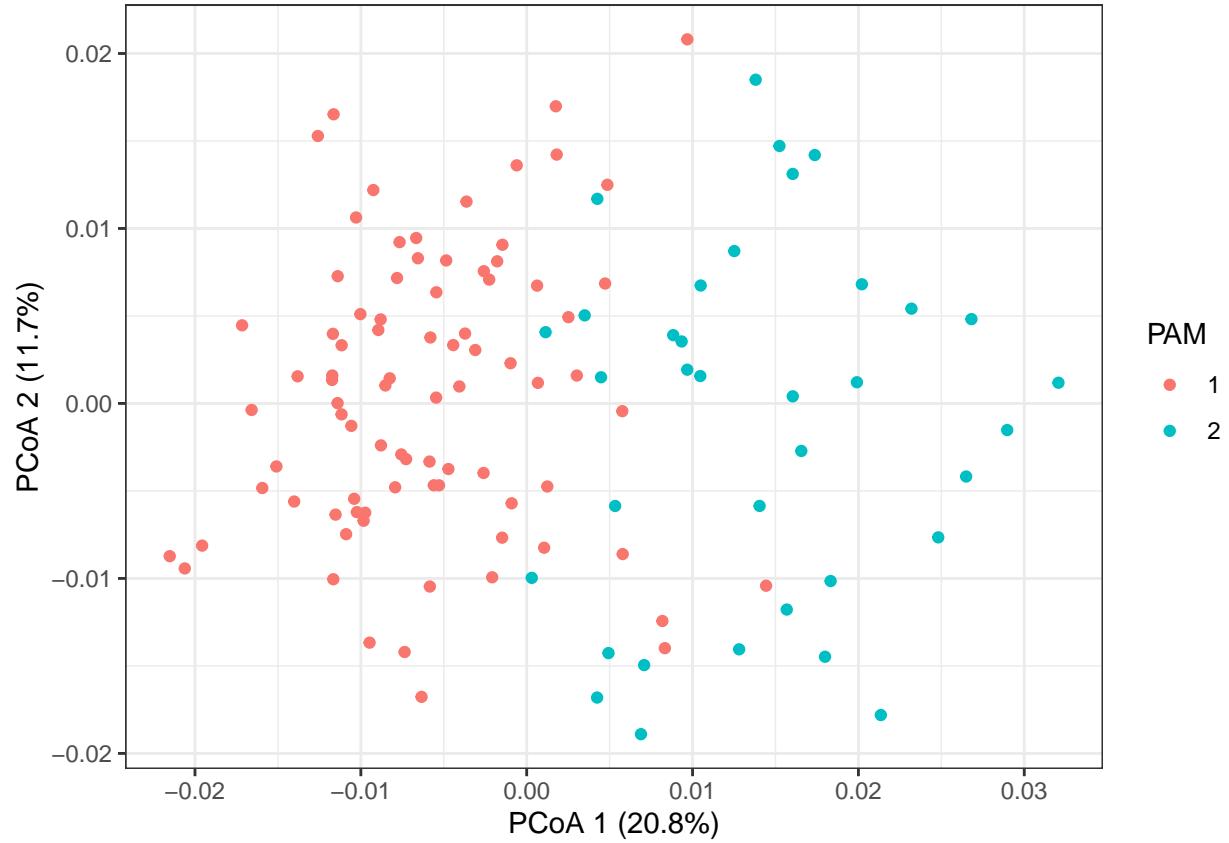
pcoa_df <- merge(pcoa_df, Samp, by = "Sample")

# Add cluster information
clusters <- factor(pam_list[[which.max(sil_width)]]$clustering)
pcoa_df <- merge(pcoa_df, clusters, by.x = "Sample", by.y = "row.names")
colnames(pcoa_df)[ncol(pcoa_df)] <- "PAM"

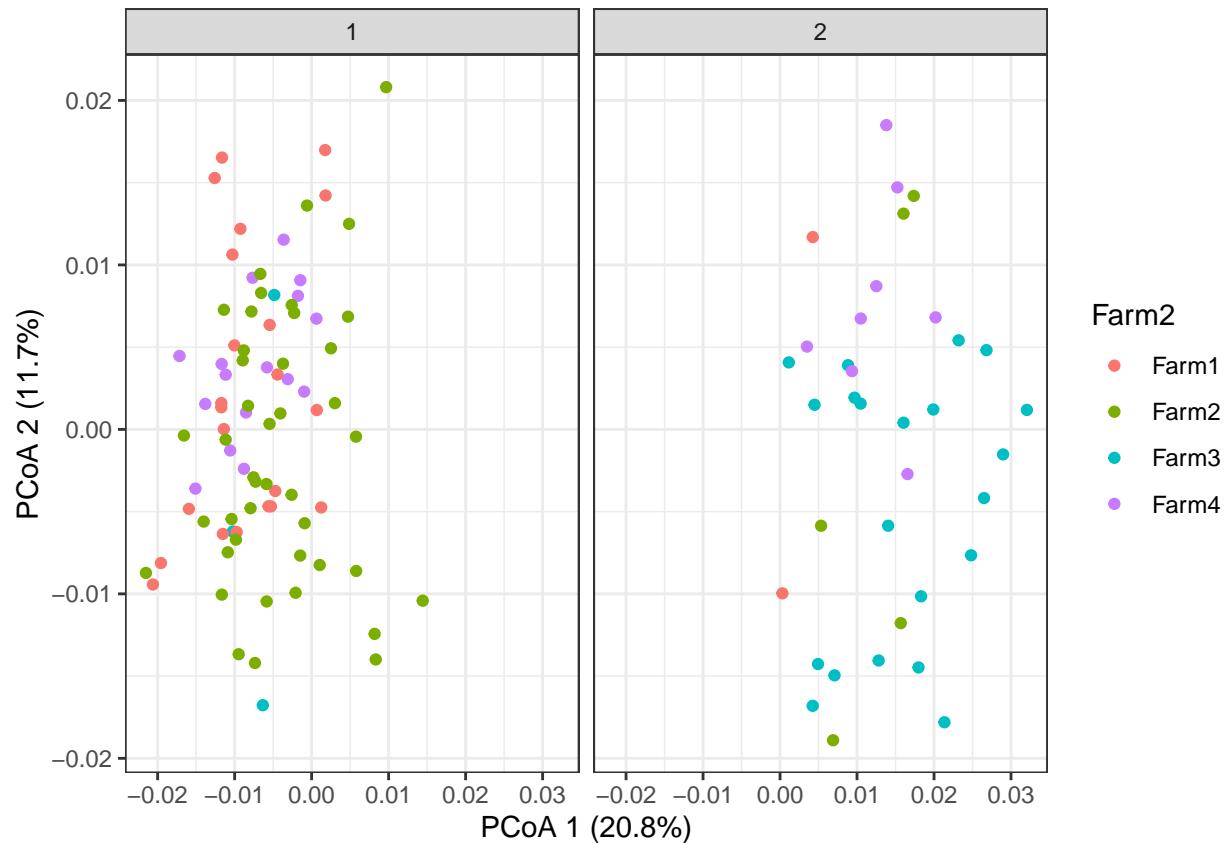
# Variance explained
ve <- pcoa_data$eig/sum(pcoa_data$eig)

# Plot
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = PAM)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)"))

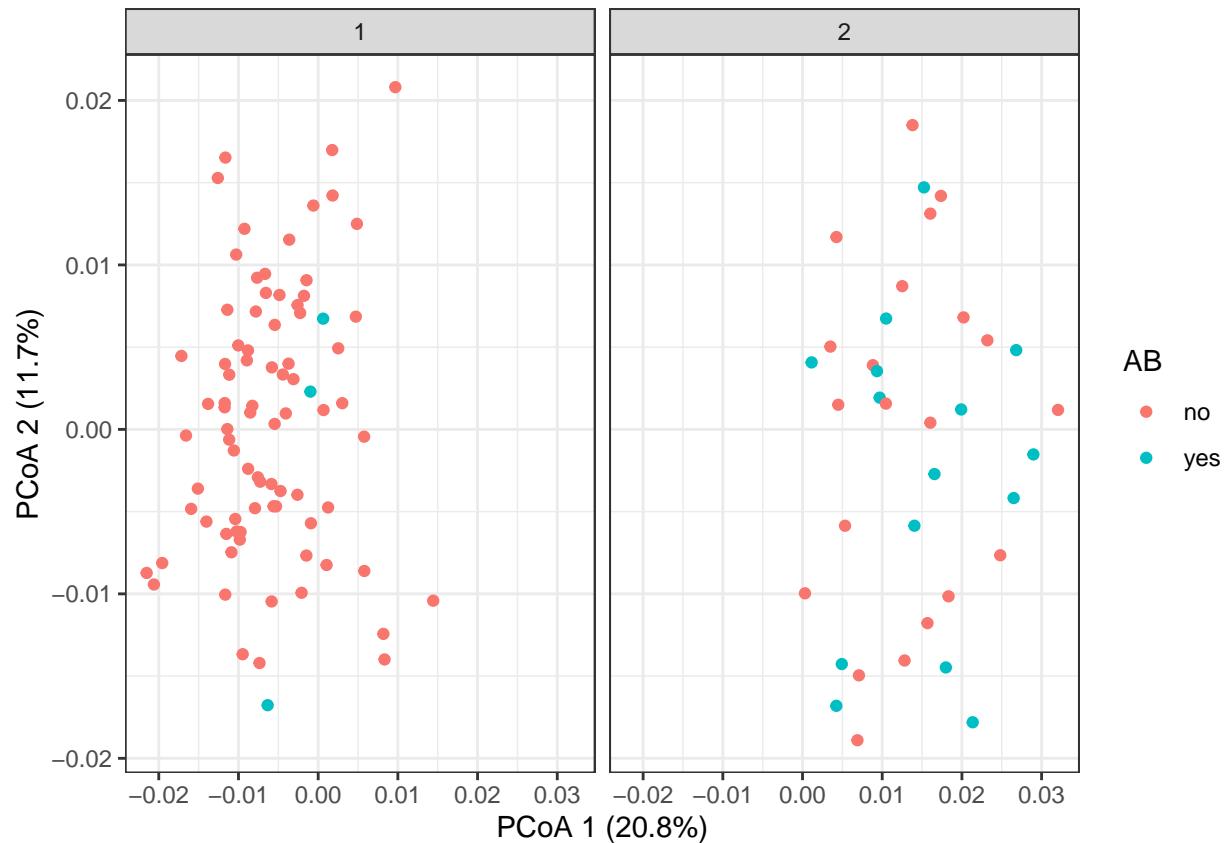
```



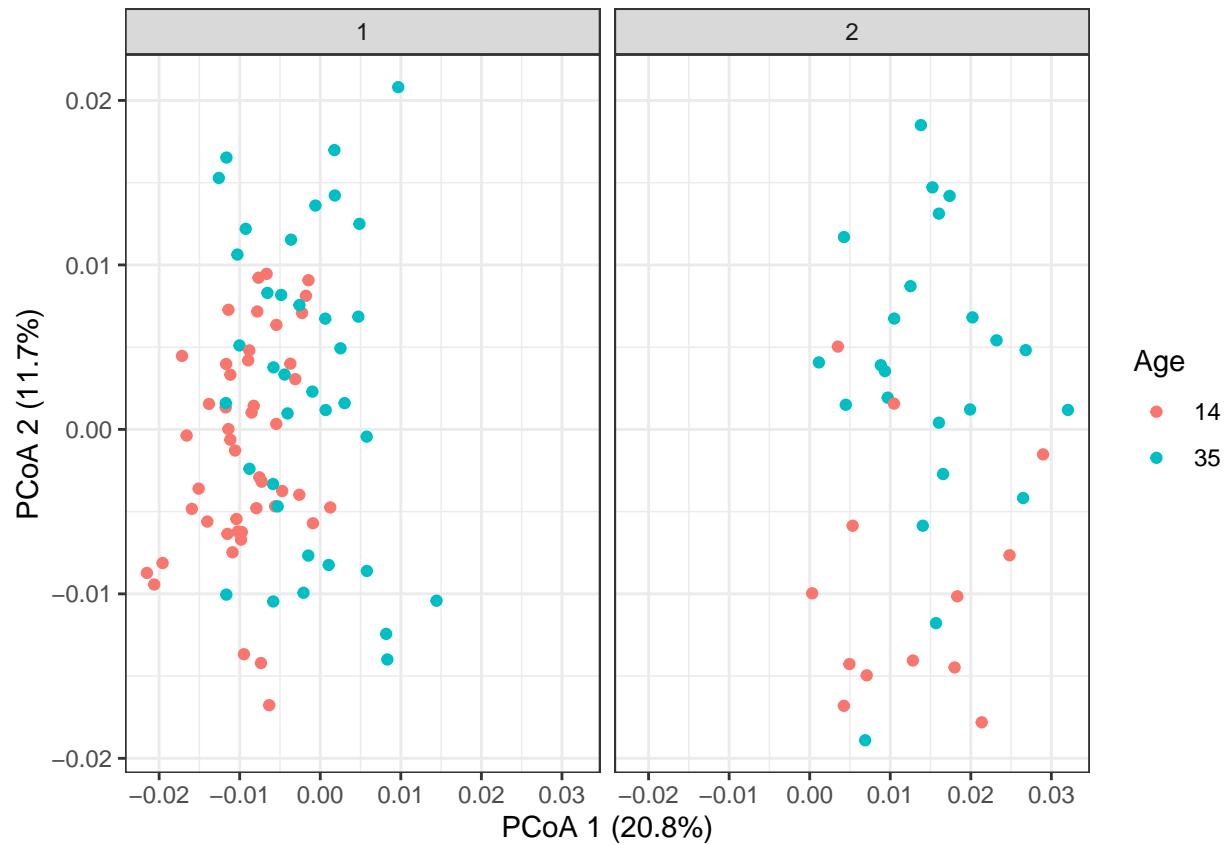
```
# facet by clusters and colour by farm
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = Farm2)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM)
```



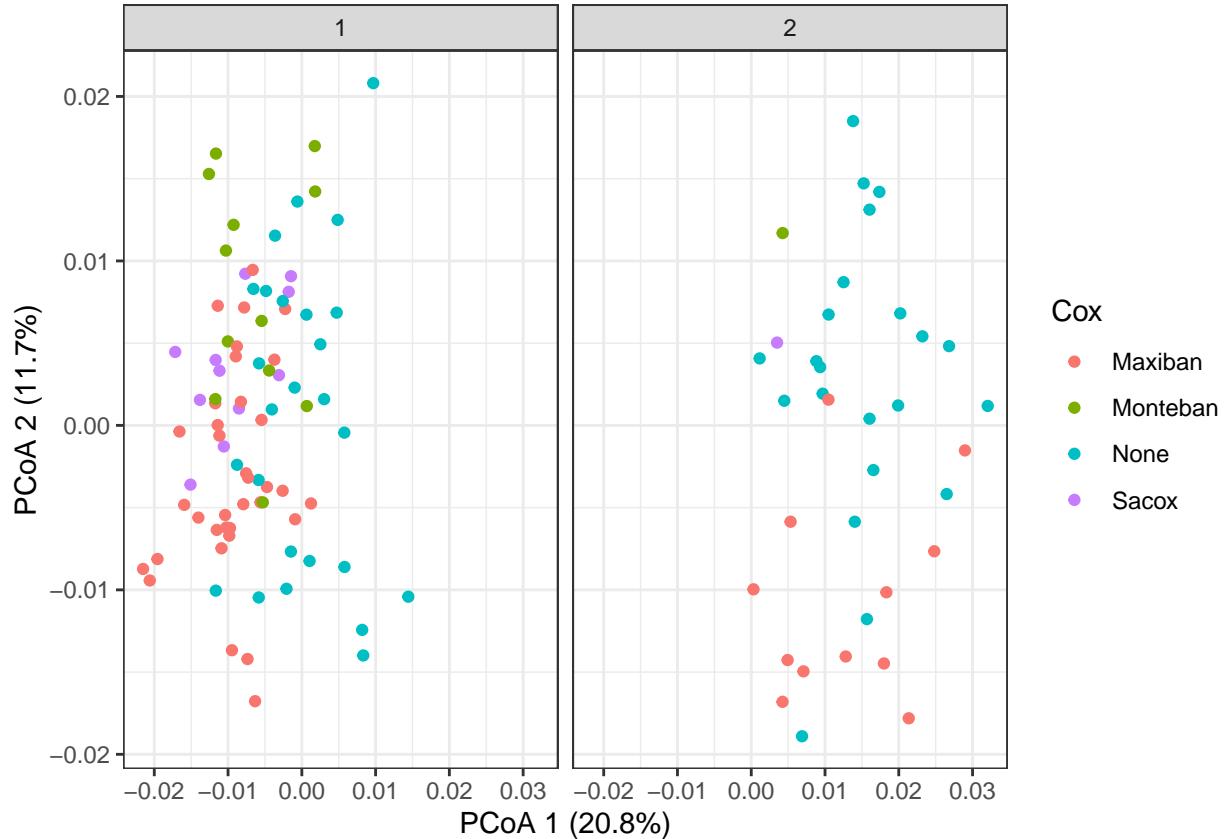
```
# facet by clusters and colour by AB
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = AB)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM)
```



```
# facet by clusters and colour by Age
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = Age)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM)
```



```
# facet by clusters and colour by Agent
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = Cox)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM)
```



```
## PCoA for Phylum data, BC with DMM, euclidian ( can change tax level with tse_dmn)
```

```
dmm_group <- calculateDMNgroup(tse_dmn,
                                variable = "Age", assay.type = "counts",
                                k = 2, seed = .Machine$integer.max)

dmm_group <- calculateDMNgroup(tse_dmn,
                                variable = "Farm2", assay.type = "counts",
                                k = 2, seed = .Machine$integer.max)
dmm_group <- calculateDMNgroup(tse_dmn,
                                variable = "AB", assay.type = "counts",
                                k = 2, seed = .Machine$integer.max)
dmm_group
```

```
## class: DMNGroup
## summary:
##      k samples taxa      NLE    LogDet    Laplace      BIC      AIC
## no   2     102    7 3831.1379 59.71807 3847.2128 3865.8252 3846.1379
## yes  2      18    7 818.2031 47.31140  828.0747  839.8809  833.2031
```

```
DirichletMultinomial::mixturewt(getBestDMNFit(tse_dmn)) # measure weights
```

```
##      pi      theta
## 1 0.628418 31.806259
## 2 0.371582  6.573224
```

```

head(DirichletMultinomial::mixture(getBestDMNFit(tse_dmn))) # sample-cluster assignment probabilities

##          [,1]      [,2]
## 2_23 0.9727179084 0.02728209
## 2_24 0.0175821347 0.98241787
## 2_25 0.0010190173 0.99898098
## 2_26 0.0010546064 0.99894539
## 2_27 0.9280199742 0.07198003
## 2_29 0.0008818687 0.99911813

head(DirichletMultinomial::fitted(getBestDMNFit(tse_dmn))) # taxa contribution

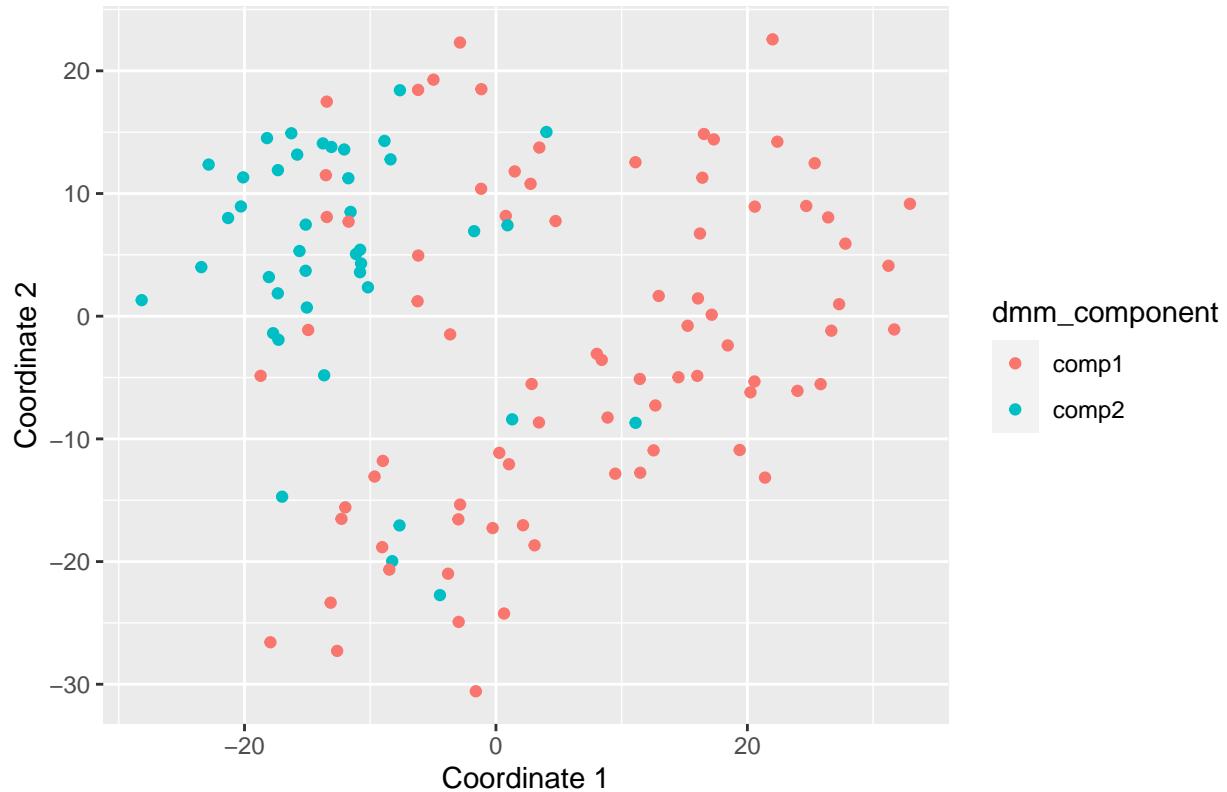
##          [,1]      [,2]
## Phylum:p__Tenericutes    1.09277350 0.0842149816
## Phylum:p__Firmicutes     28.74250221 5.6267113566
## Phylum:p__Bacteroidetes   0.89459787 0.1209397315
## Phylum:p__Proteobacteria  0.28242718 0.2800613651
## Phylum:p__Actinobacteria  0.71698835 0.4555026874
## Phylum:p__Verrucomicrobia  0.01598885 0.0002660301

prob <- DirichletMultinomial::mixture(getBestDMNFit(tse_dmn))
colnames(prob) <- c("comp1", "comp2")
vec <- colnames(prob)[max.col(prob, ties.method = "first")]
assay(tse, "pseudo") <- assay(tse, "counts") + 1
tse <- transformCounts(tse, assay.type = "pseudo", method = "relabundance")
tse <- transformCounts(tse, "relabundance", method = "clr")
df <- calculateMDS(tse, assay.type = "clr", method = "euclidean")
euclidean_pcoa_df <- data.frame(
  pcoa1 = df[, 1],
  pcoa2 = df[, 2])
euclidean_dmm_pcoa_df <- cbind(euclidean_pcoa_df,
                                   dmm_component = vec)

ggplot(data = euclidean_dmm_pcoa_df, aes(x = pcoa1, y = pcoa2, color = dmm_component)) +
  geom_point() +
  labs(x = "Coordinate 1",
       y = "Coordinate 2",
       title = "PCoA with Aitchison distances")

```

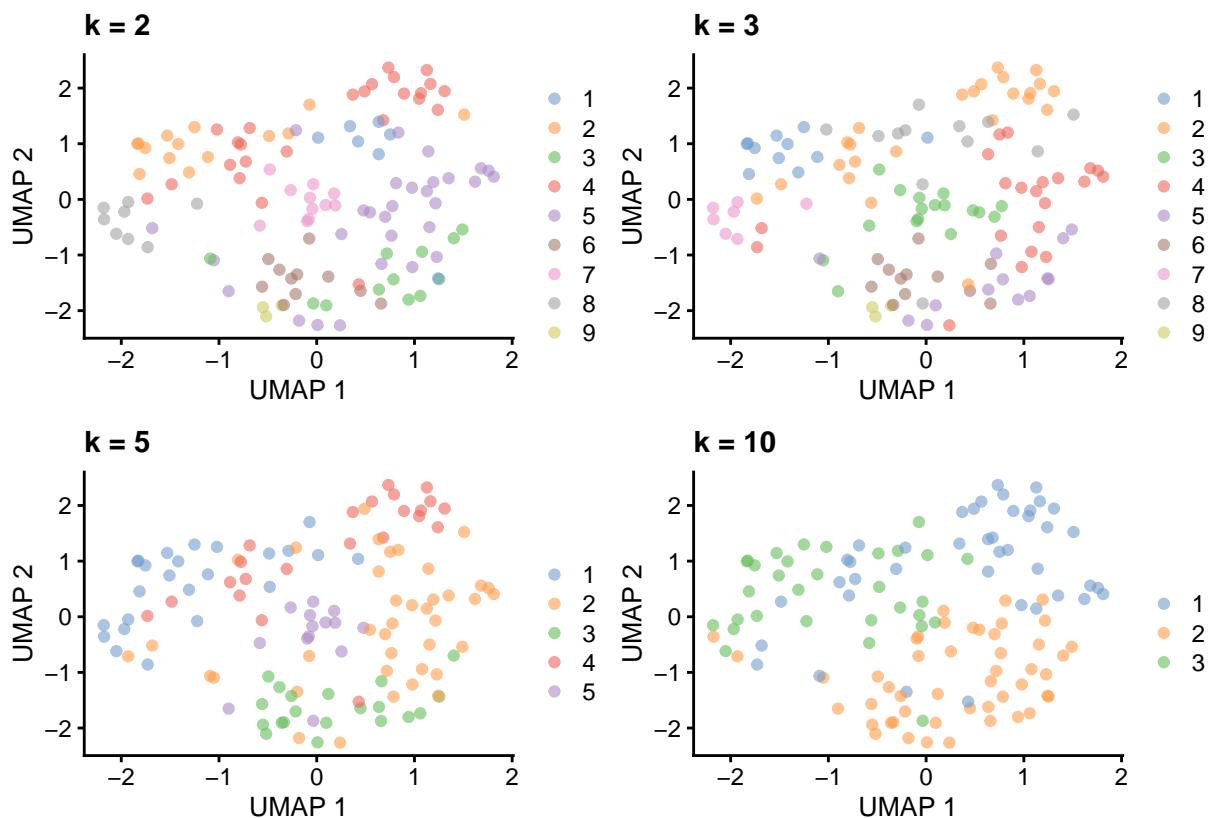
PCoA with Aitchison distances



```
## UMAP with different ks
```

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subset16S)
tse <- transformCounts(tse, method = "rclr")
tse <- runUMAP(tse, name = "UMAP", assay.type = "rclr")
k <- c(2, 3, 5, 10)
ClustAndPlot <- function(x) {
  # Creating the graph and running the short random walks algorithm
  graph_clusters <- clusterRows(t(assays(tse)$rclr), NNGraphParam(k = x))

  # Results of the clustering as a color for each sample
  plotUMAP(tse, colour_by = I(graph_clusters)) +
    labs(title = paste0("k = ", x))
}
plots <- lapply(k, ClustAndPlot)
(plots[[1]] + plots[[2]]) / (plots[[3]] + plots[[4]])
```



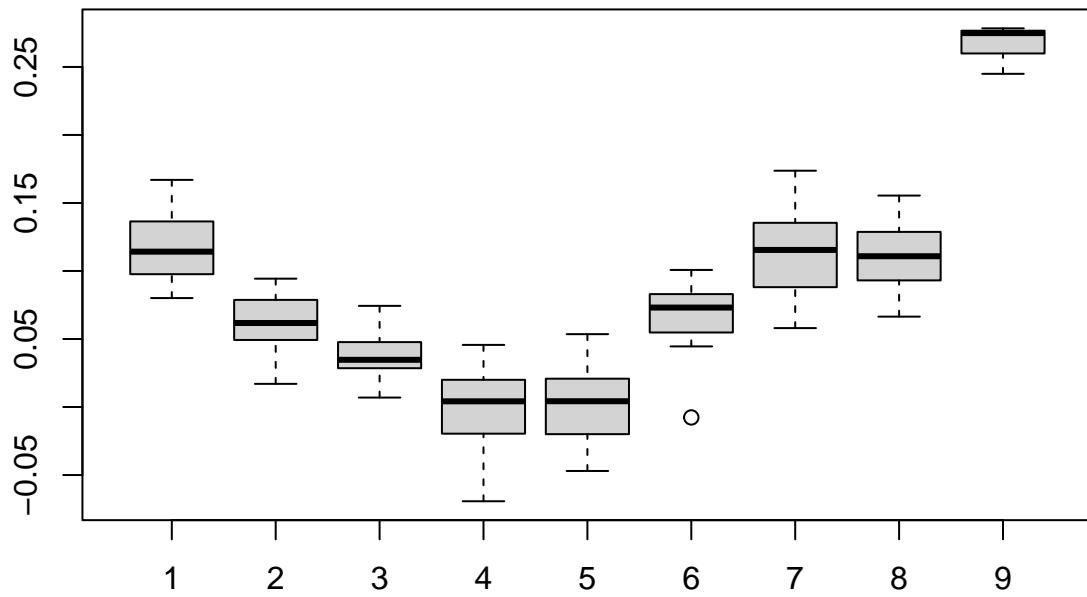
```
# boxplots
ClustDiagPlot <- function(x) {
  # Getting the clustering results
  graph_clusters <- clusterRows(t(assays(tse)$rclr), NNGraphParam(k = x))

  # Computing the diagnostic info
  sil <- approxSilhouette(t(assays(tse)$rclr), graph_clusters)

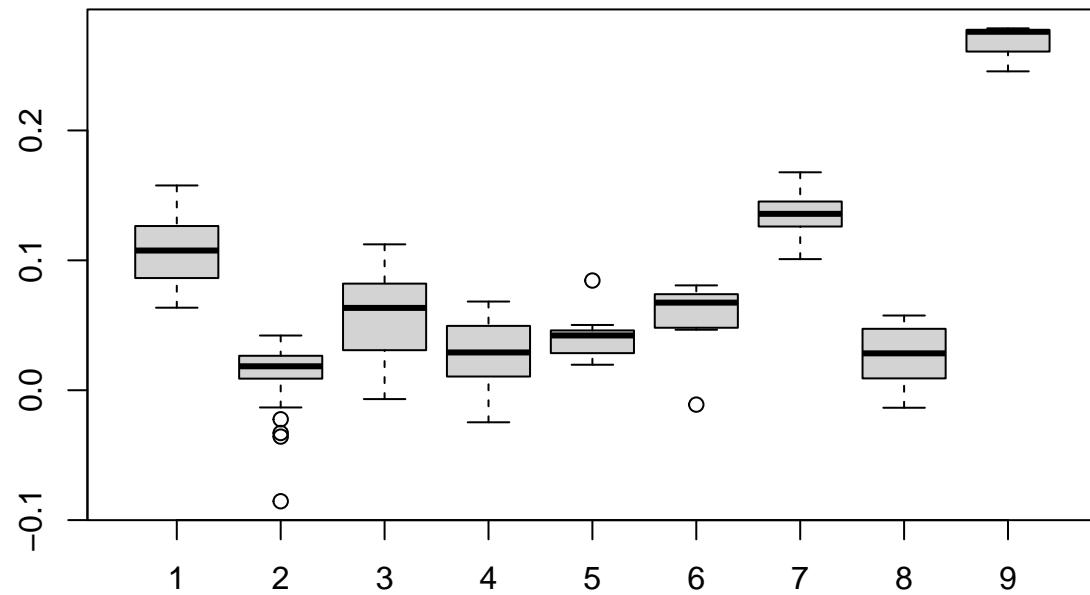
  # Plotting as a boxplot to observe cluster separation
  boxplot(split(sil$width, graph_clusters), main = paste0("k = ", x))
}

# Applying the function for different k values
res <- lapply(k, ClustDiagPlot)
```

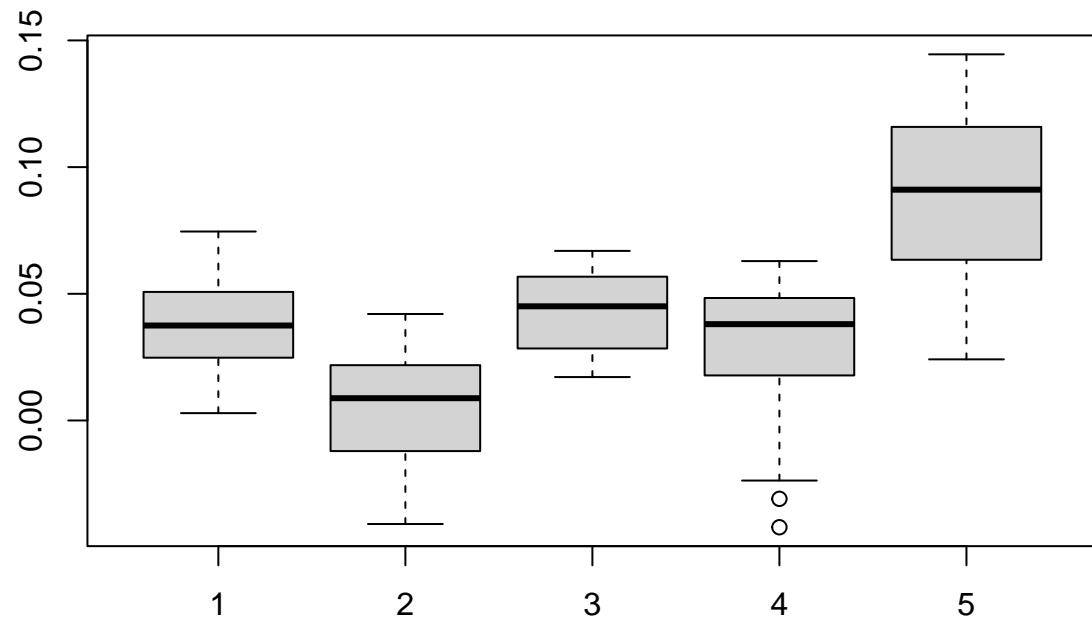
k = 2



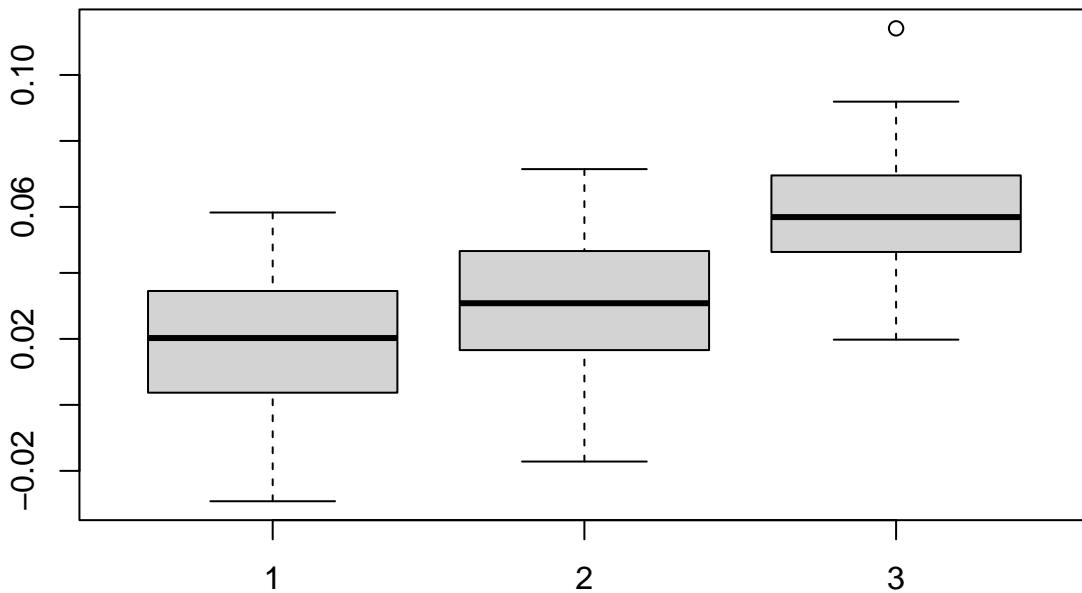
k = 3



k = 5



k = 10



Metagenomic data

Loading data

```
### Loading a subset of metagenomic data into phyloseq format
subsetMG= import_biom("kraken2_output.biom") # this imports a .biom created by kraken2-biom containing

# We rewrite the sample names to a format filtering out Firm and firm and the first underscore so that
sample_names(subsetMG) = sapply(regmatches(sample_names(subsetMG), regexpr("_", sample_names(subsetMG)))

# Because the names in both metadata sets do not completely overlap, we need to manually edit one of the
sample_names(subsetMG)[68] = "4_65"

# reading in and combining metadata from 16S and metagenomic origins, adding missing underscores
firm_names = read_excel("./Metagenomic/FIRM_MetaData.xlsx")
firm_names = firm_names[,-2] # Remove wrongful Raw_data_name column, to avoid confusion

meta_data = read.csv("MetaData.csv", header = TRUE, sep = ",")
meta_data_MG = dplyr::right_join(firm_names, meta_data, by="SampleID")

# using Sample_Unique as rownames so we can match the two sets in phyloseq
rownames(meta_data_MG) = meta_data_MG$Sample_Unique
```

```

# now we'll also add in microbial load
microbial_load = read.table("bacterial_load_kraken2.tab", sep = "\t", header = TRUE)
microbial_load$Sample_Unique = sapply(regmatches(microbial_load$Sample_Unique, regexpr("_", microbial_load$Sample_Unique)), function(x) x[1])
microbial_load$Sample_Unique[68] = "4_65"
meta_data_MG = dplyr::right_join(meta_data_MG, microbial_load, by="Sample_Unique")

# creating tree and making phyloseq components, adding tree and sample data components to phyloseq
set.seed("878") # setting seed for reproducibility purposes
random_tree = rtree(ntaxa(subsetMG), rooted=TRUE, tip.label=taxa_names(subsetMG))
meta_data_MG = sample_data(meta_data_MG)
rownames(meta_data_MG) = meta_data_MG$Sample_Unique
subsetMG = merge_phyloseq(subsetMG, meta_data_MG, random_tree)
class(subsetMG)

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

# set Rank names
colnames(tax_table(subsetMG)) = c("Domain", "Phylum", "Class", "Order", "Family", "Genus", "Species")
rank_names(subsetMG)

## [1] "Domain"   "Phylum"   "Class"    "Order"    "Family"   "Genus"    "Species"

### overview data
datatable(tax_table(subsetMG))

subsetMG # 7058 taxa

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 7058 taxa and 120 samples ]
## sample_data() Sample Data: [ 120 samples by 34 sample variables ]
## tax_table() Taxonomy Table: [ 7058 taxa by 7 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 7058 tips and 7057 internal nodes ]

# filter out non bacterial domains (no chloroplast, mitochondrial "taxa" present)
subsetMG <- subset_taxa(subsetMG, Domain!="k_Archaea")
subsetMG <- subset_taxa(subsetMG, Domain!="k_Viruses")
subsetMG <- subset_taxa(subsetMG, Domain!="k_Eukaryota")

subsetMG # 6355 taxa

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 6355 taxa and 120 samples ]
## sample_data() Sample Data: [ 120 samples by 34 sample variables ]
## tax_table() Taxonomy Table: [ 6355 taxa by 7 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 6355 tips and 6354 internal nodes ]

# Amount of different taxa present.
sort(table(tax_table(subsetMG)[, "Phylum"]))

```

```

##          p__Balneolota           p__Caldisericota
##                           1                   1
##          p__Calditrichota        p__Candidatus Bipolaricaulota
##                           1                   1
##          p__Candidatus Omnitrophota
##                           1
##          p__Coprothermobacterota
##                           1
##          p__Lentisphaerota
##                           1
##          p__Armatimonadota
##                           2
##          p__Kiritimatiellota
##                           2
##          p__Elusimicrobiota
##                           3
##          p__Candidatus Saccharibacteria
##                           5
##          p__Gemmatimonadota
##                           6
##          p__Nitrospirota
##                           7
##          p__Chlamydiota
##                           9
##          p__Synergistota
##                           15
##          p__Chlorobiota
##                           18
##          p__Acidobacteriota
##                           28
##          p__Fusobacteriota
##                           29
##          p__Deinococcota
##                           44
##          p__Planctomycetota
##                           65
##          p__Campylobacterota
##                           76
##          p__Thermodesulfobacteriota
##                           122
##          p__Bacillota
##                           1240
##          p__Pseudomonadota
##                           2383

```

```
sort(table(tax_table(subsetMG) [, "Order"]))
```

```

##          o__Acanthopleuribacterales      o__Acidothermales
##                           1                   1
##          o__Actinopolysporales          o__Balneolales
##                           1                   1
##          o__Bryobacterales            o__Caldilineales

```

```

##                               1                               1
##          o__ Caldisericales           o__ Calditrichales
##                               1                               1
##          o__ Candidatus Brocadiales   o__ Candidatus Izemoplasmatales
##                               1                               1
##  o__ Candidatus Saccharimonadales o__ Candidatus Sulfidibacteriales
##                               1                               1
##          o__ Catenulisporales         o__ Chroococcidiopsidales
##                               1                               1
##          o__ Chrysiogenales          o__ Chthonomonadales
##                               1                               1
##          o__ Coprothermobacteriales  o__ Desulfarculales
##                               1                               1
##          o__ Desulfobaccales        o__ Desulfomonilales
##                               1                               1
##          o__ Desulfurellales        o__ Egibacterales
##                               1                               1
##          o__ Egicoccales          o__ Elusimicrobiales
##                               1                               1
##          o__ Euzebyales            o__ Ferrovales
##                               1                               1
##          o__ Fibrobacteriales      o__ Fimbriimonadales
##                               1                               1
##          o__ Gloeomargaritales     o__ Haliangiales
##                               1                               1
##          o__ Hydrogenophilales    o__ Immundisolibacteriales
##                               1                               1
##          o__ Kineosporiales        o__ Kiritimatiellales
##                               1                               1
##          o__ Koleobacteriales      o__ Kordiimonadales
##                               1                               1
##          o__ Ktedonobacteriales    o__ Lentisphaerales
##                               1                               1
##          o__ Limisphaerales        o__ Limnochordales
##                               1                               1
##          o__ Magnetococcales       o__ Mesoaciditogales
##                               1                               1
##          o__ Nannocystales         o__ Nitrospiniales
##                               1                               1
##          o__ Phototrophicales      o__ Salinisphaerales
##                               1                               1
##          o__ Sneathiellales        o__ Sphaerobacteriales
##                               1                               1
##          o__ Sporichthyales        o__ Syntrophales
##                               1                               1
##          o__ Tepidisphaerales      o__ Thermoleophilales
##                               1                               1
##          o__ Thermomicrobiales     o__ Thermostichales
##                               1                               1
##          o__ Thermotomaculales     o__ Tichowtungiales
##                               1                               1
##          o__ Trueperales          o__ Bacteriovoracales
##                               1                               2
##  o__ Candidatus Pelagibacteriales o__ Dehalococcoidales

```

##		2		2
##	o__Desulfurobacteriales	2	o__Dictyoglomales	2
##		2		2
##	o__Emcibacteriales	2	o__Endomicrobiales	2
##		2		2
##	o__Kangiellales	2	o__Methylacidiphilales	2
##		2		2
##	o__Parvularculales	2	o__Pleurocapsales	2
##		2		2
##	o__Puniceicoccales	2	o__Saprospirales	2
##		2		2
##	o__Syntrophobacteriales	2	o__Thermodesulfovibrionales	2
##		2		2
##	o__Anaerolineales	3	o__Candidatus Babeliales	3
##		3		3
##	o__Candidatus Nanopelagicales	3	o__Candidatus Nanosynbacteriales	3
##		3		3
##	o__Glycomycetales	3	o__Jatrophihabitantales	3
##		3		3
##	o__Jiangellales	3	o__Kosmotogales	3
##		3		3
##	o__Mariprofundales	3	o__Miltoncostaeales	3
##		3		3
##	o__Natranaerobiales	3	o__Phycisphaerales	3
##		3		3
##	o__Silvanigrellales	3	o__Tepidiformales	3
##		3		3
##	o__Vicinamibacteriales	3	o__Acidiferrobacteriales	4
##		3		4
##	o__Bdellovibrionales	4	o__Bradymonadales	4
##		4		4
##	o__Cardiobacteriales	4	o__Chlamydiales	4
##		4		4
##	o__Maricaulales	4	o__Nakamurellales	4
##		4		4
##	o__Nevskiales	4	o__Orbales	4
##		4		4
##	o__Rhodothermales	4	o__Thermodesulfobacteriales	4
##		4		4
##	o__Thermosediminibacteriales	4	o__Aquificales	5
##		4		5
##	o__Brachyspirales	5	o__Chloroflexales	5
##		5		5
##	o__Gloeobacteriales	5	o__Nitrospirales	5
##		5		5
##	o__Parachlamydiales	5	o__Acidithiobacillales	6
##		5		6
##	o__Deferribacteriales	6	o__Gemmatimonadales	6
##		6		6
##	o__Holosporales	6	o__Hyphomonadales	6
##		6		6
##	o__Isosphaerales	6	o__Marinilabiliales	6
##		6		6
##	o__Nautiliales		o__Petrotogaes	

##		6		6
##	o__Geodermatophilales	7	o__Halanaerobiales	7
##				
##	o__Moorellales	7	o__Opitutales	7
##				
##	o__Sedimentisphaerales	7	o__Thermotogales	7
##				
##	o__Acidaminococcales	8	o__Frankiales	8
##				
##	o__Gemmatales	8	o__Oscillatoriales	8
##				
##	o__Solirubrobacterales	8	o__Acidimicrobiales	9
##				
##	o__Polyangiales	9	o__Acholeplasmatales	10
##				
##	o__Leptospirales	10	o__Rubrobacterales	10
##				
##	o__Chroococcales	11	o__Desulfobulbales	11
##				
##	o__Mycoplasmoidales	11	o__Legionellales	12
##				
##	o__Verrucomicrobiales	13	o__Desulfuromonadales	14
##				
##	o__Rickettsiales	14	o__Synergistales	15
##				
##	o__Pseudanabaenales	16	o__Veillonellales	16
##				
##	o__Desulfobacterales	17	o__Entomoplasmatales	17
##				
##	o__Planctomycetales	17	o__Selenomonadales	17
##				
##	o__Chlorobiales	18	o__Terriglobales	18
##				
##	o__Thermales	18	o__Eggerthellales	20
##				
##	o__Thiotrichales	20	o__Coriobacteriales	21
##				
##	o__Geobacterales	22	o__Pirellulales	22
##				
##	o__Chitinophagales	23	o__Deinococcales	25
##				
##	o__Methylococcales	25	o__Nostocales	25
##				
##	o__Tissierellales	25	o__Myxococcales	26
##				
##	o__Thermoanaerobacterales	26	o__Aeromonadales	27
##				
##	o__Fusobacteriales	29	o__Mycoplasmatales	30
##				
##	o__Erysipelotrichales	32	o__Cellvibrionales	34
##				
##	o__Pasteurellales	35	o__Sphingobacteriales	35
##				
##	o__Streptosporangiales		o__Nitrosomonadales	

##		36		37
##	o__Rhodocyclales		o__	38
##		37		
##	o__Bifidobacteriales		o__Caulobacterales	41
##		39		
##	o__Spirochaetales		o__Synechococcales	47
##		44		
##	o__Desulfovibrionales		o__Chromatiales	49
##		48		
##	o__Moraxellales		o__Vibrionales	52
##		50		
##	o__Actinomycetales		o__Alteromonadales	58
##		58		
##	o__Pseudonocardiales		o__Campylobacterales	69
##		67		
##	o__Neisseriales		o__Micromonosporales	74
##		70		
##	o__Oceanospirillales		o__Cytophagales	84
##		78		
##	o__Rhodospirillales		o__Xanthomonadales	102
##		91		
##	o__Propionibacteriales		o__Bacteroidales	116
##		108		
##	o__Flavobacteriales		o__Rhodobacterales	160
##		147		
##	o__Sphingomonadales		o__Pseudomonadales	190
##		174		
##	o__Enterobacterales		o__Kitasatosporales	256
##		254		
##	o__Lactobacillales		o__Mycobacteriales	303
##		296		
##	o__Eubacteriales		o__Burkholderiales	342
##		308		
##	o__Hyphomicrobiales		o__Micrococcales	436
##		362		
##	o__Bacillales			
##		483		

```
# sort(table(tax_table(subsetMG)[, "Family"])) -> too many to show
```

```
# Check the amount of unique Orders in samples which have and have not been treated with antibiotics
subsetMG %>% ps_filter(AB == "no") %>% get_taxa_unique("Order") # 200 different orders for non AB treat
```

## [1]	"o__Hyphomicrobiales"	"o__Burkholderiales"
## [3]	"o__Cytophagales"	"o__Holosporales"
## [5]	"o__Lactobacillales"	"o__Bacillales"
## [7]	"o__Sphingobacteriales"	"o__Vibrionales"
## [9]	"o__Pseudomonadales"	"o__Chromatiales"
## [11]	"o__Micrococcales"	"o__Mycoplasmatales"
## [13]	"o__Kitasatosporales"	"o__Rhodobacterales"
## [15]	"o__Eubacteriales"	"o__Desulfuromonadales"
## [17]	"o__Syntrophobacterales"	"o__Nitrosomonadales"
## [19]	"o__Caulobacterales"	"o__Actinomycetales"
## [21]	"o__Opitutales"	"o__Tissierellales"

```

## [23] "o__Nostocales"
## [25] "o__Neisseriales"
## [27] "o__Enterobacteriales"
## [29] "o__Methylococcales"
## [31] "o__Natrananaerobiales"
## [33] "o__Sphingomonadales"
## [35] "o__Xanthomonadales"
## [37] "o__Cellvibrionales"
## [39] "o__Pseudonocardiales"
## [41] "o__Bacteroidales"
## [43] "o__Oceanospirillales"
## [45] "o__Campylobacteriales"
## [47] "o__Aeromonadales"
## [49] "o__Desulfobacteriales"
## [51] "o__Hyphomonadales"
## [53] "o__Synechococcales"
## [55] "o__Flavobacteriales"
## [57] "o__Chlorobiales"
## [59] "o__Legionellales"
## [61] "o__Terriglobales"
## [63] "o__Spirochaetales"
## [65] "o__Tepidiformales"
## [67] "o__Candidatus Nanosynbacteriales"
## [69] "o__Moraxellales"
## [71] "o__Entomoplasmatales"
## [73] "o__Desulfovibrionales"
## [75] "o__Mycoplasmidales"
## [77] "o__Geobacterales"
## [79] "o__Rhodocyclales"
## [81] "o__Planctomyctales"
## [83] "o__Erysipelotrichales"
## [85] "o__Acidimicrobiales"
## [87] "o__Leptospirales"
## [89] "o__Thiotrichales"
## [91] "o__Thermoanaerobacteriales"
## [93] "o__Saprospirales"
## [95] "o__Desulfobulbales"
## [97] "o__Thermales"
## [99] "o__Miltoncostaeales"
## [101] "o__Ktedonobacteriales"
## [103] "o__Oscillatoriiales"
## [105] "o__Nautiliales"
## [107] "o__Sedimentisphaerales"
## [109] "o__Acholeplasmatales"
## [111] "o__Candidatus Babeliales"
## [113] "o__Pleurocapsales"
## [115] "o__Nitrospirales"
## [117] "o__Emcibacteriales"
## [119] "o__Gemmimonadales"
## [121] "o__Egibacteriales"
## [123] "o__Brachyspirales"
## [125] "o__Gemmatales"
## [127] "o__Halanaerobiales"
## [129] "o__Desulfarculales"
"o__Veillonellales"
"o__Myxococcales"
"o__Propionibacteriales"
"o__Alteromonadales"
"o__Bifidobacteriales"
"o__Pasteurellales"
"o__Micromonosporales"
"o__Rhodospirillales"
"o__Acidaminococcales"
"o__Chitinophagales"
"o__Deinococcales"
"o__Pseudanabaenales"
"o__Mycobacteriales"
"o__Bryobacteriales"
"o__Streptosporangiales"
"o__Thermosediminibacteriales"
"o__Solirubrobacteriales"
"o__"
"o__Acidiferrobacteriales"
"o__Polyangiales"
"o__Verrucomicrobiales"
"o__Rubrobacteriales"
"o__Frankiales"
"o__Eggerthellales"
"o__Selenomonadales"
"o__Pirellulales"
"o__Fusobacteriales"
"o__Chlamydiales"
"o__Methylacidiphilales"
"o__Deferrribacteriales"
"o__Parachlamydiales"
"o__Acidithiobacillales"
"o__Coriobacteriales"
"o__Rickettsiales"
"o__Chroococcales"
"o__Cardiobacteriales"
"o__Nevskiales"
"o__Kordiimonadales"
"o__Synergistales"
"o__Petrotogales"
"o__Puniceicoccales"
"o__Thermotogales"
"o__Kineosporiales"
"o__Anaerolineales"
"o__Chloroflexales"
"o__Nakamurellales"
"o__Phototrophicales"
"o__Nitrospiniales"
"o__Bradymonadales"
"o__Jatrophihabitantales"
"o__Sphaerobacteriales"
"o__Moorellales"
"o__Gloeobacteriales"
"o__Isosphaerales"

```

```

## [131] "o__Orbales"
## [133] "o__Maricaulales"
## [135] "o__Kosmotogales"
## [137] "o__Geodermatophilales"
## [139] "o__Marinilabiliales"
## [141] "o__Jiangellales"
## [143] "o__Bdellovibrionales"
## [145] "o__Ferrovales"
## [147] "o__Thermomicrobiales"
## [149] "o__Silvanigrellales"
## [151] "o__Euzebyales"
## [153] "o__Tepidisphaerales"
## [155] "o__Thermoleophilales"
## [157] "o__Phycisphaerales"
## [159] "o__Candidatus Sulfidibacteriales"
## [161] "o__Tichowtungiales"
## [163] "o__Caldisericales"
## [165] "o__Immundisolibacteriales"
## [167] "o__Vicinamibacteriales"
## [169] "o__Syntrophales"
## [171] "o__Desulfurellales"
## [173] "o__Trueperales"
## [175] "o__Haliangiales"
## [177] "o__Candidatus Pelagibacteriales"
## [179] "o__Hydrogenophilales"
## [181] "o__Kiritimatiellales"
## [183] "o__Catenulisporales"
## [185] "o__Actinopolysporales"
## [187] "o__Nannocystales"
## [189] "o__Calditrichales"
## [191] "o__Limnochordales"
## [193] "o__Egicoccales"
## [195] "o__Coprothermobacteriales"
## [197] "o__Chroococcidiopsidales"
## [199] "o__Mesoaciditogales"
"o__Dictyoglomales"
"o__Aquificales"
"o__Koleobacterales"
"o__Thermodesulfobacteriales"
"o__Thermostichales"
"o__Rhodothermales"
"o__Acanthopleuribacteriales"
"o__Glycomycetales"
"o__Candidatus Nanopelagicales"
"o__Sneathiellales"
"o__Endomicrobiales"
"o__Lentisphaerales"
"o__Kangiellales"
"o__Parvularculales"
"o__Desulfobaccales"
"o__Sporichthyales"
"o__Dehalococcoidales"
"o__Desulfurobacteriales"
"o__Mariprofundales"
"o__Acidothamales"
"o__Caldilineales"
"o__Salinisphaerales"
"o__Bacteriovoracales"
"o__Chrysiogenales"
"o__Desulfomonilales"
"o__Elusimicrobiales"
"o__Thermodesulfovibrionales"
"o__Gloeomargaritales"
"o__Candidatus Izemoplasmatales"
"o__Chthonomonadales"
"o__Limisphaerales"
"o__Fimbriimonadales"
"o__Fibrobacterales"
"o__Balneolales"
"o__Magnetococcales"

```

```
subsetMG %>% ps_filter(AB == "yes") %>% get_taxa_unique("Order") # 159 different orders for AB treated
```

```

## [1] "o__Burkholderiales"
## [3] "o__Hyphomicrobiales"
## [5] "o__Lactobacillales"
## [7] "o__Synechococcales"
## [9] "o__Pseudomonadales"
## [11] "o__Mycoplasmatales"
## [13] "o__Rhodobacterales"
## [15] "o__Syntrophobacterales"
## [17] "o__Tissierellales"
## [19] "o__Neisseriales"
## [21] "o__Enterobacterales"
## [23] "o__Bifidobacteriales"
## [25] "o__Actinomycetales"
## [27] "o__Pseudonocardiales"
## [29] "o__Bacteroidales"
## [31] "o__Rhodospirillales"
"o__Cytophagales"
"o__Flavobacteriales"
"o__Bacillales"
"o__Vibrionales"
"o__Chromatiales"
"o__Kitasatosporales"
"o__Eubacteriales"
"o__Opitutales"
"o__Veillonellales"
"o__Myxococcales"
"o__Propionibacteriales"
"o__Xanthomonadales"
"o__Mycobacteriales"
"o__Acidaminococcales"
"o__Oceanospirillales"
"o__Aeromonadales"

```

```

## [33] "o__Desulfobacterales"
## [35] "o__Micrococcales"
## [37] "o__Spirochaetales"
## [39] "o__Micromonosporales"
## [41] "o__Methylococcales"
## [43] "o__Legionellales"
## [45] "o__Terriglobales"
## [47] "o__Verrucomicrobiales"
## [49] "o__Alteromonadales"
## [51] "o__Sphingomonadales"
## [53] "o__Rickettsiales"
## [55] "o__Acidiferrobacterales"
## [57] "o__Entomoplasmatales"
## [59] "o__Desulfovibrionales"
## [61] "o__Mycoplasmoidales"
## [63] "o__Rhodocyclales"
## [65] "o__Nitrosomonadales"
## [67] "o__Planctomycetales"
## [69] "o__Caulobacterales"
## [71] "o__Acidimicrobiales"
## [73] "o__Leptospirales"
## [75] "o__Streptosporangiales"
## [77] "o__Nautiliales"
## [79] "o__Cardiobacterales"
## [81] "o__Miltoncostaeales"
## [83] "o__Chitinophagales"
## [85] "o__Sphingobacterales"
## [87] "o__Puniceicoccales"
## [89] "o__Thermales"
## [91] "o__Kineosporiales"
## [93] "o__Anaerolineales"
## [95] "o__Nakamurellales"
## [97] "o__Parachlamydiales"
## [99] "o__Oscillatoriiales"
## [101] "o__Nitrospiniales"
## [103] "o__Frankiales"
## [105] "o__Gemmatales"
## [107] "o__Moorellales"
## [109] "o__Desulfarculales"
## [111] "o__Isosphaerales"
## [113] "o__Thermosediminibacterales"
## [115] "o__Tepidiformales"
## [117] "o__Geodermatophilales"
## [119] "o__Bdellovibrionales"
## [121] "o__Rhodothermales"
## [123] "o__Nevskiales"
## [125] "o__Thermotogales"
## [127] "o__Pseudanabaenales"
## [129] "o__Chroococcales"
## [131] "o__Parvularculales"
## [133] "o__Tichowtungiales"
## [135] "o__Dehalococcoidales"
## [137] "o__Marinilabiliales"
## [139] "o__Candidatus Babeliales"

## [33] "o__Bryobacterales"
## [35] "o__Pasteurellales"
## [37] "o__Solirubrobacterales"
## [39] "o__Phycisphaerales"
## [41] "o__"
## [43] "o__Campylobacterales"
## [45] "o__Polyangiales"
## [47] "o__Rubrobacterales"
## [49] "o__Nostocales"
## [51] "o__Candidatus Brocadiales"
## [53] "o__Candidatus Saccharimonadales"
## [55] "o__Eggerthellales"
## [57] "o__Selenomonadales"
## [59] "o__Pirellulales"
## [61] "o__Geobacterales"
## [63] "o__Deinococcales"
## [65] "o__Aquificales"
## [67] "o__Erysipelotrichales"
## [69] "o__Candidatus Nanopelagicales"
## [71] "o__Acidithiobacillales"
## [73] "o__Coriobacterales"
## [75] "o__Moraxellales"
## [77] "o__Desulfuromonadales"
## [79] "o__Desulfobulbales"
## [81] "o__Fusobacterales"
## [83] "o__Cellvibrionales"
## [85] "o__Thermoanaerobacterales"
## [87] "o__Synergistales"
## [89] "o__Sedimentisphaerales"
## [91] "o__Acholeplasmatales"
## [93] "o__Chloroflexales"
## [95] "o__Nitrospirales"
## [97] "o__Thermodesulfobacterales"
## [99] "o__Emcibacterales"
## [101] "o__Bradymonadales"
## [103] "o__Sphaerobacterales"
## [105] "o__Chlorobiiales"
## [107] "o__Halanaerobiales"
## [109] "o__Gemmamimonadales"
## [111] "o__Thiotrichales"
## [113] "o__Gloeobacterales"
## [115] "o__Koleobacterales"
## [117] "o__Brachyspirales"
## [119] "o__Acanthopleuribacterales"
## [121] "o__Orbales"
## [123] "o__Petrogales"
## [125] "o__Glycomycetales"
## [127] "o__Sneathiellales"
## [129] "o__Euzebyales"
## [131] "o__Desulfobaccales"
## [133] "o__Sporichthyales"
## [135] "o__Immundisolibacterales"
## [137] "o__Vicinamibacterales"
## [139] "o__Thermodesulfovibrionales"

```

```

## [141] "o__Caldilineales"
## [143] "o__Candidatus Pelagibacterales" "o__Haliangiales"
## [145] "o__Thermotomaculales" "o__Kosmotogales"
## [147] "o__Kiritimatiellales" "o__Desulfomonilales"
## [149] "o__Nannocystales" "o__Gloeomargaritales"
## [151] "o__Candidatus Nanosynbacterales" "o__Candidatus Izemoplasmatales"
## [153] "o__Limnochordales" "o__Limisphaerales"
## [155] "o__Egicoccales" "o__Fimbriimonadales"
## [157] "o__Fibrobacterales" "o__Mesoaciditogales"
## [159] "o__Maricaulales"

```

```
subsetMG %>% get_taxa_unique("Order") # 203 different order in total, so 3 orders are not found in non
```

```

## [1] "o__Hyphomicrobiales" "o__Burkholderiales"
## [3] "o__Cytophagales" "o__Holosporales"
## [5] "o__Flavobacteriales" "o__Lactobacillales"
## [7] "o__Bacillales" "o__Synechococcales"
## [9] "o__Sphingobacteriales" "o__Vibrionales"
## [11] "o__Pseudomonadales" "o__Chromatiales"
## [13] "o__Micrococcales" "o__Mycoplasmatales"
## [15] "o__Kitasatosporales" "o__Rhodobacterales"
## [17] "o__Eubacteriales" "o__Desulfuromonadales"
## [19] "o__Syntrophobacterales" "o__Nitrosomonadales"
## [21] "o__Caulobacterales" "o__Actinomycetales"
## [23] "o__Opitutales" "o__Tissierellales"
## [25] "o__Nostocales" "o__Veillonellales"
## [27] "o__Neisseriales" "o__Myxococcales"
## [29] "o__Enterobacterales" "o__Propionibacteriales"
## [31] "o__Methylococcales" "o__Alteromonadales"
## [33] "o__Natrananaerobiales" "o__Bifidobacteriales"
## [35] "o__Sphingomonadales" "o__Pasteurellales"
## [37] "o__Xanthomonadales" "o__Micromonosporales"
## [39] "o__Cellvibrionales" "o__Rhodospirillales"
## [41] "o__Mycobacteriales" "o__Pseudonocardiales"
## [43] "o__Acidaminococcales" "o__Bacteroidales"
## [45] "o__Chitinophagales" "o__Oceanospirillales"
## [47] "o__Deinococcales" "o__Campylobacterales"
## [49] "o__Pseudanabaenales" "o__Aeromonadales"
## [51] "o__Desulfovobacterales" "o__Bryobacterales"
## [53] "o__Hyphomonadales" "o__Streptosporangiales"
## [55] "o__Spirochaetales" "o__Thermosediminibacterales"
## [57] "o__Solirubrobacterales" "o__Chlorobiales"
## [59] "o__Phycisphaerales" "o__"
## [61] "o__Legionellales" "o__Acidiferrobacterales"
## [63] "o__Terriglobales" "o__Polyangiales"
## [65] "o__Verrucomicrobiales" "o__Tepidiformales"
## [67] "o__Rubrobacterales" "o__Candidatus Nanosynbacterales"
## [69] "o__Candidatus Brocadiales" "o__Frankiales"
## [71] "o__Rickettsiales" "o__Candidatus Saccharimonadales"
## [73] "o__Moraxellales" "o__Eggerthellales"
## [75] "o__Entomoplasmatales" "o__Selenomonadales"
## [77] "o__Desulfovibrionales" "o__Pirellulales"
## [79] "o__Mycoplasmodiales" "o__Fusobacteriales"
## [81] "o__Geobacterales" "o__Rhodocyclales"

```

```

## [83] "o__Aquificales"
## [85] "o__Methylacidiphilales"
## [87] "o__Deferribacterales"
## [89] "o__Candidatus Nanopelagicales"
## [91] "o__Acidimicrobiales"
## [93] "o__Leptospirales"
## [95] "o__Thiotrichales"
## [97] "o__Nautiales"
## [99] "o__Saprospirales"
## [101] "o__Desulfobulbales"
## [103] "o__Thermales"
## [105] "o__Miltoncostaeales"
## [107] "o__Ktedonobacterales"
## [109] "o__Oscillatoriiales"
## [111] "o__Thermotogales"
## [113] "o__Kineosporiales"
## [115] "o__Anaerolineales"
## [117] "o__Chloroflexales"
## [119] "o__Nakamurellales"
## [121] "o__Phototrophicales"
## [123] "o__Emcibacterales"
## [125] "o__Gemmimonadales"
## [127] "o__Egibacterales"
## [129] "o__Brachyspirales"
## [131] "o__Gemmatales"
## [133] "o__Halanaerobiales"
## [135] "o__Desulfarculales"
## [137] "o__Orbales"
## [139] "o__Maricaulales"
## [141] "o__Koleobacterales"
## [143] "o__Marinilabiliales"
## [145] "o__Jiangellales"
## [147] "o__Bdellovibrionales"
## [149] "o__Ferrovales"
## [151] "o__Thermomicrobiales"
## [153] "o__Sneathiellales"
## [155] "o__Endomicrobiales"
## [157] "o__Lentisphaerales"
## [159] "o__Kangiellales"
## [161] "o__Candidatus Sulfidibacteriales"
## [163] "o__Tichowtungiales"
## [165] "o__Caldisericiales"
## [167] "o__Immundisolibacterales"
## [169] "o__Vicinamibacterales"
## [171] "o__Syntrophales"
## [173] "o__Desulfurellales"
## [175] "o__Caldilineales"
## [177] "o__Salinisphaerales"
## [179] "o__Bacteriovoracales"
## [181] "o__Chrysioigenales"
## [183] "o__Thermotomaculales"
## [185] "o__Kiritimatiellales"
## [187] "o__Catenulisporales"
## [189] "o__Gloeomargaritales"
"o__Chlamydiales"
"o__Planctomycetales"
"o__Erysipelotrichales"
"o__Parachlamydiales"
"o__Acidithiobacillales"
"o__Coriobacteriales"
"o__Thermoanaerobacterales"
"o__Chroococcales"
"o__Cardiobacteriales"
"o__Nevskiales"
"o__Kordimonadales"
"o__Synergistales"
"o__Petrotogaless"
"o__Puniceicoccales"
"o__Sedimentisphaerales"
"o__Achopleplasmatales"
"o__Candidatus Babeliales"
"o__Pleurocapsales"
"o__Nitrospirales"
"o__Thermodesulfobacteriales"
"o__Nitrospinales"
"o__Bradymonadales"
"o__Jatrophihabitantales"
"o__Sphaerobacterales"
"o__Moorellales"
"o__Gloeobacterales"
"o__Isosphaerales"
"o__Dictyoglomales"
"o__Kosmotogales"
"o__Geodermatophilales"
"o__Thermostichales"
"o__Rhodothermales"
"o__Acanthopleuribacterales"
"o__Glycomycetales"
"o__Silvanigrellales"
"o__Euzebyales"
"o__Tepidisphaerales"
"o__Thermoleophilales"
"o__Parvularculales"
"o__Desulfobaccales"
"o__Sporichthyales"
"o__Dehalococcoidales"
"o__Desulfurobacteriales"
"o__Mariprofundales"
"o__Acidothermales"
"o__Thermodesulfovibrionales"
"o__Trueperales"
"o__Haliangiales"
"o__Candidatus Pelagibacterales"
"o__Hydrogenophilales"
"o__Desulfomonilales"
"o__Elusimicrobiales"
"o__Actinopolysporales"
"o__Nannocystales"

```

```

## [191] "o__Candidatus Izemoplasmatales"      "o__Calditrichales"
## [193] "o__Chthonomonadales"                   "o__Limnochordales"
## [195] "o__Limisphaerales"                     "o__Egicoccales"
## [197] "o__Fimbriimonadales"                  "o__Coprothermobacteriales"
## [199] "o__Fibrobacterales"                   "o__Chroococcidiopsidales"
## [201] "o__Balneolales"                      "o__Mesoaciditogales"
## [203] "o__Magnetococcales"

# Check the amount of unique Species in samples which have and have not been treated with antibiotics (will
#subsetMG %>% ps_filter(AB == "no") %>% get_taxa_unique("Species") # 4464 different orders for non AB treated
#subsetMG %>% ps_filter(AB == "yes") %>% get_taxa_unique("Species") # 2347 different orders for AB treated
#subsetMG %>% get_taxa_unique("Species") # 4706 different order in total, so 242 species are not found

# Check the amount of unique taxa in samples which have and have not been treated with antibiotics (will
#subsetMG %>% ps_filter(AB == "no") # 6014 different taxa for non AB treated
#subsetMG %>% ps_filter(AB == "yes") # 3148 different taxa for AB treated
# 6355 different taxa in total, so 341 taxa are not found in non AB

# Stable "Farm2R1S1" has the three lowest sampling depths of the dataset, the other nine samples are fine
subsetMG %>% ps_filter(FarmRoundStable == c("Farm2R1S1")) %>% sample_sums() %>% sort()

##      5_54     4_56     4_41     4_38     4_40     4_36     4_57     4_54     4_39     5_55
## 2164863 2171475 2197133 2384366 2692826 2780965 2796239 2906132 3078035 3401482
##      4_55     4_37
## 4528417 5182762

# factorizing variables as not to create problems with visualization later down the line
sample_data(subsetMG)$Cluster = as.factor(sample_data(subsetMG)$Cluster)
sample_data(subsetMG)$FlockSize = as.factor(sample_data(subsetMG)$FlockSize)
sample_data(subsetMG)$AgeParentStock = as.factor(sample_data(subsetMG)$AgeParentStock)
sample_data(subsetMG)$Age = as.factor(sample_data(subsetMG)$Age)
sample_data(subsetMG)$LibraryNumber = as.factor(sample_data(subsetMG)$LibraryNumber)

# add stable column with shorter names
sample_data(subsetMG)$FarmRoundStable = as.factor(sample_data(subsetMG)$FarmRoundStable)
subsetMG@sam_data$Stables = revalue(sample_data(subsetMG)$FarmRoundStable, c("Farm1R1S1"="Stable1", "Farm2R1S1"="Stable2", "Farm2R2S1"="Stable5", "Farm3R1S1"="Stable6", "Farm3R2S1"="Stable7", "Farm4R1S1"="Stable9", "Farm4R2S1"="Stable8"))

# Shortening agent names
subsetMG@sam_data$Cox[subsetMG@sam_data$Cox == "narasinandnicarbazin(maxiban)"] = "Maxiban"
subsetMG@sam_data$Cox[subsetMG@sam_data$Cox == "narasin(monteban)"] = "Monteban"
subsetMG@sam_data$Cox[subsetMG@sam_data$Cox == "salinomycin(Sacox120microGranulate)"] = "Sacox"

```

Abundances and heatmaps

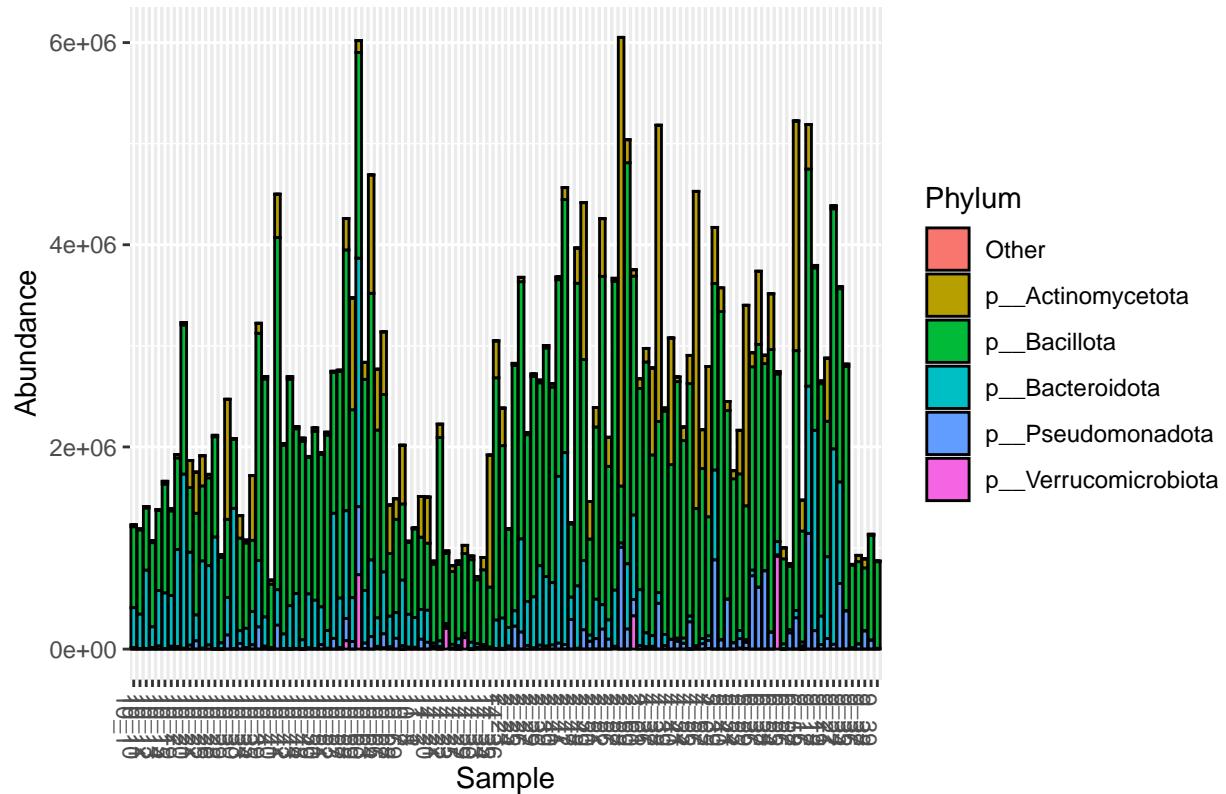
```

# absolute abundances, since there are a lot of phyla (43), we will only include the top 5 phyla

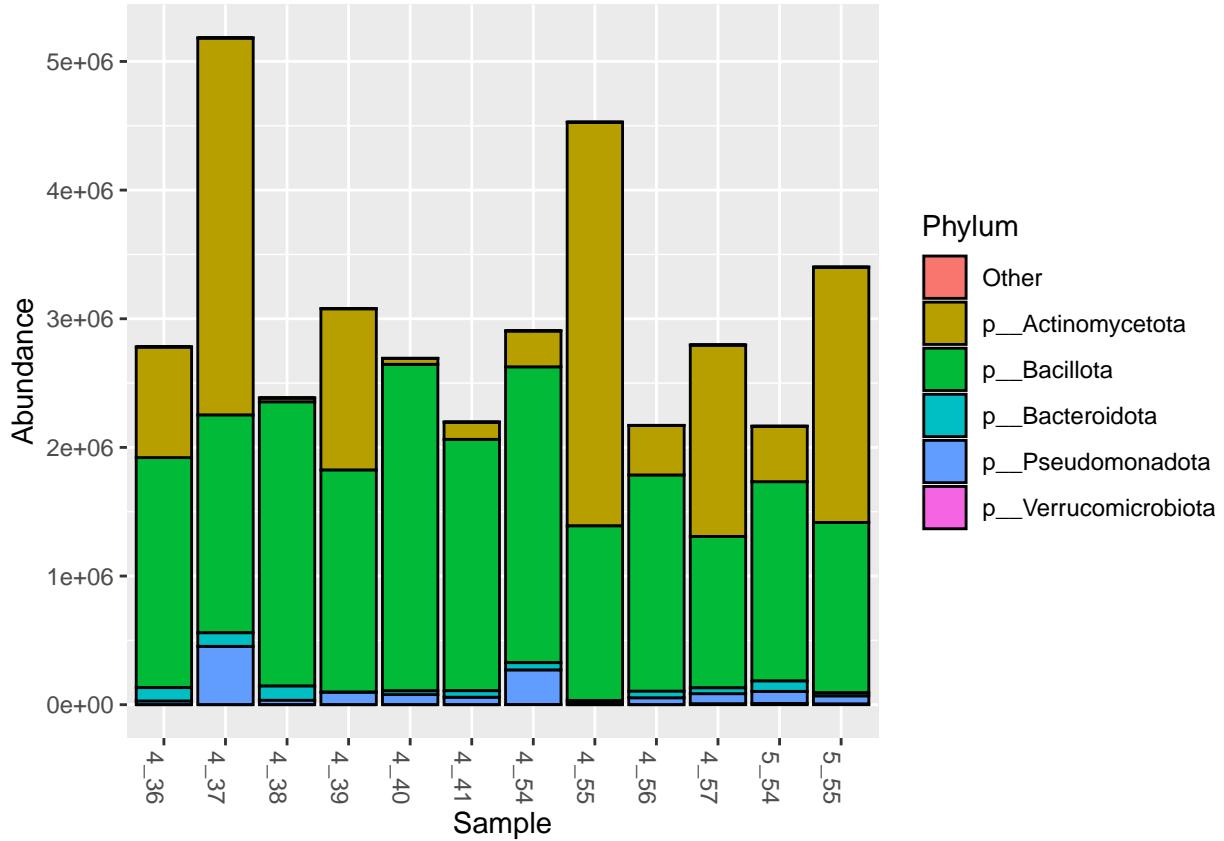
subsetMG %>% aggregate_top_taxa2("Phylum", top = 5) %>% plot_bar(fill="Phylum", title = "Absolute abundance")

```

Absolute abundances per sample

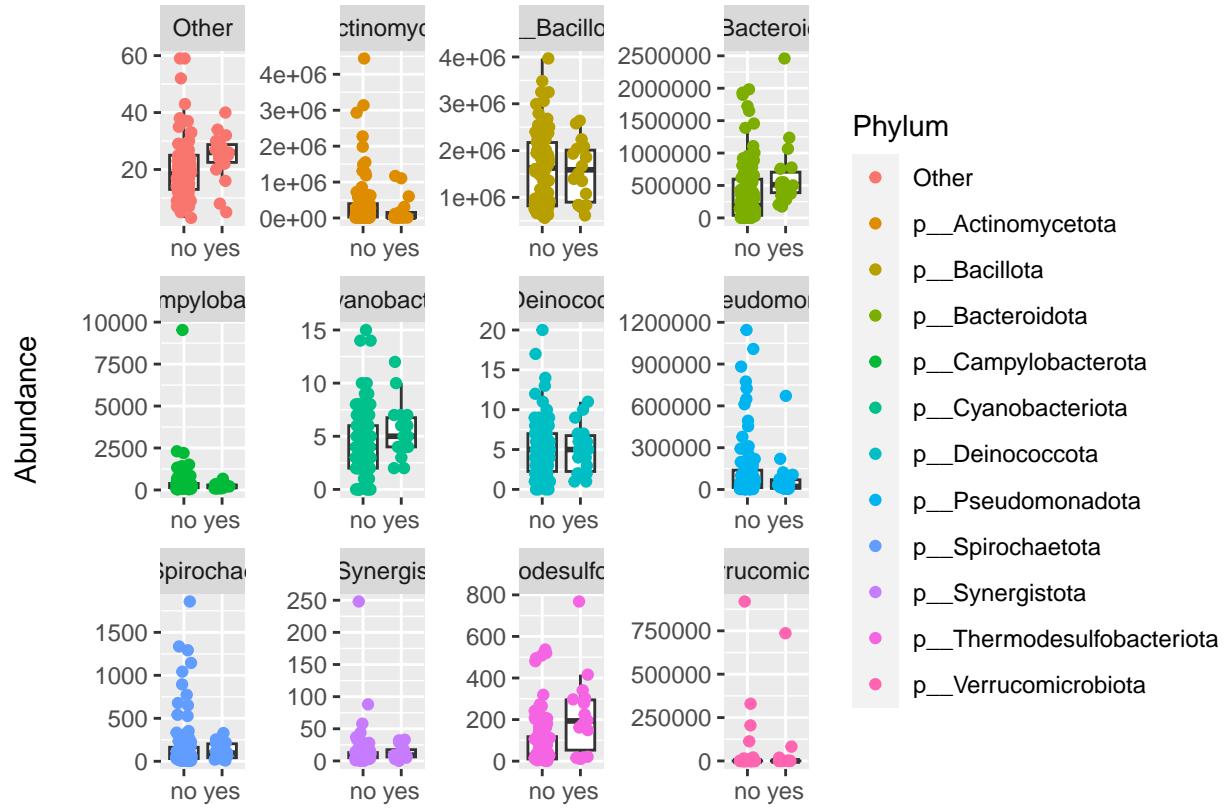


```
# for plotting abundances of specific stables
subsetMG %>% aggregate_top_taxa2("Phylum", top = 5) %>% ps_filter(FarmRoundStable == c("Farm2R1S1")) %>%
```

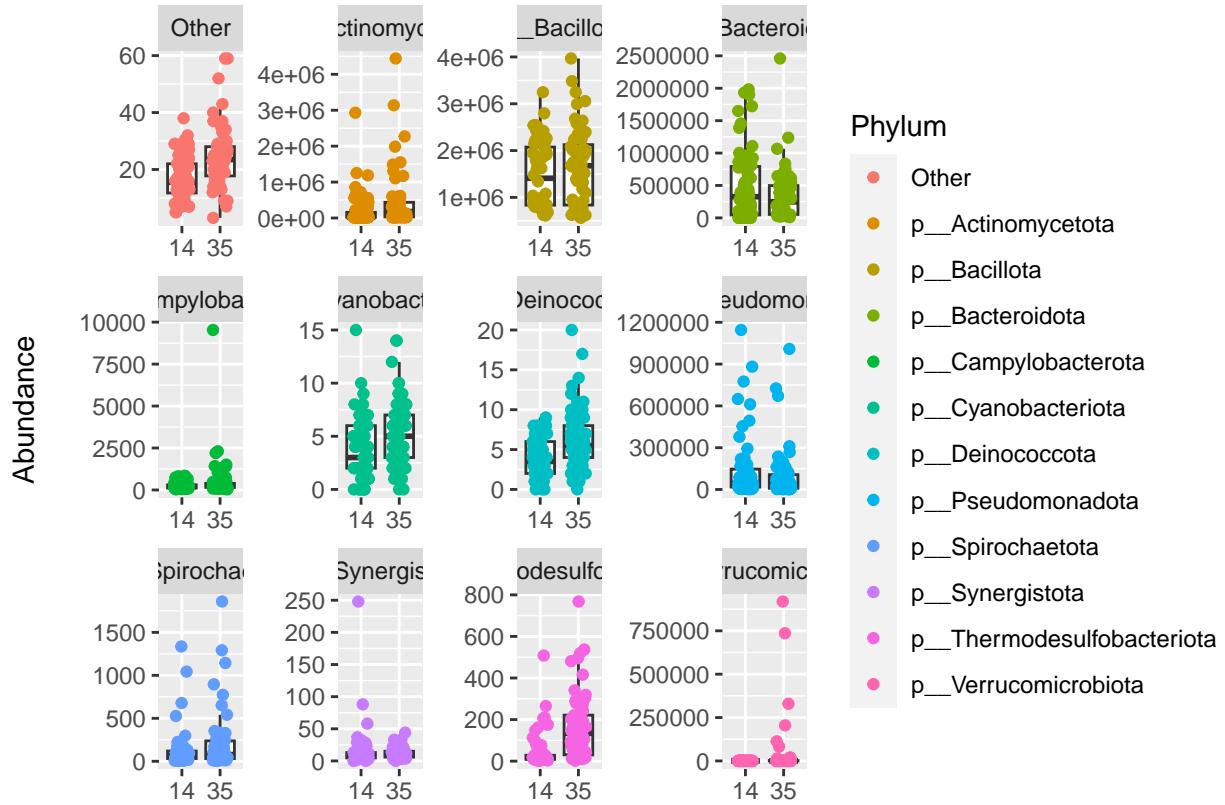


```
# visualisation on AB at Phylum level, more data for samples which have not been treated with AB, but a
ps_prim <- subsetMG %>% aggregate_top_taxa2("Phylum", top = 11) %>% phyloseq:::tax_glom("Phylum")
taxa_names(ps_prim) <- phyloseq:::tax_table(ps_prim)[, "Phylum"]

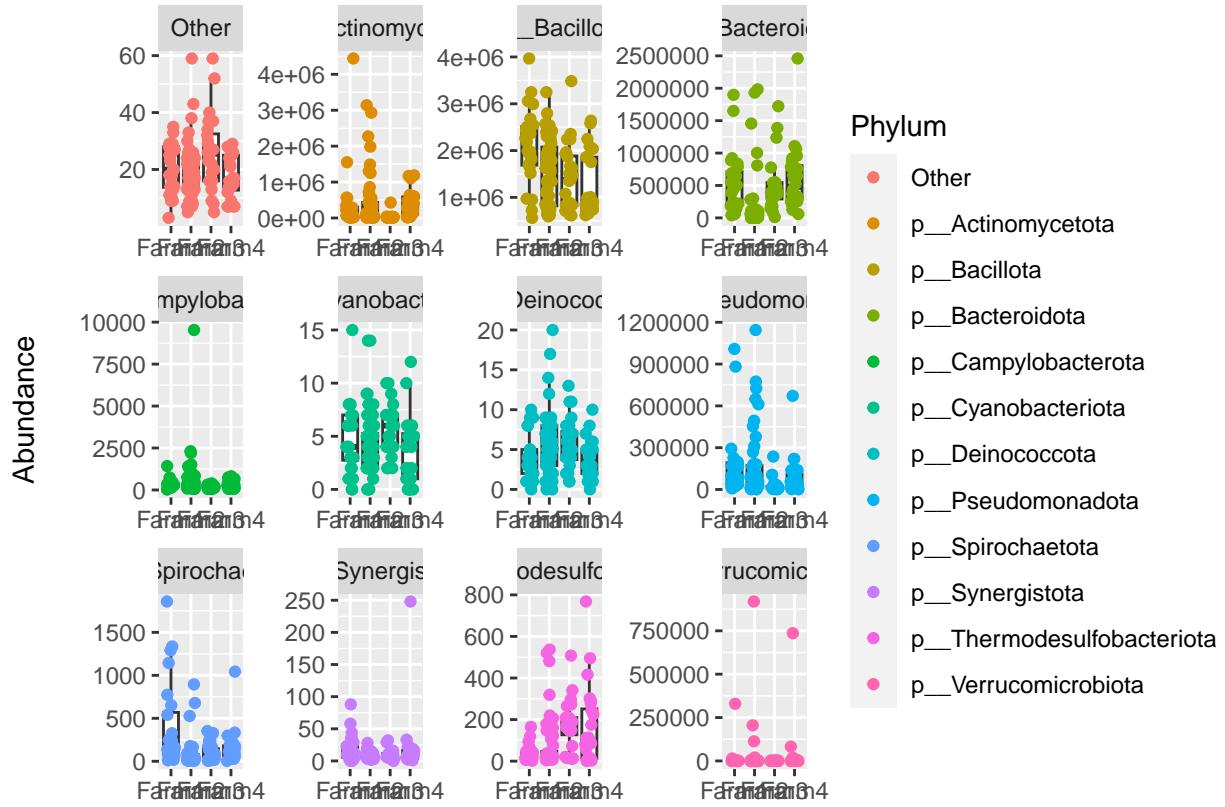
psmelt(ps_prim) %>% #AB
  ggplot(aes(x = AB, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ OTU, scales = "free")
```



```
psmelt(ps_prim) %>% #Age
ggplot(aes(x = Age, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ OTU, scales = "free")
```



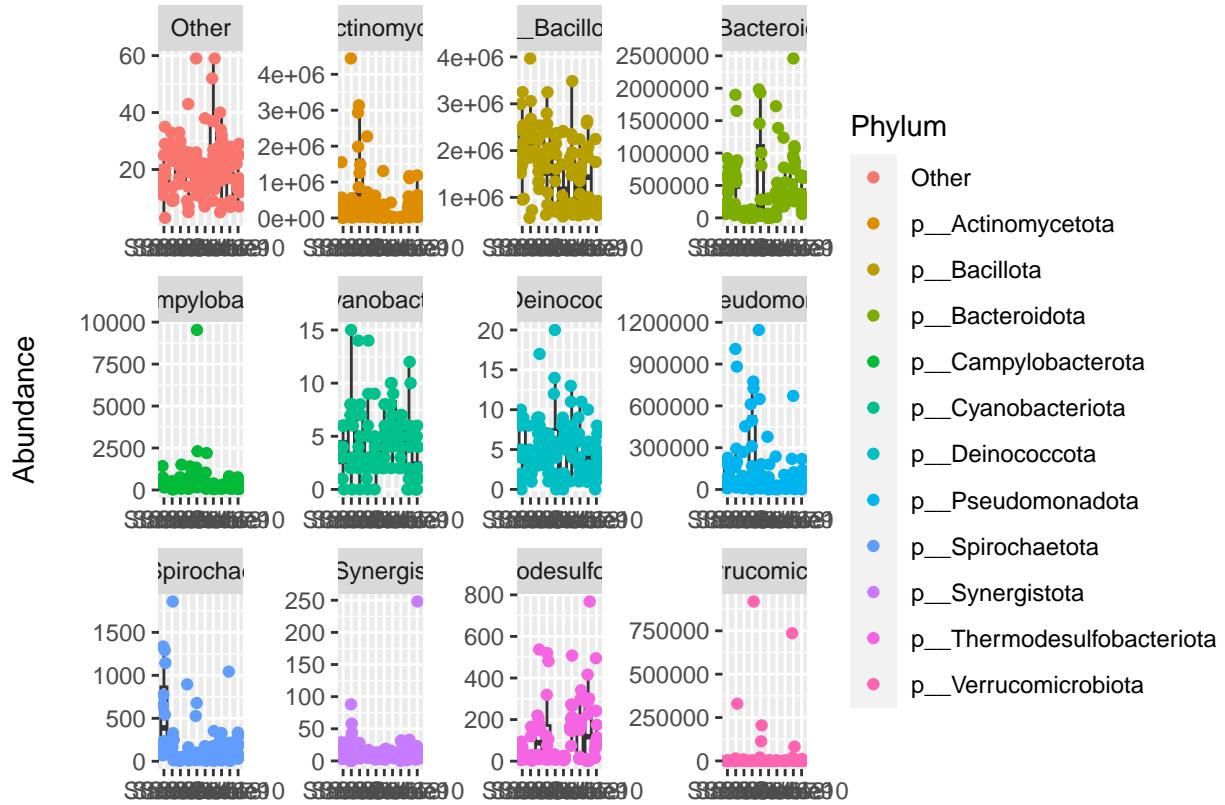
```
psmelt(ps_prim) %>% #Farm
ggplot(aes(x = Farm2, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ OTU, scales = "free")
```



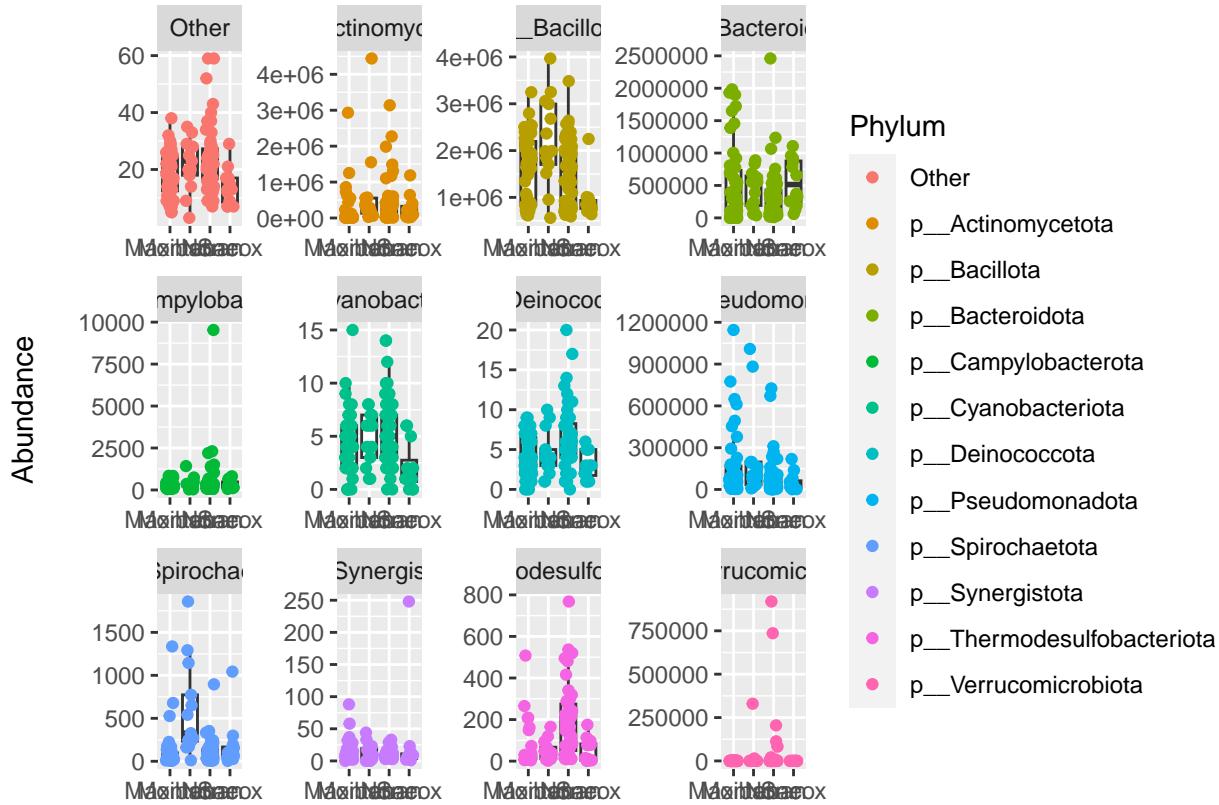
```

psmelt(ps_prim) %>% #Stable
ggplot(aes(x = Stables, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ OTU, scales = "free")

```



```
psmelt(ps_prim) %>% #Agent
ggplot(aes(x = Cox, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ OTU, scales = "free")
```



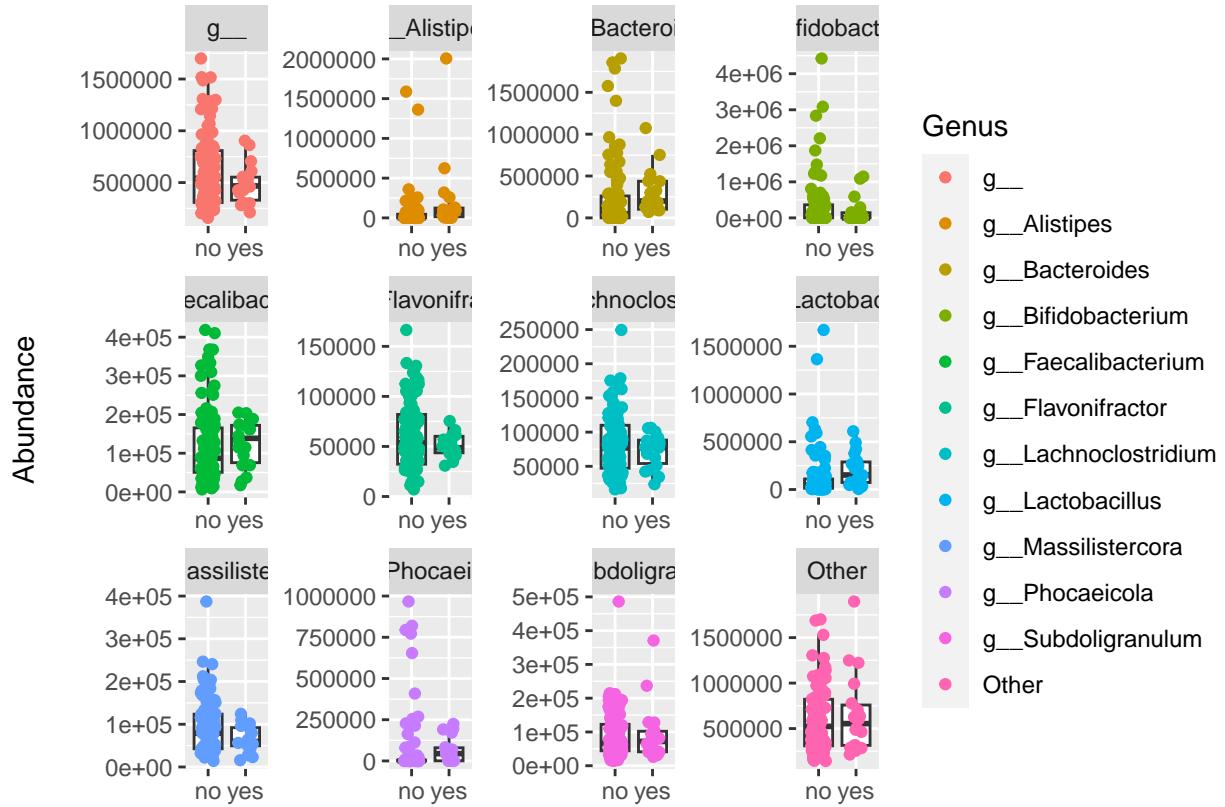
visualisation on AB at Genus level, more data for samples which have not been treated with AB, but also many more samples in this group (18 vs 102)

```

ps_prim <- subsetMG %>% aggregate_top_taxa2("Genus", top = 13) %>% phyloseq:::tax_glm("Genus")
# some top hits will not work properly, there are in actuality only 11 genera being selected above
# this is because of problems within the taxonomy info, empty taxonomies etc will be found, with tax_fi:
# the problem is that when replacing these taxonomies with their higher ranks you are not looking at th:
# therefore we opt to not include these taxonomies and rather skip these unknown taxonomies
taxa_names(ps_prim) <- phyloseq:::tax_table(ps_prim)[, "Genus"]

psmelt(ps_prim) %>% # AB
  ggplot(aes(x = AB, y = Abundance)) +
    geom_boxplot(outlier.shape = NA) +
    geom_jitter(aes(color = Genus), height = 0, width = .2) +
    labs(x = "", y = "Abundance\n") +
    facet_wrap(~ OTU, scales = "free")

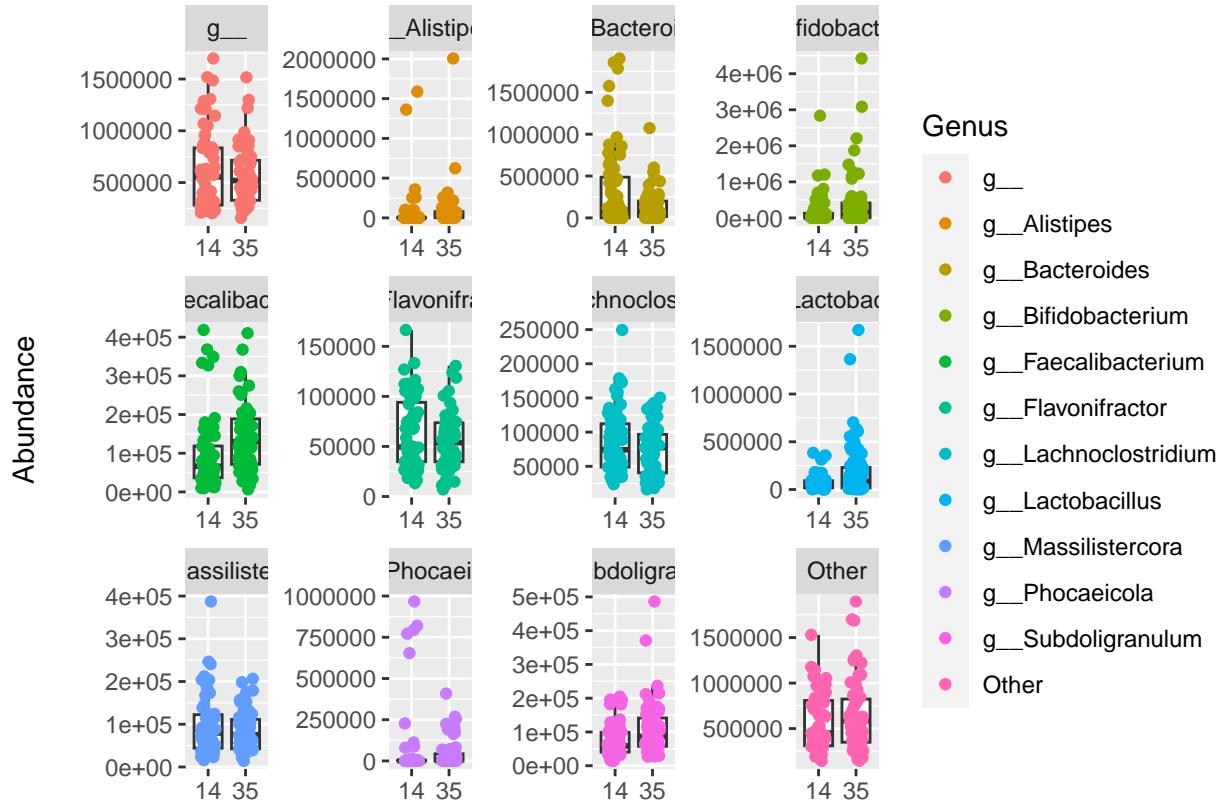
```



```

psmelt(ps_prim) %>% #Age
ggplot(aes(x = Age, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Genus), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ OTU, scales = "free")

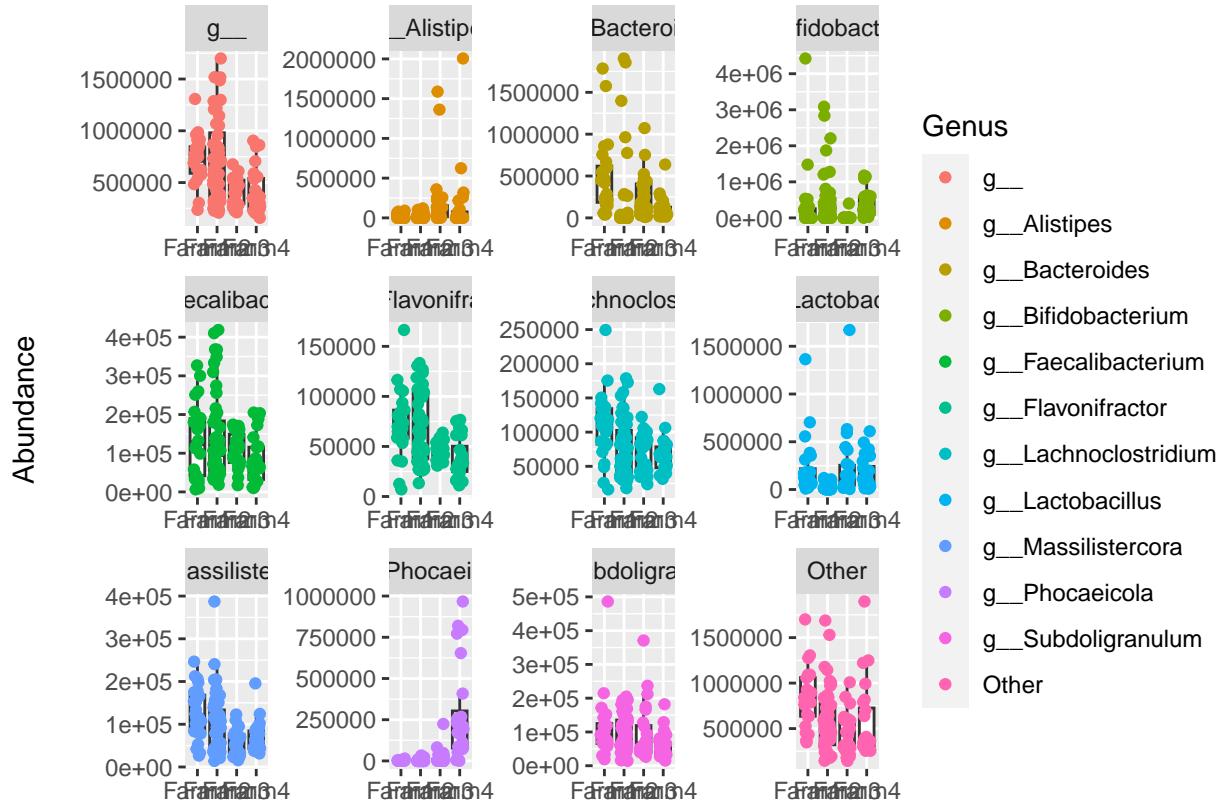
```



```

psmelt(ps_prim) %>% #Farm
ggplot(aes(x = Farm2, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Genus), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ OTU, scales = "free")

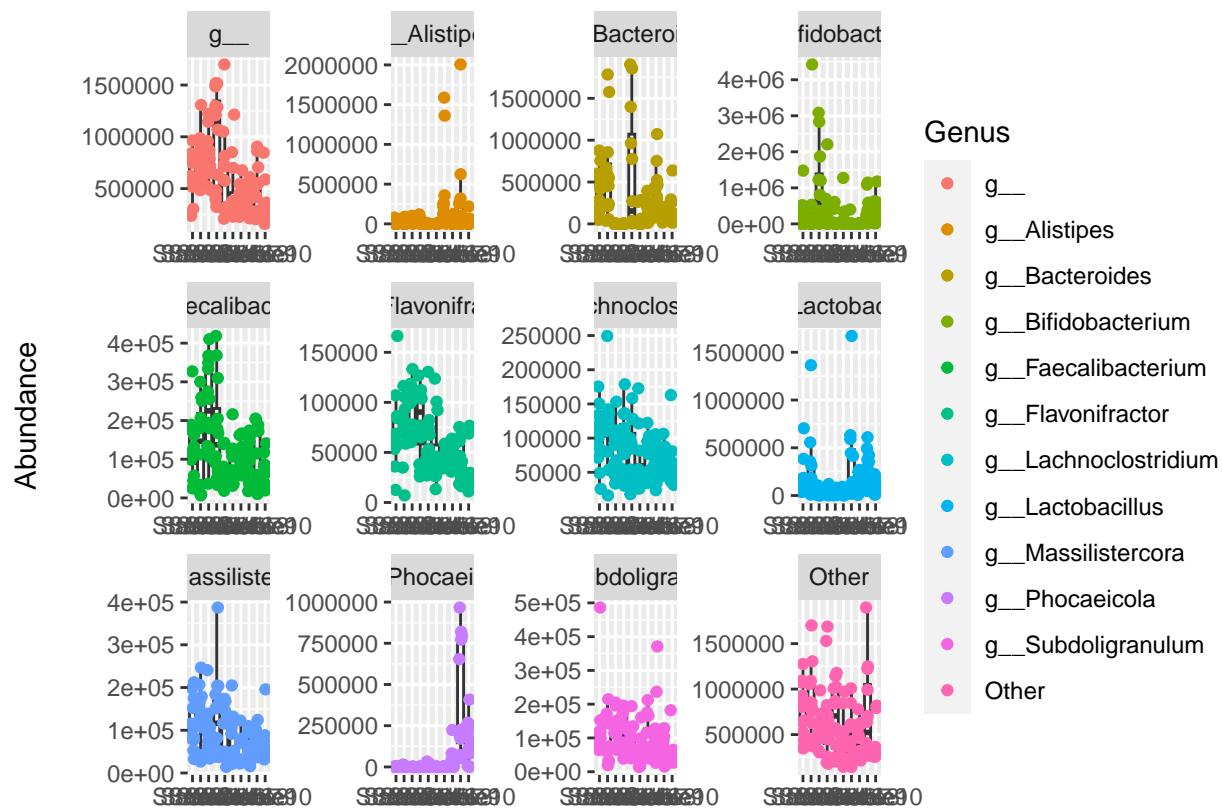
```



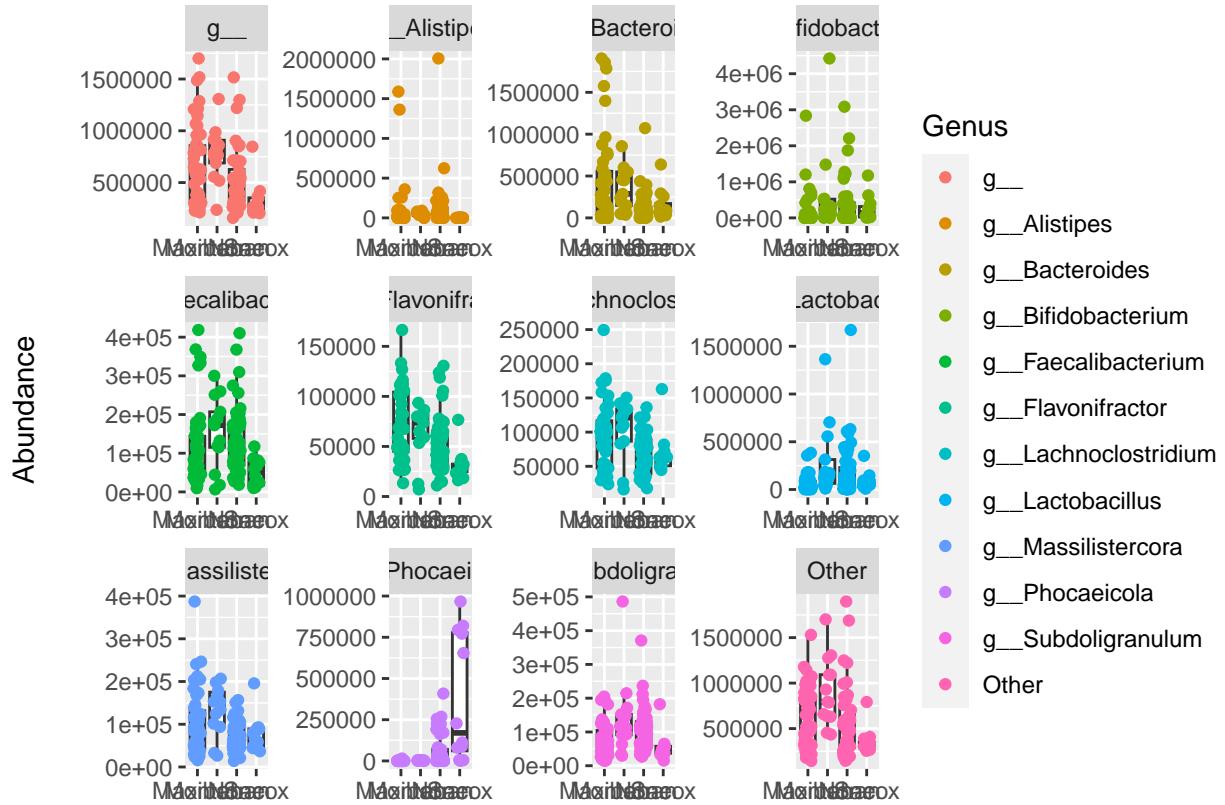
```

psmelt(ps_prim) %>% #Stable
ggplot(aes(x = Stables, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Genus), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ OTU, scales = "free")

```



```
psmelt(ps_prim) %>% #Agent
ggplot(aes(x = Cox, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Genus), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ OTU, scales = "free")
```

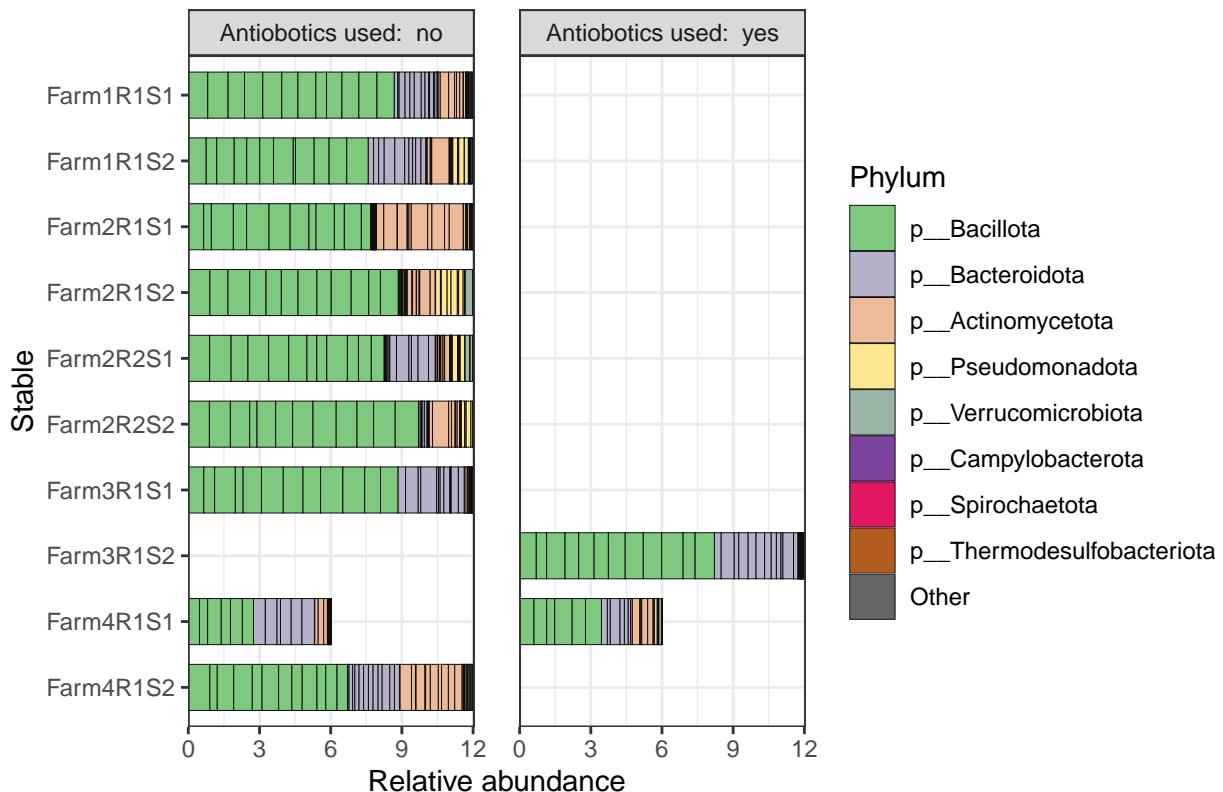


```
# Plots of relative abundances, fixing some genes that are clustered in the data twice, showing top 12
```

```
# Relative abundance for both stable and antibiotics used
```

```
subsetMG %>% tax_fix() %>%
  ps_arrange(FarmRoundStable) %>%
  ps_mutate(
    FarmRoundStable = factor(FarmRoundStable, rev(unique(FarmRoundStable)))
  ) %>%
  comp_barplot(
    tax_level = "Phylum", bar_width = 0.7, sample_order = "asis",
    palette = colorRampPalette(brewer.pal(8, "Accent"))(9),
    x = "FarmRoundStable") +
  facet_wrap(
    facets = vars(AB), labeller = as_labeller(~ paste("Antibiotics used: ", .)),
    scales = "fixed"
  ) +
  coord_flip() +
  labs(x = "Stable", y = "Relative abundance") +
  scale_y_continuous(expand = expansion(add = c(0, 0.05))) +
  theme_bw() +
  theme(panel.spacing.x = unit(6, "mm")) +
  ggtitle("Relative abundance of Phyla by stable and antibiotics used")
```

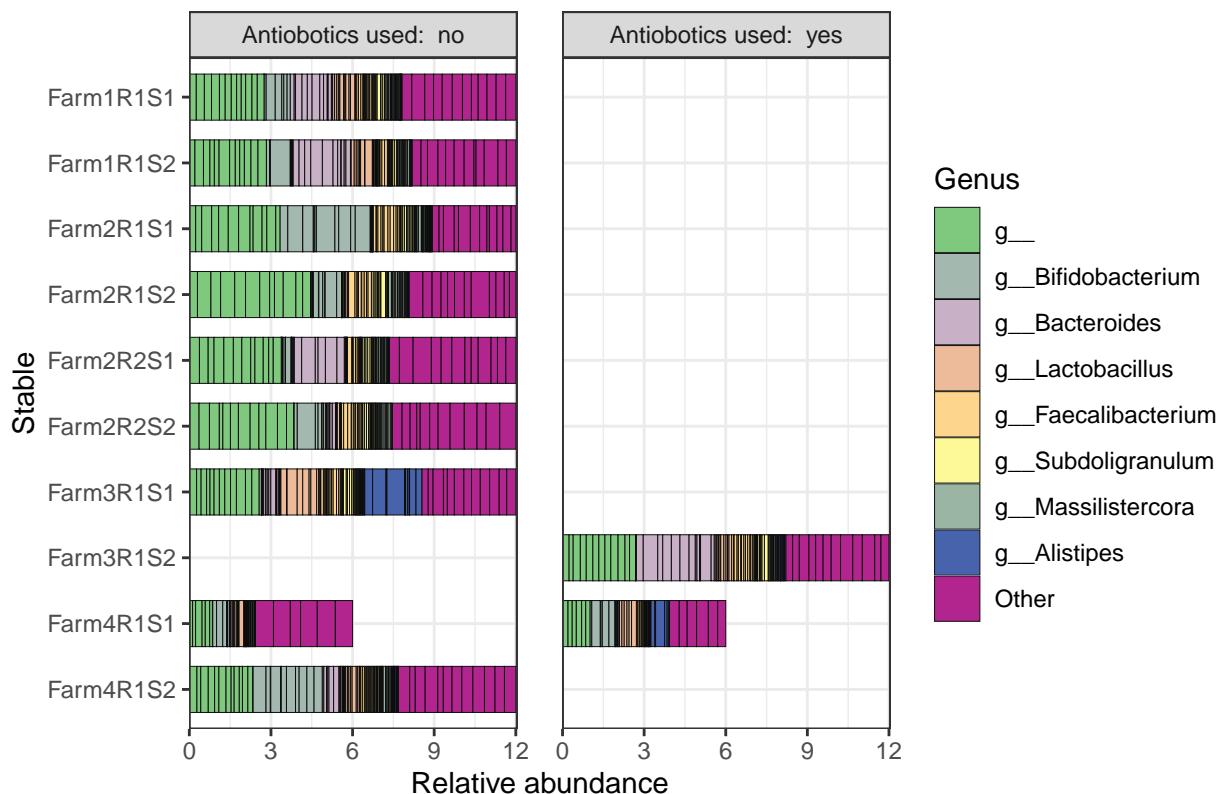
Relative abundance of Phyla by stable and antibiotics used



Same plot but with Genus

```
subsetMG %>% aggregate_top_taxa2("Genus", top = 10) %>% phyloseq::tax_glom("Genus") %>%
  ps_arrange(FarmRoundStable) %>%
  ps_mutate(
    FarmRoundStable = factor(FarmRoundStable, rev(unique(FarmRoundStable))))
  ) %>%
  comp_barplot(
    tax_level = "Genus", bar_width = 0.7, sample_order = "asis",
    palette = colorRampPalette(brewer.pal(8,"Accent"))(13),
    x = "FarmRoundStable",
    n_taxa = 12, other_name = "Other ARG", merge_other = F) +
  facet_wrap(
    facets = vars(AB), labeller = as_labeller(~ paste("Antibiotics used: ", .)),
    scales = "fixed"
  ) +
  coord_flip() +
  labs(x = "Stable", y = "Relative abundance") +
  scale_y_continuous(expand = expansion(add = c(0, 0.05))) +
  theme_bw() +
  theme(panel.spacing.x = unit(6, "mm")) +
  ggtitle("Relative abundance of Genera by stable and antibiotics used")
```

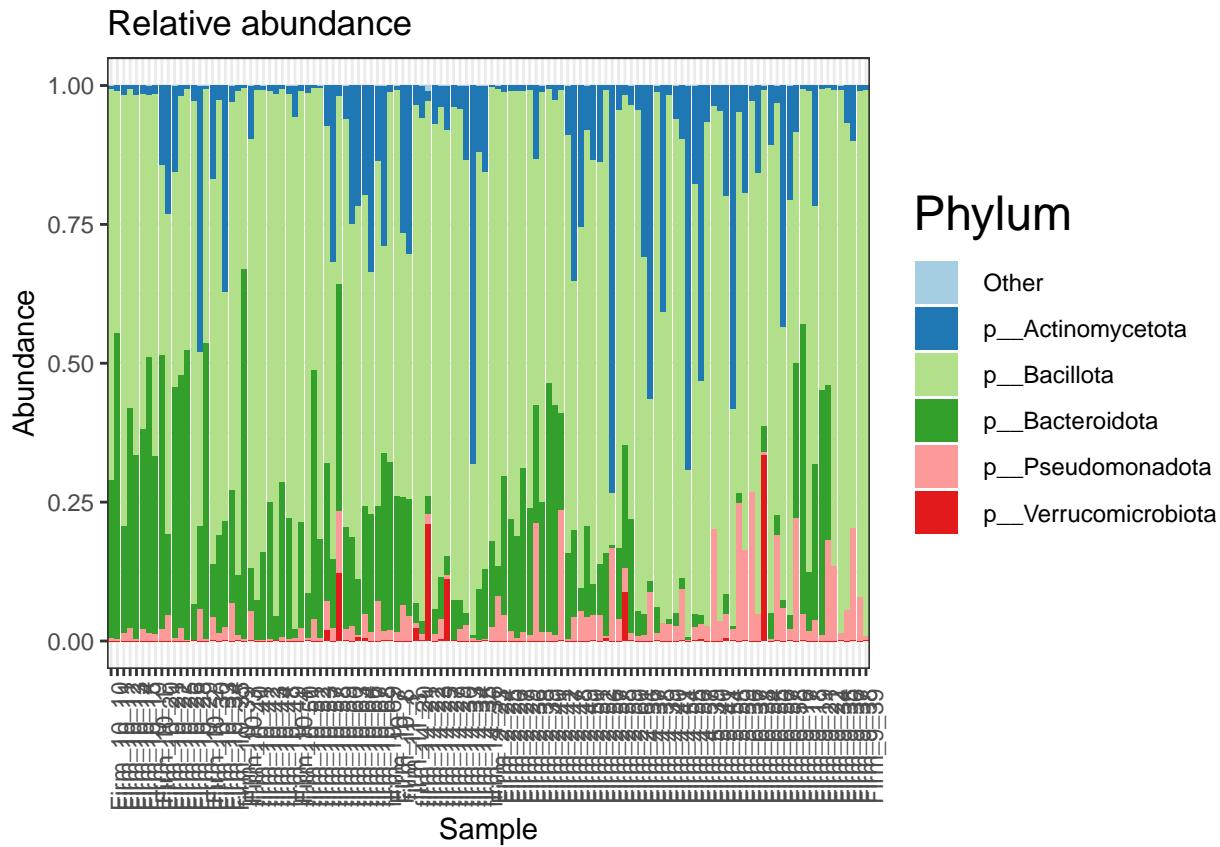
Relative abundance of Genera by stable and antibiotics used



```
# relabundance with other category
```

```
subsetMG.rel <- subsetMG %>% aggregate_top_taxa2("Phylum", top = 5) %>% microbiome::transform("compositional")

plot_composition(subsetMG.rel, x.label = "Id") + theme(legend.position = "bottom") +
  scale_fill_brewer("Phylum", palette = "Paired") + theme_bw() + theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Relative abundance") + theme(legend.title = element_text(size = 18))
```



```
# instead we will plot both 16S and MG data in the same figure

# get the samples in the same order
sample_names(subset16S) = sample_names(subsetMG)

dataset1 = ps_filter(subset16S)
dataset2 = ps_filter(subsetMG)

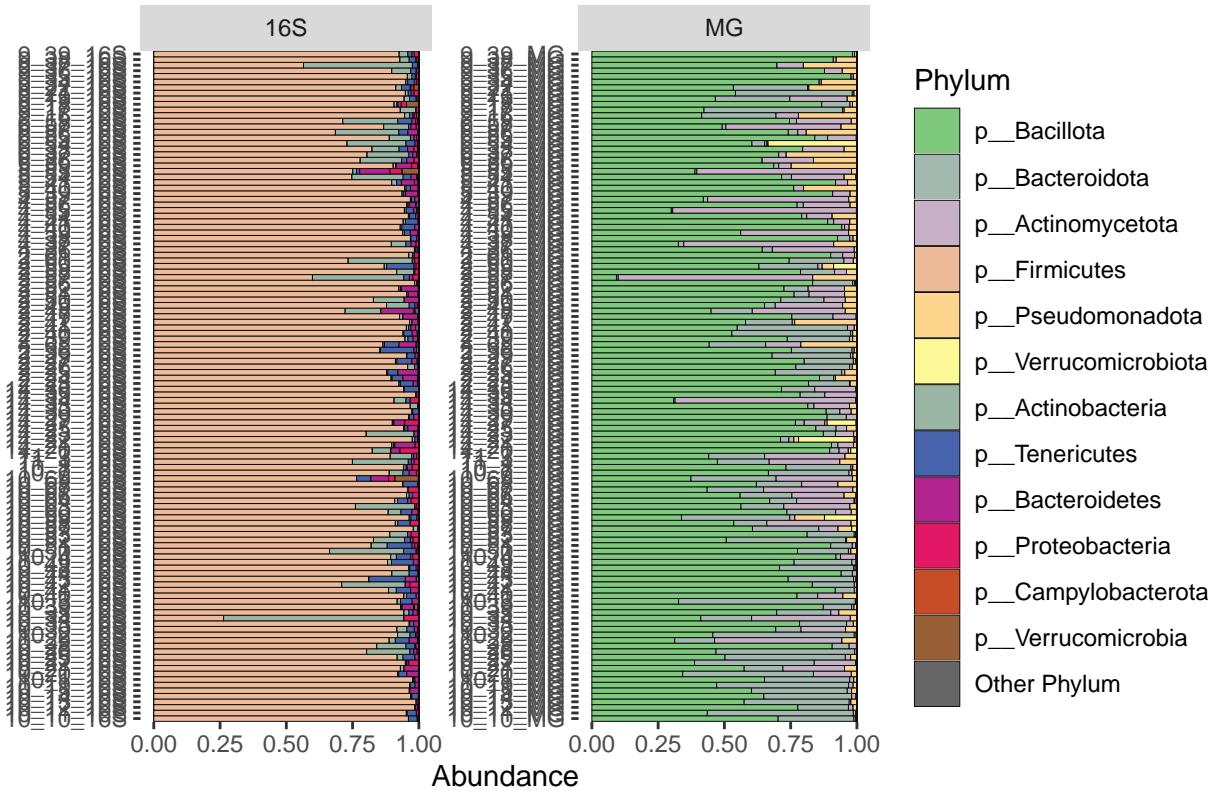
dataset1 %<>% ps_mutate(dataset = "16S")
dataset2 %<>% ps_mutate(dataset = "MG")

sample_names(dataset1) <- paste(sample_names(dataset1), "16S", sep="_")
sample_names(dataset2) <- paste(sample_names(dataset2), "MG", sep="_")

combined <- phyloseq::merge_phyloseq(
  dataset1 %>% tax_agg("Phylum") %>% ps_get(),
  dataset2 %>% tax_agg("Phylum") %>% ps_get()
)

combined %>%
  comp_barplot("Phylum", facet_by = "dataset", n_taxa = 12, palette = colorRampPalette(brewer.pal(8,"Ac"),
    other_name = "Other Phylum", merge_other = F, sample_order = "asis") +
  coord_flip() + ggtitle("Metataxonomic vs metagenomic relative abundances of Phyla")
```

Metataxonomic vs metagenomic relative abundances of Phyla



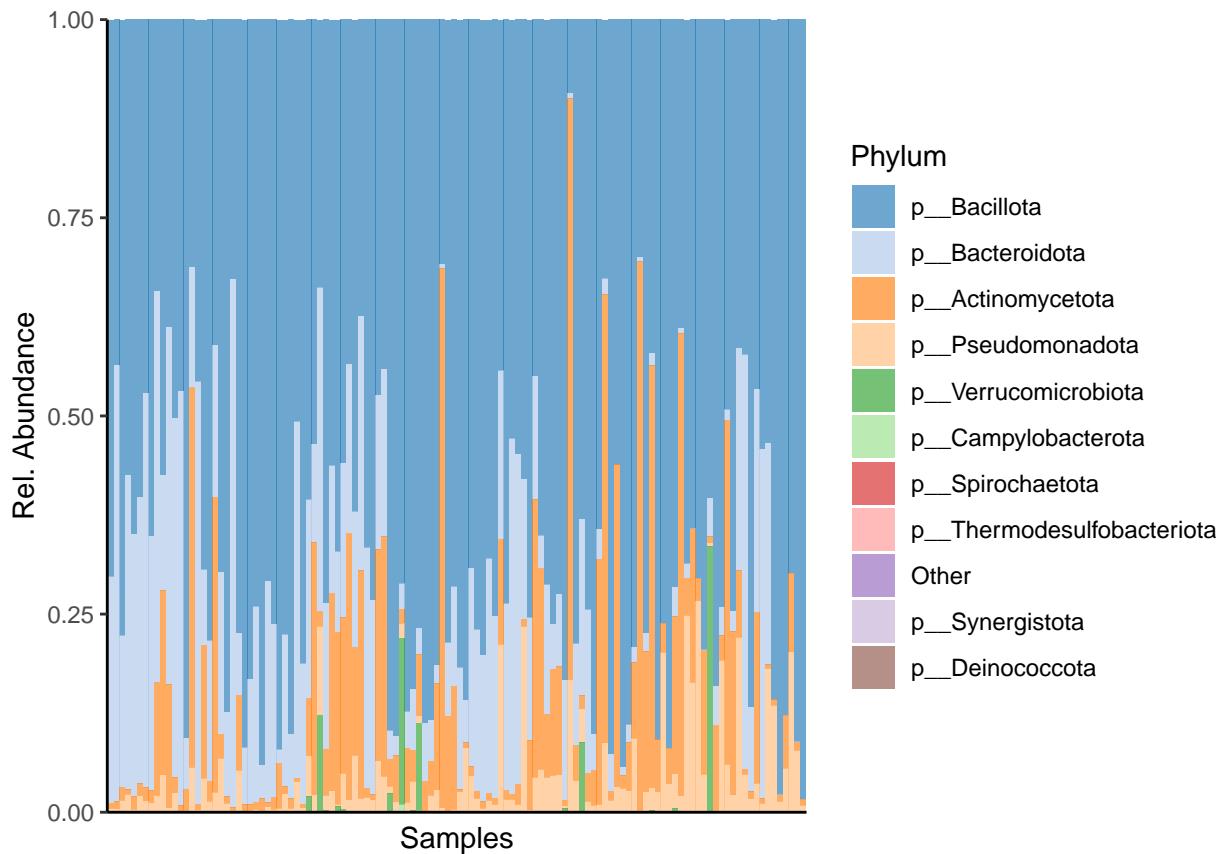
```
### rel abundance on phylum level (old version without other phyla)
```

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)

tse <- transformCounts(tse, method = "relabundance")

tse_phylum <- agglomerateByRank(tse,
                                   rank = "Phylum",
                                   onRankOnly = TRUE)
tse_phylum <- transformCounts(tse_phylum,
                               assay.type = "counts",
                               method = "relabundance")
top_taxa <- getTopTaxa(tse_phylum,
                        top = 10,
                        assay.type = "relabundance")
phylum_renamed <- lapply(rowData(tse_phylum)$Phylum,
                          function(x){if (x %in% top_taxa) {x} else {"Other"}})
rowData(tse_phylum)$Phylum <- as.character(phylum_renamed)

# rel abundance figures, can order by specific taxa
miaViz::plotAbundance(tse_phylum,
                       assay.type = "relabundance",
                       rank = "Phylum",
                       order_rank_by = "abund")
```



```
tse_phylum$Farm2 = as.factor(tse_phylum$Farm2)
tse_phylum$AB = as.factor(tse_phylum$AB)

# Add AB plot on top

plots <- miaViz:::plotAbundance(tse_phylum,
                                    assay.type = "relabundance",
                                    rank = "Phylum",
                                    order_rank_by = "abund",
                                    #                                         order_sample_by = "o_Clostridiales",
                                    order_sample_by = "AB",
                                    features = "AB")

plots[[1]] <- plots[[1]] +
  theme(legend.key.size = unit(0.3, 'cm'),
        legend.text = element_text(size = 6),
        legend.title = element_text(size = 8))
plots[[2]] <- plots[[2]] +
  theme(legend.key.height = unit(0.3, 'cm'),
        legend.key.width = unit(0.3, 'cm'),
        legend.text = element_text(size = 6),
        legend.title = element_text(size = 8),
        legend.direction = "vertical")

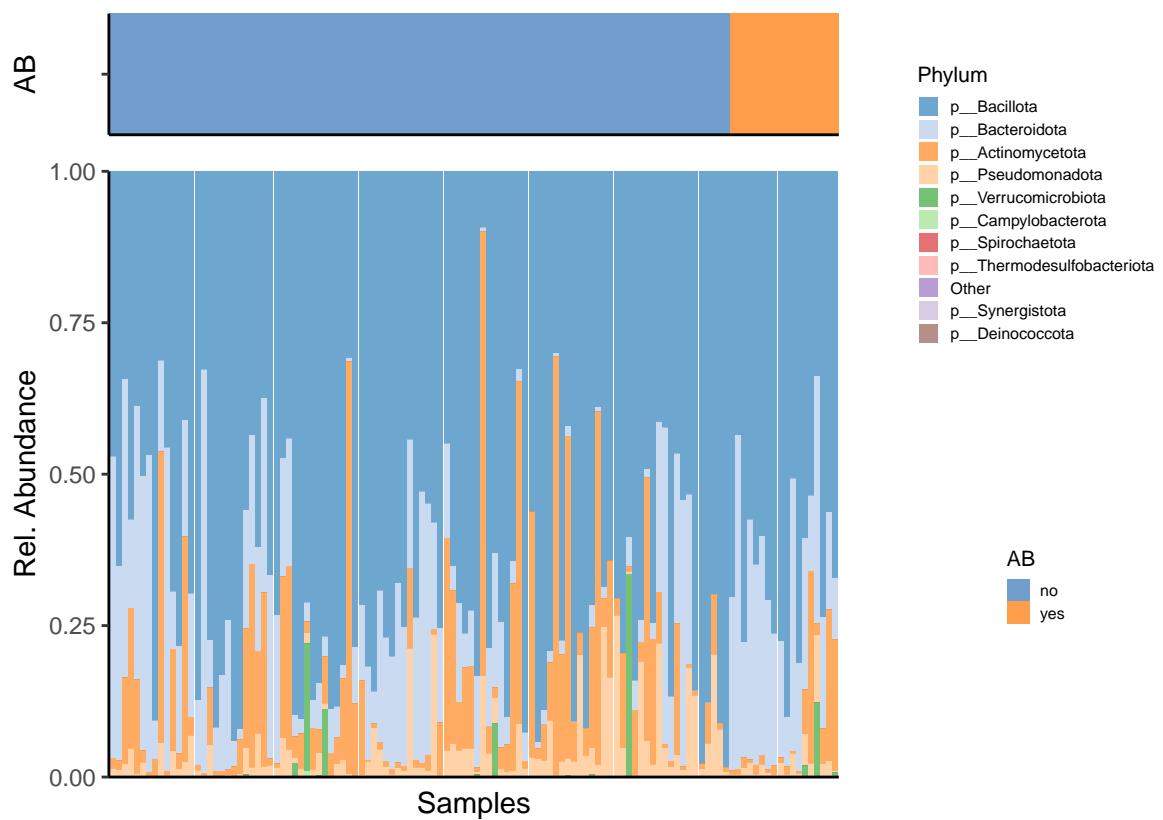
legend <- wrap_plots(as_ggplot(get_legend(plots[[1]])), as_ggplot(get_legend(plots[[2]])), ncol = 1)
plots[[1]] <- plots[[1]] + theme(legend.position = "none")
```

```

plots[[2]] <- plots[[2]] + theme(legend.position = "none", axis.title.x=element_blank())

plot <- wrap_plots(plots[[2]], plots[[1]], ncol = 1, heights = c(2, 10))
wrap_plots(plot, legend, nrow = 1, widths = c(2, 1))

```



heatmaps on phylum level

```

tse = subsetMG %>% aggregate_top_taxa2("Phylum", top = 11) %>% phyloseq::tax_glm("Phylum") %>% makeTree

tse <- transformCounts(tse, method = "relabundance")

tse_phylum <- agglomerateByRank(tse,
                                    rank = "Phylum",
                                    onRankOnly = TRUE)

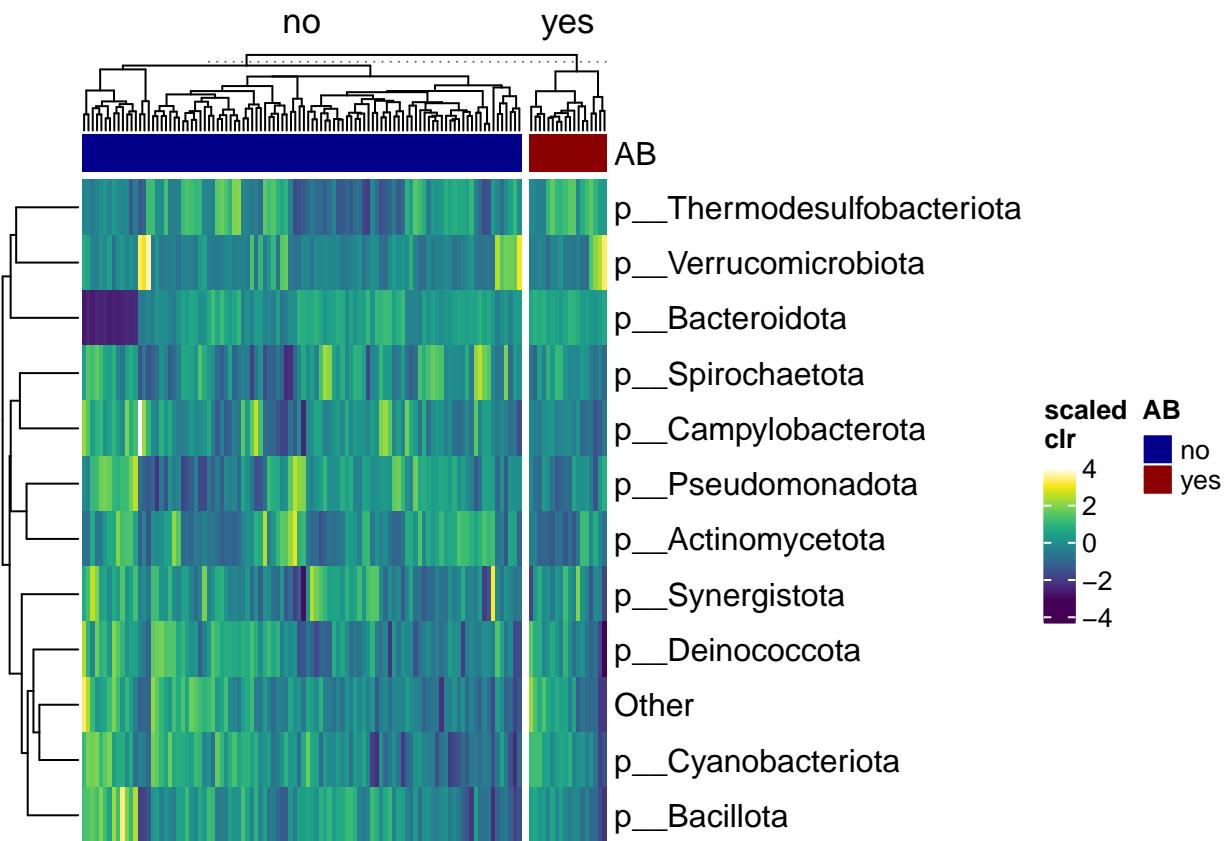
tse_phylum <- transformCounts(tse_phylum, MARGIN = "samples", method = "clr", assay.type = "counts", ps)
tse_phylum <- transformCounts(tse_phylum, assay.type = "clr",
                               MARGIN = "features",
                               method = "z", name = "clr_z")

top_taxa <- getTopTaxa(tse_phylum, top = 10)
#tse_phylum <- tse_phylum[top_taxa, ]

```

```
# Phylum AB heatmap
tse_phylum@metadata$anno_colors$AB = c(yes = "darkred", no = "darkblue")

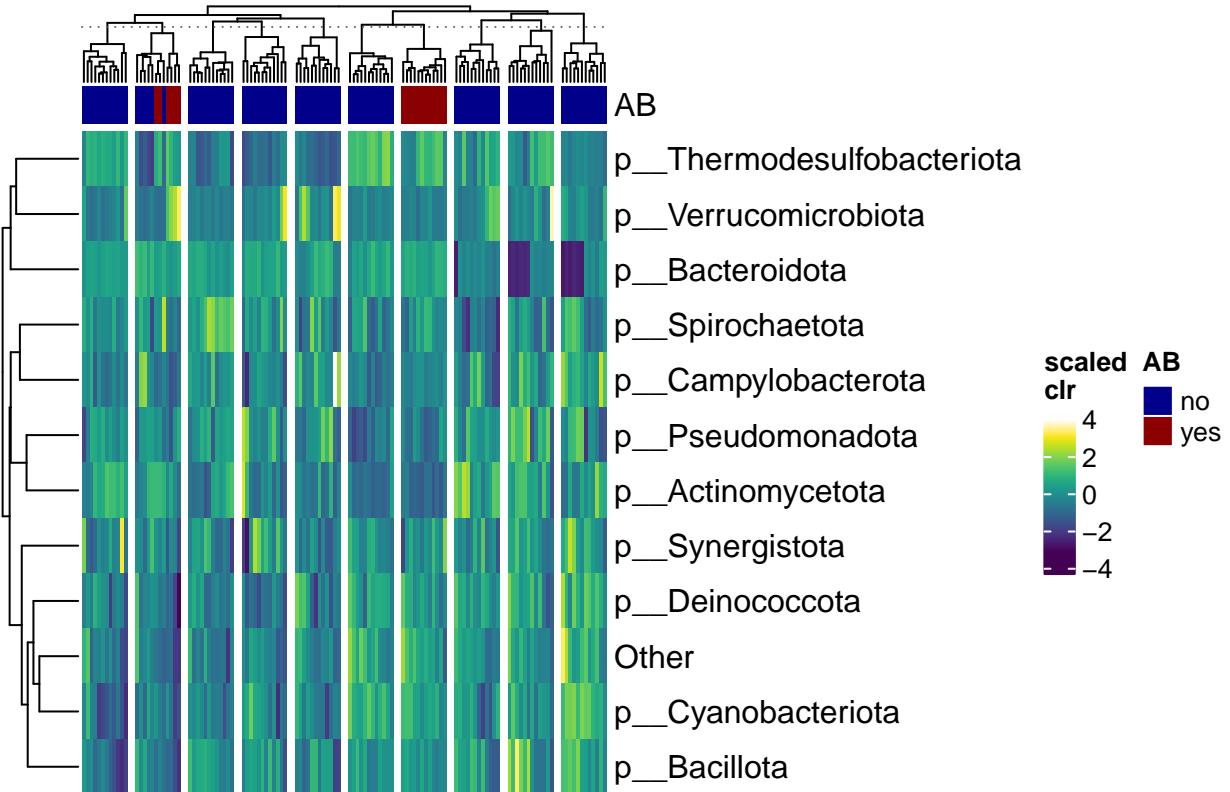
sechm(tse_phylum,
      features = rownames(tse_phylum),
      assayName = "clr",
      do.scale = TRUE,
      top_annotation = "AB",
      gaps_at = "AB",
      hmcols = viridis(256),
      cluster_cols = TRUE, cluster_rows = TRUE)
```



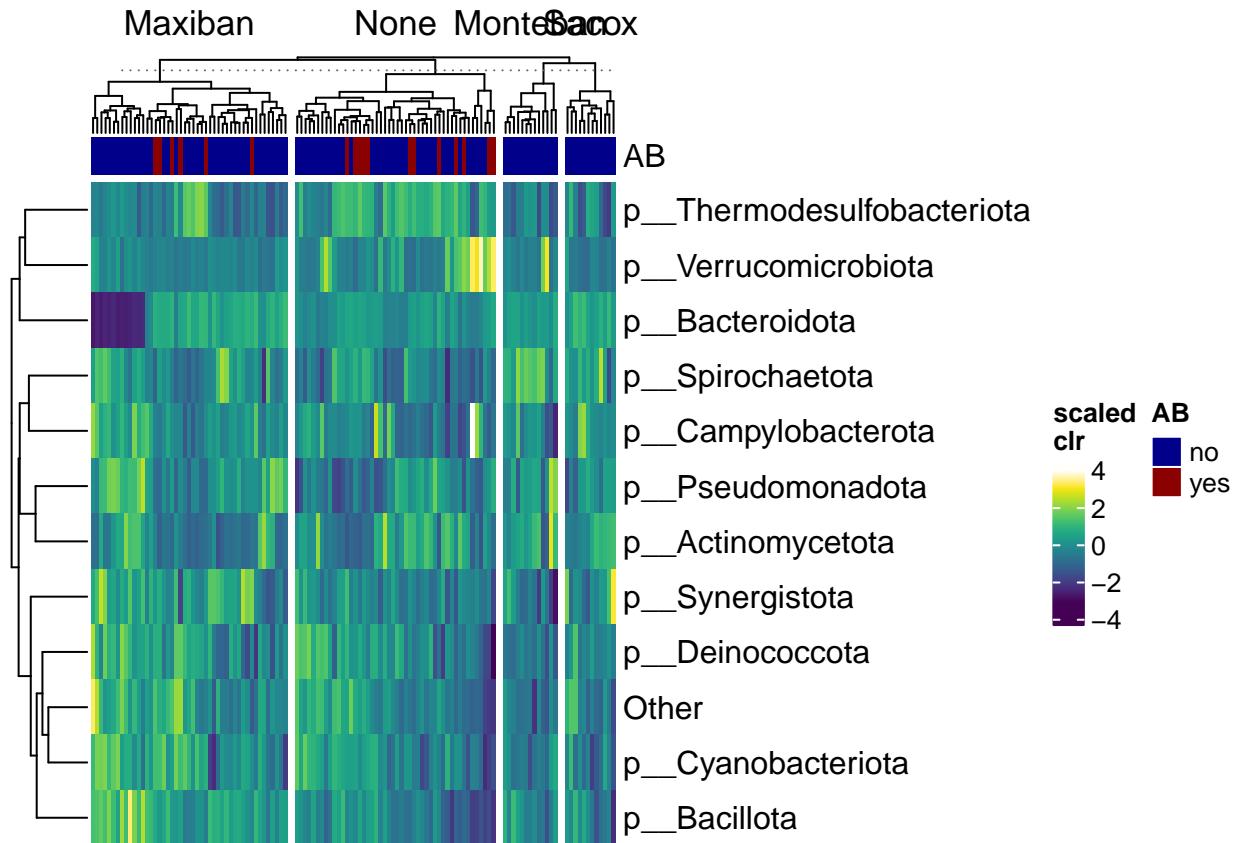
```
# heatmap with AB and stable
tse_phylum@metadata$anno_colors$AB = c(yes = "darkred", no = "darkblue")

sechm(tse_phylum,
      features = rownames(tse_phylum),
      assayName = "clr",
      do.scale = TRUE,
      top_annotation = c("AB"),
      gaps_at = "Stables",
      hmcols = viridis(256),
      cluster_cols = TRUE, cluster_rows = TRUE)
```

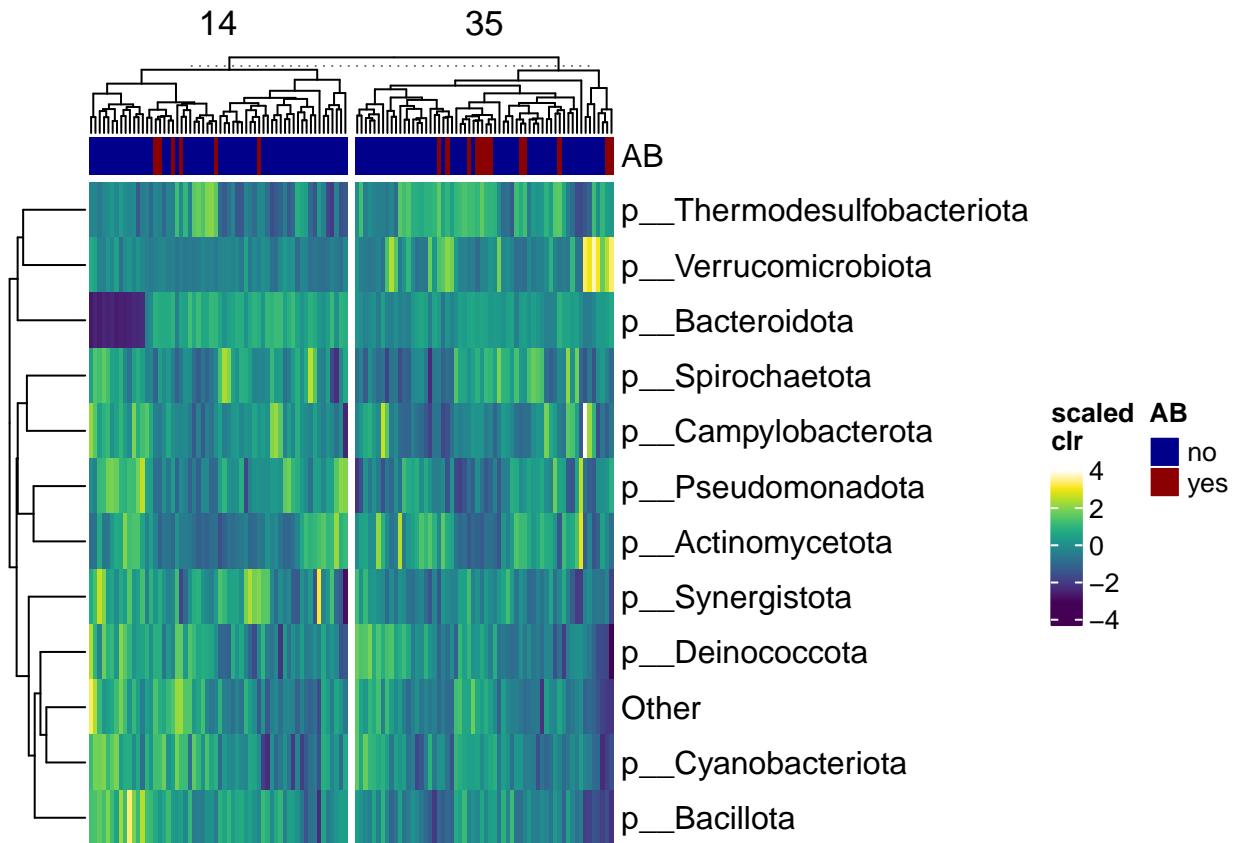
Stable1 Stable2 Stable3 Stable4 Stable5 Stable6



```
# AB and agent
sechm(tse_phylum,
  features = rownames(tse_phylum),
  assayName = "clr",
  do.scale = TRUE,
  top_annotation = c("AB"),
  gaps_at = "Cox",
  hmcols = viridis(256),
  cluster_cols = TRUE, cluster_rows = TRUE)
```

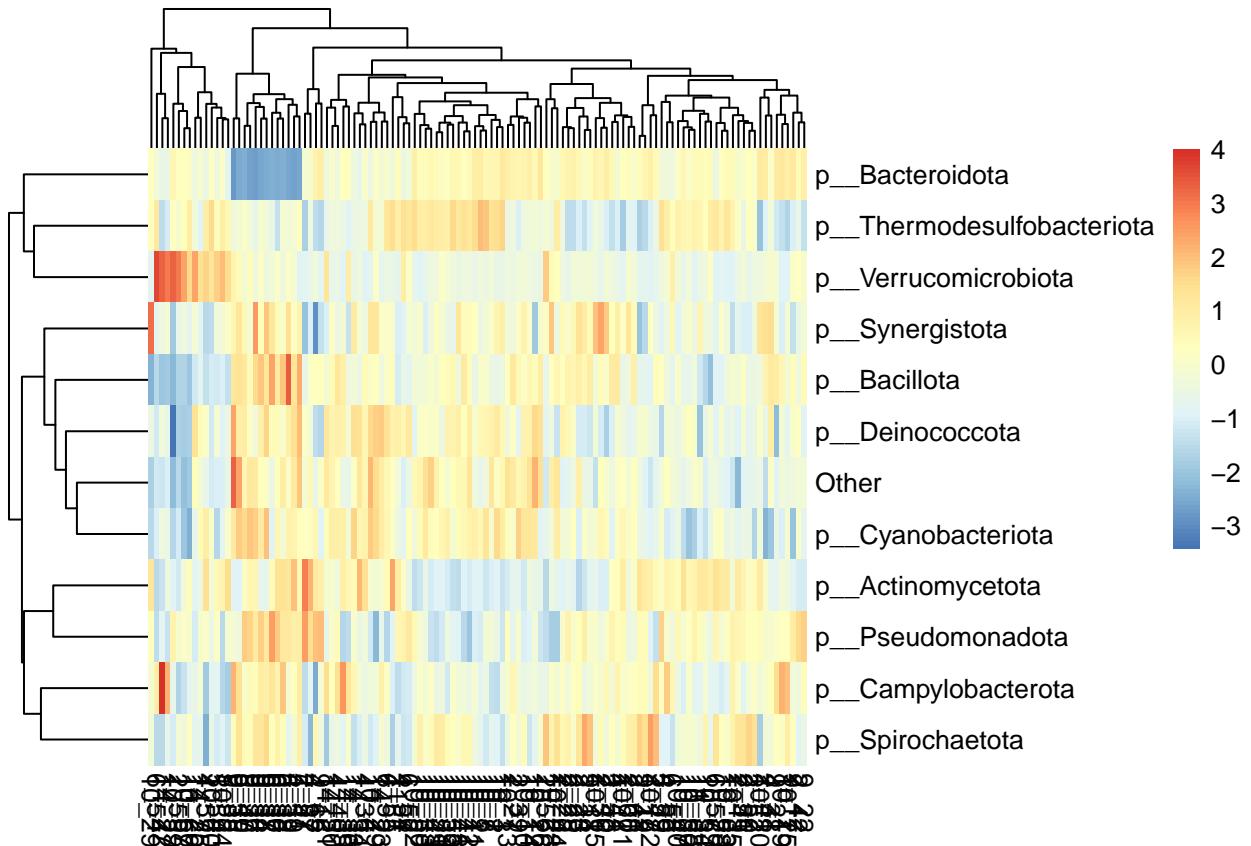


```
# AB and age, we do see interesting shifts here
sechm(tse_phylum,
  features = rownames(tse_phylum),
  assayName = "clr",
  do.scale = TRUE,
  top_annotation = c("AB"),
  gaps_at = "Age",
  hmcols = viridis(256),
  cluster_cols = TRUE, cluster_rows = TRUE)
```



```
# Phylum heatmap
mat <- assay(tse_phylum, "clr_z")

pheatmap(mat)
```



Phylum heatmap hierachal clustering with AB, Clustering both samples and features hierarchically

```

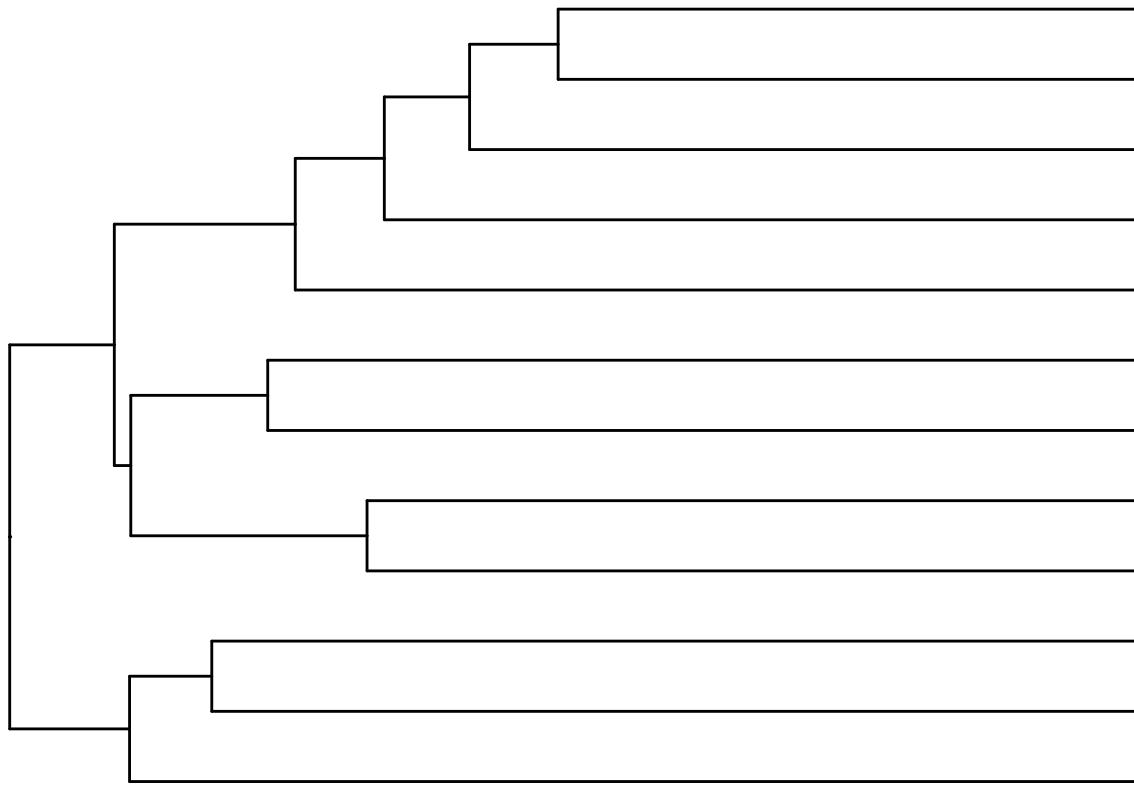
taxa_hclust <- hclust(dist(mat), method = "complete")

# Creates a phylogenetic tree
taxa_tree <- as.phylo(taxa_hclust)

# Plot taxa tree
taxa_tree <- ggtree(taxa_tree) +
  theme(plot.margin=margin(0,0,0,0)) # removes margins

taxa_tree # based on this three, we'll create two clusters

```



```

# Get order of taxa in plot
taxa_ordered <- get_taxa_name(taxa_tree)

taxa_clusters <- cutree(tree = taxa_hclust, k = 3) # 3 clusters based on tree figure

# Converts into data frame
taxa_clusters <- data.frame(clusters = taxa_clusters)
taxa_clusters$clusters <- factor(taxa_clusters$clusters)

# Order data so that it's same as in phylo tree
taxa_clusters <- taxa_clusters[taxa_ordered, , drop = FALSE]

```

```

# Prints taxa and their clusters
taxa_clusters

```

##	clusters
## p__Cyanobacteriota	1
## Other	1
## p__Deinococcota	1
## p__Bacillota	1
## p__Synergistota	1
## p__Spirochaetota	2
## p__Campylobacterota	2
## p__Pseudomonadota	2
## p__Actinomycetota	2
## p__Verrucomicrobiota	3

```

## p__Thermodesulfobacteriota      3
## p__Bacteroidota                3

rowData(tse_phylum)$clusters <- taxa_clusters[order(match(rownames(taxa_clusters), rownames(tse_phylum)))]  

# Prints taxa and their clusters
rowData(tse_phylum)$clusters  
  

## [1] 1 2 1 3 2 1 1 2 2 1 3 3
## Levels: 1 2 3  
  

sample_hclust <- hclust(dist(t(mat)), method = "complete")  
  

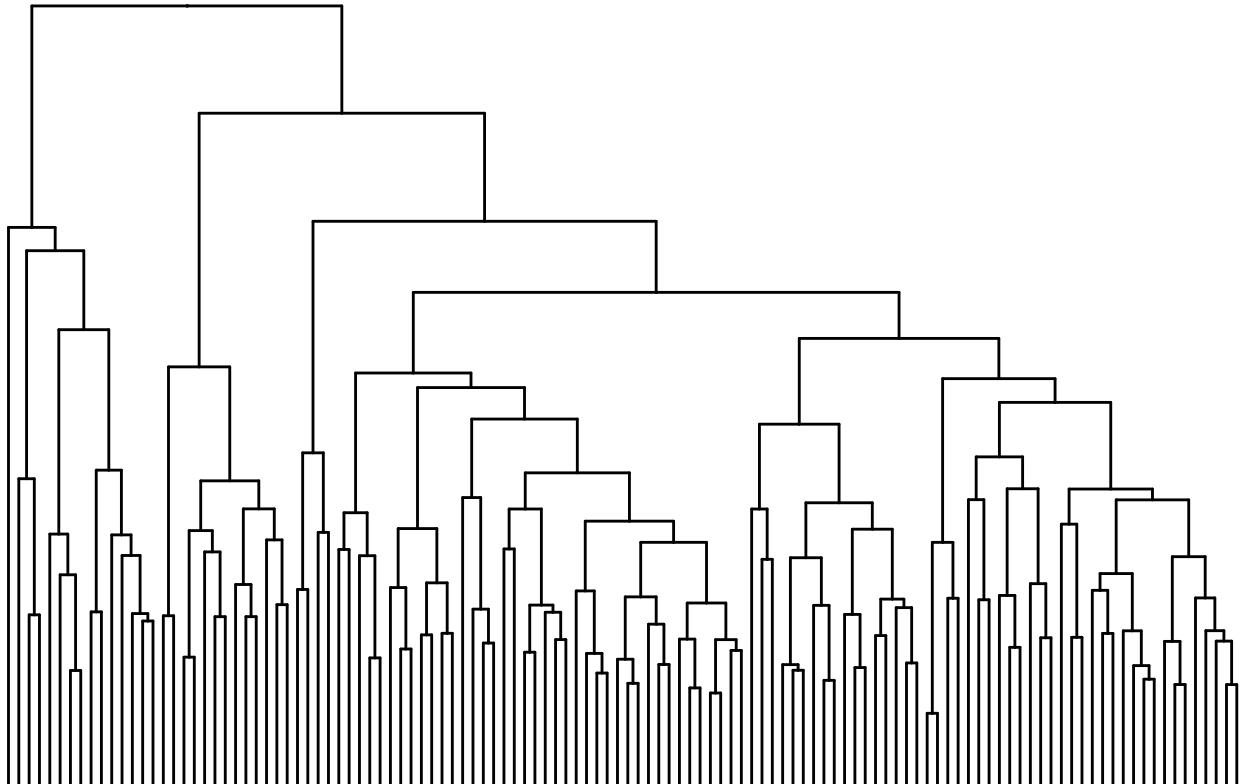
# Creates a phylogenetic tree
sample_tree <- as.phylo(sample_hclust)  
  

# Plot sample tree
sample_tree <- ggtree(sample_tree) + layout_dendrogram() +
  theme(plot.margin=margin(0,0,0,0)) # removes margins  
  

# Get order of samples in plot
samples_ordered <- rev(get_taxa_name(sample_tree))  
  

# to view the tree, run
sample_tree

```



```

# Creates clusters
sample_clusters <- factor(cutree(tree = sample_hclust, k = 2)) # 2 clusters based on methods in Cluster

# Converts into data frame
sample_data <- data.frame(clusters = sample_clusters)

# Order data so that it's same as in phylo tree
sample_data <- sample_data[samples_ordered, , drop = FALSE]

# Order data based on
tse_phylum <- tse_phylum[ , rownames(sample_data)]

# Add sample type data
sample_data$sample_types <- colData(tse_phylum)$AB

sample_data

```

	clusters	sample_types
## 10_29	2	no
## 6_54	2	no
## 14_22	2	no
## 14_27	2	no
## 10_59	2	yes
## 2_59	2	no
## 10_57	2	yes
## 10_66	2	no
## 14_20	2	no
## 14_25	2	no
## 4_57	2	no
## 10_60	2	yes
## 5_54	2	no
## 10_64	2	yes
## 5_55	2	no
## 9_35	1	no
## 9_39	1	no
## 5_40	1	no
## 9_37	1	no
## 9_36	1	no
## 9_34	1	no
## 9_38	1	no
## 6_37	1	no
## 6_36	1	no
## 6_38	1	no
## 5_39	1	no
## 4_39	1	no
## 5_41	1	no
## 2_57	1	no
## 4_37	1	no
## 4_65	1	no
## 9_16	1	no
## 14_35	1	no
## 14_36	1	no
## 14_21	1	no

## 14_30	1	no
## 4_40	1	no
## 14_33	1	no
## 14_34	1	no
## 4_36	1	no
## 10_4	1	no
## 14_29	1	no
## 14_23	1	no
## 6_55	1	no
## 4_55	1	no
## 11_1	1	no
## 4_54	1	no
## 6_56	1	no
## 10_12	1	yes
## 10_26	1	no
## 10_13	1	yes
## 2_27	1	no
## 2_39	1	no
## 10_10	1	yes
## 10_14	1	yes
## 10_52	1	yes
## 10_50	1	yes
## 10_33	1	no
## 10_43	1	no
## 10_11	1	yes
## 10_2	1	no
## 10_8	1	no
## 10_7	1	no
## 10_1	1	no
## 10_3	1	no
## 10_48	1	yes
## 10_41	1	no
## 10_42	1	no
## 10_49	1	yes
## 10_51	1	yes
## 10_39	1	no
## 10_53	1	yes
## 2_56	1	no
## 10_28	1	no
## 10_44	1	no
## 2_26	1	no
## 2_25	1	no
## 2_29	1	no
## 2_23	1	no
## 2_24	1	no
## 9_18	1	no
## 10_15	1	yes
## 2_36	1	no
## 2_40	1	no
## 2_58	1	no
## 4_41	1	no
## 10_21	1	no
## 2_60	1	no
## 2_61	1	no

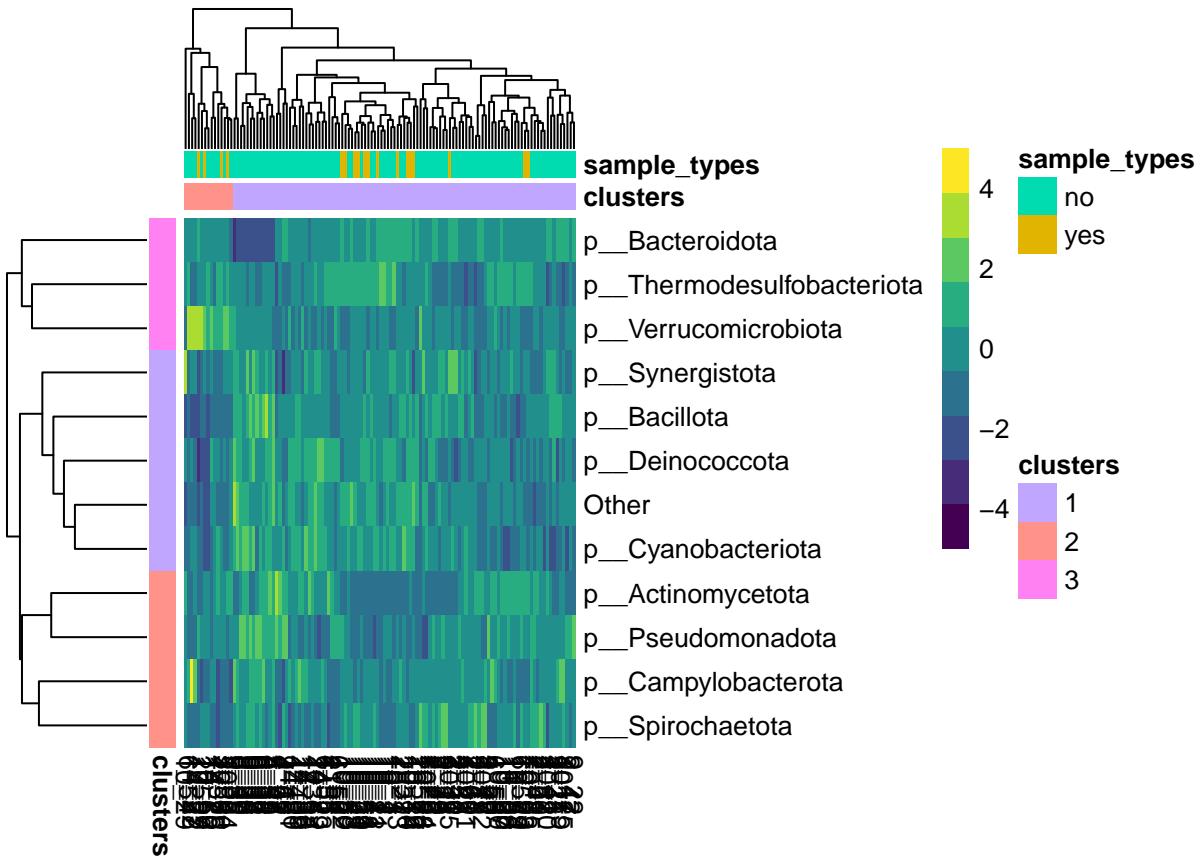
```

## 2_48      1      no
## 9_19      1      no
## 10_22     1      no
## 2_47      1      no
## 10_20     1      no
## 2_41      1      no
## 4_38      1      no
## 9_21      1      no
## 10_19     1      no
## 10_25     1      no
## 9_17      1      no
## 2_42      1      no
## 9_22      1      no
## 5_59      1      no
## 4_56      1      no
## 6_58      1      no
## 11_3      1      no
## 10_34     1      no
## 10_69     1      no
## 10_40      1      no
## 10_67     1      no
## 10_30     1      no
## 10_68     1      no
## 6_57      1      no
## 10_58     1      yes
## 10_63     1      yes
## 10_35     1      no
## 2_49      1      no
## 2_51      1      no
## 2_50      1      no
## 2_52      1      no

breaks <- seq(-ceiling(max(abs(mat))), ceiling(max(abs(mat))),
               length.out = ifelse( max(abs(mat))>5, 2*ceiling(max(abs(mat))), 10 ) )
#colors <- colorRampPalette(c("darkblue", "blue", "white", "red", "darkred"))(length(breaks)-1) replace

pheatmap(mat, annotation_row = taxa_clusters,
         annotation_col = sample_data,
         breaks = breaks,
         color = colorRampPalette(viridis(256))(length(breaks)-1))

```

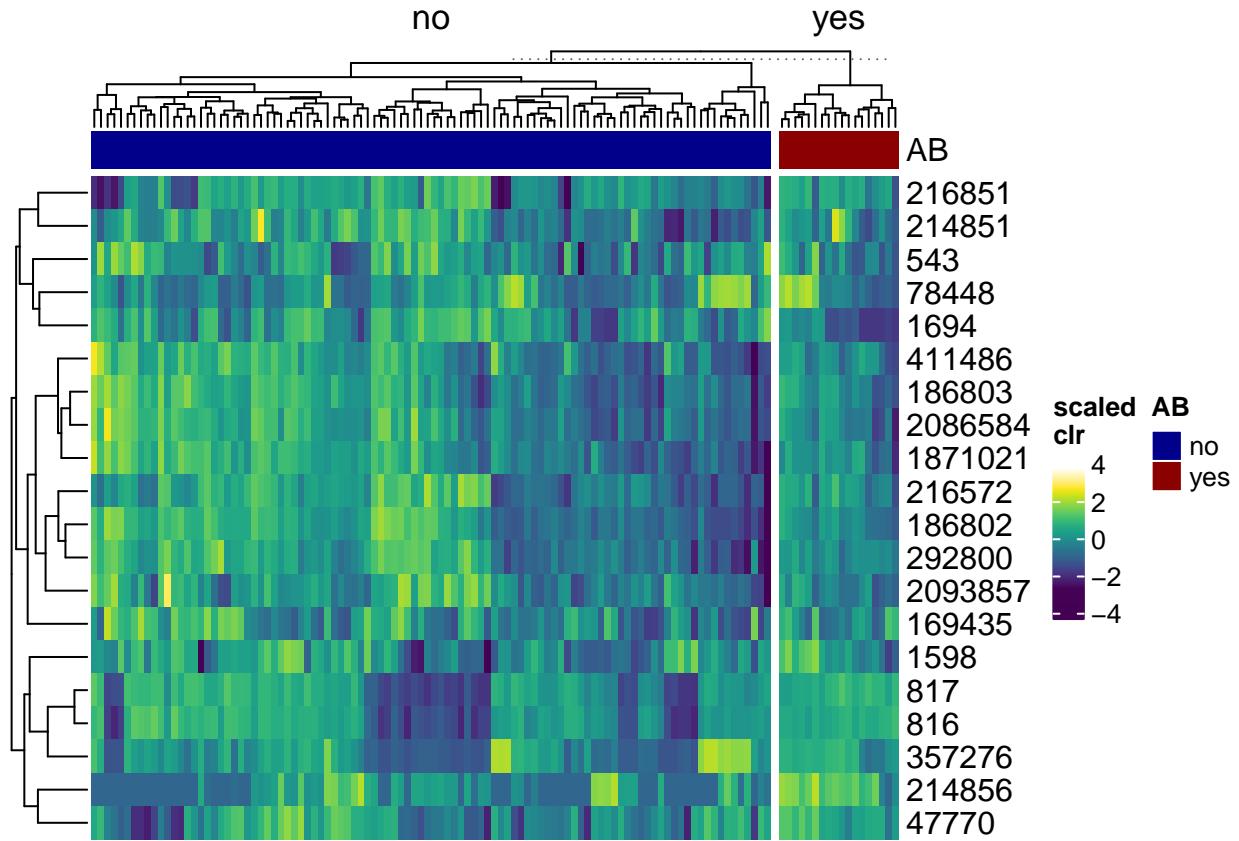


heatmaps on OTU level

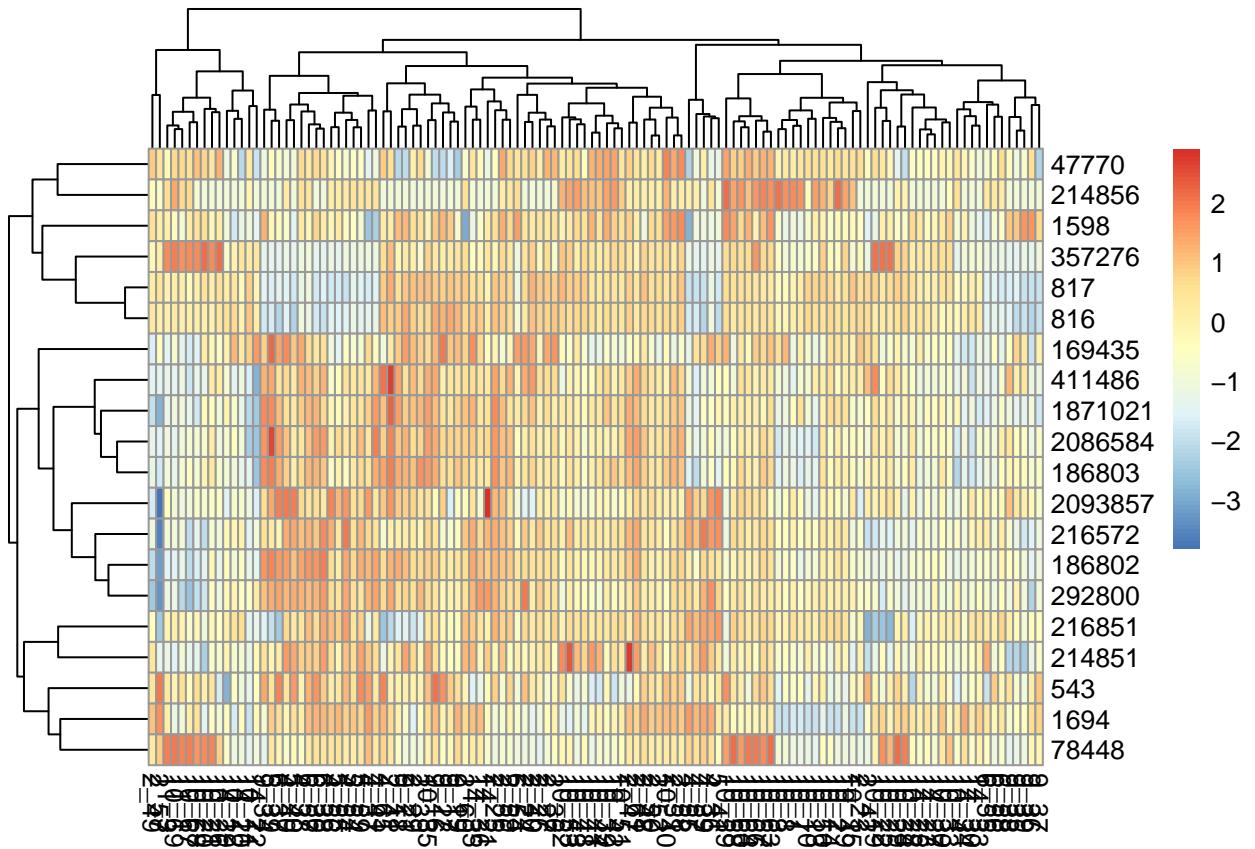
```
tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)
tse <- transformCounts(tse, method = "relabundance")
tse <- transformCounts(tse, MARGIN = "samples", method = "clr", assay.type = "counts", pseudocount=1)
tse <- transformCounts(tse, assay.type = "clr",
                      MARGIN = "features",
                      method = "z", name = "clr_z")
top_taxa <- getTopTaxa(tse, top = 20)
tse <- tse[top_taxa, ]

# OTU heatmap AB
tse@metadata$anno_colors$AB = c(yes = "darkred", no ="darkblue")

sechm(tse,
      features = rownames(tse),
      assayName = "clr",
      do.scale = TRUE,
      top_annotation = c("AB"),
      gaps_at = "AB",
      hmcols = viridis(256),
      cluster_cols = TRUE, cluster_rows = TRUE)
```



```
# OTU heatmap
mat <- assay(tse, "clr_z")
pheatmap(mat)
```



```

# OTU heatmap hierachal clustering with AB

# Clustering both samples and features hierarchically

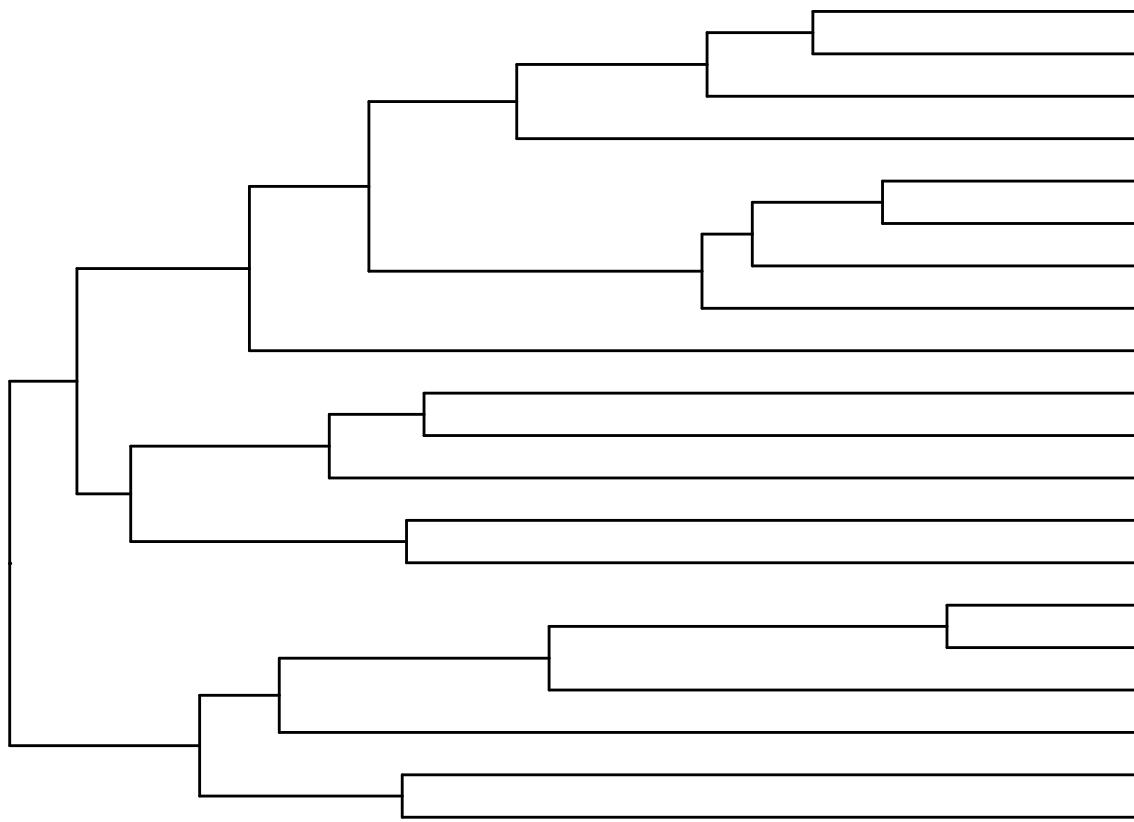
taxa_hclust <- hclust(dist(mat), method = "complete")

# Creates a phylogenetic tree
taxa_tree <- as.phylo(taxa_hclust)

# Plot taxa tree
taxa_tree <- ggtree(taxa_tree) +
  theme(plot.margin=margin(0,0,0,0)) # removes margins

taxa_tree # based on this three, we'll create two clusters

```



```

# Get order of taxa in plot
taxa_ordered <- get_taxa_name(taxa_tree)

taxa_clusters <- cutree(tree = taxa_hclust, k = 2) # 2 clusters based on methods in Clustering.R script

# Converts into data frame
taxa_clusters <- data.frame(clusters = taxa_clusters)
taxa_clusters$clusters <- factor(taxa_clusters$clusters)

# Order data so that it's same as in phylo tree
taxa_clusters <- taxa_clusters[taxa_ordered, , drop = FALSE]

# Prints taxa and their clusters
taxa_clusters

##      clusters
## 292800      1
## 186802      1
## 216572      1
## 2093857     1
## 186803      1
## 2086584     1
## 1871021     1
## 411486      1
## 169435      1
## 78448       1

```

```

## 1694          1
## 543           1
## 214851        1
## 216851        1
## 816            2
## 817            2
## 357276        2
## 1598           2
## 214856        2
## 47770          2

# Prints taxa and their clusters
rowData(tse_phylum)$clusters

## [1] 1 2 1 3 2 1 1 2 2 1 3 3
## Levels: 1 2 3

sample_hclust <- hclust(dist(t(mat)), method = "complete")

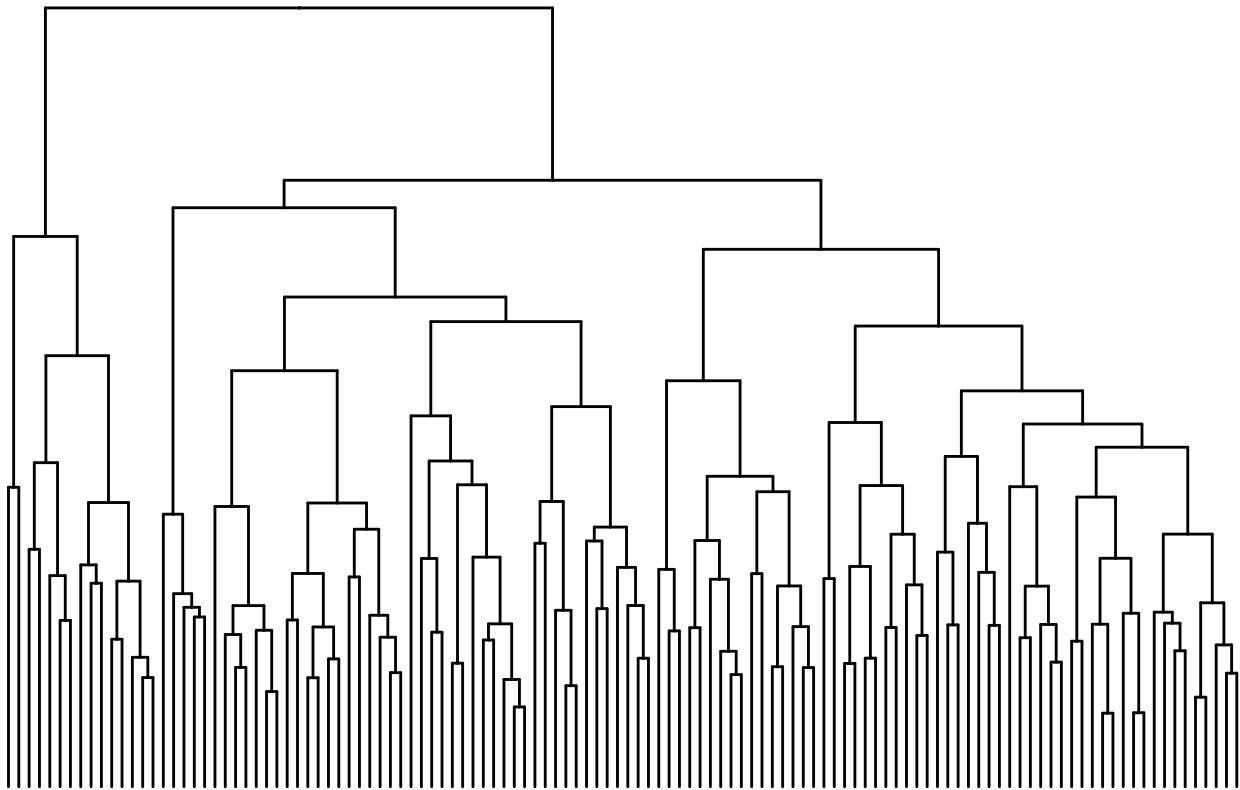
# Creates a phylogenetic tree
sample_tree <- as.phylo(sample_hclust)

# Plot sample tree
sample_tree <- ggtree(sample_tree) + layout_dendrogram() +
  theme(plot.margin=margin(0,0,0,0)) # removes margins

# Get order of samples in plot
samples_ordered <- rev(get_taxa_name(sample_tree))

# to view the tree, run
sample_tree

```



```

# Creates clusters
sample_clusters <- factor(cutree(tree = sample_hclust, k = 2))

# Converts into data frame
sample_data <- data.frame(clusters = sample_clusters)

# Order data so that it's same as in phylo tree
sample_data <- sample_data[samples_ordered, , drop = FALSE]

# Order data based on
tse_phylum <- tse_phylum[ , rownames(sample_data)]

# Add sample type data
sample_data$sample_types <- colData(tse_phylum)$AB

sample_data

##      clusters sample_types
## 2_49          2        no
## 2_57          2        no
## 10_11         2       yes
## 14_22         2        no
## 14_29         2        no
## 10_12         2       yes
## 14_20         2        no
## 10_20         2        no

```

## 10_21	2	no
## 10_26	2	no
## 10_67	2	no
## 10_68	2	no
## 11_3	2	no
## 10_69	2	no
## 11_1	2	no
## 4_55	1	no
## 4_57	1	no
## 4_39	1	no
## 4_36	1	no
## 5_41	1	no
## 10_59	1	yes
## 10_58	1	yes
## 10_60	1	yes
## 10_64	1	yes
## 10_66	1	no
## 10_57	1	yes
## 10_63	1	yes
## 10_10	1	yes
## 10_13	1	yes
## 10_3	1	no
## 10_8	1	no
## 10_1	1	no
## 10_7	1	no
## 10_15	1	yes
## 2_23	1	no
## 10_50	1	yes
## 10_14	1	yes
## 10_2	1	no
## 10_49	1	yes
## 2_42	1	no
## 10_19	1	no
## 10_22	1	no
## 10_25	1	no
## 10_29	1	no
## 10_34	1	no
## 10_28	1	no
## 10_30	1	no
## 10_33	1	no
## 14_25	1	no
## 14_23	1	no
## 14_27	1	no
## 9_36	1	no
## 9_37	1	no
## 9_38	1	no
## 9_35	1	no
## 9_39	1	no
## 6_55	1	no
## 6_56	1	no
## 6_58	1	no
## 10_4	1	no
## 14_34	1	no
## 14_30	1	no

## 14_33	1	no
## 9_34	1	no
## 5_39	1	no
## 6_37	1	no
## 4_40	1	no
## 5_40	1	no
## 4_38	1	no
## 6_57	1	no
## 6_36	1	no
## 6_38	1	no
## 4_37	1	no
## 4_41	1	no
## 4_56	1	no
## 5_55	1	no
## 4_54	1	no
## 5_54	1	no
## 5_59	1	no
## 4_65	1	no
## 2_41	1	no
## 9_18	1	no
## 9_21	1	no
## 2_29	1	no
## 2_39	1	no
## 10_35	1	no
## 9_16	1	no
## 9_22	1	no
## 9_17	1	no
## 9_19	1	no
## 2_60	1	no
## 14_35	1	no
## 14_36	1	no
## 14_21	1	no
## 2_25	1	no
## 2_56	1	no
## 2_59	1	no
## 6_54	1	no
## 2_24	1	no
## 2_40	1	no
## 2_26	1	no
## 2_27	1	no
## 2_36	1	no
## 2_47	1	no
## 2_61	1	no
## 2_48	1	no
## 2_50	1	no
## 2_52	1	no
## 10_40	1	no
## 2_51	1	no
## 2_58	1	no
## 10_52	1	yes
## 10_53	1	yes
## 10_43	1	no
## 10_48	1	yes
## 10_41	1	no

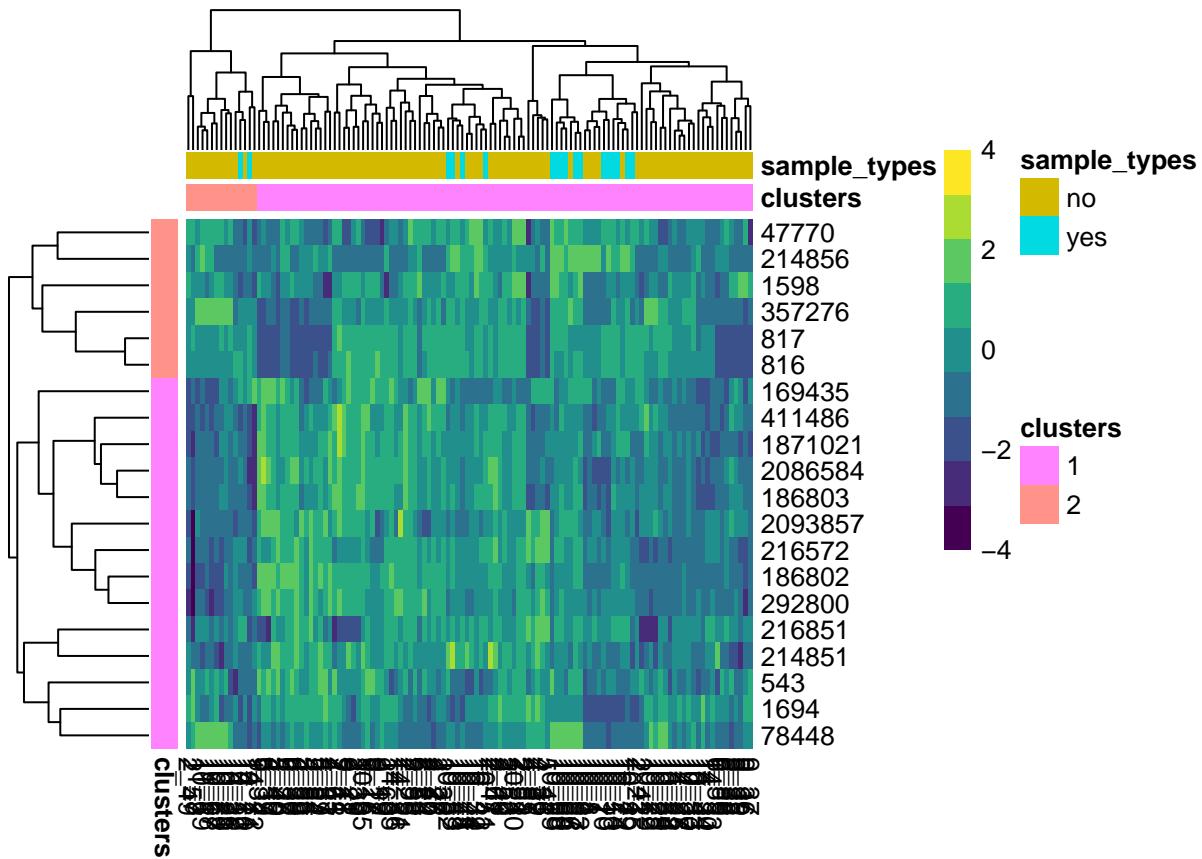
```

## 10_44      1       no
## 10_39      1       no
## 10_42      1       no
## 10_51      1      yes

breaks <- seq(-ceiling(max(abs(mat))), ceiling(max(abs(mat))),
               length.out = ifelse( max(abs(mat))>5, 2*ceiling(max(abs(mat))), 10 ) )

pheatmap(mat, annotation_row = taxa_clusters,
          annotation_col = sample_data,
          breaks = breaks,
          color = colorRampPalette(viridis(256))(length(breaks)-1))

```

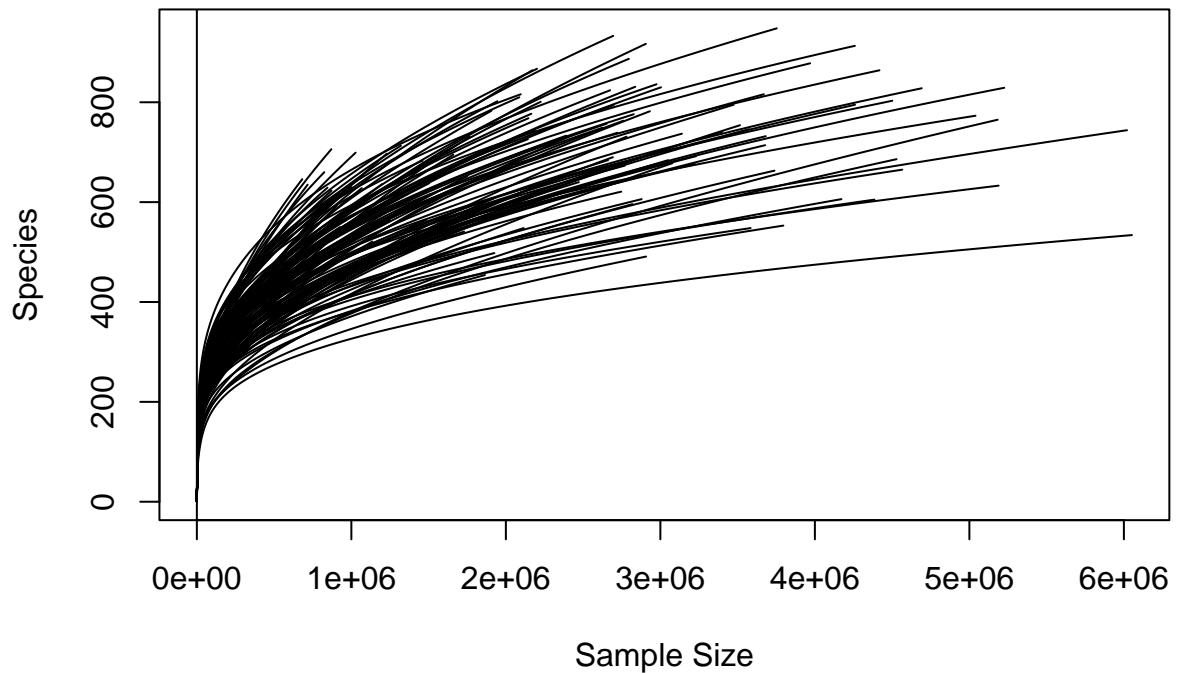


```

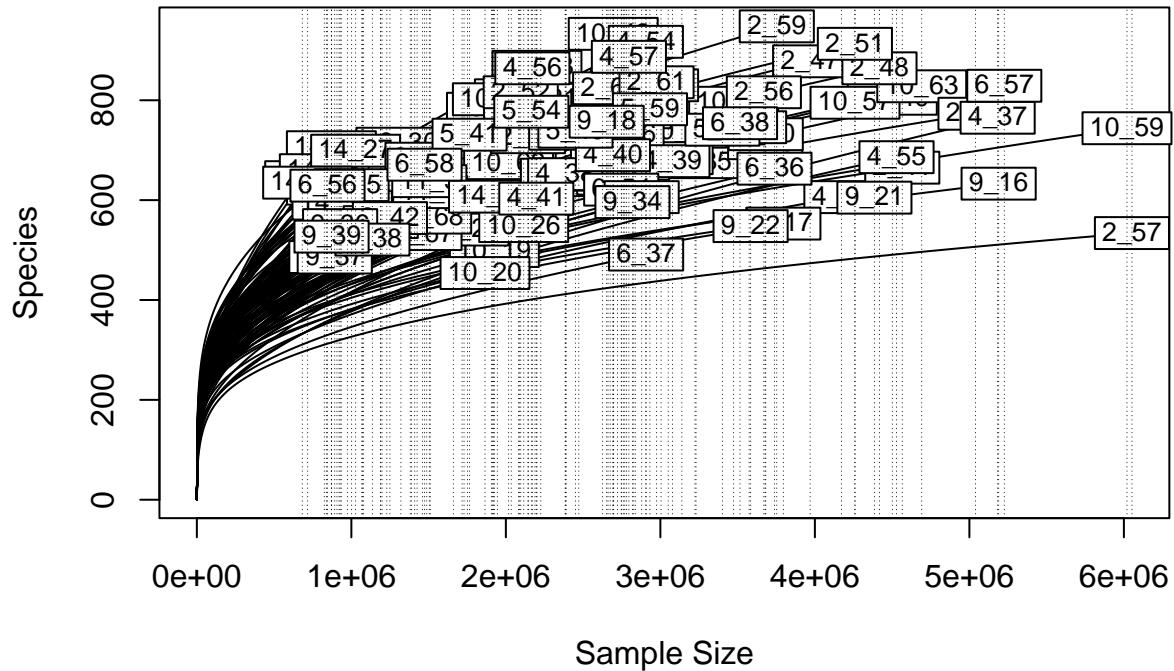
## Alpha diversity

otu_tab <- t(abundances(subsetMG))
# rarefaction curve - takes a long time
vegan::rarecurve(otu_tab,
                  step = 5000, label = FALSE,
                  sample = min(rowSums(otu_tab),
                               col = "blue", cex = 0.6))

```



```
# we can add lines to show sampling depths
rarecurve(otu_tab, step=5000)
abline(v=sample_sums(subsetMG), lty='dotted', lwd=0.5)
```



```
# we do see that many samples do not plateau
```

```
summary(goods(otu_tab)) # there are on average 0.01% singletons in this data
```

```
##      no.sing      no.seqs      goods
##  Min.   :154.0   Min.   : 682316   Min.   : 99.96
##  1st Qu.:224.8   1st Qu.:1451674  1st Qu.: 99.98
##  Median :265.5   Median :2305179  Median : 99.99
##  Mean   :264.0   Mean   :2475820  Mean   : 99.99
##  3rd Qu.:302.5   3rd Qu.:3093301  3rd Qu.: 99.99
##  Max.   :421.0   Max.   :6052117  Max.   :100.00
```

```
#rarefy to equal library size or not?
```

```
lib.div <- microbiome::alpha(subsetMG, index = "all")
lib.div$ReadsPerSample <- sample_sums(subsetMG)
colnames(lib.div)
```

```
## [1] "observed"                  "chao1"
## [3] "diversity_inverse_simpson" "diversity_gini_simpson"
## [5] "diversity_shannon"         "diversity_fisher"
## [7] "diversity_coverage"        "evenness_camargo"
## [9] "evenness_pielou"           "evenness_simpson"
## [11] "evenness_evar"             "evenness_bulla"
```

```

## [13] "dominance_dbp"
## [15] "dominance_absolute"
## [17] "dominance_simpson"
## [19] "dominance_gini"
## [21] "rarity_low_abundance"
## [23] "ReadsPerSample"

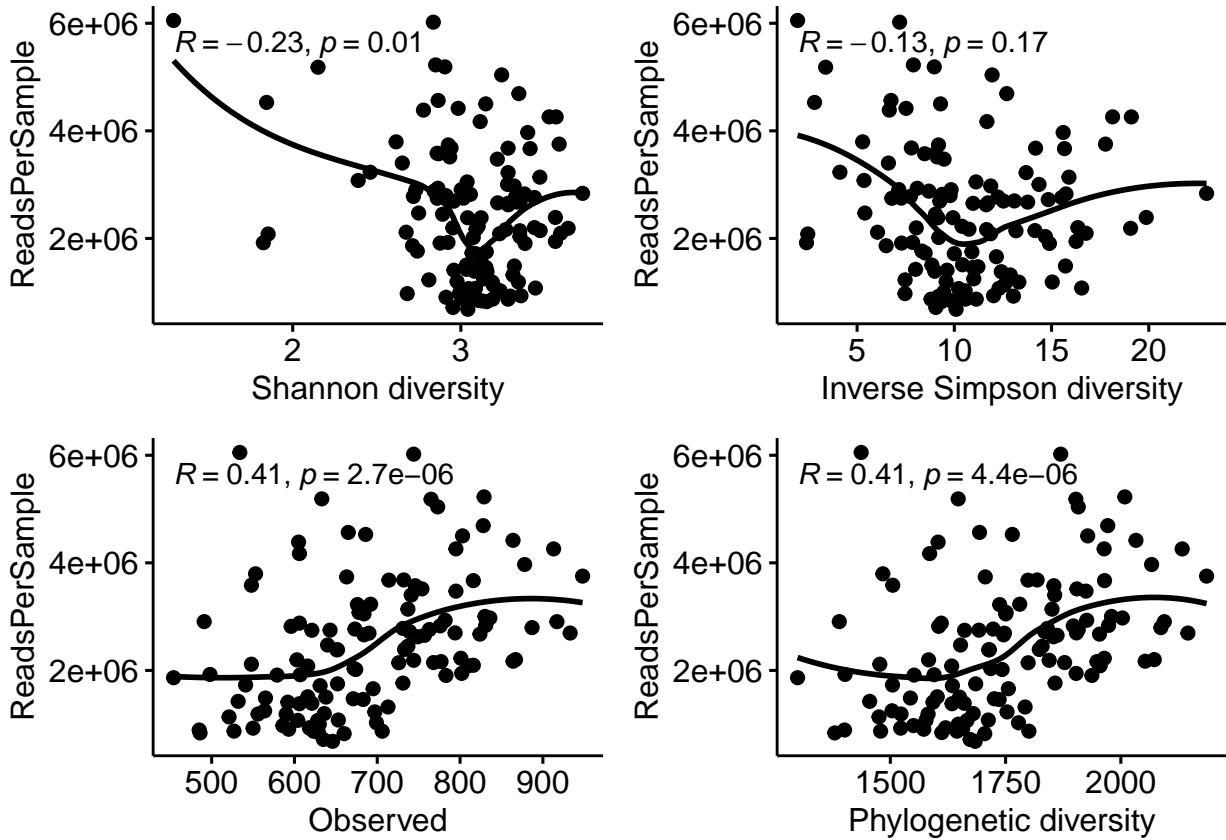
p1 = ggscatter(lib.div, "diversity_shannon", "ReadsPerSample", xlab = "Shannon diversity", add = "loess",
  stat_cor(method = "pearson")
p2 = ggscatter(lib.div, "diversity_inverse_simpson", "ReadsPerSample", xlab = "Inverse Simpson diversity",
  stat_cor(method = "pearson")
p3 = ggscatter(lib.div, "observed", "ReadsPerSample", xlab = "Observed", add = "loess") +
  stat_cor(method = "pearson")

df.pd <- pd(t(as.data.frame(subsetMG@otu_table)), subsetMG@phy_tree, include.root=T) # transposing for use
lib.div$Phylogenetic_Diversity <- df.pd$PD

p4 = ggscatter(lib.div, "Phylogenetic_Diversity", "ReadsPerSample", xlab = "Phylogenetic diversity", add = "loess",
  stat_cor(method = "pearson"))

ggarrange(p1, p2, p3, p4, ncol = 2, nrow = 2)

```



remove samples with lower sequencing depth? we can see that observed and phylogenetic diversity increase with library size

```
summary(sample_sums(subsetMG))
```

```
##      Min. 1st Qu. Median     Mean 3rd Qu.    Max.
## 682316 1451674 2305179 2475820 3093301 6052117
```

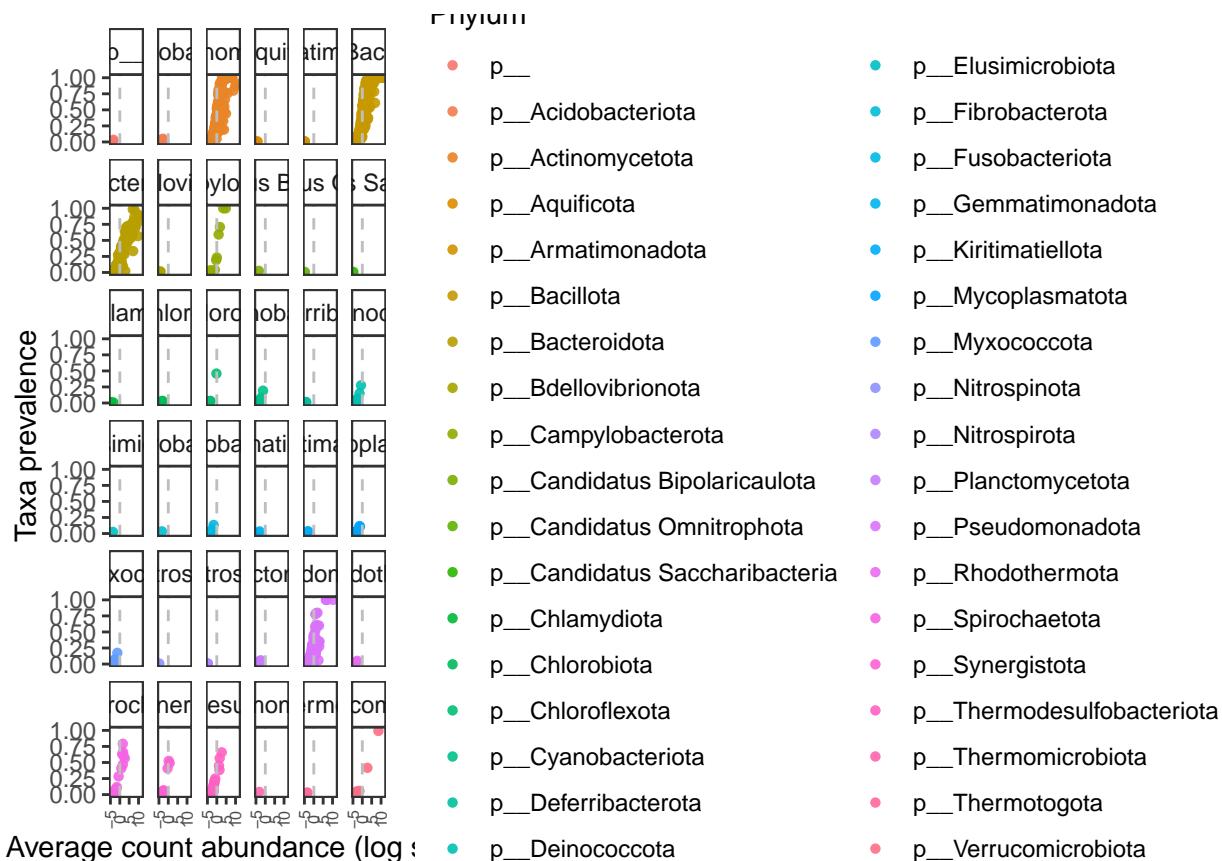
```
set.seed(1337)
```

```
ps0.rar <- rarefy_even_depth(subsetMG, sample.size = 682316, rngseed = 456)
```

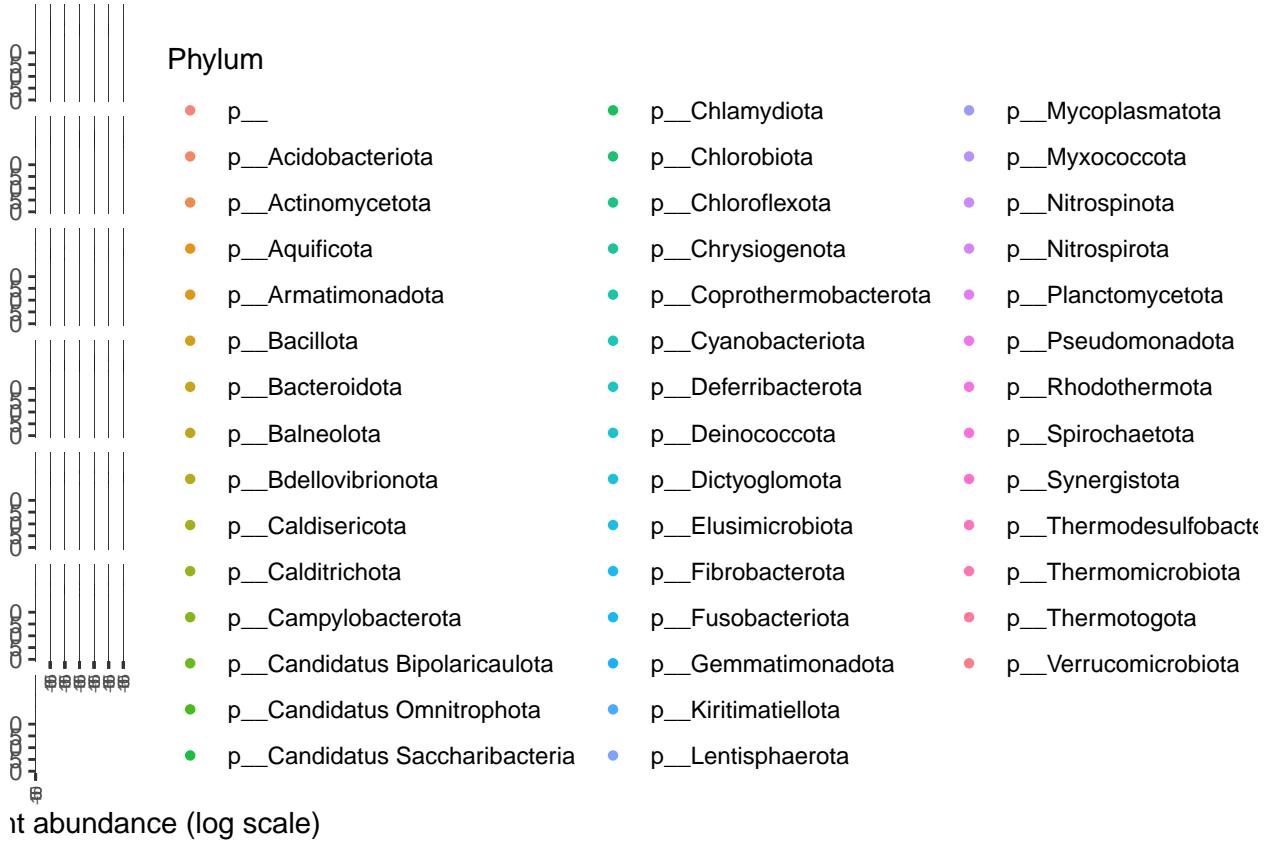
```
ps0.rar
```

```
## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 4189 taxa and 120 samples ]
## sample_data() Sample Data: [ 120 samples by 35 sample variables ]
## tax_table() Taxonomy Table: [ 4189 taxa by 7 taxonomic ranks ]
## phy_tree() Phylogenetic Tree: [ 4189 tips and 4188 internal nodes ]
```

```
plot_taxa_prevalence(ps0.rar, "Phylum")
```



```
plot_taxa_prevalence(subsetMG, "Phylum")
```



```
hmp.div <- microbiome::alpha(ps0.rar, index = "all") # using ps0.rar since we rarefy
datatable(hmp.div)
```

```
hmp.meta <- meta(ps0.rar) # using ps0.rar since rarefied
hmp.meta$sam_name <- rownames(hmp.meta)
hmp.div$sam_name <- rownames(hmp.div)
div.df <- merge(hmp.div,hmp.meta, by = "sam_name")
colnames(div.df)
```

```
## [1] "sam_name"                      "observed"
## [3] "chao1"                          "diversity_inverse_simpson"
## [5] "diversity_gini_simpson"        "diversity_shannon"
## [7] "diversity_fisher"                "diversity_coverage"
## [9] "evenness_camargo"               "evenness_pielou"
## [11] "evenness_simpson"                "evenness_evar"
## [13] "evenness_bulla"                  "dominance_dbp"
## [15] "dominance_dmn"                  "dominance_absolute"
## [17] "dominance_relative"              "dominance_simpson"
## [19] "dominance_core_abundance"       "dominance_gini"
## [21] "rarity_log_modulo_skewness"     "rarity_low_abundance"
## [23] "rarity_rare_abundance"          "Id"
## [25] "SampleIdentifier"                "ResCap"
## [27] "Conc...ng..pl."                 "SampleID"
## [29] "LibraryNumber"                  "Sample_Unique"
```

```

## [31] "LibraryName"                 "Farm"
## [33] "Farm2"                      "Stable"
## [35] "FarmRoundStable"             "Days"
## [37] "Age"                         "Sname"
## [39] "WeightAnimal"                "Gender"
## [41] "AgeParentStock"               "Hatchery"
## [43] "Researcher"                  "AB"
## [45] "Abday"                       "FlockSize"
## [47] "FeedF"                        "FeedType"
## [49] "FeedProducent"                "Cox"
## [51] "OPG"                          "Cluster"
## [53] "LitterType"                  "Metagenomics"
## [55] "ReadPerc"                     "ReadTot"
## [57] "Conc...ng..pl..1"              "Stables"

div.df2 <- div.df[, c("Cox", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon")]
colnames(div.df2) <- c("Agent", "Inverse Simpson", "Gini-Simpson", "Shannon", "Observed", "Coverage", "Metagenomics")

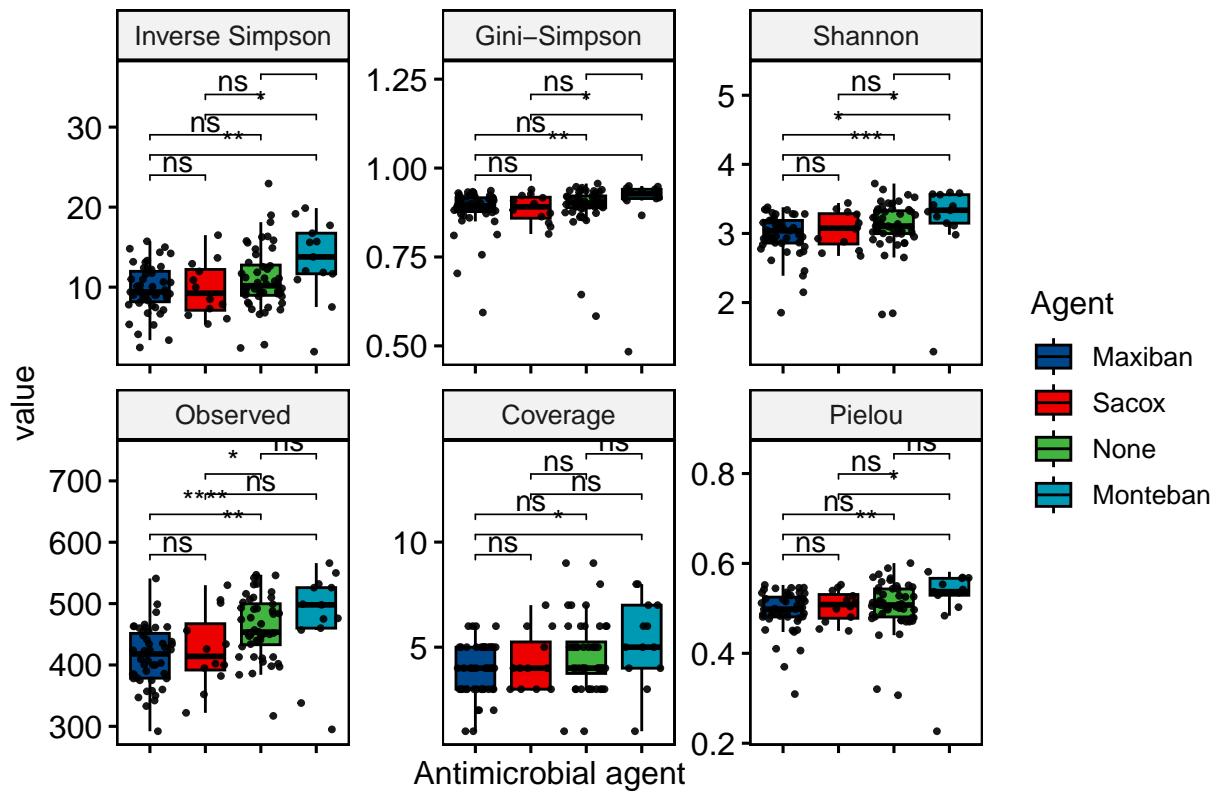
div_df_melt <- reshape2::melt(div.df2)

lev = c("Maxiban", "Sacox", "Monteban", "None")
L.pairs <- combn(seq_along(lev), 2, simplify = FALSE, FUN = function(i) lev[i])

ggboxplot(div_df_melt, x = "Agent", y = "value",
          fill = "Agent",
          palette = "lancet",
          legend= "right",
          facet.by = "variable",
          scales = "free",
          xlab = "Antimicrobial agent",
          title = "Alpha diversity metrics by microbial agent",
          outlier.shape = NA) +
  rremove("x.text") + stat_compare_means(
    comparisons = L.pairs,
    label = "p.signif"
  ) + geom_jitter(size = 0.7, alpha = 0.9)

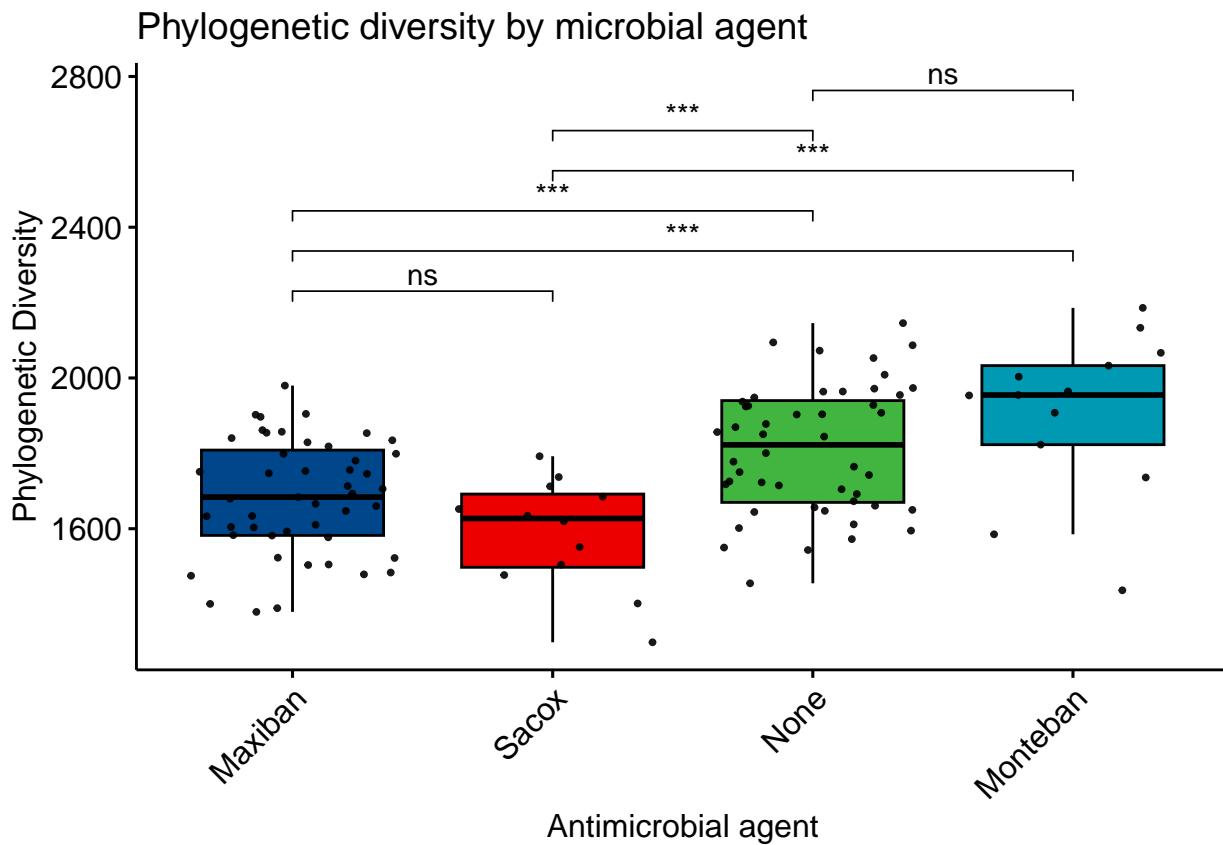
```

Alpha diversity metrics by microbial agent



```
hmp.meta$Phylogenetic_Diversity <- df.pd$PD
```

```
ggboxplot(hmp.meta,
  x = "Cox",
  y = "Phylogenetic_Diversity",
  fill = "Cox",
  palette = "lancet",
  ylab = "Phylogenetic Diversity",
  xlab = "Antimicrobial agent",
  legend = "right",
  title = "Phylogenetic diversity by microbial agent",
  outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) +
  stat_compare_means(
    comparisons = L.pairs,
    label = "p.signif"
  ) + geom_jitter(size = 0.7, alpha = 0.9)
```



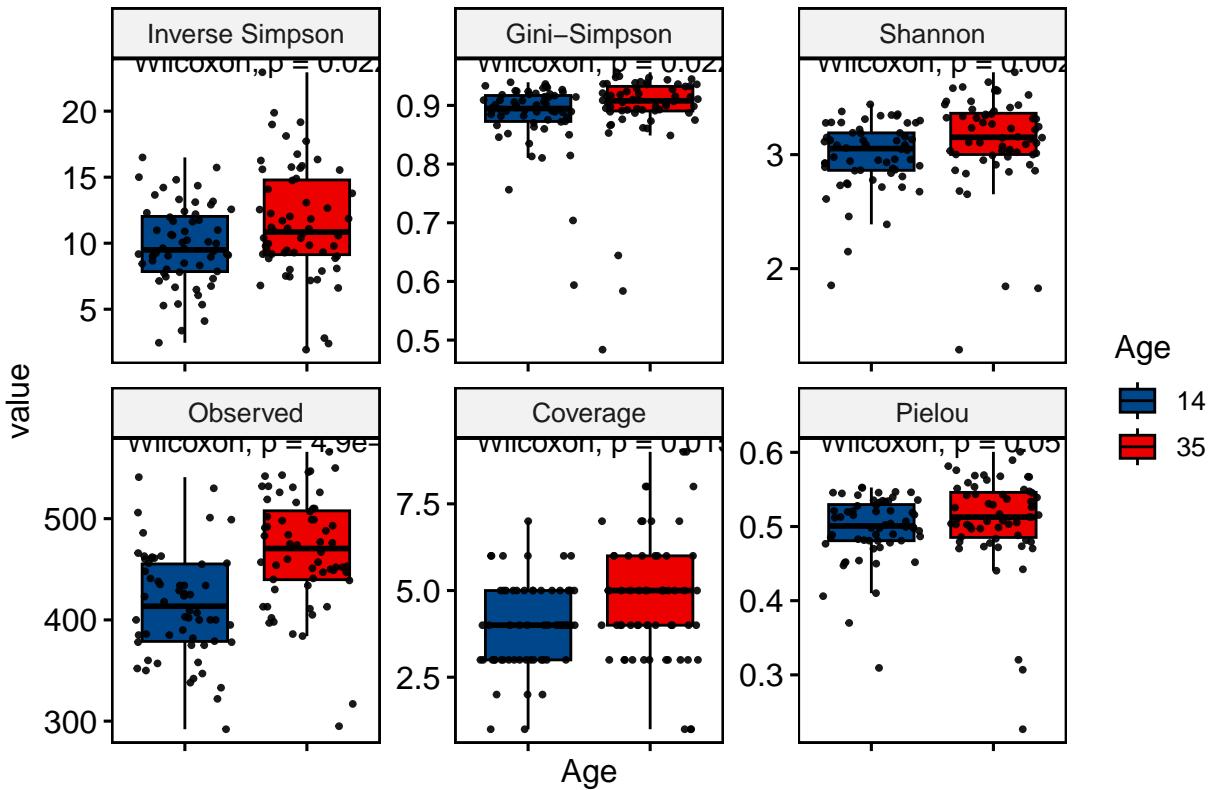
```
# age / days

div.df2 <- div.df[, c("Age", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon")]
colnames(div.df2) <- c("Age", "Inverse Simpson", "Gini-Simpson", "Shannon", "Observed", "Coverage", "Pi")

div.df2$Age = as.factor(div.df2$Age)
div_df_melt <- reshape2::melt(div.df2)

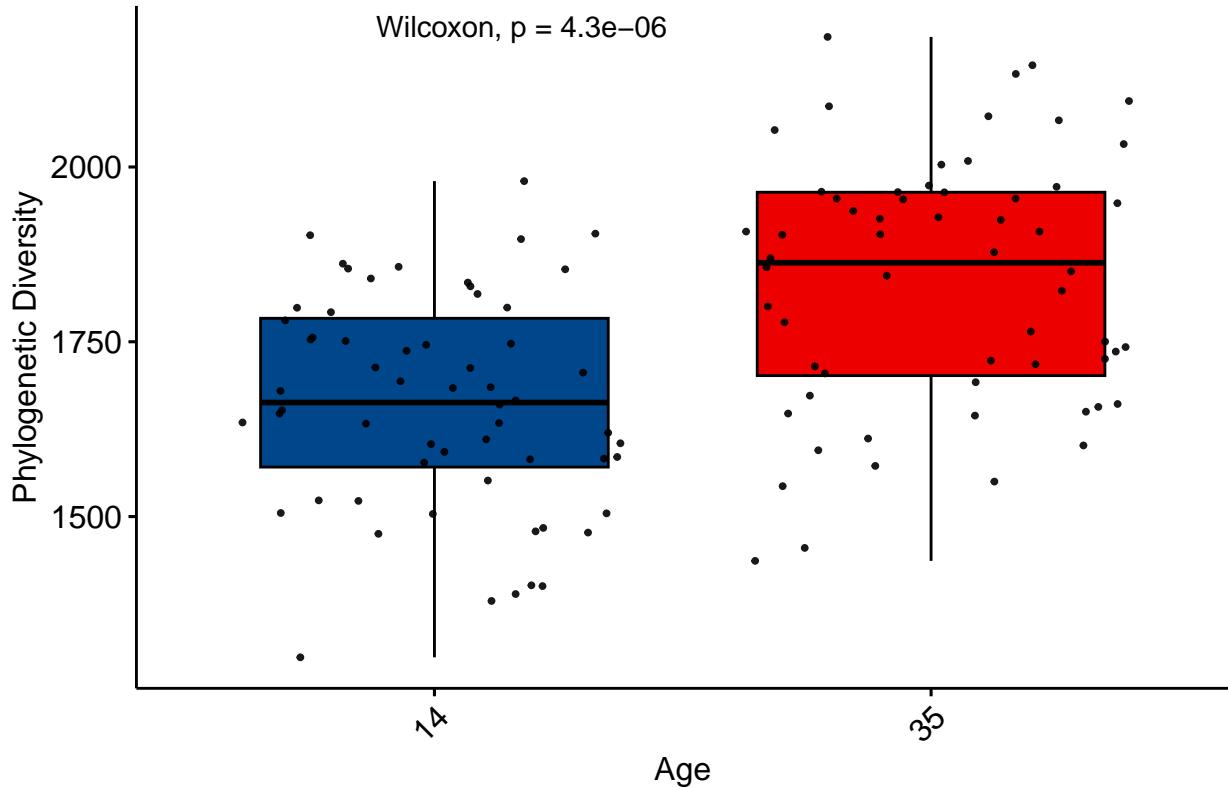
ggboxplot(div_df_melt, x = "Age", y = "value",
          fill = "Age",
          palette = "lancet",
          legend= "right",
          facet.by = "variable",
          scales = "free",
          title = "Alpha diversity metrics by age",
          outlier.shape = NA) +
  rremove("x.text") + stat_compare_means() + geom_jitter(size = 0.7, alpha = 0.9)
```

Alpha diversity metrics by age



```
ggboxplot(hmp.meta,
  x = "Age",
  y = "Phylogenetic_Diversity",
  fill = "Age",
  palette = "lancet",
  ylab = "Phylogenetic Diversity",
  xlab = "Age",
  legend = "right",
  title = "Phylogenetic diversity by age",
  outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) +
  stat_compare_means(paired = TRUE) + geom_jitter(size = 0.7, alpha = 0.9)
```

Phylogenetic diversity by age



```
# farms / company

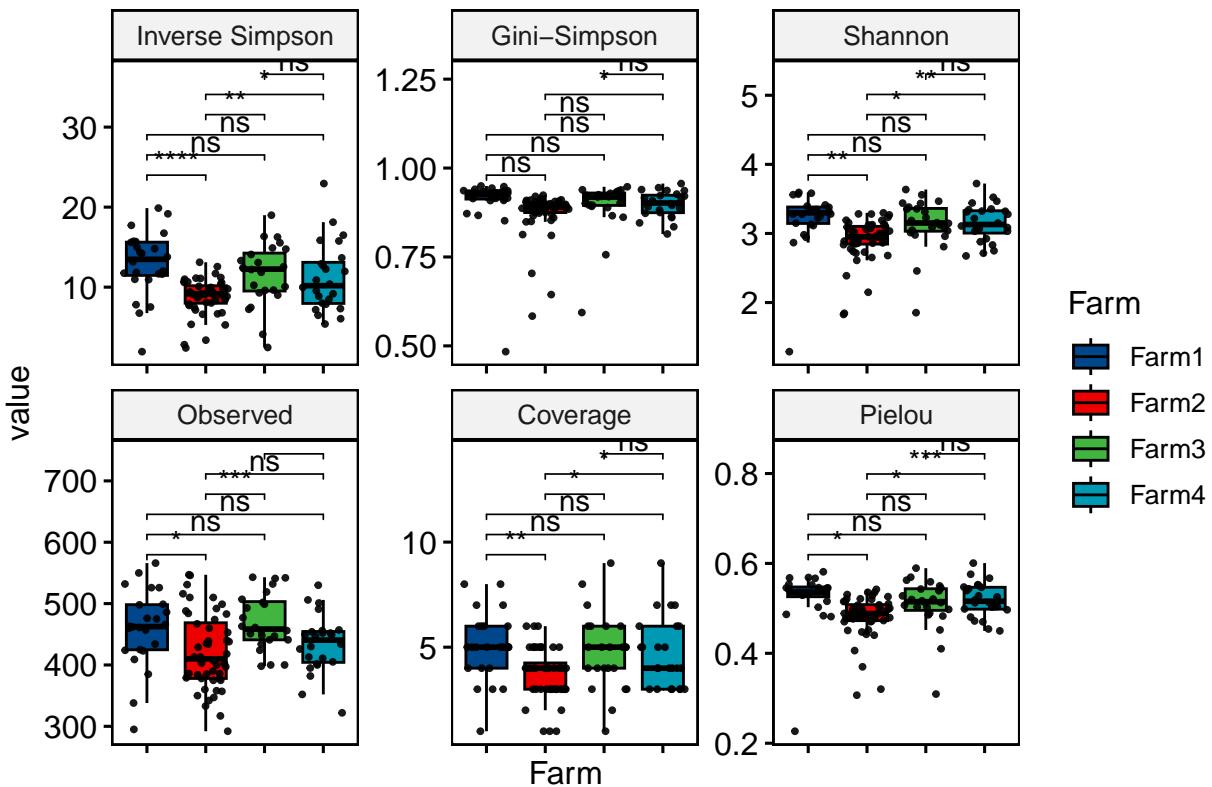
div.df2 <- div.df[, c("Farm2", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon")]
colnames(div.df2) <- c("Farm", "Inverse Simpson", "Gini-Simpson", "Shannon", "Observed", "Coverage", "P")

div_df_melt <- reshape2::melt(div.df2)

lev = c("Farm1", "Farm2", "Farm3", "Farm4")
L.pairs <- combn(seq_along(lev), 2, simplify = FALSE, FUN = function(i) lev[i])

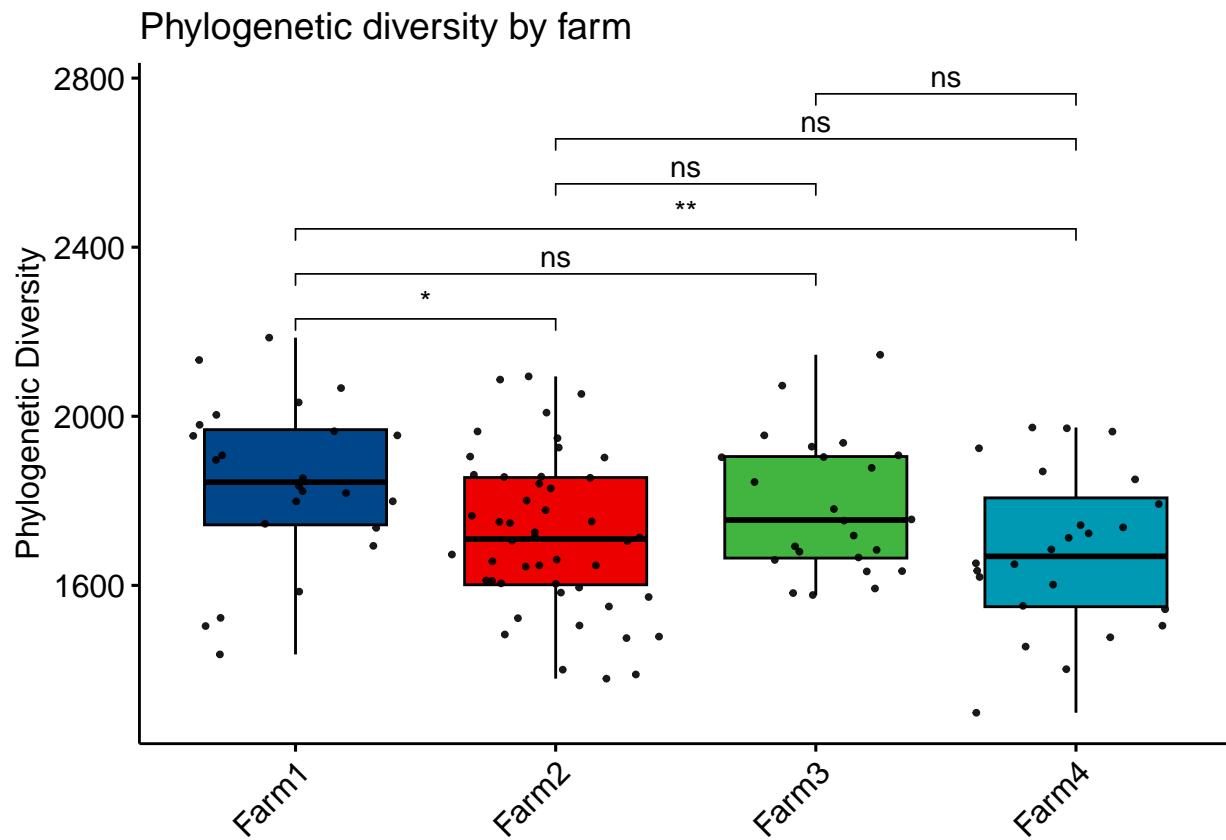
ggboxplot(div_df_melt, x = "Farm", y = "value",
          fill = "Farm",
          palette = "lancet",
          legend= "right",
          facet.by = "variable",
          scales = "free",
          order = lev,
          title = "Alpha diversity metrics by farm",
          outlier.shape = NA) + rotate_x_text() + rremove("x.text") +
stat_compare_means(method = "t.test",
                    comparisons = L.pairs,
                    label = "p.signif"
) + geom_jitter(size = 0.7, alpha = 0.9)
```

Alpha diversity metrics by farm



```
ggboxplot(hmp.meta,
  x = "Farm2",
  y = "Phylogenetic_Diversity",
  fill = "Farm2",
  palette = "lancet",
  ylab = "Phylogenetic Diversity",
  xlab = "Farm",
  order = lev,
  legend = "right",
  title = "Phylogenetic diversity by farm",
  outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12), axis.title
```

stat_compare_means(
 comparisons = L.pairs,
 label = "p.signif"
) + geom_jitter(size = 0.7, alpha = 0.9)



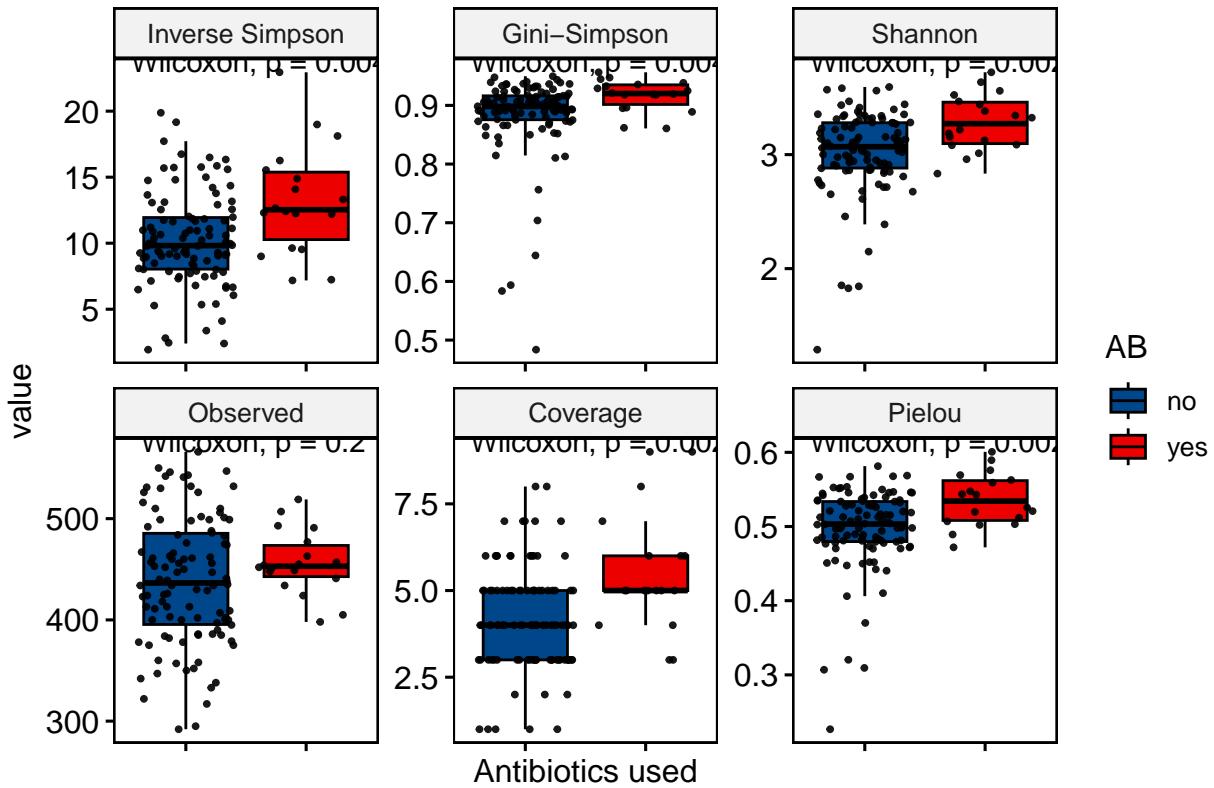
```
# based on AB

div.df2 <- div.df[, c("AB", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon",
colnames(div.df2) <- c("AB", "Inverse Simpson", "Gini-Simpson", "Shannon", "Observed", "Coverage", "Pie")

div_df_melt <- reshape2::melt(div.df2)

ggboxplot(div_df_melt, x = "AB", y = "value",
fill = "AB",
palette = "lancet",
legend= "right",
facet.by = "variable",
scales = "free",
xlab = "Antibiotics used",
title = "Alpha diversity metrics by antibiotic usage",
outlier.shape = NA) +
rremove("x.text") + stat_compare_means() + geom_jitter(size = 0.7, alpha = 0.9)
```

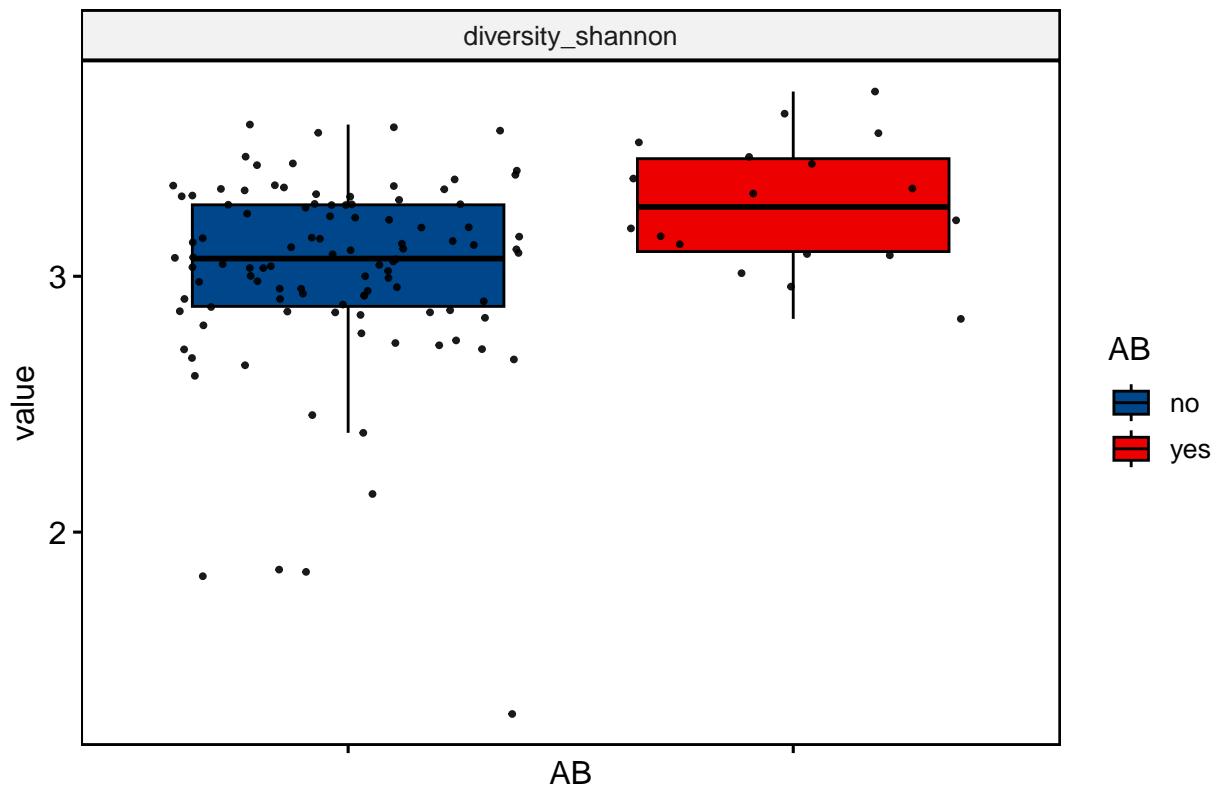
Alpha diversity metrics by antibiotic usage



```
div.df2 <- div.df[, c("AB", "diversity_shannon")]
div_df_melt <- reshape2::melt(div.df2)

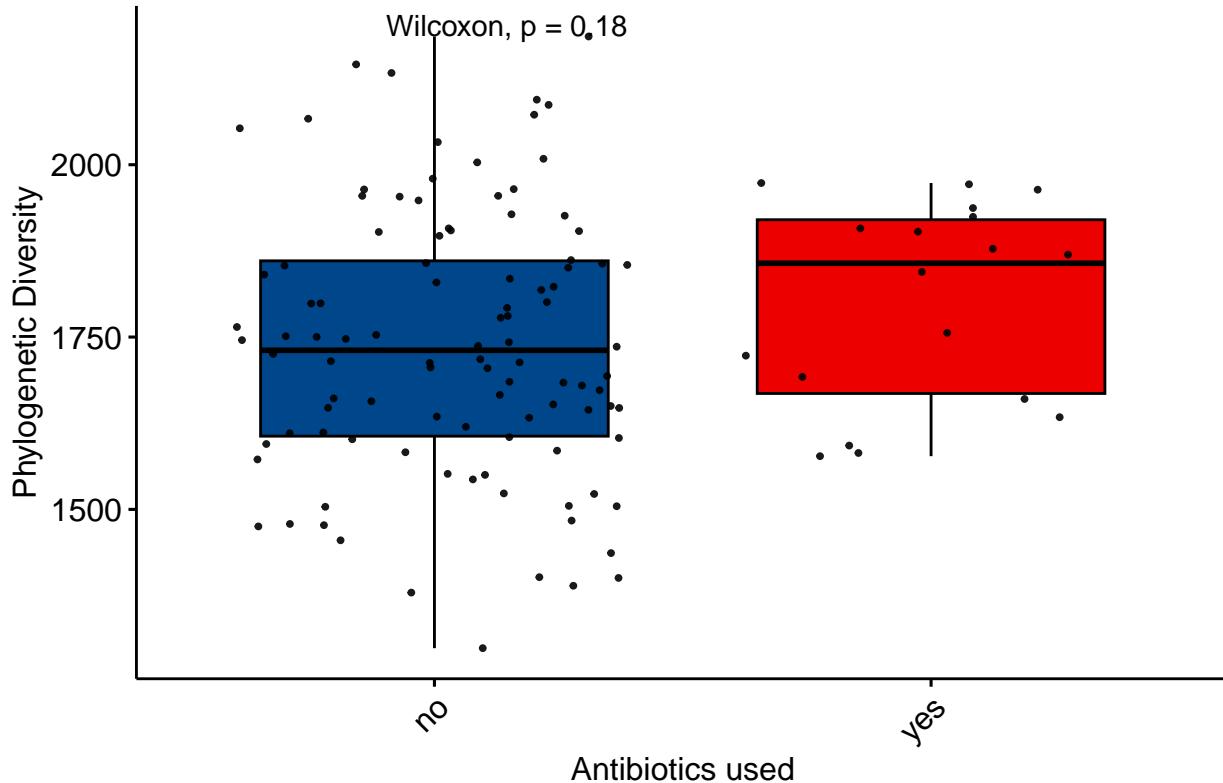
ggboxplot(div_df_melt, x = "AB", y = "value",
          fill = "AB",
          palette = "lancet",
          legend= "right",
          facet.by = "variable",
          scales = "free",
          title = "Shannon diversity by antibiotic usage",
          outlier.shape = NA) +
  rremove("x.text") + geom_jitter(size = 0.7, alpha = 0.9)
```

Shannon diversity by antibiotic usage



```
ggboxplot(hmp.meta,
  x = "AB",
  y = "Phylogenetic_Diversity",
  fill = "AB",
  palette = "lancet",
  order = c("no","yes"),
  ylab = "Phylogenetic Diversity",
  xlab = "Antibiotics used",
  legend = "right",
  title = "Phylogenetic diversity by antibiotic usage",
  outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) +
  stat_compare_means() + geom_jitter(size = 0.7, alpha = 0.9)
```

Phylogenetic diversity by antibiotic usage



```
# based on stable and age

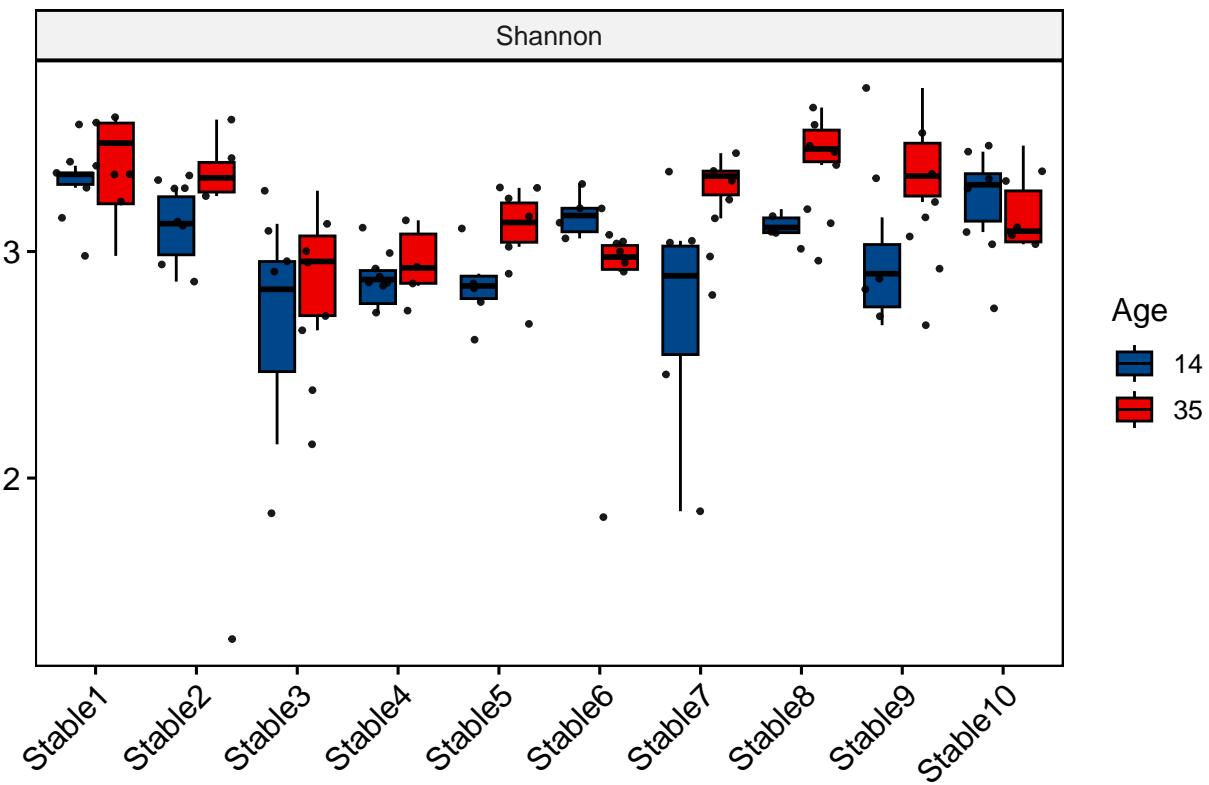
div.df2 <- div.df[, c("Stables", "Age", "diversity_shannon")]
colnames(div.df2) <- c("Stable", "Age", "Shannon")

div_df_melt <- reshape2::melt(div.df2)

lev = c("Stable1", "Stable2", "Stable3", "Stable4", "Stable5", "Stable6", "Stable7", "Stable8", "Stable9", "Stab

ggboxplot(div_df_melt, x = "Stable", y = "value",
  fill = "Age",
  palette = "lancet",
  legend= "right",
  facet.by = "variable",
  scales = "free",
  order = lev,
  title = "Shannon diversity by stable and age",
  xlab = FALSE,
  ylab = FALSE,
  outlier.shape = NA) + rotate_x_text() +
  theme(axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) + geom_jitter(size = 0.7, alpha = 0.5)
```

Shannon diversity by stable and age

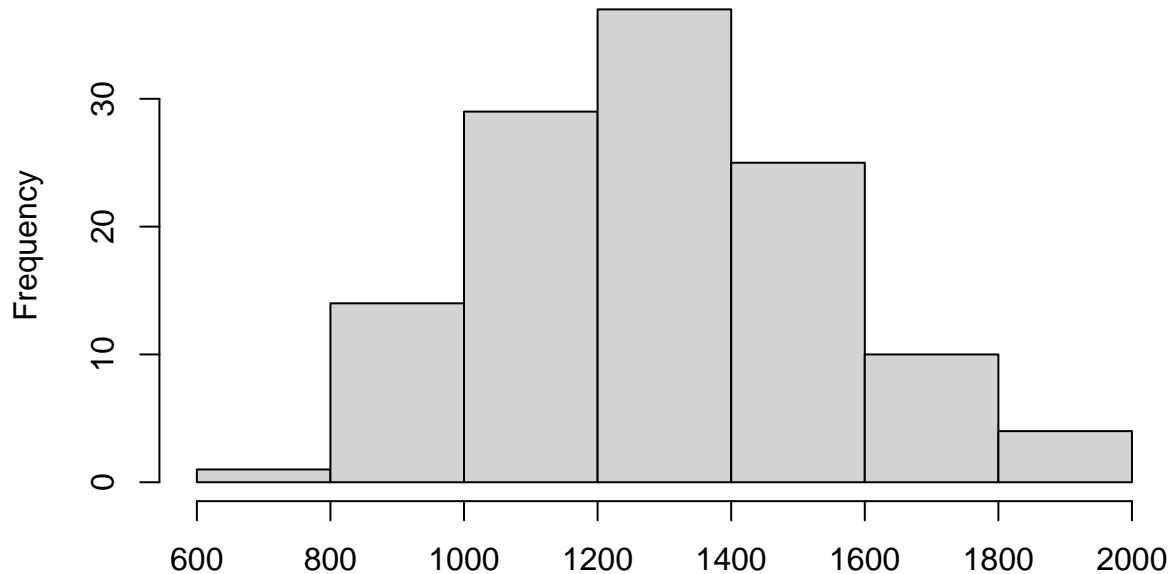


Looking at significance

Checking for normality

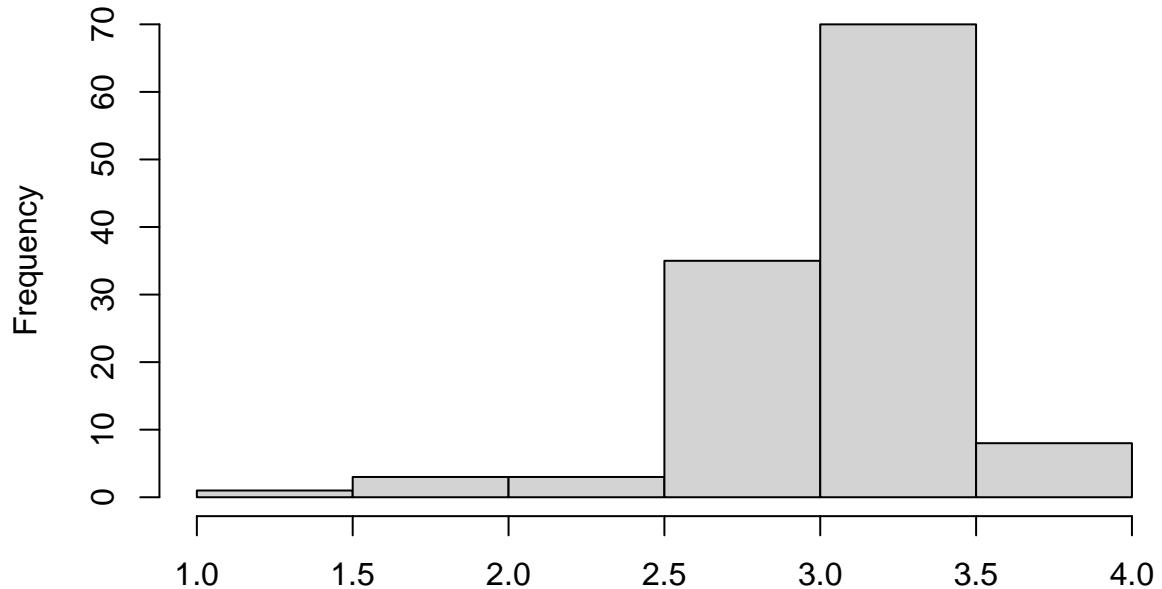
```
hist(lib.div$chao1, main="Observed richness", xlab="")
```

Observed richness



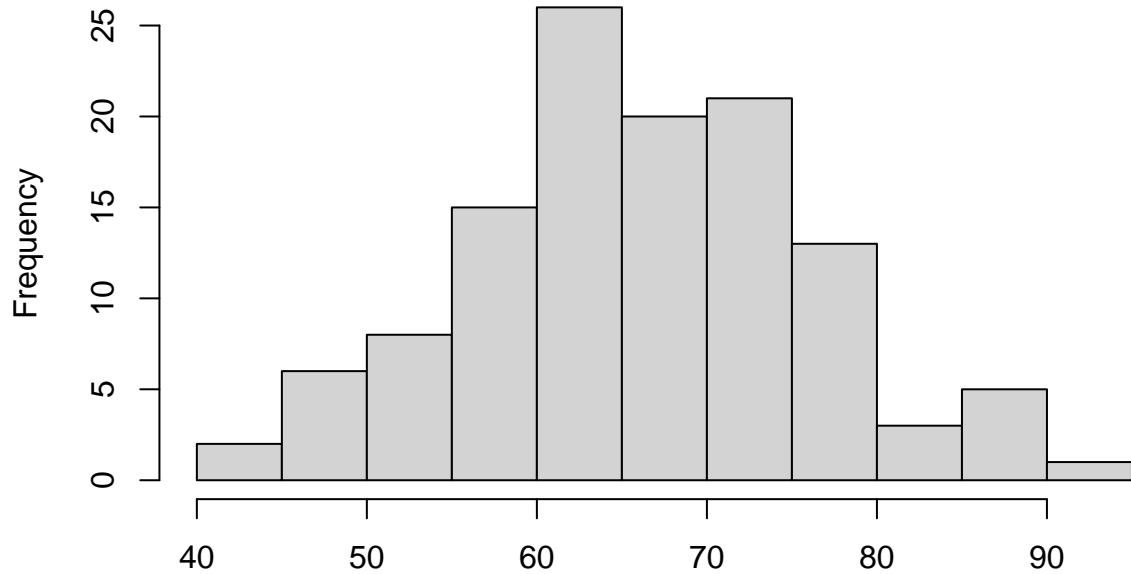
```
hist(lib.div$diversity_shannon, main="Shannon diversity", xlab="")
```

Shannon diversity



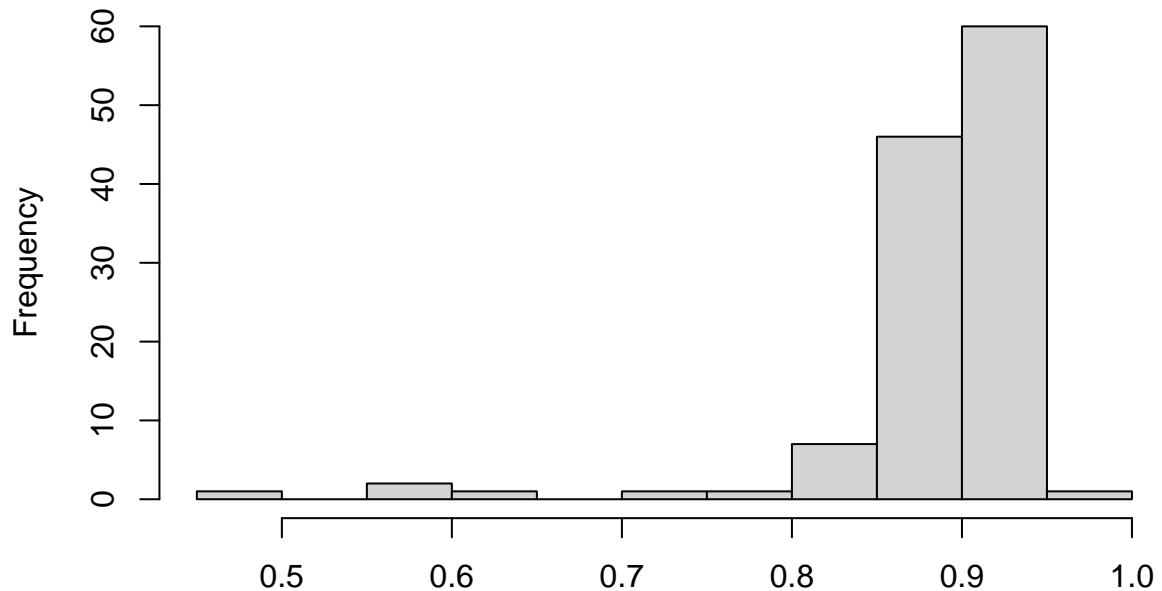
```
hist(lib.div$diversity_fisher, main="Fisher diversity", xlab="")
```

Fisher diversity



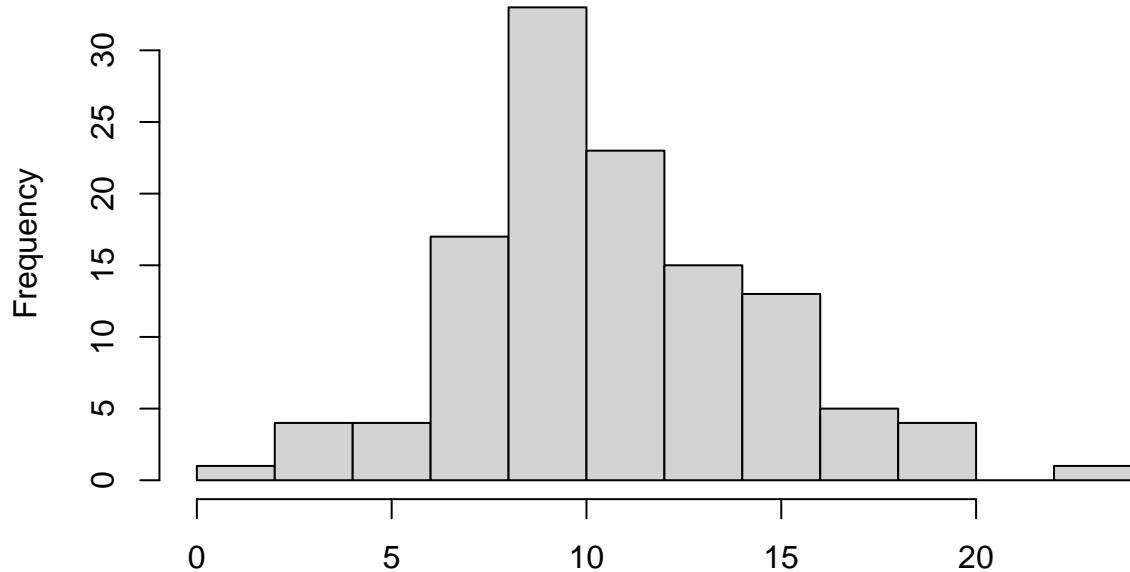
```
hist(lib.div$diversity_gini_simpson, main="Gini-Simpson diversity", xlab="")
```

Gini–Simpson diversity



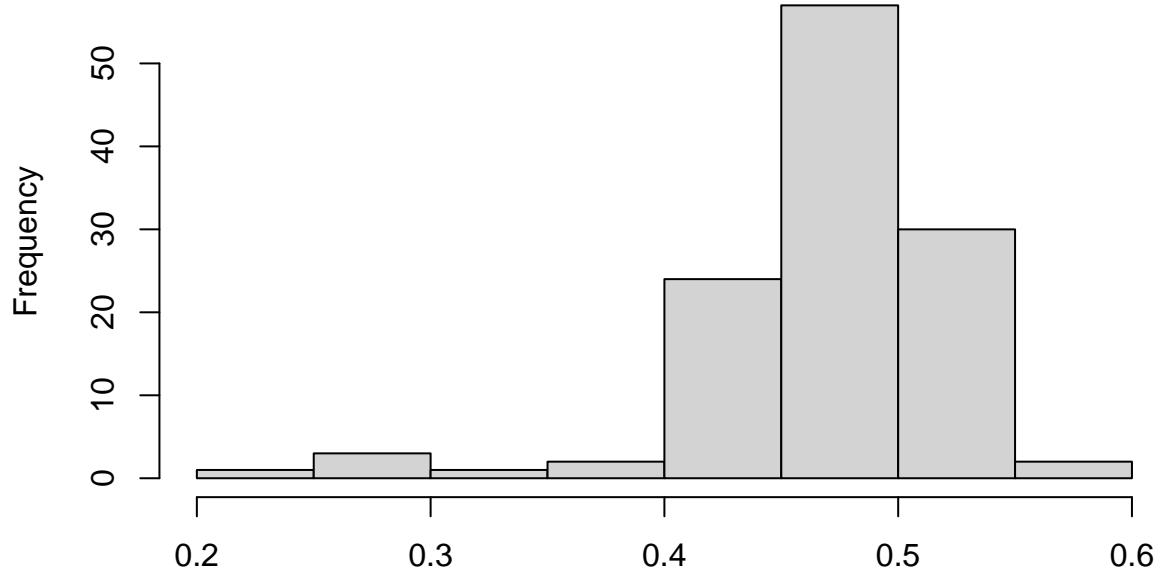
```
hist(lib.div$diversity_inverse_simpson, main="Inverse Simpson evenness", xlab="")
```

Inverse Simpson evenness



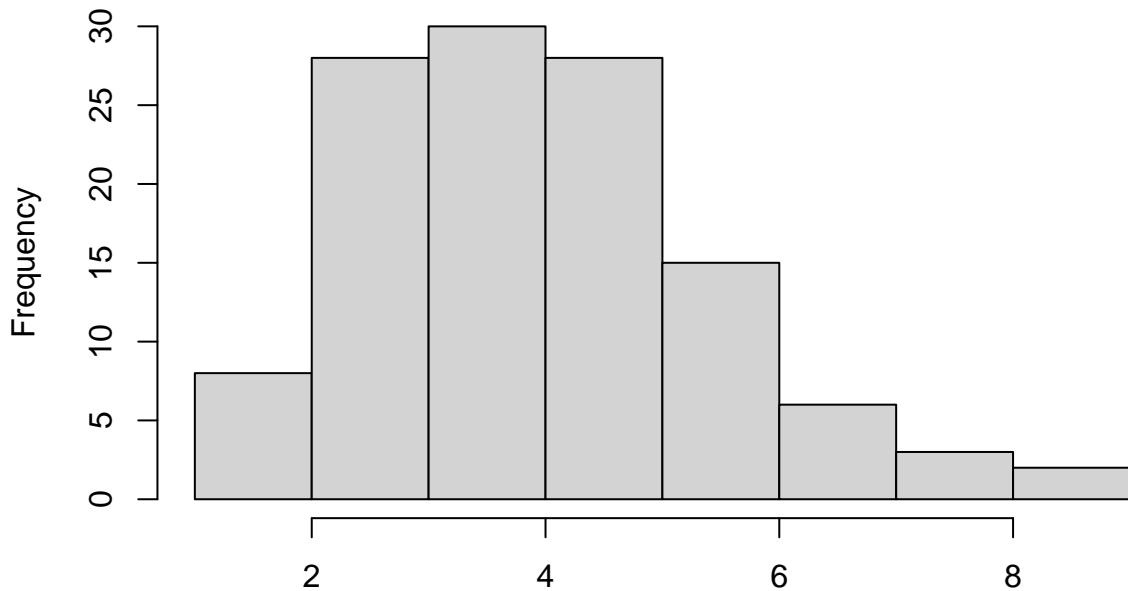
```
hist(lib.div$evenness_pielou, main="Pielou evenness", xlab="")
```

Pielou evenness



```
hist(lib.div$diversity_coverage, main="Coverage diversity", xlab="")
```

Coverage diversity



```
# If data is normally distributed we can use ANOVA / t-tests, if not we will use Kruskal-Wallis tests  
# In this case, the data seems roughly normally distributed for some metrics, we can use Shapiro-Wilk t  
shapiro.test(lib.div$chao1) # test deems it normally distributed p>0,05
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: lib.div$chao1  
## W = 0.9823, p-value = 0.1163  
  
shapiro.test(lib.div$diversity_shannon) # test deems this measure not normally distributed p<0,05  
  
##  
## Shapiro-Wilk normality test  
##  
## data: lib.div$diversity_shannon  
## W = 0.86849, p-value = 6.494e-09  
  
shapiro.test(lib.div$diversity_fisher) # test deems this measure normally distributed p>0,05  
  
##  
## Shapiro-Wilk normality test  
##  
## data: lib.div$diversity_fisher  
## W = 0.99471, p-value = 0.9349
```

```

shapiro.test(lib.div$diversity_gini_simpson) # test deems this measure not normally distributed p<0,05

## 
## Shapiro-Wilk normality test
## 
## data: lib.div$diversity_gini_simpson
## W = 0.61577, p-value = 3.025e-16

shapiro.test(lib.div$diversity_inverse_simpson) # test deems this measure normally distributed p>0,05

## 
## Shapiro-Wilk normality test
## 
## data: lib.div$diversity_inverse_simpson
## W = 0.9815, p-value = 0.09819

shapiro.test(lib.div$evenness_pielou) # test deems this measure not normally distributed p<0,05

## 
## Shapiro-Wilk normality test
## 
## data: lib.div$evenness_pielou
## W = 0.83508, p-value = 2.943e-10

shapiro.test(lib.div$diversity_coverage) # test deems this measure not normally distributed p<0,05

## 
## Shapiro-Wilk normality test
## 
## data: lib.div$diversity_coverage
## W = 0.94699, p-value = 0.0001308

shapiro.test(lib.div$Phylogenetic_Diversity) # test deems this measure normally distributed p>0,05

## 
## Shapiro-Wilk normality test
## 
## data: lib.div$Phylogenetic_Diversity
## W = 0.9921, p-value = 0.7302

# Based on shapiro-wilk tests we will assume normality for some measures
# The variables that we are interested in are the Age, which Farm the samples are from, and whether ant

# We will run ANOVAs for the normally distributed variables

# Age

# Normally distributed with only 2 levels, so we can use t-tests :

t.test(lib.div$chao1 ~ sample_data(subsetMG)$Age) # significant

```

```

##  

## Welch Two Sample t-test  

##  

## data: lib.div$chao1 by sample_data(subsetMG)$Age  

## t = -3.1449, df = 118, p-value = 0.002102  

## alternative hypothesis: true difference in means between group 14 and group 35 is not equal to 0  

## 95 percent confidence interval:  

## -228.4688 -51.9177  

## sample estimates:  

## mean in group 14 mean in group 35  

## 1225.059 1365.252

t.test(lib.div$diversity_fisher ~ sample_data(subsetMG)$Age) # significant

##  

## Welch Two Sample t-test  

##  

## data: lib.div$diversity_fisher by sample_data(subsetMG)$Age  

## t = -6.1818, df = 115.32, p-value = 9.866e-09  

## alternative hypothesis: true difference in means between group 14 and group 35 is not equal to 0  

## 95 percent confidence interval:  

## -13.154462 -6.770287  

## sample estimates:  

## mean in group 14 mean in group 35  

## 61.21639 71.17876

t.test(lib.div$diversity_inverse_simpson ~ sample_data(subsetMG)$Age) # significant

##  

## Welch Two Sample t-test  

##  

## data: lib.div$diversity_inverse_simpson by sample_data(subsetMG)$Age  

## t = -2.5431, df = 107.24, p-value = 0.01241  

## alternative hypothesis: true difference in means between group 14 and group 35 is not equal to 0  

## 95 percent confidence interval:  

## -3.0692244 -0.3803386  

## sample estimates:  

## mean in group 14 mean in group 35  

## 9.807183 11.531964

t.test(lib.div$Phylogenetic_Diversity ~ sample_data(subsetMG)$Age) # significant

##  

## Welch Two Sample t-test  

##  

## data: lib.div$Phylogenetic_Diversity by sample_data(subsetMG)$Age  

## t = -5.6686, df = 115.06, p-value = 1.084e-07  

## alternative hypothesis: true difference in means between group 14 and group 35 is not equal to 0  

## 95 percent confidence interval:  

## -233.7142 -112.6742  

## sample estimates:  

## mean in group 14 mean in group 35  

## 1662.490 1835.684

```

```

# Non-normally distributed

wilcox.test(lib.div$diversity_shannon ~ sample_data(subsetMG)$Age) # shannon diversity seems to significant

## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$diversity_shannon by sample_data(subsetMG)$Age
## W = 1230, p-value = 0.002798
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(lib.div$diversity_gini_simpson ~ sample_data(subsetMG)$Age) # significant

## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$diversity_gini_simpson by sample_data(subsetMG)$Age
## W = 1362, p-value = 0.02166
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(lib.div$evenness_pielou ~ sample_data(subsetMG)$Age) # not significant

## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$evenness_pielou by sample_data(subsetMG)$Age
## W = 1464, p-value = 0.07825
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(lib.div$diversity_coverage ~ sample_data(subsetMG)$Age) # significant

## 
## Wilcoxon rank sum test with continuity correction
## 
## data: lib.div$diversity_coverage by sample_data(subsetMG)$Age
## W = 1357, p-value = 0.01758
## alternative hypothesis: true location shift is not equal to 0

# For age, the groups seems significantly different in all metrics except simpson evenness.

# Antibiotics

t.test(lib.div$observed ~ sample_data(subsetMG)$AB) # not significant

## 
## Welch Two Sample t-test
## 
## data: lib.div$observed by sample_data(subsetMG)$AB
## t = -1.3665, df = 27.68, p-value = 0.1828
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0

```

```

## 95 percent confidence interval:
## -79.54352 15.90299
## sample estimates:
## mean in group no mean in group yes
## 683.5686 715.3889

t.test(lib.div$diversity_fisher ~ sample_data(subsetMG)$AB) # not significant

## Welch Two Sample t-test
## data: lib.div$diversity_fisher by sample_data(subsetMG)$AB
## t = -1.4372, df = 30.898, p-value = 0.1607
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0
## 95 percent confidence interval:
## -7.003503 1.213738
## sample estimates:
## mean in group no mean in group yes
## 65.76334 68.65823

t.test(lib.div$diversity_inverse_simpson ~ sample_data(subsetMG)$AB) # significant

## Welch Two Sample t-test
## data: lib.div$diversity_inverse_simpson by sample_data(subsetMG)$AB
## t = -2.9341, df = 21.663, p-value = 0.00776
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0
## 95 percent confidence interval:
## -5.2046485 -0.8917236
## sample estimates:
## mean in group no mean in group yes
## 10.21235 13.26053

t.test(lib.div$Phylogenetic_Diversity ~ sample_data(subsetMG)$AB) # not significant

## Welch Two Sample t-test
## data: lib.div$Phylogenetic_Diversity by sample_data(subsetMG)$AB
## t = -1.5033, df = 28.718, p-value = 0.1437
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0
## 95 percent confidence interval:
## -139.87692 21.38891
## sample estimates:
## mean in group no mean in group yes
## 1740.200 1799.444

# Non-normally distributed

wilcox.test(lib.div$diversity_shannon ~ sample_data(subsetMG)$AB) # shannon diversity does seem to sign

```

```

##  

## Wilcoxon rank sum test with continuity correction  

##  

## data: lib.div$diversity_shannon by sample_data(subsetMG)$AB  

## W = 511, p-value = 0.002812  

## alternative hypothesis: true location shift is not equal to 0  

wilcox.test(lib.div$diversity_gini_simpson ~ sample_data(subsetMG)$AB) # significant  

##  

## Wilcoxon rank sum test with continuity correction  

##  

## data: lib.div$diversity_gini_simpson by sample_data(subsetMG)$AB  

## W = 532, p-value = 0.004608  

## alternative hypothesis: true location shift is not equal to 0  

wilcox.test(lib.div$evenness_pielou ~ sample_data(subsetMG)$AB) # significant  

##  

## Wilcoxon rank sum test with continuity correction  

##  

## data: lib.div$evenness_pielou by sample_data(subsetMG)$AB  

## W = 533, p-value = 0.004715  

## alternative hypothesis: true location shift is not equal to 0  

wilcox.test(lib.div$diversity_coverage ~ sample_data(subsetMG)$AB) # significant  

##  

## Wilcoxon rank sum test with continuity correction  

##  

## data: lib.div$diversity_coverage by sample_data(subsetMG)$AB  

## W = 521, p-value = 0.002891  

## alternative hypothesis: true location shift is not equal to 0  

# AB does seems to significantly differ in their alpha diversities for some metrics  

# Farm has more than 2 levels, so we will use ANOVAs for normally distributed metrics  

aov.chao1.farm = aov(lib.div$chao1 ~ sample_data(subsetMG)$Farm2)  

summary(aov.chao1.farm) # not significant  

##  

## Df Sum Sq Mean Sq F value Pr(>F)  

## sample_data(subsetMG)$Farm2 3 318620 106207 1.686 0.174  

## Residuals 116 7305522 62979  

TukeyHSD(aov.chao1.farm)  

## Tukey multiple comparisons of means  

## 95% family-wise confidence level  

##
```

```

## Fit: aov(formula = lib.div$chao1 ~ sample_data(subsetMG)$Farm2)
##
## $`sample_data(subsetMG)$Farm2`
##   diff      lwr      upr     p adj
## Farm2-Farm1  35.75578 -127.78341 199.29497 0.9408364
## Farm3-Farm1 102.29156  -86.54724 291.13035 0.4944681
## Farm4-Farm1 -55.33683 -244.17563 133.50196 0.8705219
## Farm3-Farm2  66.53578  -97.00341 230.07497 0.7141256
## Farm4-Farm2 -91.09261 -254.63180  72.44658 0.4698852
## Farm4-Farm3 -157.62839 -346.46718  31.21040 0.1361106

aov.fisher.farm = aov(lib.div$diversity_fisher ~ sample_data(subsetMG)$Farm2)
summary(aov.fisher.farm)

##                               Df Sum Sq Mean Sq F value    Pr(>F)
## sample_data(subsetMG)$Farm2  3   1192   397.3   4.197 0.00737 ***
## Residuals                  116  10979   94.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(aov.fisher.farm) # only not significant between 1 and 3, 2 and 3 and 4 and 2

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lib.div$diversity_fisher ~ sample_data(subsetMG)$Farm2)
##
## $`sample_data(subsetMG)$Farm2`
##   diff      lwr      upr     p adj
## Farm2-Farm1 -5.8453541 -12.185302 0.49459396 0.0820817
## Farm3-Farm1 -0.6703669  -7.991108 6.65037453 0.9951902
## Farm4-Farm1 -7.9730716 -15.293813 -0.65233010 0.0270581
## Farm3-Farm2  5.1749872  -1.164961 11.51493528 0.1504613
## Farm4-Farm2 -2.1277174  -8.467666  4.21223066 0.8178769
## Farm4-Farm3 -7.3027046 -14.623446  0.01803685 0.0508199

aov.invsimp.farm = aov(lib.div$diversity_inverse_simpson ~ sample_data(subsetMG)$Farm2)
summary(aov.invsimp.farm)

##                               Df Sum Sq Mean Sq F value    Pr(>F)
## sample_data(subsetMG)$Farm2  3   313.9   104.6   8.646 3.16e-05 ***
## Residuals                  116  1403.7    12.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(aov.invsimp.farm) # same as above

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lib.div$diversity_inverse_simpson ~ sample_data(subsetMG)$Farm2)

```

```

## 
## $`sample_data(subsetMG)$Farm2` 
##      diff      lwr      upr     p adj
## Farm2-Farm1 -4.1224050 -6.38928706 -1.8555229 0.0000359
## Farm3-Farm1 -1.3736284 -3.99119838  1.2439415 0.5219668
## Farm4-Farm1 -1.7609692 -4.37853914  0.8566007 0.3010859
## Farm3-Farm2  2.7487766  0.48189449  5.0156586 0.0106724
## Farm4-Farm2  2.3614358  0.09455373  4.6283179 0.0377428
## Farm4-Farm3 -0.3873408 -3.00491070  2.2302292 0.9804004

# Non-normally distributed

kruskal.test(lib.div$diversity_shannon ~ sample_data(subsetMG)$Farm2) # shannon diversity seems to sign

## 
## Kruskal-Wallis rank sum test
## 
## data: lib.div$diversity_shannon by sample_data(subsetMG)$Farm2
## Kruskal-Wallis chi-squared = 28.692, df = 3, p-value = 2.599e-06

pairwise.wilcox.test(lib.div$diversity_shannon, sample_data(subsetMG)$Farm2, p.adjust.method="fdr") # d

## 
## Pairwise comparisons using Wilcoxon rank sum exact test
## 
## data: lib.div$diversity_shannon and sample_data(subsetMG)$Farm2
## 
##      Farm1     Farm2     Farm3
## Farm2 1.4e-06 - 
## Farm3 0.2958  0.0012 - 
## Farm4 0.1658  0.0040  0.7670
## 
## P value adjustment method: fdr

kruskal.test(lib.div$diversity_gini_simpson ~ sample_data(subsetMG)$Farm2) # significant

## 
## Kruskal-Wallis rank sum test
## 
## data: lib.div$diversity_gini_simpson by sample_data(subsetMG)$Farm2
## Kruskal-Wallis chi-squared = 24.217, df = 3, p-value = 2.251e-05

pairwise.wilcox.test(lib.div$diversity_gini_simpson, sample_data(subsetMG)$Farm2, p.adjust.method="fdr")

## 
## Pairwise comparisons using Wilcoxon rank sum exact test
## 
## data: lib.div$diversity_gini_simpson and sample_data(subsetMG)$Farm2
## 
##      Farm1     Farm2     Farm3
## Farm2 5.9e-06 - 

```

```

## Farm3 0.2760 0.0013 -
## Farm4 0.1277 0.1191 0.4188
##
## P value adjustment method: fdr

kruskal.test(lib.div$evenness_pielou ~ sample_data(subsetMG)$Farm2) # significant

##
## Kruskal-Wallis rank sum test
##
## data: lib.div$evenness_pielou by sample_data(subsetMG)$Farm2
## Kruskal-Wallis chi-squared = 24.117, df = 3, p-value = 2.361e-05

pairwise.wilcox.test(lib.div$evenness_pielou, sample_data(subsetMG)$Farm2, p.adjust.method="fdr") # dsan

##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$evenness_pielou and sample_data(subsetMG)$Farm2
##
##      Farm1     Farm2     Farm3
## Farm2 0.00015 - -
## Farm3 0.25779 0.00257 -
## Farm4 0.42169 0.00151 0.66038
##
## P value adjustment method: fdr

# agent also has more than 2 levels, so we will use ANOVAs for normally distributed metrics

aov.chao1.agent = aov(lib.div$chao1 ~ sample_data(subsetMG)$Cox)
summary(aov.chao1.agent)

##
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## sample_data(subsetMG)$Cox 3  840841  280280   4.793 0.00348 ***
## Residuals                 116 6783300   58477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

TukeyHSD(aov.chao1.agent) # only significant between sacox and none

##
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lib.div$chao1 ~ sample_data(subsetMG)$Cox)
##
## $`sample_data(subsetMG)$Cox`
##          diff      lwr      upr      p adj
## Monteban-Maxiban  49.26113 -148.26828 246.79053 0.9153336
## None-Maxiban     116.75992 -12.59085 246.11068 0.0922262
## Sacox-Maxiban   -160.76117 -364.63580 43.11346 0.1740828
## None-Monteban    67.49879 -129.58430 264.58188 0.8086228
## Sacox-Monteban  -210.02230 -462.36133 42.31674 0.1379306
## Sacox-None       -277.52109 -480.96332 -74.07885 0.0030310

```

```
aov.fisher.agent = aov(lib.div$diversity_fisher ~ sample_data(subsetMG)$Cox)
summary(aov.fisher.agent)
```

```
##                                Df Sum Sq Mean Sq F value    Pr(>F)
## sample_data(subsetMG)$Cox     3   2974   991.2    12.5 3.85e-07 ***
## Residuals                     116   9198    79.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aov.fisher.agent) # only not significant maxiban & sacox and none & monteban
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lib.div$diversity_fisher ~ sample_data(subsetMG)$Cox)
##
## $`sample_data(subsetMG)$Cox`
##          diff      lwr      upr      p adj
## Monteban-Maxiban 11.356193  4.082611 18.629775 0.0004951
## None-Maxiban     8.113833  3.350778 12.876888 0.0001201
## Sacox-Maxiban   -3.677067 -11.184298  3.830164 0.5795288
## None-Monteban   -3.242360 -10.499507  4.014788 0.6502941
## Sacox-Monteban  -15.033260 -24.325086 -5.741435 0.0002846
## Sacox-None       -11.790900 -19.282209 -4.299591 0.0004380
```

```
# Non-normally distributed
```

```
kruskal.test(lib.div$diversity_shannon ~ sample_data(subsetMG)$Cox) # shannon diversity seems to signif
```

```
##
## Kruskal-Wallis rank sum test
##
## data: lib.div$diversity_shannon by sample_data(subsetMG)$Cox
## Kruskal-Wallis chi-squared = 13.227, df = 3, p-value = 0.00417
```

```
pairwise.wilcox.test(lib.div$diversity_shannon, sample_data(subsetMG)$Cox, p.adjust.method="fdr") # dif
```

```
##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$diversity_shannon and sample_data(subsetMG)$Cox
##
##          Maxiban Monteban None
## Monteban 0.0033  -    -
## None      0.0642  0.0642  -
## Sacox     0.5831  0.0642  0.5116
##
## P value adjustment method: fdr
```

```

kruskal.test(lib.div$diversity_gini_simpson ~ sample_data(subsetMG)$Cox) # significant

##
## Kruskal-Wallis rank sum test
##
## data: lib.div$diversity_gini_simpson by sample_data(subsetMG)$Cox
## Kruskal-Wallis chi-squared = 9.8825, df = 3, p-value = 0.01959

pairwise.wilcox.test(lib.div$diversity_gini_simpson, sample_data(subsetMG)$Cox, p.adjust.method="fdr") ...

##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$diversity_gini_simpson and sample_data(subsetMG)$Cox
##
##          Maxiban Monteban None
## Monteban 0.020   -      -
## None     0.207   0.069   -
## Sacox    0.802   0.069   0.371
##
## P value adjustment method: fdr

kruskal.test(lib.div$diversity_inverse_simpson ~ sample_data(subsetMG)$Cox) # not significant

##
## Kruskal-Wallis rank sum test
##
## data: lib.div$diversity_inverse_simpson by sample_data(subsetMG)$Cox
## Kruskal-Wallis chi-squared = 9.8825, df = 3, p-value = 0.01959

pairwise.wilcox.test(lib.div$diversity_inverse_simpson, sample_data(subsetMG)$Cox, p.adjust.method="fdr") ...

##
## Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$diversity_inverse_simpson and sample_data(subsetMG)$Cox
##
##          Maxiban Monteban None
## Monteban 0.020   -      -
## None     0.207   0.069   -
## Sacox    0.802   0.069   0.371
##
## P value adjustment method: fdr

kruskal.test(lib.div$evenness_pielou ~ sample_data(subsetMG)$Cox) # not significant

##
## Kruskal-Wallis rank sum test
##
## data: lib.div$evenness_pielou by sample_data(subsetMG)$Cox
## Kruskal-Wallis chi-squared = 6.7791, df = 3, p-value = 0.07928

```

```

pairwise.wilcox.test(lib.div$evenness_pielou, sample_data(subsetMG)$Cox, p.adjust.method="fdr") # not shown

##
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$evenness_pielou and sample_data(subsetMG)$Cox
##
##          Maxiban Monteban None
## Monteban 0.094   -      -
## None      0.305   0.258   -
## Sacox     0.476   0.308   0.949
##
## P value adjustment method: fdr

# Different ordination methods based on BC dissimilarity (runs very slowly for MG)

dist = "bray"
ord_meths = c("DCA", "CCA", "RDA", "NMDS", "MDS", "PCoA", "DPCoA")
plist = llply(as.list(ord_meths), function(i, physeq, dist){
  ordi = ordinate(subsetMG, method=i, distance=dist)
  plot_ordination(subsetMG, ordi, "samples", color="Age", shape = "AB")
}, subsetMG, dist)

##
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2345122
## Run 1 stress 0.2435759
## Run 2 stress 0.2403301
## Run 3 stress 0.2465643
## Run 4 stress 0.2347155
## ... Procrustes: rmse 0.02288774 max resid 0.1640683
## Run 5 stress 0.2446542
## Run 6 stress 0.2476091
## Run 7 stress 0.2379555
## Run 8 stress 0.2412041
## Run 9 stress 0.2428403
## Run 10 stress 0.2424341
## Run 11 stress 0.2378768
## Run 12 stress 0.2515049
## Run 13 stress 0.24204
## Run 14 stress 0.2465702
## Run 15 stress 0.2440242
## Run 16 stress 0.2413374
## Run 17 stress 0.258972
## Run 18 stress 0.2354474
## Run 19 stress 0.2404408
## Run 20 stress 0.247721
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      20: stress ratio > sratmax

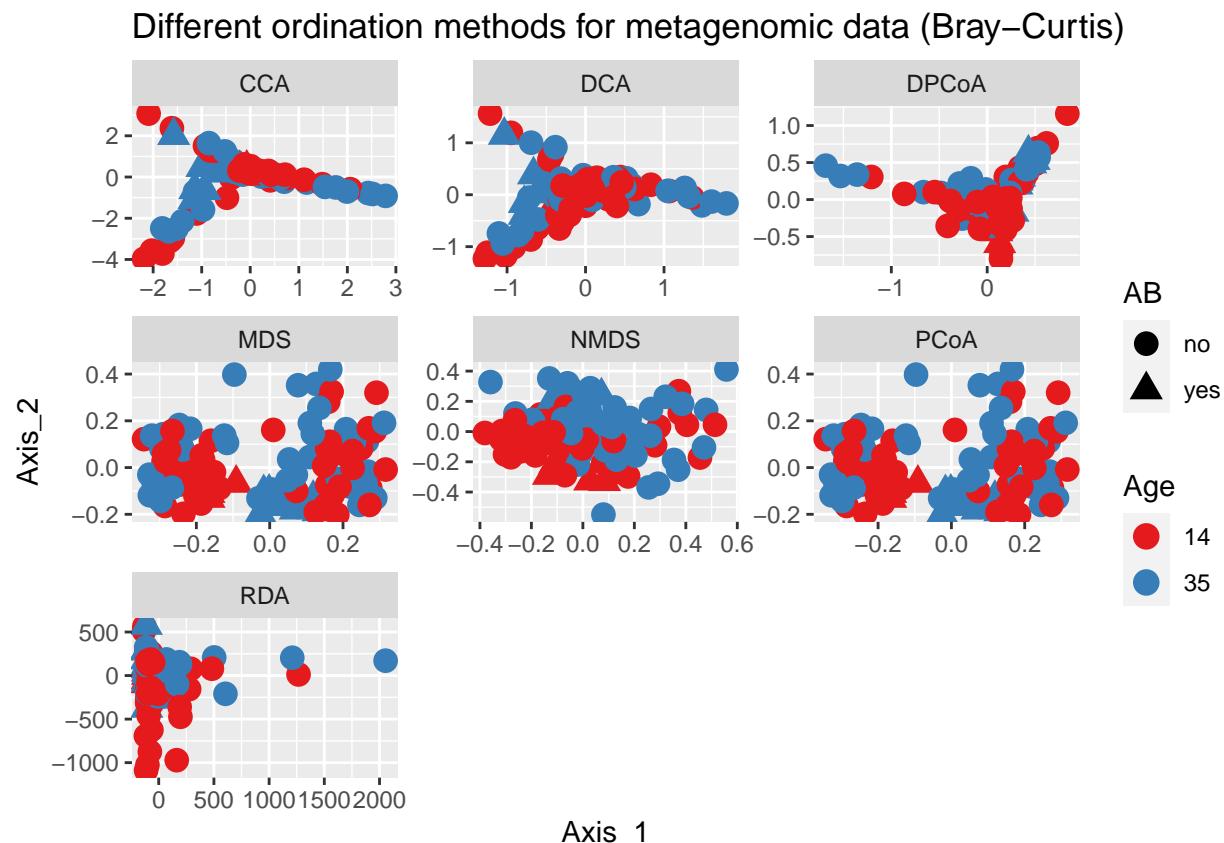
names(plist) <- ord_meths

```

```

pdatatable = ldply(plist, function(x){
  df = x$data[, 1:2]
  colnames(df) = c("Axis_1", "Axis_2")
  return(cbind(df, x$data))
})
names(pdatatable)[1] = "method"
ggplot(pdatatable, aes(Axis_1, Axis_2, color=Age, shape=AB)) +
  geom_point(size=4) +
  facet_wrap(~method, scales="free") +
  scale_fill_brewer(type="qual", palette="Set1") +
  scale_colour_brewer(type="qual", palette="Set1") +
  ggtitle("Different ordination methods for metagenomic data (Bray-Curtis)")

```



```
### PCoAs for different methods
```

```

# functionize plotting pcoa
plot_pcoa_ordination <- function(data, pcoa, var, title) {
  p <- plot_ordination(data, pcoa, color = var, shape = "AB") +
    geom_point(size = 3) +
    labs(title = title, color = var, shape = "Antibiotics used")

  return(p)
}

pcoa_bc = ordinate(subsetMG, "PCoA", "bray")
pcoa_unifrac = ordinate(subsetMG, "PCoA", "unifrac")

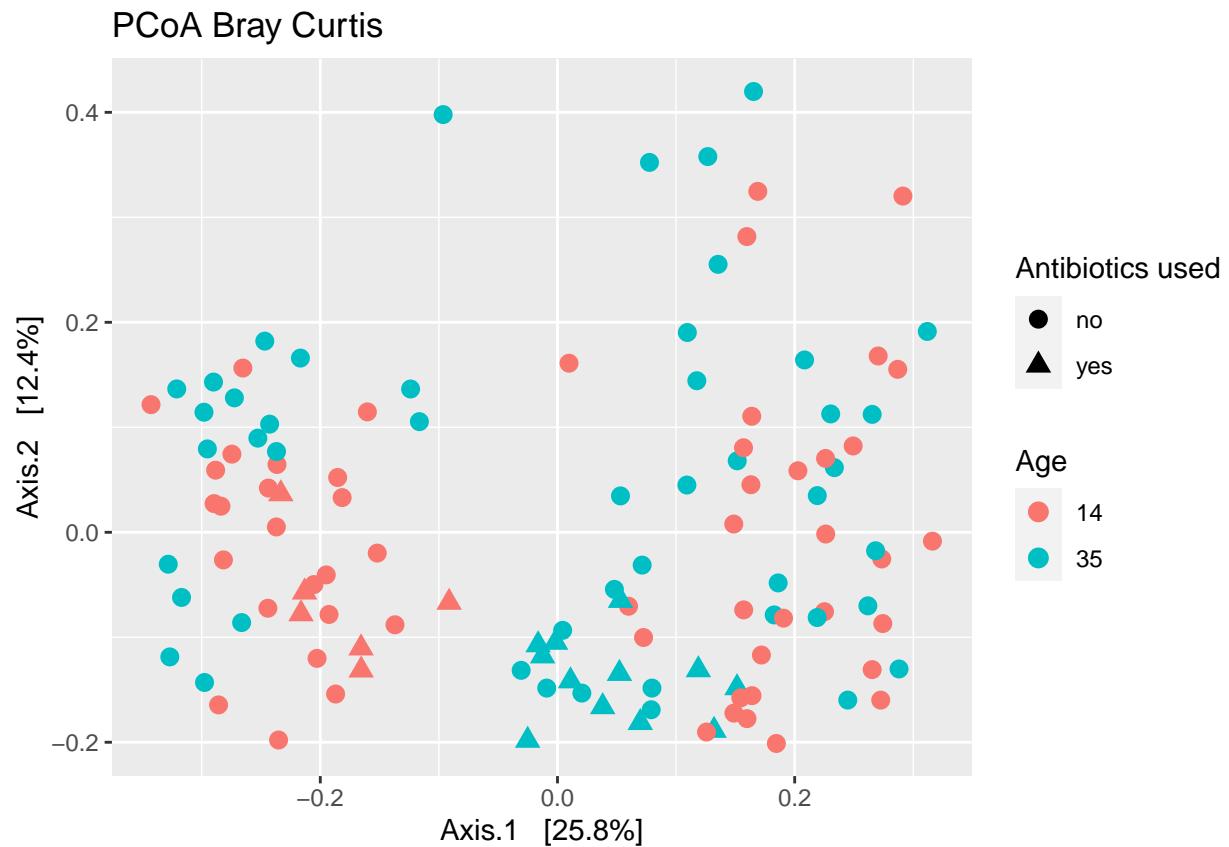
```

```

pcoa_wunifrac = ordinate(subsetMG, "PCoA", "wunifrac")
pcoa_jsd = ordinate(subsetMG, "PCoA", "jsd")
pcoa_jaccard = ordinate(subsetMG, "PCoA", "jaccard", binary=TRUE)

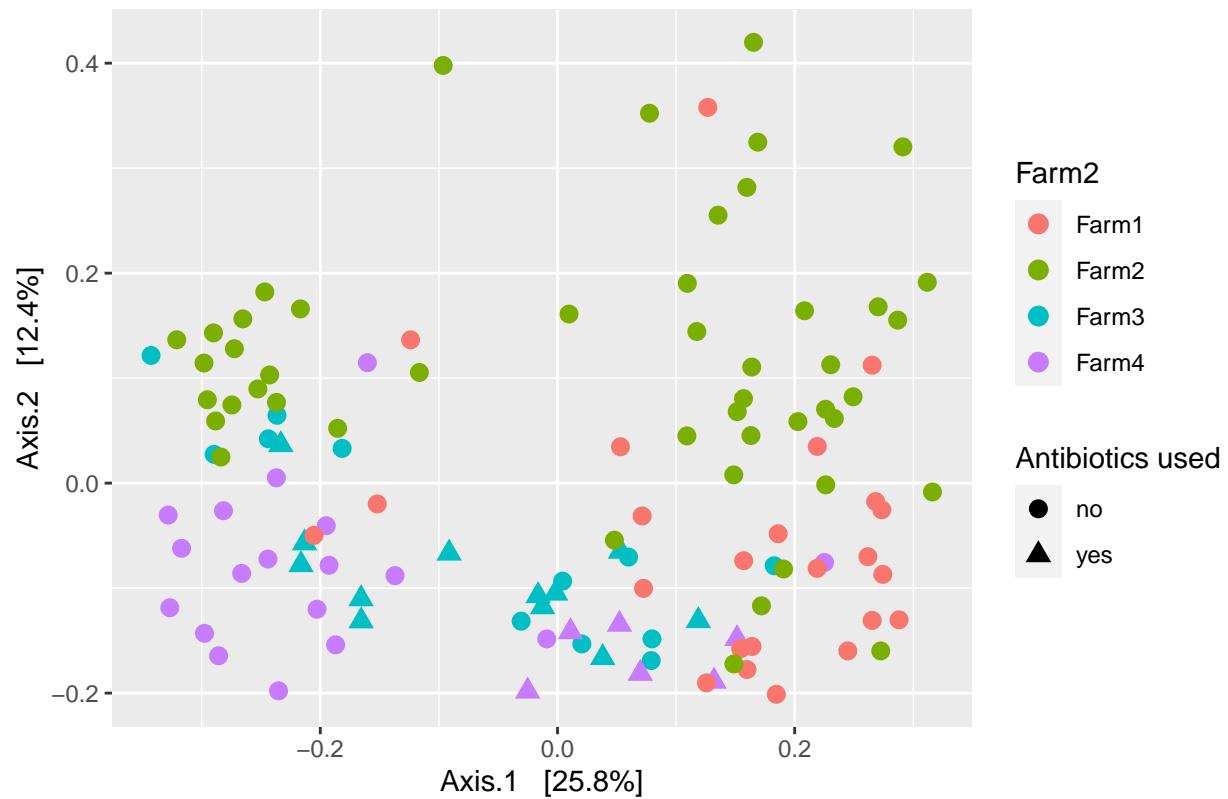
```

```
plot_pcoa_ordination(subsetMG, pcoa_bc, "Age", "PCoA Bray Curtis")
```



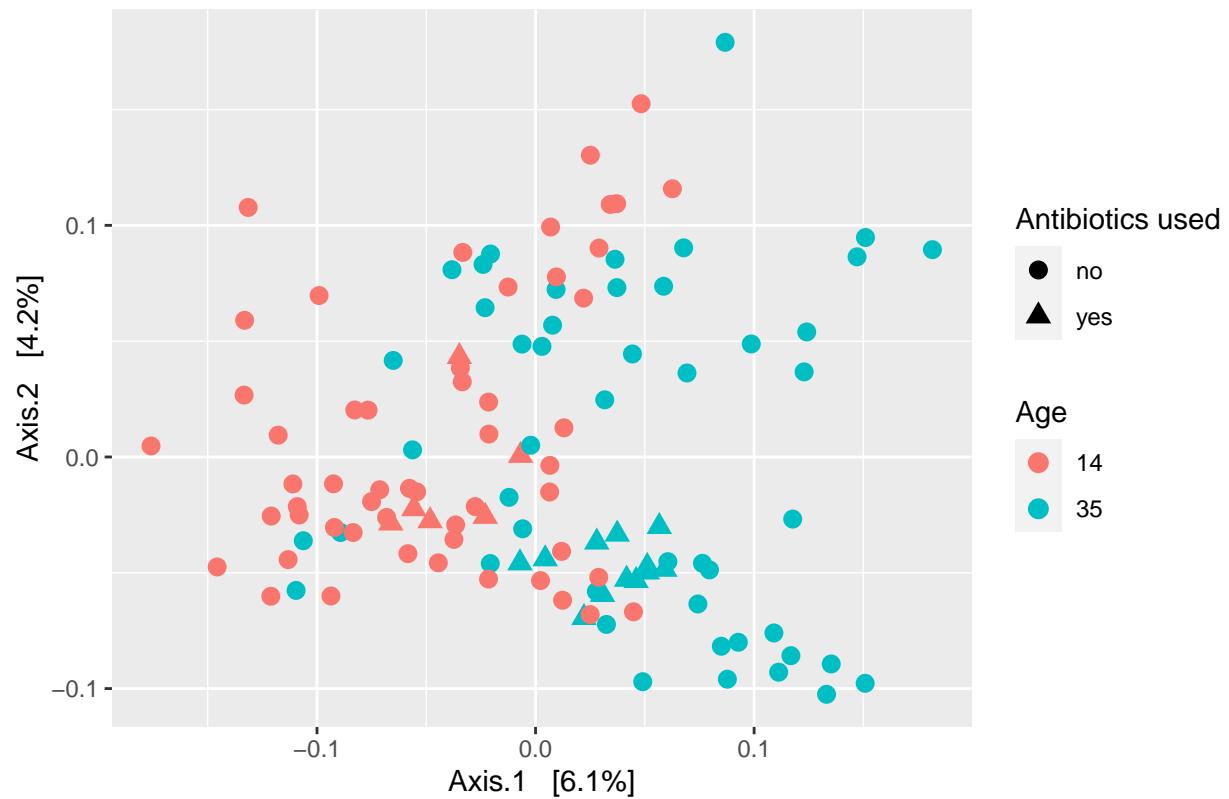
```
plot_pcoa_ordination(subsetMG, pcoa_bc, "Farm2", "PCoA Bray Curtis")
```

PCoA Bray Curtis



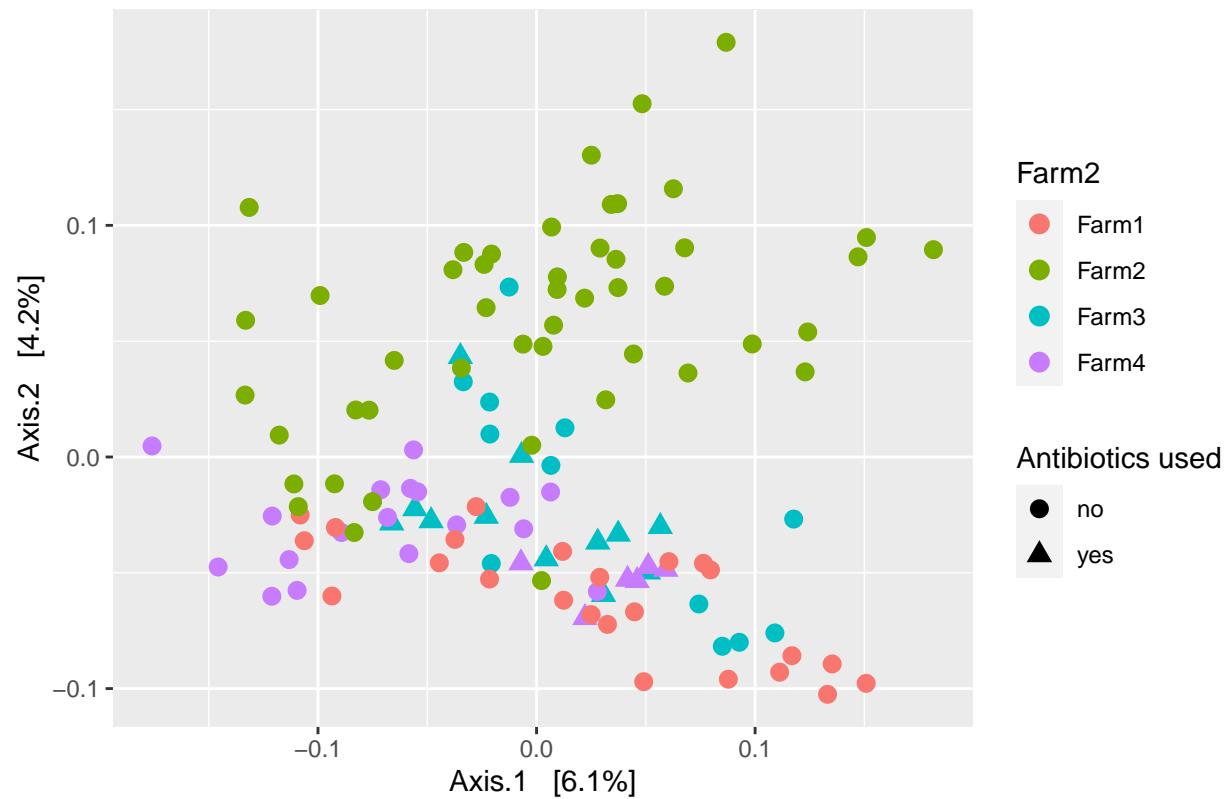
```
plot_pcoa_ordination(subsetMG, pcoa_unifrac, "Age", "PCoA Unifrac")
```

PCoA Unifrac



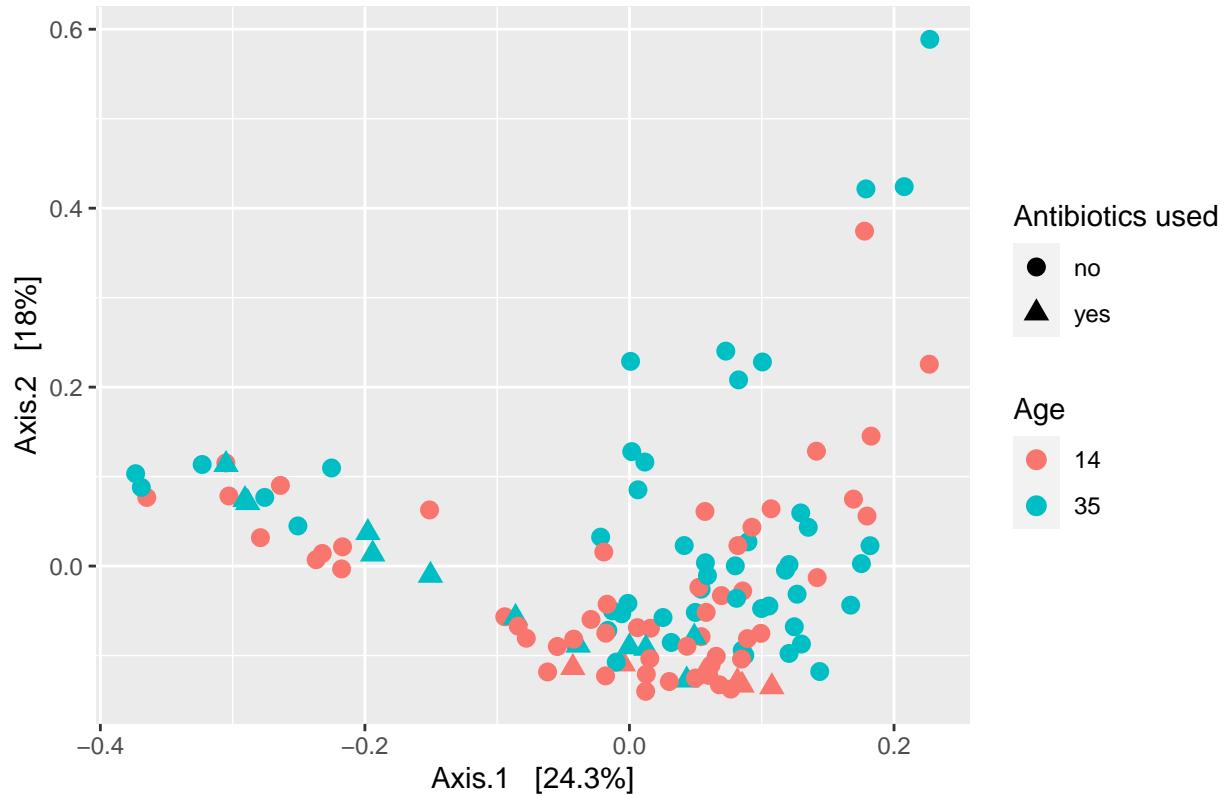
```
plot_pcoa_ordination(subsetMG, pcoa_unifrac, "Farm2", "PCoA Unifrac")
```

PCoA Unifrac



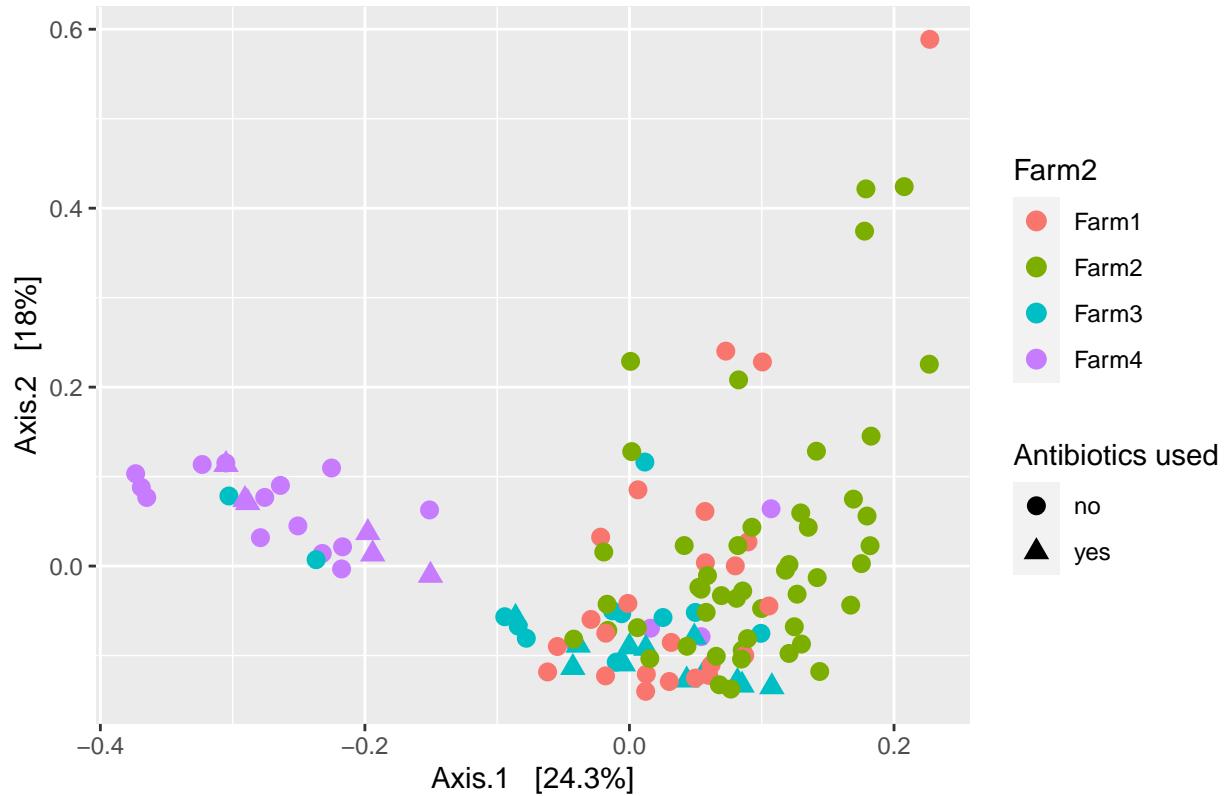
```
plot_pcoa_ordination(subsetMG, pcoa_wunifrac, "Age", "PCoA Weighted Unifrac")
```

PCoA Weighted Unifrac



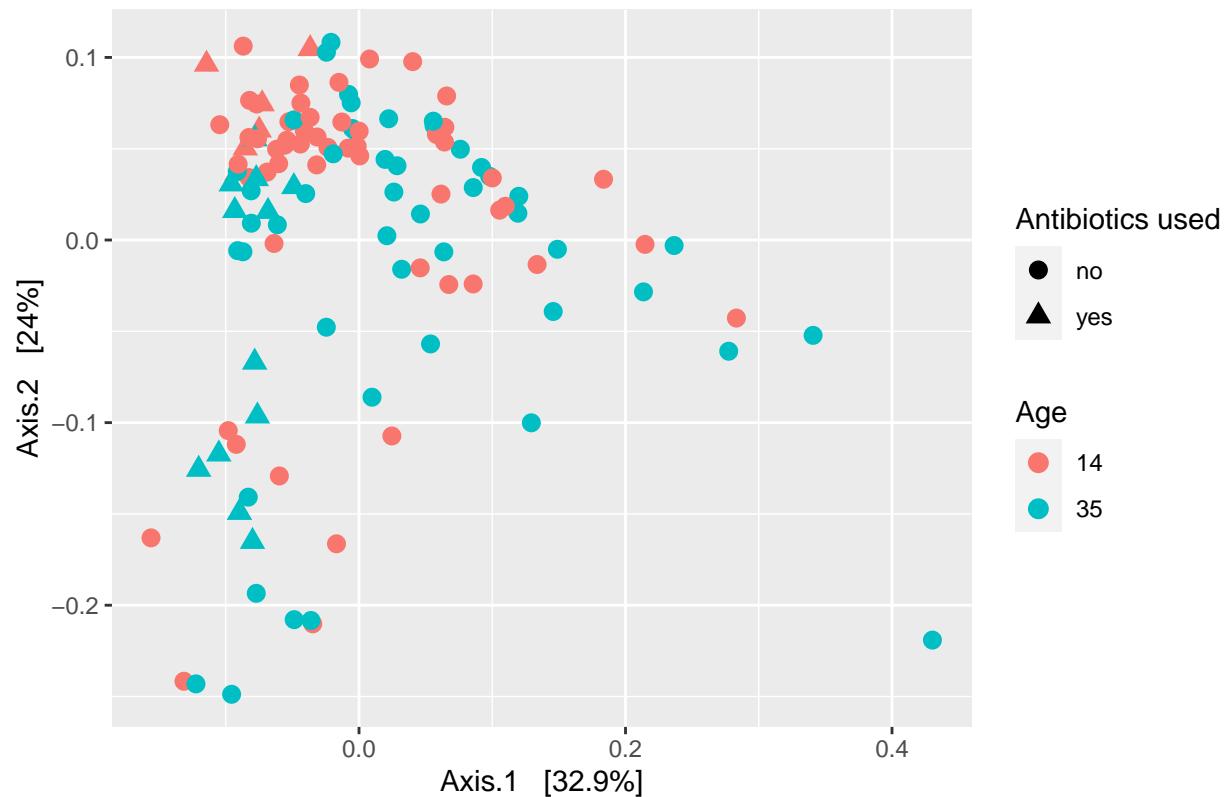
```
plot_pcoa_ordination(subsetMG, pcoa_wunifrac, "Farm2", "PCoA Weighted Unifrac")
```

PCoA Weighted Unifrac



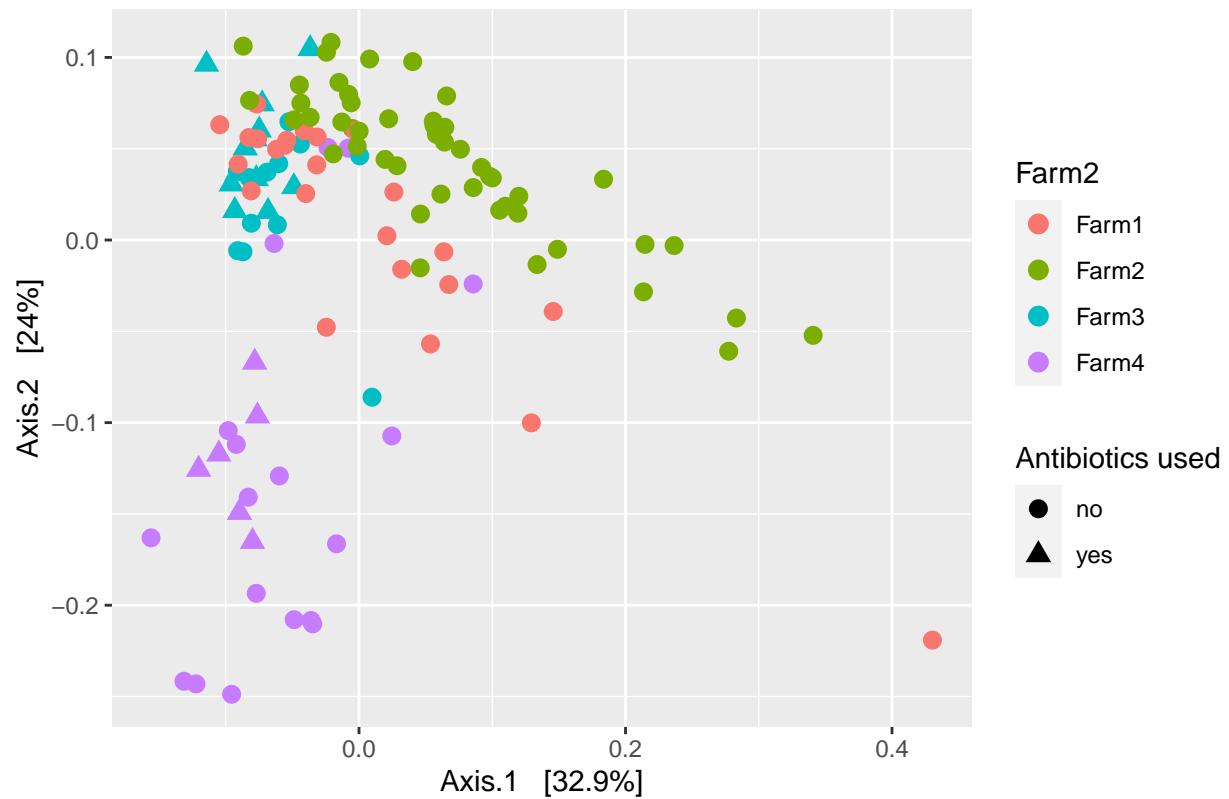
```
plot_pcoa_ordination(subsetMG, pcoa_jsd, "Age", "PCoA Jensen-Shannon Divergence")
```

PCoA Jensen–Shannon Divergence



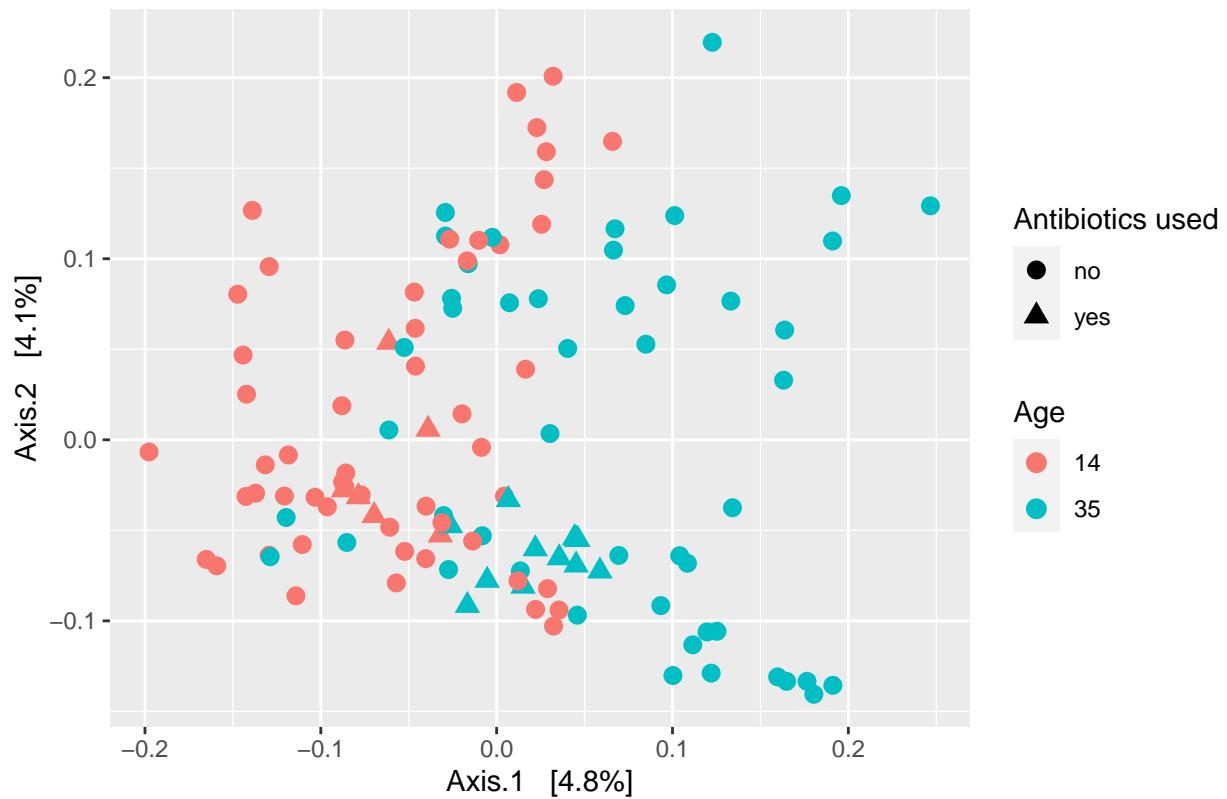
```
plot_pcoa_ordination(subsetMG, pcoa_jsd, "Farm2", "PCoA Jensen–Shannon Divergence")
```

PCoA Jensen–Shannon Divergence



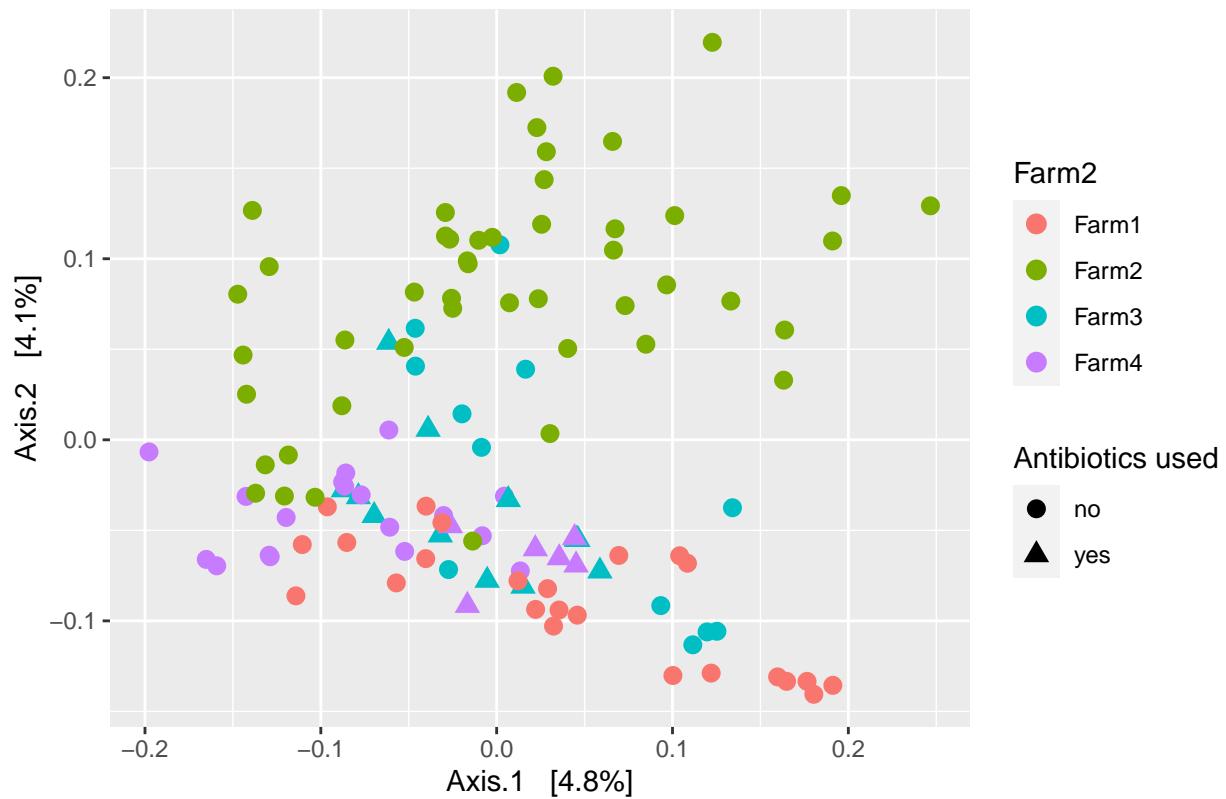
```
plot_pcoa_ordination(subsetMG, pcoa_jaccard, "Age", "PCoA Jaccard")
```

PCoA Jaccard



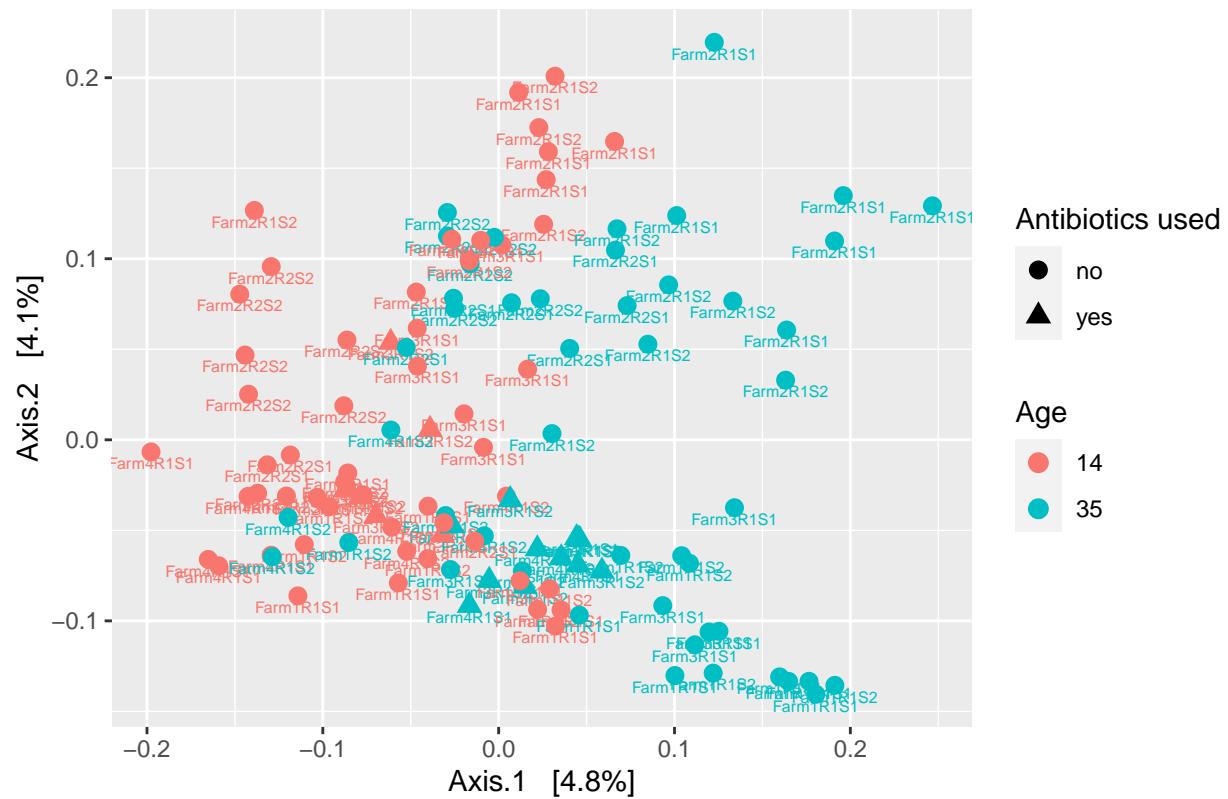
```
plot_pcoa_ordination(subsetMG, pcoa_jaccard, "Farm2", "PCoA Jaccard")
```

PCoA Jaccard



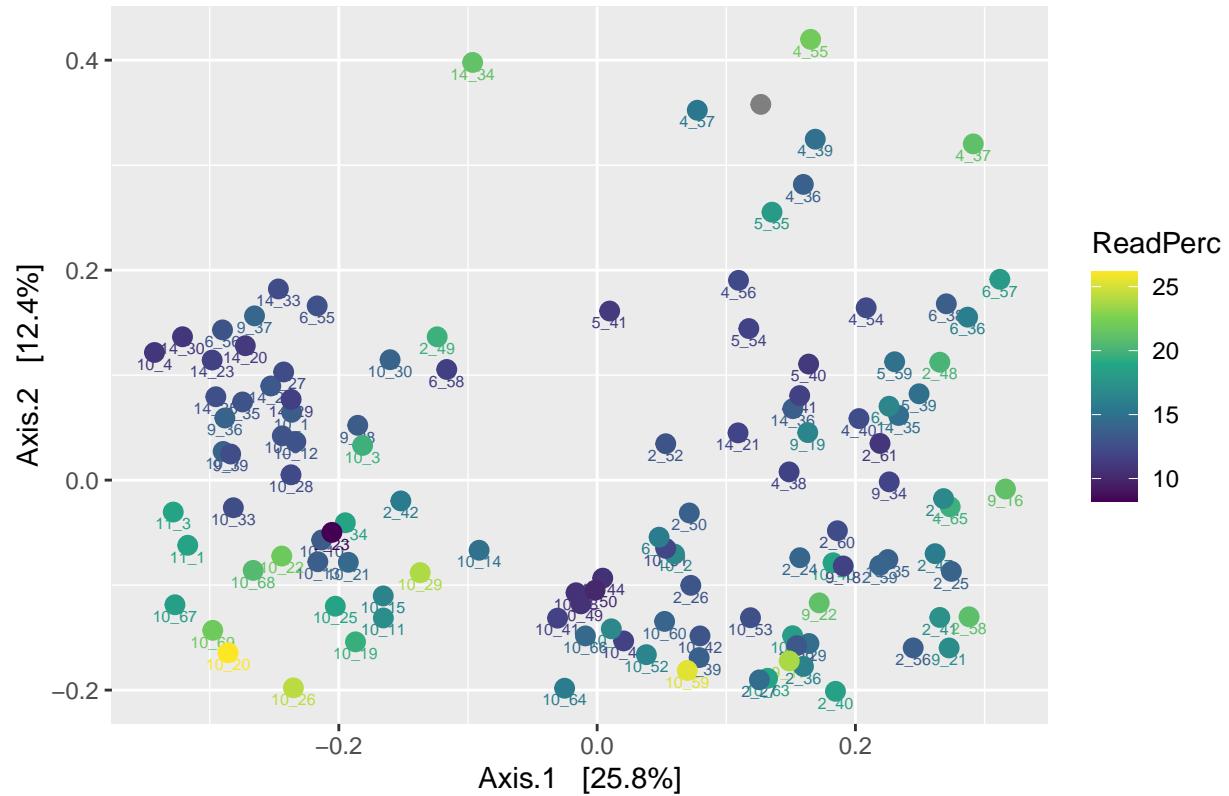
```
plot_ordination(subsetMG, pcoa_jaccard, color = "Age", shape = "AB", label = "FarmRoundStable") +
  geom_point(size = 3) + labs(title = "PCoA Jaccard Age", color = "Age", shape = "Antibiotics used")
```

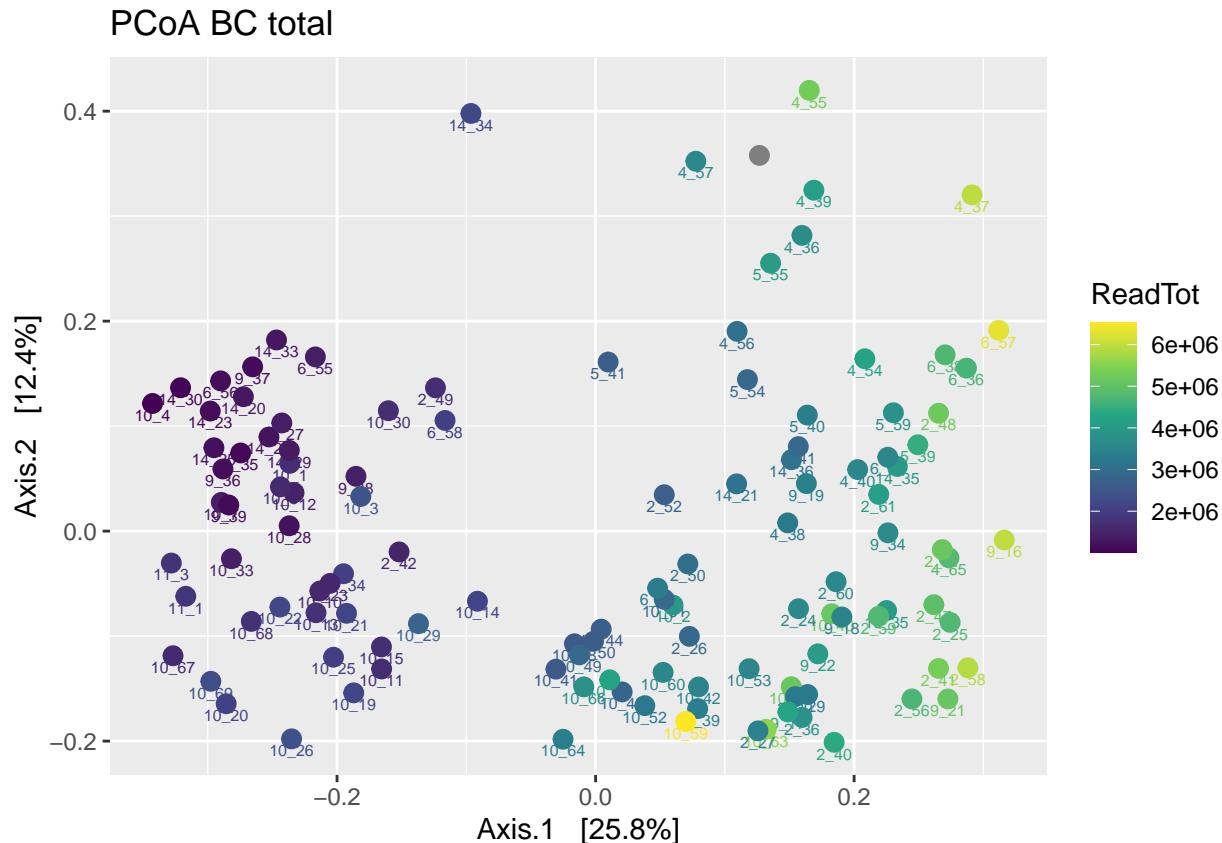
PCoA Jaccard Age



```
# BC plots for looking at percentage and total amount of bacterial reads mapped, removing 2_57 which is
subsetMG %>% subset_samples(Sample_Unique != "2_57") %>% plot_ordination(ordinate(subsetMG, "PCoA", "bra
geom_point(size = 3) + labs(title = "PCoA BC percentage", color = "ReadPerc") +
scale_colour_viridis_c()
```

PCoA BC percentage





```
pcoa_bc2 = subsetMG %>% subset_samples(Sample_Unique != "2_57") %>% ordinate("PCoA", "bray")  
  
pcoa_bc
```

```

## $correction
## [1] "none" "1"
##
## $note
## [1] "No correction was applied to the negative eigenvalues"
##
## $values
##      Eigenvalues  Relative_eig Rel_corr_eig Broken_stick Cum_corr_eig
## 1  5.0122631098  2.575116e-01  0.1442111904  4.535705e-02   0.1442112
## 2  2.4211831094  1.243915e-01  0.0716372884  3.688247e-02   0.2158485
## 3  2.2625283598  1.162404e-01  0.0671935065  3.264518e-02   0.2830420
## 4  1.5393597128  7.908663e-02  0.0469381804  2.982032e-02   0.3299802
## 5  1.2371269732  6.355903e-02  0.0384729037  2.770168e-02   0.3684531
## 6  1.0574339359  5.432706e-02  0.0334398576  2.600676e-02   0.4018929
## 7  0.8366167786  4.298229e-02  0.0272549607  2.459434e-02   0.4291479
## 8  0.7377230898  3.790150e-02  0.0244850343  2.338368e-02   0.4536329
## 9  0.5529466406  2.840836e-02  0.0193096064  2.232436e-02   0.4729425
## 10 0.5102202993  2.621324e-02  0.0181128786  2.138274e-02   0.4910554
## 11 0.4367418812  2.243819e-02  0.0160548119  2.053528e-02   0.5071102
## 12 0.3955216029  2.032044e-02  0.0149002677  1.976487e-02   0.5220105
## 13 0.3889305168  1.998182e-02  0.0147156571  1.905865e-02   0.5367261

```

## 14	0.3310193136	1.700655e-02	0.0130936145	1.840676e-02	0.5498198
## 15	0.3288015932	1.689261e-02	0.0130314981	1.780143e-02	0.5628513
## 16	0.3059531834	1.571875e-02	0.0123915340	1.723646e-02	0.5752428
## 17	0.2821519665	1.449593e-02	0.0117248826	1.670680e-02	0.5869677
## 18	0.2412035196	1.239215e-02	0.0105779521	1.620830e-02	0.5975456
## 19	0.2212217053	1.136556e-02	0.0100182788	1.573749e-02	0.6075639
## 20	0.2109057631	1.083556e-02	0.0097293382	1.529146e-02	0.6172932
## 21	0.2004926895	1.030058e-02	0.0094376771	1.486773e-02	0.6267309
## 22	0.1851614428	9.512913e-03	0.0090082622	1.446418e-02	0.6357392
## 23	0.1619634892	8.321088e-03	0.0083585076	1.407897e-02	0.6440977
## 24	0.1529578528	7.858411e-03	0.0081062676	1.371051e-02	0.6522040
## 25	0.1421444024	7.302856e-03	0.0078033922	1.335740e-02	0.6600073
## 26	0.1314408746	6.752948e-03	0.0075035957	1.301842e-02	0.6675109
## 27	0.1291466790	6.635081e-03	0.0074393373	1.269247e-02	0.6749503
## 28	0.1194555221	6.137185e-03	0.0071678964	1.237860e-02	0.6821182
## 29	0.1069897415	5.496739e-03	0.0068187407	1.207594e-02	0.6889369
## 30	0.0922023830	4.737019e-03	0.0064045596	1.178371e-02	0.6953415
## 31	0.0885749237	4.550653e-03	0.0063029576	1.150122e-02	0.7016444
## 32	0.0830681021	4.267733e-03	0.0061487163	1.122785e-02	0.7077932
## 33	0.0766228890	3.936602e-03	0.0059681915	1.096302e-02	0.7137613
## 34	0.0665413520	3.418650e-03	0.0056858164	1.070621e-02	0.7194472
## 35	0.0648063843	3.329513e-03	0.0056372214	1.045696e-02	0.7250844
## 36	0.0610635733	3.137221e-03	0.0055323886	1.021483e-02	0.7306168
## 37	0.0588788549	3.024979e-03	0.0054711965	9.979427e-03	0.7360880
## 38	0.0558151807	2.867578e-03	0.0053853856	9.750384e-03	0.7414734
## 39	0.0513661477	2.639003e-03	0.0052607721	9.527369e-03	0.7467341
## 40	0.0474807092	2.439384e-03	0.0051519443	9.310072e-03	0.7518861
## 41	0.0452233458	2.323409e-03	0.0050887175	9.098208e-03	0.7569748
## 42	0.0435298491	2.236403e-03	0.0050412842	8.891511e-03	0.7620161
## 43	0.0376446224	1.934042e-03	0.0048764441	8.689735e-03	0.7668925
## 44	0.0343705865	1.765834e-03	0.0047847412	8.492652e-03	0.7716773
## 45	0.0311868846	1.602267e-03	0.0046955684	8.300048e-03	0.7763728
## 46	0.0291039810	1.495255e-03	0.0046372281	8.111724e-03	0.7810101
## 47	0.0273568650	1.405495e-03	0.0045882929	7.927494e-03	0.7855983
## 48	0.0236610502	1.215618e-03	0.0044847763	7.747184e-03	0.7900831
## 49	0.0214674042	1.102916e-03	0.0044233342	7.570631e-03	0.7945065
## 50	0.0198267967	1.018628e-03	0.0043773822	7.397680e-03	0.7988838
## 51	0.0181447270	9.322092e-04	0.0043302689	7.228189e-03	0.8032141
## 52	0.0159879585	8.214024e-04	0.0042698597	7.062020e-03	0.8074840
## 53	0.0150556961	7.735062e-04	0.0042437478	6.899048e-03	0.8117277
## 54	0.0133628625	6.865347e-04	0.0041963330	6.739150e-03	0.8159240
## 55	0.0112287553	5.768921e-04	0.0041365586	6.582214e-03	0.8200606
## 56	0.0074836656	3.844832e-04	0.0040316618	6.428130e-03	0.8240923
## 57	0.0068721916	3.530679e-04	0.0040145350	6.276799e-03	0.8281068
## 58	0.0052137576	2.678637e-04	0.0039680837	6.128122e-03	0.8320749
## 59	0.0030942061	1.589689e-04	0.0039087169	5.982008e-03	0.8359836
## 60	0.0017191419	8.832318e-05	0.0038702025	5.838372e-03	0.8398538
## 61	0.0013366369	6.867148e-05	0.0038594889	5.697129e-03	0.8437133
## 62	0.0001892802	9.724519e-06	0.0038273524	5.558201e-03	0.8475406
## 63	0.0000000000	0.0000000e+00	0.0037433543	5.421514e-03	0.8512840
## 64	-0.0028096757	-1.443508e-04	0.0037301303	5.286997e-03	0.8550141
## 65	-0.0032818076	-1.686072e-04	0.0037149149	5.154582e-03	0.8587290
## 66	-0.0038250358	-1.965162e-04	0.0036881138	5.024204e-03	0.8624172
## 67	-0.0047819084	-2.456768e-04	0.0036823985	4.895801e-03	0.8660996

```

## 68 -0.0049859569 -2.561601e-04 0.0036057765 4.769315e-03 0.8697053
## 69 -0.0077215677 -3.967057e-04 0.0035908852 4.644689e-03 0.8732962
## 70 -0.0082532240 -4.240202e-04 0.0035442985 4.521869e-03 0.8768405
## 71 -0.0099164929 -5.094729e-04 0.0035329641 4.400804e-03 0.8803735
## 72 -0.0103211614 -5.302633e-04 0.0034938824 4.281443e-03 0.8838674
## 73 -0.0117164796 -6.019496e-04 0.0034840516 4.163741e-03 0.8873514
## 74 -0.0120674659 -6.199819e-04 0.0034538587 4.047651e-03 0.8908053
## 75 -0.0131454310 -6.753638e-04 0.0034262208 3.933130e-03 0.8942315
## 76 -0.0141321805 -7.260594e-04 0.0034060468 3.820135e-03 0.8976375
## 77 -0.0148524438 -7.630639e-04 0.0033134096 3.708628e-03 0.9009510
## 78 -0.0181598375 -9.329856e-04 0.0032981028 3.598568e-03 0.9042491
## 79 -0.0187063304 -9.610624e-04 0.0032726584 3.489920e-03 0.9075217
## 80 -0.0196147618 -1.007734e-03 0.0032560863 3.382647e-03 0.9107778
## 81 -0.0202064292 -1.038132e-03 0.0032141586 3.276715e-03 0.9139920
## 82 -0.0217033590 -1.115039e-03 0.0031784661 3.172090e-03 0.9171704
## 83 -0.0229776775 -1.180508e-03 0.0031462240 3.068742e-03 0.9203166
## 84 -0.0241288035 -1.239649e-03 0.0031114634 2.966638e-03 0.9234281
## 85 -0.0253698508 -1.303409e-03 0.0030810809 2.865750e-03 0.9265092
## 86 -0.0264545841 -1.359139e-03 0.0030701763 2.766050e-03 0.9295794
## 87 -0.0268439095 -1.379141e-03 0.0030156511 2.667508e-03 0.9325950
## 88 -0.0287906000 -1.479155e-03 0.0030076622 2.570099e-03 0.9356027
## 89 -0.0290758253 -1.493809e-03 0.0029703252 2.473797e-03 0.9385730
## 90 -0.0304088552 -1.562295e-03 0.0029320737 2.378577e-03 0.9415051
## 91 -0.0317745347 -1.632459e-03 0.0029037392 2.284415e-03 0.9444088
## 92 -0.0327861525 -1.684432e-03 0.0028177107 2.191288e-03 0.9472265
## 93 -0.0358575981 -1.842231e-03 0.0028101184 2.099173e-03 0.9500367
## 94 -0.0361286628 -1.856158e-03 0.0027912126 2.008049e-03 0.9528279
## 95 -0.0368036495 -1.890836e-03 0.0027103116 1.917893e-03 0.9555382
## 96 -0.0396920272 -2.039230e-03 0.0026737616 1.828687e-03 0.9582119
## 97 -0.0409969612 -2.106273e-03 0.0026674007 1.740411e-03 0.9608793
## 98 -0.0412240590 -2.117940e-03 0.0026324065 1.653044e-03 0.9635117
## 99 -0.0424734470 -2.182129e-03 0.0025955535 1.566569e-03 0.9661073
## 100 -0.0437891942 -2.249728e-03 0.0025552754 1.480967e-03 0.9686626
## 101 -0.0452272287 -2.323608e-03 0.0024677670 1.396221e-03 0.9711303
## 102 -0.0483515098 -2.484122e-03 0.0024523025 1.312314e-03 0.9735826
## 103 -0.0489036344 -2.512489e-03 0.0023839486 1.229230e-03 0.9759666
## 104 -0.0513440486 -2.637868e-03 0.0023086771 1.146953e-03 0.9782753
## 105 -0.0540314421 -2.775936e-03 0.0022554839 1.065466e-03 0.9805308
## 106 -0.0559305772 -2.873507e-03 0.0021530668 9.847562e-04 0.9826838
## 107 -0.0595871379 -3.061368e-03 0.0020950043 9.048074e-04 0.9847788
## 108 -0.0616601232 -3.167870e-03 0.0020476252 8.256057e-04 0.9868264
## 109 -0.0633516828 -3.254776e-03 0.0018974510 7.471374e-04 0.9887239
## 110 -0.0687132984 -3.530236e-03 0.0018622185 6.693890e-04 0.9905861
## 111 -0.0699711921 -3.594862e-03 0.0017752460 5.923474e-04 0.9923614
## 112 -0.0730763400 -3.754393e-03 0.0015320825 5.159999e-04 0.9938934
## 113 -0.0817579177 -4.200421e-03 0.0014688977 4.403340e-04 0.9953623
## 114 -0.0840137848 -4.316319e-03 0.0013099217 3.653378e-04 0.9966723
## 115 -0.0896896483 -4.607924e-03 0.0011551201 2.909994e-04 0.9978274
## 116 -0.0952164730 -4.891872e-03 0.0009832039 2.173074e-04 0.9988106
## 117 -0.1013543334 -5.207212e-03 0.0008436656 1.442507e-04 0.9996543
## 118 -0.1063362212 -5.463163e-03 0.0003457452 7.181844e-05 1.0000000
## 119 -0.1241132934 -6.376484e-03 0.0000000000 0.000000e+00 1.0000000
## 120 -0.1364573115 -7.010674e-03 0.0000000000 0.000000e+00 1.0000000
##      Cumul_br_stick

```

```
## 1      0.04535705
## 2      0.08223952
## 3      0.11488470
## 4      0.14470503
## 5      0.17240671
## 6      0.19841347
## 7      0.22300781
## 8      0.24639149
## 9      0.26871585
## 10     0.29009859
## 11     0.31063387
## 12     0.33039874
## 13     0.34945739
## 14     0.36786415
## 15     0.38566558
## 16     0.40290205
## 17     0.41960885
## 18     0.43581715
## 19     0.45155463
## 20     0.46684609
## 21     0.48171382
## 22     0.49617800
## 23     0.51025697
## 24     0.52396747
## 25     0.53732488
## 26     0.55034329
## 27     0.56303577
## 28     0.57541437
## 29     0.58749031
## 30     0.59927402
## 31     0.61077524
## 32     0.62200309
## 33     0.63296611
## 34     0.64367233
## 35     0.65412929
## 36     0.66434412
## 37     0.67432355
## 38     0.68407393
## 39     0.69360130
## 40     0.70291137
## 41     0.71200958
## 42     0.72090109
## 43     0.72959083
## 44     0.73808348
## 45     0.74638353
## 46     0.75449525
## 47     0.76242275
## 48     0.77016993
## 49     0.77774056
## 50     0.78513824
## 51     0.79236643
## 52     0.79942845
## 53     0.80632750
## 54     0.81306665
```

```
## 55      0.81964886
## 56      0.82607699
## 57      0.83235379
## 58      0.83848191
## 59      0.84446392
## 60      0.85030229
## 61      0.85599942
## 62      0.86155762
## 63      0.86697914
## 64      0.87226614
## 65      0.87742072
## 66      0.88244492
## 67      0.88734072
## 68      0.89211004
## 69      0.89675473
## 70      0.90127660
## 71      0.90567740
## 72      0.90995884
## 73      0.91412259
## 74      0.91817024
## 75      0.92210337
## 76      0.92592350
## 77      0.92963213
## 78      0.93323070
## 79      0.93672062
## 80      0.94010326
## 81      0.94337998
## 82      0.94655207
## 83      0.94962081
## 84      0.95258745
## 85      0.95545320
## 86      0.95821925
## 87      0.96088676
## 88      0.96345686
## 89      0.96593065
## 90      0.96830923
## 91      0.97059364
## 92      0.97278493
## 93      0.97488411
## 94      0.97689215
## 95      0.97881005
## 96      0.98063874
## 97      0.98237915
## 98      0.98403219
## 99      0.98559876
## 100     0.98707972
## 101     0.98847595
## 102     0.98978826
## 103     0.99101749
## 104     0.99216444
## 105     0.99322991
## 106     0.99421467
## 107     0.99511947
## 108     0.99594508
```

```

## 109      0.99669222
## 110      0.99736160
## 111      0.99795395
## 112      0.99846995
## 113      0.99891029
## 114      0.99927562
## 115      0.99956662
## 116      0.99978393
## 117      0.99992818
## 118      1.00000000
## 119      1.00000000
## 120      1.00000000
##
## $vectors
##          Axis.1     Axis.2     Axis.3     Axis.4     Axis.5
## 10_10   -0.213270881 -0.057044942  0.137398760  0.072062165  0.1171862547
## 10_11   -0.165509227 -0.131309892  0.114019365  0.299489611  0.1547289805
## 10_12   -0.233280569  0.036568819  0.169431013  0.069559793  0.0573545740
## 10_13   -0.216254599 -0.077734078  0.093171531  0.039059409  0.1388689296
## 10_14   -0.091386677 -0.066784703  0.141554052  0.027474969  0.0851366719
## 10_15   -0.165623029 -0.110290236  0.110700638  0.159136716  0.0651610240
## 10_19   -0.187081064 -0.154048679 -0.144938156  0.142860624 -0.1578668656
## 10_1    -0.236508396  0.064580907  0.123412617 -0.136910710  0.1369349547
## 10_20   -0.285780680 -0.164423875 -0.332426544  0.058270380 -0.1789320618
## 10_21   -0.192669542 -0.078275882 -0.160596676 -0.049489386 -0.1679894746
## 10_22   -0.244184887 -0.072319108 -0.259344526  0.045475780 -0.2029473956
## 10_25   -0.202722715 -0.120107342 -0.122419730  0.100943049 -0.1936595612
## 10_26   -0.235179803 -0.197864861 -0.213557059  0.046817454 -0.0434172165
## 10_28   -0.237031934  0.005082417  0.111019276  0.060313400 -0.0470952718
## 10_29   -0.137010005 -0.088111409 -0.219450524  0.052986205 -0.0379733713
## 10_2    0.059910728 -0.070439860  0.042441722 -0.195934126  0.1907000278
## 10_30   -0.160313806  0.114701936  0.026908369  0.002718166 -0.0300904055
## 10_33   -0.281684884 -0.026293955  0.058541705  0.042149826  0.0028622697
## 10_34   -0.194977755 -0.040602906 -0.151496656  0.077290646 -0.0308148546
## 10_35   0.225131822 -0.075626287  0.076134822  0.088853700 -0.0638045214
## 10_39   0.078999538 -0.169035665  0.021376405 -0.123623359  0.1200309372
## 10_3    -0.181578101  0.033034625  0.026749762 -0.101381517  0.2491735453
## 10_40   0.182696940 -0.078698674 -0.135276164 -0.015139675  0.1756848492
## 10_41   -0.030647012 -0.131484186  0.051215228 -0.134626896  0.1070411711
## 10_42   0.079658429 -0.148388684  0.026094078 -0.190050892  0.1446676486
## 10_43   0.020545775 -0.153167780  0.096393409 -0.031425316  0.1229067820
## 10_44   0.004358239 -0.093377730  0.048009206 -0.164163268  0.0667967892
## 10_48   -0.016329622 -0.107282471  0.135172152  0.014535999  0.0890001159
## 10_49   -0.012528275 -0.117872139  0.067665288 -0.119009276  0.0764318802
## 10_4    -0.342730663  0.121569583  0.123076270 -0.005740733  0.0249483128
## 10_50   -0.001582568 -0.104906967  0.079829399 -0.076633368  0.0300744823
## 10_51   0.053108332 -0.065013791  0.094643638 -0.135730951  0.0024909230
## 10_52   0.038114683 -0.166244760  0.056513885  0.089890994  0.0935364412
## 10_53   0.118758400 -0.131042451  0.081383852 -0.095556883  0.0719952287
## 10_57   0.151364741 -0.148182959 -0.153461708 -0.201196627  0.0508873743
## 10_58   0.010977178 -0.141635032 -0.218433124 -0.223523272  0.0264238524
## 10_59   0.069820509 -0.181305860 -0.209986963 -0.028869987  0.2887437871
## 10_60   0.052231668 -0.134567535 -0.071719772 -0.187178205  0.0386775776
## 10_63   0.131990518 -0.188753824 -0.211514012 -0.202568001  0.0575027929

```

```

## 10_64 -0.025166292 -0.198227587 -0.148712687 -0.172643136 0.0317027341
## 10_66 -0.009066998 -0.148385402 -0.205338337 -0.178400207 -0.0162968134
## 10_67 -0.326891231 -0.118702719 -0.347854467 -0.030908953 -0.1360101059
## 10_68 -0.266328730 -0.085999812 -0.279856161 -0.030801849 -0.0727222220
## 10_69 -0.297617932 -0.143066099 -0.326390478 -0.091848327 -0.0598934066
## 10_7 -0.289543527 0.027338331 0.088744195 -0.072991233 0.1244225256
## 10_8 -0.243776525 0.042145191 0.132421723 -0.055727679 0.1075396710
## 11_1 -0.316996889 -0.062070803 -0.262166486 -0.036265425 -0.1096998056
## 11_3 -0.328214222 -0.030421057 -0.286095847 -0.035383253 -0.1666506046
## 14_20 -0.272438563 0.128031767 0.165914571 -0.051025389 0.0111965598
## 14_21 0.109111447 0.045003518 0.123710929 -0.105330011 -0.1415368166
## 14_22 -0.252664481 0.089668164 0.118861906 0.108841136 0.0041309913
## 14_23 -0.298078891 0.114375349 0.135296961 0.009408762 -0.0194622273
## 14_25 -0.295232182 0.079410009 0.129074073 0.022521689 -0.0118010490
## 14_27 -0.242718213 0.103001784 0.140572670 -0.013349866 -0.0124969213
## 14_29 -0.236929189 0.076901403 0.182159251 -0.009640614 0.0149189266
## 14_30 -0.321038674 0.136552459 0.148211519 -0.005973124 0.0193455444
## 14_33 -0.246814590 0.182073784 0.120303005 -0.030431915 0.0005382726
## 14_34 -0.096413081 0.397852869 -0.193678716 0.148906821 0.1556891478
## 14_35 0.233455304 0.061572478 0.077314637 -0.006225587 -0.0599682486
## 14_36 0.151537912 0.068049231 0.075578721 0.027376338 -0.0204736778
## 2_23 -0.205350325 -0.049760219 0.114668235 0.079233008 0.0341522464
## 2_24 0.156991170 -0.073909378 0.055750756 -0.004574774 -0.0359222458
## 2_25 0.274297958 -0.086931437 0.069147393 0.085304011 -0.0728993632
## 2_26 0.072538895 -0.100228881 0.123908344 0.004432621 0.0153669612
## 2_27 0.125655104 -0.190265670 0.085303867 0.025038835 0.0652961869
## 2_29 0.164020164 -0.155609072 0.123933607 0.122460599 -0.0239151134
## 2_36 0.159757742 -0.177494098 0.085878904 0.049130811 0.0433259400
## 4_65 0.273401669 -0.025401909 -0.118866313 0.242781781 0.0598281872
## 2_39 0.154556972 -0.157632346 0.097111054 0.091917292 -0.0486929993
## 2_40 0.184546594 -0.201153362 0.073605211 0.143804512 0.0349122255
## 2_41 0.265339362 -0.130792760 0.036506048 0.145938329 -0.0951328026
## 2_42 -0.151990130 -0.019833596 0.056308946 0.236341887 -0.0264267081
## 2_47 0.261718616 -0.070062464 -0.018997313 0.005569149 0.0280314807
## 2_48 0.265340382 0.112306749 -0.143050056 0.115925519 0.1345172195
## 2_49 -0.123852306 0.136488008 -0.187339350 0.002331844 0.0958516569
## 2_50 0.071377823 -0.031273133 -0.005272866 -0.054279164 0.0429069992
## 2_51 0.268233417 -0.017578729 -0.120728368 -0.089360363 0.0669703799
## 2_52 0.053128405 0.034647293 0.023622806 -0.052094864 0.0118328570
## 2_56 0.244750175 -0.159832113 0.047973854 -0.072952899 0.0325943522
## 2_57 0.126738400 0.357877294 -0.458956860 0.289630737 0.2485416309
## 2_58 0.287991793 -0.130178841 -0.073688200 -0.007096028 0.0883399916
## 2_59 0.218945416 -0.081187543 0.012433311 -0.042212230 -0.0531056184
## 2_60 0.186047580 -0.048218864 0.113923579 -0.002193047 -0.0262609645
## 2_61 0.219062977 0.034935641 0.049534873 -0.131847516 -0.1282990272
## 4_36 0.159665562 0.281664484 -0.018810130 -0.062573834 0.0306685851
## 4_37 0.291258897 0.320259460 -0.210543006 0.097828709 0.0460396745
## 4_38 0.148642316 0.007839413 0.116432078 -0.111307313 -0.1299639588
## 4_39 0.168896016 0.324728546 -0.092656960 -0.033278434 0.0518330830
## 4_40 0.202746300 0.058615034 0.139024318 -0.101508615 -0.1612286862
## 4_41 0.156848987 0.080503360 0.107553393 -0.078830437 -0.1707576977
## 4_54 0.208322808 0.164144695 0.039070763 -0.108336127 -0.1438792323
## 4_55 0.165305865 0.419845202 -0.214535347 -0.002475694 0.1757873772
## 4_56 0.109481335 0.190276378 0.051982903 -0.140389440 -0.0656564602

```

```

## 4_57  0.077562396  0.352315829 -0.083661091 -0.078892150  0.0625282409
## 5_39  0.249278488  0.082220929  0.041863119  0.020882441 -0.2065294507
## 5_40  0.163788696  0.110538532  0.082902745 -0.026380589 -0.1259672836
## 5_41  0.009711905  0.161055539  0.138631618 -0.127947742 -0.0643917123
## 5_54  0.117492978  0.144374423  0.042196351 -0.101162870 -0.0697908537
## 5_55  0.135411452  0.255196983 -0.101371658 -0.075061872  0.0274786325
## 5_59  0.230379555  0.112734739  0.027438287 -0.050502122 -0.1134251828
## 6_36  0.286886292  0.155230872 -0.086296607  0.018025690 -0.0609742450
## 6_37  0.226058333  0.070353758  0.040147868  0.083507324 -0.1774601045
## 6_38  0.270386908  0.167920692 -0.026111455 -0.055574172 -0.1088336098
## 6_54  0.047993264 -0.054356076  0.068245057 -0.062806481  0.0350684478
## 6_55  -0.216762431  0.165936017  0.145589044 -0.096108602  0.0012848190
## 6_56  -0.290049463  0.143084513  0.074234741  0.037307510 -0.0345751517
## 6_57  0.311838632  0.191294052 -0.094841959 -0.064572083 -0.0805200507
## 6_58  -0.116440802  0.105392448  0.065350733 -0.144391884  0.0023828734
## 9_16  0.316158501 -0.008438574 -0.058042659  0.243475041  0.0337805786
## 9_17  0.148757027 -0.172258099  0.037604059  0.279495808  0.0400009739
## 9_18  0.190575685 -0.081774394  0.123466239  0.061338948 -0.1276632040
## 9_19  0.163086163  0.045298956 -0.014453243  0.227124146  0.0637922891
## 9_21  0.272561493 -0.159847987  0.086939270  0.203043026 -0.0490439244
## 9_22  0.171948819 -0.116914275  0.028181762  0.262876064  0.0103591725
## 9_34  0.226174933 -0.001666040  0.069126124  0.027126873 -0.1807594445
## 9_35  -0.274543399  0.074435428  0.142473164  0.054417555 -0.0460606942
## 9_36  -0.288258122  0.059167152 -0.007526775  0.044507475 -0.0235503639
## 9_37  -0.265286147  0.156455696  0.016893134  0.115798230 -0.0790163929
## 9_38  -0.185354064  0.052198452  0.097566518  0.030219837 -0.1018923602
## 9_39  -0.283748288  0.024860672  0.086920336  0.070651116 -0.0247078084
##          Axis.6      Axis.7      Axis.8      Axis.9      Axis.10
## 10_10 -0.1060083292  0.051392152  0.007167341 -5.085258e-02  0.024024101
## 10_11 -0.0500739476  0.149832421 -0.010077310  2.696505e-02  0.100870641
## 10_12  0.0065385714  0.089472458  0.020920251  6.371619e-02  0.041371039
## 10_13 -0.0390326198  0.050841976 -0.018724792 -9.813843e-02  0.015422454
## 10_14 -0.0893799571  0.089277093 -0.025866283 -3.656543e-02 -0.031698569
## 10_15 -0.0884369777  0.040581687 -0.011688486 -7.054789e-02 -0.031907685
## 10_19 -0.0521477555 -0.028187057 -0.281062810 -8.619121e-02  0.014403368
## 10_1  0.0878204458  0.081031767 -0.033436887 -1.083088e-01  0.009520175
## 10_20 -0.0124354515  0.007451076 -0.206475106  7.609865e-03  0.012753276
## 10_21 -0.0035239858  0.070415642  0.046792122  1.918137e-02 -0.063482807
## 10_22 -0.0487785808 -0.004661151 -0.205152437 -2.875959e-02  0.025269291
## 10_25 -0.0395674800 -0.038989957 -0.226818160 -7.325298e-02 -0.063284888
## 10_26 -0.0437891904 -0.144423693 -0.285582801  1.942720e-02 -0.002754562
## 10_28 -0.0126268241 -0.120818381 -0.014061525 -2.111807e-03 -0.059879462
## 10_29 -0.0376475158  0.190604177  0.260889937 -4.029587e-03 -0.061248045
## 10_2  0.1928752376  0.117619314 -0.139453535 -1.396103e-01  0.071653428
## 10_30 -0.0819368305 -0.074281000  0.053282638 -2.140081e-02  0.030660907
## 10_33 -0.0179867269 -0.073401780  0.025739626  2.231813e-02 -0.082771187
## 10_34 -0.1273661725  0.146279420  0.178859020 -1.034450e-02  0.056455957
## 10_35 -0.0360585774  0.022597425  0.057342392 -1.177655e-01 -0.024091123
## 10_39 -0.0065640129 -0.151445409 -0.012482888  4.103959e-02 -0.015526815
## 10_3  0.3119153308  0.145974787 -0.116151432 -1.825379e-01  0.126972344
## 10_40 -0.0168612353 -0.230986113  0.061464559  2.271476e-02  0.023034826
## 10_41 -0.0526771169 -0.175557301 -0.016229704  5.734284e-02 -0.075427955
## 10_42 -0.0086296365 -0.116757425 -0.052966349  2.032819e-03 -0.017715277
## 10_43 -0.0730538833  0.007987719 -0.015689468 -2.230619e-02 -0.027616989

```

```

## 10_44 -0.0617315749 -0.152272223 -0.038625019  4.235980e-02 -0.052481884
## 10_48 -0.1361650988  0.057683824  0.020274238 -8.139054e-03 -0.079543748
## 10_49 -0.0340616478  0.002554549 -0.051664636  7.029180e-03 -0.107981942
## 10_4   0.0004655099 -0.025232083 -0.008526960 -1.741401e-02  0.030021085
## 10_50 -0.0736643766  0.006763993 -0.017853498 -2.803362e-02 -0.085192447
## 10_51 -0.0565176093 -0.047478875 -0.038334613  1.393696e-02 -0.086139509
## 10_52 -0.1129991643  0.042910327  0.033366770 -9.771940e-02 -0.087808572
## 10_53 -0.0926943026  0.022524142 -0.045440432 -4.610841e-02 -0.087604463
## 10_57  0.1494896668  0.027384655  0.047900913 -3.300437e-02  0.035600554
## 10_58 -0.0285400086  0.086898076  0.153268271  7.770373e-02 -0.090995961
## 10_59  0.4050403647 -0.053621375  0.003335627 -3.531568e-02  0.047965377
## 10_60  0.0148593698  0.016792993  0.012019513 -1.533420e-02  0.012369962
## 10_63  0.0905620972  0.066224101  0.123079374  7.709011e-02 -0.021906975
## 10_64 -0.0416313275  0.006819198  0.086007729  1.205582e-01 -0.088370479
## 10_66  0.0570998049  0.145345212  0.075029133  1.398131e-02  0.022997454
## 10_67 -0.0133210808  0.067206981  0.047262277  6.210609e-02  0.002203407
## 10_68 -0.0124778842 -0.136309984  0.060135977 -4.437659e-03  0.079642233
## 10_69 -0.0186021446  0.146317528 -0.015197358 -1.038384e-02  0.017615417
## 10_7   0.0836378895  0.048051249 -0.041935456 -1.074046e-01  0.055464552
## 10_8   0.1502806595  0.086957560 -0.035059621  4.676827e-03  0.054929395
## 11_1   0.0088829137  0.079536537  0.085917214 -1.247289e-02  0.015408053
## 11_3   0.0087806726  0.113354238  0.103051011  1.196344e-02  0.014125210
## 14_20  0.0058878633  0.033783924 -0.001368475  1.995384e-02 -0.045690589
## 14_21  0.0008769660  0.015492069 -0.056590447  1.702739e-02 -0.080548324
## 14_22  0.2228754609  0.069668685 -0.060038967  2.684399e-01  0.116089672
## 14_23 -0.0100203200 -0.028599444  0.014143915 -3.318134e-03 -0.018975331
## 14_25  0.0089196846 -0.023320001  0.020246717  1.597907e-02 -0.015823744
## 14_27  0.0403217679 -0.001222899 -0.013943980  6.576396e-02 -0.030865242
## 14_29 -0.0113719480  0.025196135  0.014374678  1.334490e-02 -0.032715578
## 14_30 -0.0183643535 -0.003552099  0.031146636 -1.509324e-02  0.007233956
## 14_33 -0.0627405996 -0.017188237  0.049097777 -2.517132e-02  0.045989584
## 14_34 -0.1114290254 -0.026986360 -0.071403504  6.017077e-02  0.038418266
## 14_35 -0.0377276780  0.069002384 -0.055654358  8.532901e-02  0.121251668
## 14_36 -0.0826075729  0.035052074 -0.022469217  3.824064e-02  0.100168322
## 2_23  -0.0153256997 -0.010029380 -0.006353450  3.653665e-02  0.036324859
## 2_24  0.0569819748 -0.028620563  0.013136011  3.825706e-02 -0.025600239
## 2_25  -0.0453599780  0.070519456  0.031819607 -7.452126e-02 -0.009005266
## 2_26  -0.1063120362  0.040791921  0.002212101 -1.210300e-02 -0.053146113
## 2_27  -0.0384914883 -0.019268312 -0.038490874  1.012388e-01  0.010087547
## 2_29  -0.0479941177  0.055831796 -0.024401128  1.865509e-02 -0.011095843
## 2_36  -0.0302443094  0.024253803 -0.033800946  1.287637e-01  0.030626668
## 4_65  -0.0014648984 -0.056742379  0.036728573 -6.348443e-02  0.016008013
## 2_39  -0.0377637390  0.017949613 -0.040326425 -3.005423e-02  0.001926501
## 2_40  -0.0442501761  0.075020479 -0.022673850  4.645455e-02  0.036689162
## 2_41  -0.0422217158  0.052661460 -0.058167640 -9.907989e-02  0.055216101
## 2_42  0.0849874301 -0.042570954  0.102185500 -1.242243e-01 -0.054577700
## 2_47  -0.1309684707 -0.051476505 -0.007247873 -3.573049e-02  0.114088281
## 2_48  -0.1334810846  0.050394012 -0.003719351 -5.480194e-02 -0.015909084
## 2_49  -0.1249709968 -0.206757832  0.098378156  1.589066e-02  0.272030873
## 2_50  -0.1144213402 -0.061659565  0.055111035 -4.915516e-02  0.062979027
## 2_51  -0.0638885620 -0.186076707  0.047831171 -3.395184e-03  0.067341396
## 2_52  -0.1162833485 -0.057629409  0.047436160 -3.315857e-02  0.042830419
## 2_56  -0.0565883589 -0.064500763 -0.035063018  2.130493e-02 -0.017333758
## 2_57  0.1301814646 -0.063914989 -0.056072557 -4.148207e-02 -0.189102547

```

```

## 2_58 -0.0605191941 -0.152813070 0.063774811 -3.982247e-02 0.042697711
## 2_59 0.0312465646 -0.002465469 -0.003566272 -3.115889e-02 0.043063162
## 2_60 -0.0900429001 0.101977747 0.001790930 -4.394091e-02 0.014674539
## 2_61 -0.0427136960 -0.010646306 -0.037667743 -6.896344e-02 0.054450456
## 4_36 -0.0502333160 0.087447492 -0.067951878 1.169836e-01 0.030963612
## 4_37 0.0890519865 0.030105613 -0.083200191 -9.989188e-03 -0.155849111
## 4_38 0.0698854099 0.013598339 -0.069879219 4.291900e-02 0.038843900
## 4_39 -0.0866963277 0.077547046 -0.041679653 4.451204e-02 -0.019214680
## 4_40 0.0853150182 0.067295022 -0.028590873 5.886504e-02 -0.003626727
## 4_41 0.0018095239 -0.024181859 0.011523174 -7.385057e-02 0.045376589
## 4_54 0.0239735233 0.014091514 0.032613975 -7.327510e-02 -0.033191046
## 4_55 -0.0911298322 0.107284853 -0.136084417 7.147668e-02 -0.165019960
## 4_56 -0.0878607971 0.017875745 0.002020435 -5.863838e-05 0.021179235
## 4_57 -0.1055595730 0.074666257 0.012669644 3.485095e-02 0.063102658
## 5_39 0.1166261576 0.009864839 -0.029971999 3.758933e-02 0.133504914
## 5_40 0.1348403001 0.012049556 0.049045662 -4.671096e-02 -0.084170580
## 5_41 0.0062664458 0.112483830 -0.007794763 6.088668e-02 -0.065375854
## 5_54 -0.0607939183 -0.020285424 0.038554739 -1.655962e-02 0.028030553
## 5_55 -0.1144878014 0.013972811 -0.009407208 4.900360e-02 0.018181049
## 5_59 0.1042523287 -0.032901730 0.062851809 -1.236904e-01 -0.073086926
## 6_36 0.0619792992 -0.084730375 0.007651563 -4.781219e-02 0.007318394
## 6_37 0.2512669871 -0.029032560 0.025741712 -3.662237e-02 -0.029040679
## 6_38 -0.0189990100 -0.009716633 -0.016165391 -3.246871e-02 0.070471298
## 6_54 0.1800190171 -0.039929447 -0.077830127 2.470799e-01 0.002378124
## 6_55 -0.0421850125 -0.016436500 0.024098577 -2.012164e-02 -0.034538455
## 6_56 0.0737123242 -0.059968166 0.126947145 -9.067994e-02 -0.019266694
## 6_57 0.0190993276 0.008053039 -0.008342288 -4.661385e-02 0.064167918
## 6_58 -0.0772587131 0.001658900 -0.017727261 1.421524e-02 -0.047436992
## 9_16 0.0688882108 -0.002437475 0.077944135 9.863357e-03 0.007898744
## 9_17 0.0349165539 0.037945708 0.090430757 6.432924e-02 -0.041530463
## 9_18 0.0463502720 -0.017262646 -0.014931861 4.543304e-02 0.020630443
## 9_19 -0.0811356339 0.070997049 0.028938443 7.477891e-02 0.096642756
## 9_21 0.0536035156 0.060270681 0.001186910 1.123287e-01 0.011499647
## 9_22 0.1732811494 0.005013465 0.087785023 8.765225e-02 -0.070258170
## 9_34 0.1769074406 -0.096323440 0.038050442 -2.941114e-02 -0.046687252
## 9_35 0.0750873266 -0.070675973 0.008721418 6.784933e-02 -0.035657556
## 9_36 0.0265721861 -0.213112785 0.068089249 2.575574e-02 0.049267314
## 9_37 0.0774764455 -0.133178964 0.153834495 -7.014021e-02 0.034762944
## 9_38 0.0416825932 -0.118047898 0.063230423 -2.288786e-02 -0.074424144
## 9_39 0.0775055164 -0.185814671 0.018685336 8.656560e-02 -0.070964786
##          Axis.11      Axis.12      Axis.13      Axis.14      Axis.15
## 10_10 -0.0814713118 7.312747e-02 0.0054236192 0.0596786221 0.0149117024
## 10_11 -0.0516747034 9.637356e-02 0.0052616015 0.0924458323 -0.0400914302
## 10_12 -0.0444608351 9.107414e-02 0.0091242013 0.0522620562 0.0096795088
## 10_13 -0.0551521367 5.344578e-02 0.0602675281 0.0040556586 0.0175064443
## 10_14 -0.0059220952 1.076811e-01 0.0525606088 0.0556536338 0.0468005203
## 10_15 -0.0282778049 -7.477701e-02 -0.0446849595 -0.0223844588 0.0277419672
## 10_19 0.0359317718 6.297250e-02 -0.0298078372 0.0004327799 0.0565993255
## 10_1 0.0284006344 -3.986310e-03 0.0467881637 -0.0296589956 -0.0658868546
## 10_20 0.0284422586 2.952631e-02 -0.0515144737 -0.0089381159 0.0003368982
## 10_21 0.0465450191 -3.088402e-02 0.0614508601 -0.0560601093 0.0073798217
## 10_22 0.0833228150 9.555356e-03 0.0089938048 0.0348819129 0.0185231689
## 10_25 0.0985948635 2.459801e-02 0.0017862323 -0.0009987119 0.0658432992
## 10_26 0.0090404616 9.281687e-02 -0.0693043244 0.0329648070 -0.0554384648

```

```

## 10_28  0.0577890706 -6.873448e-02  0.0670120463  0.0261938045  0.0572609197
## 10_29 -0.0057085131 -3.474217e-02 -0.0006924027  0.1212067769  0.0769863161
## 10_2   0.1006822605 -6.200062e-02  0.0429955295 -0.0087055525  0.0230291775
## 10_30  0.0962372687 -4.228020e-02  0.0704919590 -0.0032413681  0.0900459056
## 10_33  0.0405221818 -4.887682e-02  0.0085658084  0.0184119002  0.0333287556
## 10_34  0.0145678632  4.988201e-03  0.0713906542  0.1392455156  0.0494894371
## 10_35 -0.0495229772 -4.326866e-02 -0.0590522267 -0.0181511665  0.0132557429
## 10_39 -0.0393814261  1.518311e-02  0.0650154057  0.0304659917 -0.0246328463
## 10_3   0.1104704160 -1.224297e-01 -0.0146853852  0.0320857524 -0.0334454120
## 10_40  0.0642454581  3.325867e-02  0.0196424979  0.0769339055 -0.0128122568
## 10_41 -0.0153462041  4.947916e-03  0.0100242045 -0.0063267850 -0.0838360943
## 10_42 -0.0043515834  4.001313e-02  0.0537532742 -0.0054280947 -0.0493215296
## 10_43  0.0072780387  1.424150e-02  0.0404030841 -0.0697785046 -0.0237125554
## 10_44 -0.0135899724  5.274412e-02  0.0268063557 -0.0347092599 -0.0785392131
## 10_48  0.0229418563  6.758290e-03 -0.0419314294  0.0327880011  0.0172049959
## 10_49  0.0405589735 -5.601048e-03  0.0509331346 -0.0024279304  0.0505852647
## 10_4   -0.0781556926  3.170115e-02 -0.0093858494  0.0225583784 -0.0883265732
## 10_50  0.0435634808  1.072438e-02  0.0009410379 -0.0227612872  0.0809122276
## 10_51  0.0330010468 -2.004450e-02  0.0403729755  0.0032531284  0.0162716399
## 10_52 -0.0346013220 -1.476241e-01 -0.1549990137 -0.0846149024 -0.0208873709
## 10_53  0.0012881695  6.672354e-03 -0.0241097287 -0.0146381243  0.0111966501
## 10_57  0.0052465358  1.001249e-03 -0.0258417486 -0.0334737008  0.0003552236
## 10_58  0.0096196958 -4.305124e-02 -0.0132077922  0.0291627706  0.0371810142
## 10_59  0.0516112373 -1.493538e-02 -0.1189415923  0.0817848844  0.1109784787
## 10_60  0.0094389036 -1.631162e-02  0.0318639049 -0.1149316682  0.0684013590
## 10_63 -0.0257281801 -2.735698e-02  0.0298776420  0.0416156417 -0.0118131110
## 10_64  0.00111145349 -1.651897e-02  0.0591080657  0.1129601423 -0.0156223774
## 10_66 -0.0069287676  4.628701e-02  0.0758294154 -0.1021367886  0.0088230374
## 10_67 -0.0442000602 -6.339412e-02  0.0175115060 -0.0045916178 -0.0526143091
## 10_68 -0.0795598122  3.144200e-02 -0.0641425586 -0.1383566364 -0.0365475805
## 10_69 -0.0367527079  4.399488e-02  0.0436546630  0.0209699219 -0.0731743996
## 10_7   -0.0327276345  2.005529e-02  0.0802990770 -0.0309672028 -0.0786576978
## 10_8   -0.0062060988 -1.999543e-03  0.0739071098 -0.0810468117 -0.0483545504
## 11_1   -0.0479923167  2.121875e-02 -0.0027801654 -0.0454102278 -0.0645867814
## 11_3   -0.0090928628 -2.507241e-02 -0.0197848797 -0.0156920014 -0.0435047875
## 14_20  -0.0042284678  3.179350e-02 -0.0317249658  0.0138629159 -0.0279907424
## 14_21  0.0029200024 -3.495858e-02  0.0259953738 -0.0143430423 -0.0150564156
## 14_22 -0.1285579911  2.152324e-02 -0.1195045608 -0.0051207668  0.0623784336
## 14_23  0.0130545326 -2.387344e-02  0.0148885148  0.0137309668 -0.0136801653
## 14_25  -0.0090602373 -3.546910e-02 -0.0087583102 -0.0075890255 -0.0022062513
## 14_27  0.0088526741 -4.338712e-02 -0.0355808551  0.0193017356  0.0437090151
## 14_29 -0.0145785989  2.044758e-03  0.0290492251  0.0310848179  0.0052396337
## 14_30  -0.0453292450  3.663067e-02 -0.0175113966  0.0408386852 -0.0651481506
## 14_33  -0.0028989085  1.830224e-02  0.0071601695  0.0232847698 -0.0191967244
## 14_34  -0.0009717475 -1.297565e-01  0.1043967828  0.0107748847  0.0161961301
## 14_35  -0.0212928882  5.712336e-02  0.0526809144  0.0571969358 -0.0056969545
## 14_36  0.0435885129  2.733456e-02  0.1223316805  0.0520771588  0.0365093131
## 2_23   -0.0463360106  2.901307e-02  0.0256223932 -0.0148582281  0.0694025815
## 2_24   -0.0703591110  8.841357e-02 -0.0132064089 -0.0999728323  0.0902851229
## 2_25   -0.0271215165 -1.448721e-02 -0.0837133999  0.0172155148 -0.0477992575
## 2_26   -0.0025198476  2.263750e-02  0.0081950560 -0.0326351559  0.0498973352
## 2_27   -0.0432498999  4.785279e-02  0.0177893813 -0.0272923233 -0.0051779184
## 2_29   -0.0161081964 -8.317999e-02 -0.0008029324 -0.0249753226  0.0351724718
## 2_36   -0.0027771530  8.232039e-02 -0.0090762177 -0.0174539675 -0.0162413862

```

		Axis.16	Axis.17	Axis.18	Axis.19	Axis.20
## 4_65	0.0562819796	7.404567e-02	0.0376905936	-0.0170743189	0.0072113625	
## 2_39	-0.0587184171	-9.209758e-02	0.0050336040	-0.0241444951	0.0520006279	
## 2_40	-0.0403288029	-8.011571e-02	-0.0570942943	-0.0856355113	-0.0494390800	
## 2_41	-0.0983057775	-2.022741e-01	-0.0418496239	0.0527336063	-0.0529622387	
## 2_42	-0.0185976879	1.126362e-01	-0.0410782084	0.0272357668	0.1347795223	
## 2_47	-0.0671636130	-1.797766e-02	0.0341377184	0.0209774338	-0.0412352572	
## 2_48	-0.0881993651	-8.735985e-02	0.0049314801	-0.0945875779	0.0697189693	
## 2_49	-0.0480521477	-4.298519e-03	-0.0043157269	-0.1246739728	0.0261218565	
## 2_50	-0.0021437956	4.718068e-02	0.0228349918	-0.0715507566	0.0751394476	
## 2_51	-0.0166201398	1.042058e-02	-0.0313970378	0.0474254426	0.0133852664	
## 2_52	0.0583449817	1.608151e-05	0.0518755907	-0.0503573802	0.0914343232	
## 2_56	-0.0514206705	2.149194e-02	-0.0499748311	0.1359920704	-0.0639201278	
## 2_57	-0.1476783380	3.601613e-02	0.0525241760	-0.0060476371	-0.0225080260	
## 2_58	-0.0557482720	1.496641e-04	-0.1024517617	0.0804151303	-0.0351647942	
## 2_59	-0.0798253201	2.343980e-02	-0.1200453925	0.0251842947	0.1138644501	
## 2_60	-0.0186535523	1.537629e-02	-0.0592853432	0.0102576496	0.0255797165	
## 2_61	-0.0670979414	-3.043894e-02	0.0031407276	0.0284941677	-0.0244524367	
## 4_36	0.0117939519	1.450927e-02	0.0046333826	-0.0510173611	0.0372360066	
## 4_37	-0.1064184417	-3.665551e-02	0.0750867225	0.0090461419	0.0204386138	
## 4_38	-0.0743859551	4.065995e-03	0.0378945245	-0.0225363764	-0.0087398083	
## 4_39	-0.0495029074	-2.357072e-03	0.0068166103	-0.0809820490	-0.0105404386	
## 4_40	-0.0512089484	2.468711e-02	-0.0068425131	-0.0252863804	-0.0312736849	
## 4_41	-0.0200264556	-7.410152e-02	0.0791455988	0.0509205866	-0.0255150794	
## 4_54	-0.0016833403	5.487267e-02	-0.0756836650	0.0129427660	-0.0035465283	
## 4_55	-0.0495242910	-3.858375e-02	-0.0343438418	0.0403098201	-0.0334427079	
## 4_56	0.0452930911	5.839655e-03	-0.0350176796	-0.0169546638	0.0143291339	
## 4_57	0.1392092428	5.247804e-02	-0.1739281803	0.0306416908	-0.0241274239	
## 5_39	-0.0351177110	-6.701096e-02	0.0771190657	0.0491470005	0.0185710472	
## 5_40	-0.0286880189	1.066367e-01	-0.0046943940	-0.0476266951	-0.0036201496	
## 5_41	-0.0027888866	7.450643e-02	-0.1043798201	-0.0592786261	-0.0136179729	
## 5_54	0.0676980668	-1.922281e-02	0.0264686133	-0.0319845878	0.0436016947	
## 5_55	0.2273528567	-2.288772e-03	-0.0913500988	0.0367002708	0.0165891548	
## 5_59	-0.0013120512	1.013232e-01	0.0014070994	0.0156361643	-0.0199524168	
## 6_36	-0.0025798957	2.968852e-02	0.0874143044	-0.0052667215	-0.0093953214	
## 6_37	-0.0402050783	6.680371e-02	0.0627728346	0.0087822345	0.0261667569	
## 6_38	-0.0165808534	1.771134e-04	0.0102938587	-0.0046380017	-0.0278309455	
## 6_54	-0.0210459179	-9.507578e-02	-0.0835161643	-0.0134777145	0.1615387953	
## 6_55	-0.0043213156	-5.760802e-03	-0.0654466930	0.0125325884	-0.0930400891	
## 6_56	-0.0223432340	3.251457e-02	-0.0845179527	-0.0400805634	-0.0163937457	
## 6_57	0.0870192542	3.307033e-02	-0.1206432013	0.1067098452	-0.0284740405	
## 6_58	0.0633591251	2.717321e-02	-0.0735470886	-0.0091565181	-0.0216478739	
## 9_16	0.1164668888	7.336633e-02	0.0291433299	-0.0251944248	-0.1067809760	
## 9_17	0.1867564465	-3.670034e-02	-0.0420580518	-0.1116600935	-0.1251363619	
## 9_18	0.0042218893	-4.941190e-02	0.0973827975	0.0853266880	-0.0234076345	
## 9_19	0.1410999450	9.436572e-03	0.0851156267	-0.0429012211	0.0297104776	
## 9_21	0.0767077549	-6.769415e-02	-0.0121164055	-0.0290377867	-0.1687096640	
## 9_22	0.1274633768	1.139256e-01	0.0325191237	-0.0449136730	-0.0302691531	
## 9_34	-0.0139457547	-3.563981e-02	0.0548297081	0.0248347334	-0.0261440666	
## 9_35	0.0334400400	-5.431456e-02	0.0257538977	0.0066103517	0.0080267822	
## 9_36	-0.0337010771	-9.046220e-02	-0.0383788972	0.0179717304	-0.0388517594	
## 9_37	0.0110150137	-8.370455e-03	-0.0574540970	-0.0288745625	0.0153310647	
## 9_38	0.0709622702	-6.371792e-02	0.0262603050	-0.0442625212	0.0045756468	
## 9_39	0.0327720999	-1.163084e-01	0.0037646311	0.0706645000	-0.0246295907	

```

## 10_10 -0.0023332291 0.0356000249 -0.0356273505 0.0223717909 0.0016954134
## 10_11 0.0432467989 -0.0410510960 -0.0507233013 0.0300494531 -0.0249764843
## 10_12 -0.0033131273 -0.0448842206 0.0423310959 0.0096628554 0.0294622565
## 10_13 0.0883908952 0.0690991917 -0.0245712519 -0.0180591256 -0.0475786973
## 10_14 0.0711322027 0.0092293172 -0.0134120124 0.0402958500 0.0213654887
## 10_15 0.0469632860 0.0087434453 0.0007022688 0.0541744974 -0.0102667467
## 10_19 0.0283767581 -0.0063690680 -0.0005170752 -0.0300805526 -0.0387061515
## 10_1 0.0215086601 0.0074181035 0.0170413357 -0.0595037834 0.0362443321
## 10_20 0.0054437001 -0.0411923898 -0.0206615650 -0.0103370846 -0.0220322781
## 10_21 -0.0031972836 0.0432767200 0.0363485001 0.0904184966 0.0221585223
## 10_22 -0.1007385554 -0.0084528367 0.0214452999 -0.0211564168 -0.0229585538
## 10_25 -0.0183406269 -0.0273076981 0.0114767551 -0.0342342354 0.0410124142
## 10_26 -0.0407618416 -0.0523333577 0.0851833600 -0.0201618387 0.0148032340
## 10_28 -0.0055024935 -0.0003607640 -0.0470655869 0.0005317182 0.0242652083
## 10_29 -0.0125553461 -0.0254931522 0.0456701947 -0.0697316172 -0.0745084319
## 10_2 -0.0136412417 -0.0318010001 -0.0461493621 0.0625454694 0.0214647686
## 10_30 -0.0688093037 0.0319314665 0.0216942214 -0.0495592790 -0.0069871266
## 10_33 -0.0067458332 -0.0255270450 -0.0375458604 -0.1205589492 -0.0146920165
## 10_34 -0.0570149618 -0.0213638312 -0.0182714635 0.0547318933 0.0246499187
## 10_35 -0.0231007063 -0.0228231636 0.0833201668 -0.0043822150 -0.0246614602
## 10_39 0.0545240977 0.0044592054 0.0454503896 0.0153561458 0.0674774663
## 10_3 -0.0229331765 -0.0803362643 -0.0106868426 0.0235286754 0.0043905253
## 10_40 0.0197409835 0.0057266143 0.1573172936 -0.0221456664 -0.0121648835
## 10_41 -0.0467850376 0.0186694787 0.0189368864 0.0513970275 -0.0794500225
## 10_42 0.0166325275 -0.0092458753 0.0353919696 -0.0059910835 0.0214788915
## 10_43 0.0275235579 0.0017739882 -0.0470598960 0.0281172006 -0.0826500760
## 10_44 -0.0314986135 -0.0024253176 0.0319092723 0.0659257758 0.0121173415
## 10_48 -0.0729268515 -0.0109140188 -0.0453300957 -0.0195135384 -0.0212448895
## 10_49 0.0030336715 0.0596978126 0.0477399607 -0.1270514942 -0.0459440758
## 10_4 -0.0067233994 0.0624820942 -0.0212321680 -0.0259907725 0.0262260811
## 10_50 0.0187564976 0.0497129329 -0.0139596839 -0.0564545210 0.0017401657
## 10_51 -0.0167844153 0.0406974427 -0.0210512501 -0.0524762826 0.0028070168
## 10_52 -0.0879462425 0.0807698889 0.0932871474 0.0387768762 -0.0969165318
## 10_53 -0.0496671083 0.0207527789 -0.0276154671 -0.0594729337 -0.1245087520
## 10_57 -0.0184520596 -0.0919992226 0.0065162280 -0.0142826677 0.0349167226
## 10_58 0.0263551901 -0.0044394809 -0.0500172621 -0.0808689764 -0.0400178678
## 10_59 -0.1036435516 -0.0079469739 -0.0262679614 0.0657967368 0.0237002398
## 10_60 0.0421093314 -0.0105895205 -0.0632971566 -0.0638107907 0.0531326383
## 10_63 0.0385324314 -0.0548454102 0.0565241441 -0.0686274298 -0.0234378407
## 10_64 -0.0233754054 0.0117492818 0.0333634336 0.0776098920 0.0391736775
## 10_66 0.0502999466 0.0217670285 0.0299210055 0.0314706515 0.0224136068
## 10_67 0.0155762217 0.0261546443 -0.0828816824 0.0410033509 -0.0235100893
## 10_68 0.0070399582 -0.0086646984 -0.0243518538 0.0235150457 0.0633398343
## 10_69 0.0202843207 0.0442090296 0.0065840797 -0.0022020295 -0.0338928526
## 10_7 0.0600404695 0.0403577104 0.0194583615 -0.0824130470 -0.0340762746
## 10_8 0.0595465384 0.0164242119 0.0693317972 -0.0174411367 -0.0701214873
## 11_1 0.0217625789 0.0408162711 0.0258908123 0.0595320927 -0.0066718700
## 11_3 -0.0433293386 -0.0103902926 -0.0087271056 0.0041736282 0.0519761659
## 14_20 -0.0148071412 -0.0015399064 0.0145292888 -0.0338747423 0.0567324885
## 14_21 -0.0543095395 -0.0186106513 0.0013031731 0.0164338314 0.0193964389
## 14_22 -0.0533199559 0.0792480924 0.0403937428 -0.0942119170 -0.0119339620
## 14_23 -0.0575801288 0.0261228456 0.0205692436 0.0073096299 0.0378517448
## 14_25 -0.0327300250 0.0049277927 0.0085725274 -0.0143822532 0.0014045275
## 14_27 -0.0682373329 0.0728735892 0.0146401169 0.0261306640 0.0611416116

```

```

## 14_29 -0.0427172691 0.0331122089 0.0395641035 0.0212964254 0.0474535056
## 14_30 -0.0280993273 0.0099330606 -0.0037654576 -0.0585291192 0.0699448846
## 14_33 -0.0792723783 0.0202503514 0.0094262564 -0.0422298891 0.0387333462
## 14_34 -0.0337110330 -0.0556210219 0.0642500148 0.0010363616 -0.0059839368
## 14_35 -0.0426199057 0.0015595417 -0.0204703617 -0.0037733736 -0.0392063150
## 14_36 -0.0389375136 -0.0162050188 0.0017588123 0.0058370354 0.0371497950
## 2_23 0.0390193032 -0.0145445243 0.0002269906 0.0643512720 -0.0410078477
## 2_24 0.0705673687 0.0080120763 0.0325553872 -0.0629179900 -0.0111445940
## 2_25 0.0568359415 -0.0497233725 -0.0255288078 -0.0538468037 0.0877559363
## 2_26 0.0153288219 -0.0337960312 -0.0456196142 0.0092495223 0.0190618497
## 2_27 -0.0070444612 -0.0241224058 0.0447170486 0.0505521722 0.0039487885
## 2_29 0.0068548595 -0.0157670689 0.0209267172 0.0458550972 -0.0127144399
## 2_36 0.0310966433 -0.0279932113 0.0072726717 0.0091534089 0.0596365150
## 4_65 -0.0188772740 0.0721224259 -0.0052583807 0.0484365305 0.0138436682
## 2_39 0.0579596262 0.0113309187 0.0298729529 0.0309242523 0.0195419914
## 2_40 0.0230345286 0.0747599500 0.0600924120 0.0432765909 0.0473342593
## 2_41 0.0162474676 0.0942629206 0.0454373374 0.0178217631 0.0379867204
## 2_42 0.0819425889 -0.0868803238 0.0530639273 0.0341096930 -0.0382169602
## 2_47 -0.0358638999 -0.0023192575 -0.1009651741 -0.0082124861 -0.0536070602
## 2_48 0.0164943152 -0.0550300848 0.0337359351 -0.0249198480 0.0664497975
## 2_49 -0.0291688016 0.0134650792 -0.0619250983 -0.0138257931 -0.0201899290
## 2_50 -0.0241787432 -0.0003471372 -0.0675249198 0.0502767401 -0.0082729887
## 2_51 0.0084993121 -0.0129908849 0.0278118966 -0.0282012602 0.0373601185
## 2_52 -0.0416166927 0.0300145306 -0.0463901262 -0.0163017091 0.0168632926
## 2_56 -0.0189625918 -0.0486572378 -0.0232432063 0.0254781391 0.0197388963
## 2_57 0.0060782990 0.0922080514 -0.0754909946 -0.0283839914 0.0135697226
## 2_58 -0.0409971355 -0.0483421993 0.0563161011 -0.0254357069 0.0376119534
## 2_59 0.0352760939 0.0201767196 -0.1026149168 -0.0442208848 0.0898019281
## 2_60 -0.0369765381 -0.0600026448 -0.0297484752 0.0058338145 -0.0177929890
## 2_61 -0.0132513145 -0.0018638537 -0.0860025144 -0.0421561115 -0.0163107690
## 4_36 -0.0328590909 -0.0966391719 0.0281433492 0.0104108371 -0.0465498933
## 4_37 -0.0270015910 -0.0215683300 -0.0030245067 0.0417444057 0.0034424276
## 4_38 0.0292376867 0.0163262397 -0.0290339662 0.0427381650 -0.0316262853
## 4_39 -0.0195903329 -0.1305151652 0.0399967128 0.0107015218 -0.0431959213
## 4_40 -0.0084902352 -0.0087651048 0.0186619351 -0.0113207988 -0.0319853042
## 4_41 -0.0262220962 0.0358184634 0.0175580550 -0.0245868232 0.0393243245
## 4_54 -0.0257564688 -0.0659832523 -0.0086481883 0.0132137766 -0.0040367277
## 4_55 0.0592385374 -0.0124511009 -0.0470039354 -0.0109962753 0.0737649077
## 4_56 -0.0472530770 -0.0310471085 -0.0358312919 0.0126299758 0.0307365391
## 4_57 0.0593515702 0.0220991025 0.0294557331 0.0276589643 -0.0650024360
## 5_39 0.0392792133 0.0670753743 0.0788855931 -0.0502199758 0.0195174607
## 5_40 -0.0650309140 -0.0328636628 0.0226359617 0.0757747765 -0.0507659950
## 5_41 0.0489786309 -0.1126350691 0.0033545115 -0.0015526840 0.0459407219
## 5_54 -0.0513582686 0.0272158317 0.0178637087 0.0372509554 0.0232619517
## 5_55 0.1410569278 0.1353805595 0.0210559323 0.0302154997 0.0156358710
## 5_59 -0.1015867935 0.0548505020 -0.0544962176 0.0108826250 -0.0150430715
## 6_36 0.0189742048 0.0169887500 -0.0040880675 0.0128108555 -0.0519469112
## 6_37 -0.0291784092 0.0797633168 0.0276269462 0.0275745962 0.0066678924
## 6_38 0.0513502085 -0.0466028376 0.0264492626 -0.0198692290 -0.0710939486
## 6_54 -0.0388596287 0.0839805718 -0.0635968733 0.0353293779 -0.0502821569
## 6_55 -0.1206438282 0.0171200820 -0.0212571125 0.0554368787 -0.0705270522
## 6_56 0.0103640673 -0.0057326133 0.0580361142 -0.0208960160 0.0313288949
## 6_57 0.1557337669 0.1018724789 0.0084538393 0.0062333431 -0.0714008455
## 6_58 0.0781876158 0.0997589520 -0.0393947667 0.0723230112 0.0591339248

```

```

## 9_16 -0.0658925684 0.0315774583 -0.0644028992 -0.0562177227 0.0024628220
## 9_17 -0.0003045763 -0.0334672264 -0.0227867468 -0.0635088918 0.0557671854
## 9_18 0.0522929433 -0.0408010862 -0.0395858615 0.0109309257 0.0103681623
## 9_19 -0.0088940603 -0.0246546316 0.0471607096 0.0092045227 -0.0017920339
## 9_21 -0.0456624799 -0.0237329203 -0.0705464344 -0.0502231886 -0.0172950660
## 9_22 -0.0325063004 0.0435433298 -0.0355113951 0.0018883528 -0.0009111586
## 9_34 0.0729249426 -0.0197429639 0.0128327647 -0.0107476897 -0.0269016679
## 9_35 0.0761850676 -0.0885362773 -0.0243371840 0.0488018201 -0.0232754869
## 9_36 0.0876298982 -0.0301782307 -0.1042461218 0.0168250144 -0.0563549495
## 9_37 -0.0021675581 -0.0620757598 0.0370229333 -0.0315272026 -0.0216707463
## 9_38 0.0912691969 -0.0367469938 -0.0028957667 0.0525803533 0.0438341053
## 9_39 0.1050999601 -0.1094168101 -0.0628080615 0.0160892227 -0.0534979926
##          Axis.21      Axis.22      Axis.23      Axis.24      Axis.25
## 10_10 -0.0158079951 0.0120487737 -0.009800976 -1.081841e-03 0.015697511
## 10_11 -0.0076900423 0.0101867525 -0.026734371 -6.547222e-02 -0.037559206
## 10_12 0.0463713341 -0.0293752324 0.014747538 7.464731e-02 -0.079538920
## 10_13 -0.0184247853 0.0752503092 0.010053941 -9.426705e-02 0.064895981
## 10_14 -0.0031640635 0.0689488758 0.054275912 2.880017e-02 -0.038260393
## 10_15 -0.0100437661 -0.0049473102 0.003355939 1.688649e-03 0.114621364
## 10_19 0.0262100312 0.0075423551 0.059081735 4.021658e-02 0.040752437
## 10_1 0.0174582218 -0.0193841633 0.006025518 2.401854e-02 0.019949417
## 10_20 -0.0118153297 -0.0110660332 -0.003837340 8.164350e-03 -0.020802401
## 10_21 -0.0348132394 0.0611177686 -0.026711984 -1.117310e-04 -0.019924377
## 10_22 0.0119841094 -0.0211243519 0.060497886 6.865901e-03 -0.009792090
## 10_25 0.0642188079 0.0121055840 0.005914847 -1.999203e-02 0.001575345
## 10_26 -0.0052809497 -0.0437676331 -0.026291536 -2.278017e-02 0.030252911
## 10_28 0.0320660533 -0.0407955300 -0.018162966 -2.941813e-02 0.014218515
## 10_29 0.0146203248 -0.0631788056 0.029626811 -3.343399e-02 0.102545375
## 10_2 -0.0544397019 -0.0040463154 0.027135826 1.539817e-02 -0.011422167
## 10_30 -0.0195335708 -0.0569940082 0.026733509 -3.637156e-02 -0.013891955
## 10_33 -0.0346797602 -0.0471926207 -0.074155716 -1.068475e-02 -0.039752515
## 10_34 0.0016832657 0.0107262097 0.051732237 1.143275e-02 -0.052404049
## 10_35 0.0482815818 0.0011691227 -0.001617053 3.658733e-02 0.013968126
## 10_39 0.0141230441 -0.0032749228 0.042562476 -2.137608e-02 0.016400854
## 10_3 -0.0063900095 -0.0104239157 -0.010639232 -1.971928e-02 -0.013323250
## 10_40 -0.0159476212 0.1133176627 0.048440369 -5.019713e-02 -0.056972066
## 10_41 0.0169112831 0.0007125128 -0.027162722 -1.988533e-02 -0.031339554
## 10_42 0.0120946667 0.0175583542 0.012599183 -2.178937e-02 0.010961035
## 10_43 0.0466183452 0.0085138091 0.015925635 -1.076538e-02 -0.013598257
## 10_44 0.0453487687 -0.0374583444 -0.002431780 3.002066e-02 0.015046480
## 10_48 -0.0106776548 -0.0153171089 -0.047999813 2.410492e-02 -0.042177719
## 10_49 -0.0501215443 0.0158629810 -0.033001513 5.748707e-03 -0.081470192
## 10_4 -0.0681989613 0.0112625244 0.003846363 2.886199e-02 0.014497968
## 10_50 -0.0569143683 0.0067849694 0.014968703 1.208459e-02 0.013909322
## 10_51 -0.0575096934 0.0171583542 -0.026240039 7.274355e-02 -0.033344787
## 10_52 0.0044760619 -0.0148341865 0.004329158 -5.996709e-03 -0.093827828
## 10_53 -0.0397246742 0.0560507659 -0.019855937 -4.312649e-03 0.005625565
## 10_57 0.0420156418 -0.0243764504 -0.086290700 2.721872e-02 0.005355105
## 10_58 -0.0158862078 -0.0562089994 0.022279106 -2.286414e-02 0.018890640
## 10_59 -0.0315686572 -0.0131391556 -0.002896942 1.373462e-02 -0.008151492
## 10_60 0.0329120071 -0.0492170642 -0.033959987 -1.515377e-02 -0.022829923
## 10_63 0.0339277950 -0.0748729843 0.031709236 1.986841e-02 0.066777432
## 10_64 -0.0022149300 -0.0028767073 -0.021041923 1.334216e-02 0.009023425
## 10_66 0.0520421349 0.0303119911 0.003275292 -7.748524e-03 -0.018606014

```

```

## 10_67 -0.0360449980 0.0165982864 -0.041406445 3.946904e-02 -0.032106140
## 10_68 -0.0011339890 -0.0109129948 -0.021827283 -2.530810e-02 -0.029488477
## 10_69 0.0002927932 0.0434164110 0.025508359 -5.816514e-03 0.028272468
## 10_7 0.0092763537 0.0114766196 0.014919451 -1.884951e-02 0.015724478
## 10_8 0.0330676857 -0.0193126065 0.011161307 -1.006583e-02 0.008199639
## 11_1 -0.0733020603 0.0660763051 -0.019677675 -1.050016e-03 -0.016012689
## 11_3 0.0199973763 0.0348315007 -0.027743960 1.469712e-02 -0.026158848
## 14_20 -0.0158469236 0.0045944799 -0.031828742 6.727966e-02 -0.010136097
## 14_21 -0.0090632740 -0.0195188718 -0.073145363 -1.206802e-01 0.049941168
## 14_22 0.0552379026 0.0074526447 -0.025299797 4.007131e-03 0.024772598
## 14_23 -0.0118827621 0.0009520041 -0.009901811 -1.937819e-02 0.019881441
## 14_25 -0.0011162548 -0.0224644606 -0.069194110 -3.975382e-02 0.009208858
## 14_27 0.0059272894 0.0634232954 -0.077679469 -5.343065e-02 0.042450360
## 14_29 0.0254920657 0.0127059064 -0.046148582 -3.651287e-02 -0.003518233
## 14_30 0.0097481751 -0.0257559802 0.003107463 4.107099e-02 0.004833387
## 14_33 0.0059544654 -0.0423854246 0.070087025 3.785481e-02 -0.004857859
## 14_34 -0.0555463071 0.0663686711 -0.047186173 5.765294e-02 0.031495614
## 14_35 -0.0022732171 -0.0251641098 0.005906584 -8.321665e-03 -0.040139001
## 14_36 -0.0043850454 -0.0062560900 -0.040570971 -1.765301e-02 -0.041933836
## 2_23 0.0189337236 -0.0341214815 -0.031229146 1.721168e-03 -0.019046631
## 2_24 -0.0285227922 -0.0117734163 0.027030953 1.084164e-02 -0.006764289
## 2_25 -0.0604476577 0.0100128106 -0.036523090 -4.411057e-02 -0.023190090
## 2_26 -0.0293659996 -0.0197983101 -0.007628837 -4.771446e-03 0.013583913
## 2_27 0.0176810832 -0.0544163002 -0.009815166 6.702638e-03 0.036592740
## 2_29 0.0426676858 -0.0396887822 -0.015370910 1.700520e-02 0.016066313
## 2_36 -0.0467002539 -0.0654931592 0.010717844 9.576874e-03 0.032630368
## 4_65 -0.1197111127 -0.0482045333 -0.060415950 -8.574994e-03 0.069041587
## 2_39 -0.0096229586 -0.0124861497 -0.030023590 1.292724e-03 0.066038372
## 2_40 -0.0498729371 -0.0871644212 0.036311851 3.496462e-02 -0.055429567
## 2_41 -0.0569128806 -0.0779664578 0.032397437 -2.802329e-02 -0.061443890
## 2_42 0.0447506456 0.0027690465 -0.035939852 2.110435e-02 -0.036568748
## 2_47 0.0458492854 -0.0022019931 -0.025542821 -3.474520e-02 0.008999684
## 2_48 0.0629063536 0.0351810520 -0.006641336 9.490141e-02 0.019404754
## 2_49 0.0259181976 -0.0564409711 -0.046576447 -4.827207e-02 -0.007788873
## 2_50 0.0776943777 0.0170902113 0.061265856 -3.958342e-02 -0.049067842
## 2_51 0.0145876272 0.0171991224 -0.034745729 2.993049e-02 0.010024261
## 2_52 0.0497844215 -0.0083798459 0.030421490 1.584725e-02 -0.033413659
## 2_56 -0.0378455505 0.0235132784 -0.035097085 2.354483e-03 0.021854881
## 2_57 0.0359544925 -0.0591114335 0.025080511 -1.417645e-02 -0.026953866
## 2_58 0.0036141099 0.0381199480 0.072280579 -1.462347e-02 0.022660296
## 2_59 0.0142909721 0.1197406592 -0.039002797 1.749708e-02 0.018046873
## 2_60 0.0486788140 -0.0017775013 0.035187653 5.205656e-02 0.036465578
## 2_61 0.0076123477 0.0098477411 -0.007055204 1.674695e-02 -0.018292626
## 4_36 -0.0761702328 -0.0079966811 0.024670117 2.432481e-02 0.005838319
## 4_37 0.0766528395 0.0282129426 0.007060497 -5.919843e-02 0.001531290
## 4_38 0.0096734940 0.0046541629 0.012678505 2.456726e-02 0.040372405
## 4_39 -0.0749733263 0.0430897945 -0.010571450 2.762946e-02 -0.005156307
## 4_40 0.0300101898 0.0056436765 0.020760972 4.855477e-03 -0.018767765
## 4_41 0.0230322375 0.0270954086 0.036418533 2.568360e-02 0.022417328
## 4_54 -0.0426565966 0.0162047467 0.031294493 -8.432196e-02 -0.027198203
## 4_55 0.0383306202 -0.0086472352 0.042713563 -5.268434e-02 -0.030602672
## 4_56 -0.0497888175 -0.0238584044 0.044915859 -5.723801e-02 0.024942640
## 4_57 0.0417502478 -0.0292857380 -0.032027578 -1.014526e-02 0.021010639
## 5_39 0.0097790158 -0.0071881852 0.069385563 -4.265007e-02 -0.027224021

```

```

## 5_40  0.0016832225 -0.0332904578 -0.001230699 -3.699387e-02  0.004146804
## 5_41  -0.0777350985 -0.0430033495  0.046001603 -6.362696e-02 -0.045295959
## 5_54  0.0355721041  0.0271641865  0.033093328  5.380381e-03  0.051840508
## 5_55  0.0425237419 -0.0553660602 -0.025281349 -5.265669e-03  0.001085365
## 5_59  -0.0149558481 -0.0153134269  0.014407358  4.372226e-02  0.016202132
## 6_36  -0.0740566707 -0.0032801000 -0.053484607  8.734330e-02  0.016212597
## 6_37  -0.0334248875 -0.0527482105  0.001624340  7.947758e-03  0.022036235
## 6_38  -0.0659183073  0.0391554650 -0.030120854  3.751987e-03 -0.004806582
## 6_54  -0.0321652899  0.1050313815  0.061662845 -3.768277e-05 -0.005609489
## 6_55  0.1007018424  0.0128272804  0.017397511  2.737605e-02  0.017999736
## 6_56  -0.0067594281  0.0738393496 -0.002641153 -3.971260e-03  0.007706892
## 6_57  0.0633994750 -0.0027653085 -0.078727958  1.461417e-02 -0.025621875
## 6_58  -0.0112296085 -0.0005788337  0.049627296  7.266100e-02  0.030452399
## 9_16  -0.0016095055 -0.0167588363 -0.005066838  6.764435e-02  0.014762375
## 9_17  0.0380477778  0.0879536447  0.029128287 -1.935771e-02  0.004924266
## 9_18  0.0482707495 -0.0059949267 -0.040879049 -1.387971e-02 -0.058210792
## 9_19  -0.0354392060  0.0471762316 -0.011811399 -1.125549e-02  0.030951162
## 9_21  0.0667096824  0.0678115190  0.016152063 -1.067745e-02  0.040160863
## 9_22  -0.0226031550 -0.0091427015  0.039847447 -1.375971e-02 -0.022850912
## 9_34  0.0332112834  0.0361384750 -0.045449454  2.166687e-02 -0.030848814
## 9_35  0.0364088062 -0.0238106179 -0.007059529  2.701472e-02 -0.000872566
## 9_36  -0.0997296173 -0.0211936923  0.105180057  4.374517e-03  0.033600237
## 9_37  -0.0190125948  0.0055542888  0.036544664 -1.094603e-02 -0.011169496
## 9_38  0.0275023113  0.0009571774  0.019842110 -1.858368e-02  0.006891138
## 9_39  -0.0038686739 -0.0116194978  0.025494189  4.103525e-02 -0.020703896
##          Axis.26      Axis.27      Axis.28      Axis.29      Axis.30
## 10_10  0.030263437  4.764671e-02  3.787716e-02  0.0100970242  0.0270984982
## 10_11  -0.034981275 -8.105895e-03 -5.932651e-02  0.0357432982 -0.0789567965
## 10_12  -0.028760934 -1.166858e-02  1.764336e-02  0.0198820874  0.0419640766
## 10_13  0.051976069 -1.516320e-02 -5.661271e-03 -0.0151355615  0.0499016682
## 10_14  0.018428521 -2.935402e-02  1.396401e-02 -0.0464180948  0.0369319277
## 10_15  -0.005068534 -4.775559e-02 -8.001332e-03 -0.0693976582  0.0087892232
## 10_19  0.013656408  1.106856e-02  3.794283e-03  0.0125788437 -0.0190066000
## 10_1  -0.005732492 -6.603096e-02 -3.783943e-02  0.0061525218  0.0047228500
## 10_20  0.024305857 -2.382396e-02  4.444728e-02  0.0368334544 -0.0020807729
## 10_21  -0.024431601  4.434520e-02 -1.678626e-02 -0.0251601376 -0.0161259055
## 10_22  -0.018238917 -1.566991e-02  2.446618e-02 -0.0309238585 -0.0668038886
## 10_25  -0.004153804 -3.860870e-02 -5.515122e-02  0.0261568848  0.0461383717
## 10_26  0.055610652  4.000811e-02  2.696723e-02 -0.0002905896  0.0041884376
## 10_28  -0.043186177  4.610157e-03 -4.163587e-02 -0.0339364878 -0.0088658712
## 10_29  0.059423361  6.739815e-02  2.013099e-02 -0.0033060146 -0.0431494614
## 10_2  -0.013029999  9.477013e-02  2.173771e-02 -0.0323101653 -0.0147657137
## 10_30  0.021585015  1.058386e-02  5.362908e-03 -0.0010699516  0.0360495149
## 10_33  -0.012878702  5.230846e-02  2.110660e-02  0.0330920046  0.0988595748
## 10_34  -0.035631838 -9.688134e-04  6.866743e-03 -0.0104932347  0.0676523459
## 10_35  -0.035018515  1.483701e-03  1.642917e-02 -0.0439718196  0.0189721937
## 10_39  -0.007435762 -1.022776e-02  1.282628e-02  0.0035083528  0.0082793293
## 10_3  -0.010508615  8.120678e-02 -1.123356e-02 -0.0062415467  0.0175302498
## 10_40  -0.040878005 -4.167281e-02  2.825962e-02  0.0036558672 -0.0009491929
## 10_41  0.041112782  2.661644e-02 -1.697777e-02 -0.0338833758  0.0020505423
## 10_42  -0.008283878 -3.243298e-02  5.447685e-03  0.0002022531 -0.0024862227
## 10_43  0.009818531 -1.371101e-02  3.006045e-02 -0.0292090347 -0.0114374416
## 10_44  0.040008029  3.442840e-02 -5.716482e-05 -0.0608759913 -0.0040140112
## 10_48  0.010248828  1.398549e-03 -4.311438e-02 -0.0138662314 -0.0396959097

```

```

## 10_49 -0.023963877 -1.029519e-02 2.638583e-02 -0.0099970476 -0.0190912395
## 10_4 -0.045301912 2.192435e-02 2.496354e-02 0.0042952586 0.0093038195
## 10_50 0.016273167 2.620298e-02 -9.784698e-03 0.0336048848 0.0052522303
## 10_51 -0.026658303 3.148442e-02 -1.742413e-02 0.0289508557 -0.0856380461
## 10_52 0.033803895 -1.454396e-02 -1.849882e-02 0.0290902063 0.0122035857
## 10_53 -0.055049234 9.219634e-03 -6.956195e-02 -0.0006674499 0.0020364628
## 10_57 -0.006528085 -4.247890e-03 4.271587e-03 0.0115316393 -0.0067925679
## 10_58 0.055466613 -2.032351e-02 1.495342e-02 0.0110805372 -0.0007489709
## 10_59 0.035858597 -3.379257e-02 -1.086086e-02 -0.0190569040 0.0287850739
## 10_60 0.010923950 -6.022820e-02 -1.208335e-04 0.0275547389 0.0100801118
## 10_63 -0.021499453 1.916986e-02 -2.983652e-03 0.0344376264 -0.0319884524
## 10_64 0.047245282 -1.757863e-02 1.587439e-02 -0.0204248211 0.0205406124
## 10_66 -0.002456438 -4.958617e-02 -1.314940e-03 -0.0201121048 -0.0295505330
## 10_67 -0.096090098 1.623038e-04 -1.863792e-02 -0.0587810474 0.0072027741
## 10_68 -0.030768072 3.418407e-02 -4.604972e-02 -0.0247727116 -0.0114634701
## 10_69 0.019158323 -4.215155e-02 1.262530e-02 0.0188990800 0.0240261792
## 10_7 -0.035750312 -4.553174e-02 8.737530e-03 0.0329431254 -0.0143031655
## 10_8 -0.009492031 -5.087586e-02 -1.837126e-02 0.0274943418 -0.0231006000
## 11_1 -0.000909424 4.588009e-02 7.701039e-03 0.0273805954 0.0091992546
## 11_3 -0.015715723 -2.892549e-02 -4.552036e-02 0.0138273388 0.0198479636
## 14_20 0.015461787 -2.175395e-02 -9.473130e-03 -0.0140987489 0.0028620397
## 14_21 -0.075455374 -7.037470e-03 1.923514e-02 0.0168307353 0.0631480881
## 14_22 -0.018366326 -1.015167e-02 -1.895324e-02 -0.0295167156 -0.0003984244
## 14_23 -0.003739469 -1.048075e-03 -2.677995e-03 -0.0246404555 -0.0205128187
## 14_25 0.028374659 -3.830063e-03 3.237090e-02 0.0082936613 -0.0145042250
## 14_27 0.008945240 -2.376463e-02 1.314248e-02 -0.0356777332 -0.0477082010
## 14_29 0.037826849 1.618177e-02 3.872875e-02 -0.0171105501 -0.0313559502
## 14_30 -0.006577415 -4.324942e-03 5.382732e-03 0.0174427806 0.0228917165
## 14_33 0.014350485 -1.531668e-02 1.939900e-02 -0.0460166206 -0.0158617822
## 14_34 0.031472134 2.149159e-05 -2.495116e-02 -0.0001224042 0.0267312064
## 14_35 -0.004831192 2.209604e-02 -1.410293e-02 0.0010550612 0.0155332860
## 14_36 -0.045426001 6.285756e-03 4.804829e-02 0.0348195735 -0.0197484156
## 2_23 0.030026377 3.758247e-02 6.581831e-02 0.0024640419 -0.0139740969
## 2_24 -0.061263898 1.149688e-01 -2.987927e-03 -0.0329622006 0.0433242240
## 2_25 0.014454423 -1.270195e-02 -4.003026e-02 -0.0208294841 -0.0337030534
## 2_26 -0.048715378 1.331341e-02 5.313824e-03 -0.0188541037 0.0157134721
## 2_27 -0.014670190 4.082942e-02 2.026817e-02 -0.0277355463 0.0239654645
## 2_29 -0.043204086 -3.331597e-02 -1.735473e-02 0.0226153169 -0.0266364288
## 2_36 0.007836777 4.901981e-02 -7.209558e-02 0.0006662893 -0.0080320799
## 4_65 -0.020624805 -5.518445e-02 -3.847466e-02 0.0064893025 -0.0074778884
## 2_39 -0.035761218 -1.468819e-02 5.163998e-02 -0.0160935292 0.0084314109
## 2_40 0.022956780 9.845036e-03 1.996079e-02 0.0737353983 0.0067157363
## 2_41 0.031024320 -4.220958e-02 3.485508e-02 0.0096410955 0.0101561987
## 2_42 -0.007540631 6.308018e-03 8.653415e-03 0.0005332244 0.0068346515
## 2_47 -0.016322748 1.453514e-02 -4.560700e-02 -0.0248650066 0.0097127060
## 2_48 0.001953798 -2.317575e-03 -1.464022e-02 -0.0032456212 -0.0030184988
## 2_49 0.033820070 -2.454661e-03 3.557093e-02 0.0045055231 -0.0203660386
## 2_50 0.036616290 8.316657e-03 -8.806263e-03 0.0164974593 0.0022303857
## 2_51 -0.015904206 -1.770139e-03 -1.112463e-02 0.0209821572 0.0003768028
## 2_52 0.016537309 -1.880900e-02 -3.354779e-03 -0.0115783594 -0.0103241444
## 2_56 0.024839731 -7.010282e-03 -4.203476e-02 0.0595289562 -0.0274533560
## 2_57 -0.022734702 1.004902e-02 -2.538313e-03 -0.0172002139 -0.0032228892
## 2_58 -0.089987147 -2.269986e-02 -2.413416e-02 -0.0072499415 0.0180614291
## 2_59 0.072473504 -7.686702e-03 3.082945e-02 0.0221409737 0.0039467470

```

```

## 2_60 -0.028416069 -2.045732e-02 1.734075e-03 -0.0135557938 -0.0247067372
## 2_61 0.029598084 -1.998434e-02 -6.729383e-03 -0.0541141759 -0.0033942982
## 4_36 -0.005069883 -4.911295e-02 4.213281e-03 0.0092463812 0.0342550365
## 4_37 -0.024331163 3.467848e-02 2.292585e-02 0.0610590421 -0.0164809158
## 4_38 0.009623408 2.578447e-02 3.045061e-02 0.0689659374 0.0235099023
## 4_39 0.030790232 -7.324001e-03 -7.943909e-03 -0.0191060374 0.0080516464
## 4_40 0.024319884 4.112281e-02 -8.671304e-02 0.0187089236 0.0295963769
## 4_41 0.016631930 1.179374e-02 1.372769e-02 0.0133567132 -0.0194929502
## 4_54 0.004525023 4.126913e-03 2.251823e-02 -0.0103433887 0.0092253468
## 4_55 0.016434844 3.551450e-02 -2.990898e-05 -0.0116273360 -0.0283039619
## 4_56 -0.048198547 -1.883902e-02 8.749531e-03 0.0191851041 0.0109287135
## 4_57 0.028347406 -2.981228e-02 -7.794654e-03 0.0128523212 0.0196481955
## 5_39 0.050820102 4.269787e-02 -9.542856e-02 -0.0775533704 0.0145287052
## 5_40 -0.037334872 -2.261312e-02 3.866131e-02 0.0143033809 0.0031080457
## 5_41 0.022497276 -6.757579e-02 4.177804e-02 -0.0450978616 -0.0240836612
## 5_54 -0.052167278 3.042603e-02 6.701406e-03 0.0660972948 -0.0125225101
## 5_55 -0.039414109 5.441755e-04 -1.551107e-02 -0.0187739184 0.0134674456
## 5_59 0.021215101 -1.496091e-02 -2.442669e-02 0.0125044884 -0.0188585932
## 6_36 0.061931188 -1.885173e-02 1.653551e-02 -0.0053579722 0.0220850244
## 6_37 -0.012731316 -5.042053e-02 1.107971e-03 -0.0140978199 -0.0107856018
## 6_38 0.044161034 1.078433e-02 1.019606e-02 -0.0545363299 -0.0054712836
## 6_54 -0.019953189 -3.544919e-02 2.170182e-02 0.0167415467 -0.0071101455
## 6_55 0.010787009 7.563739e-03 -1.755009e-02 0.0315515661 0.0025823481
## 6_56 0.005879541 3.794875e-02 -2.315828e-02 0.0749036181 0.0044139165
## 6_57 -0.048191427 2.587107e-02 1.850173e-02 -0.0109827862 0.0038733084
## 6_58 0.001853231 5.319278e-02 1.655534e-02 0.0142037742 -0.0243869239
## 9_16 -0.005725921 -3.575503e-02 6.684564e-02 -0.0096350624 0.0327378686
## 9_17 0.001300733 3.979317e-02 8.427953e-02 -0.0312414191 -0.0093843589
## 9_18 0.025370985 -7.059555e-03 2.942660e-02 0.0495363993 -0.0188920948
## 9_19 0.022236490 1.374760e-02 -3.485654e-02 0.0427233240 -0.0465273570
## 9_21 0.011906429 1.337414e-02 2.679918e-02 -0.0174379288 0.0509762001
## 9_22 0.0462228510 7.974060e-03 -6.242739e-02 0.0103433303 -0.0026237058
## 9_34 0.020575200 3.232773e-03 3.510155e-02 -0.0200484047 -0.0329289925
## 9_35 -0.002758525 -4.708420e-02 -1.991448e-02 -0.0307558039 -0.0068465698
## 9_36 -0.022047000 -7.334813e-04 2.125946e-02 0.0375710715 -0.0141347677
## 9_37 -0.008912140 3.303537e-02 1.464123e-02 -0.0200094003 -0.0639971244
## 9_38 0.078951316 -9.456220e-03 -8.492595e-02 0.0377637338 0.0451494544
## 9_39 -0.025344574 -3.182011e-02 -2.086353e-02 0.0035215621 -0.0001342933
##          Axis.31      Axis.32      Axis.33      Axis.34      Axis.35
## 10_10 -0.0216157186 -0.0014796128 3.724202e-03 -0.0146148471 0.032687485
## 10_11 0.0239525812 0.0052179231 5.384619e-03 0.0074516610 -0.008927484
## 10_12 -0.0221704769 0.0073494917 -1.808826e-03 0.0289439673 0.042063348
## 10_13 -0.0254874859 -0.0158994817 2.398100e-02 0.0233138801 -0.066943703
## 10_14 0.0243998186 -0.0291232830 1.379005e-03 -0.0099877187 -0.030731524
## 10_15 -0.0381668473 0.0357124416 -2.516154e-02 -0.0320695127 0.002265042
## 10_19 0.0058881988 -0.0327888304 9.305025e-03 0.0011744779 0.008220742
## 10_1 0.0373106684 0.0112715875 -3.514994e-02 -0.0391186169 0.011437611
## 10_20 -0.0281004956 -0.0007675236 -4.899576e-02 -0.0548187637 0.003449188
## 10_21 -0.0174675204 -0.0065572018 -4.782740e-02 0.0050980831 0.025475160
## 10_22 0.0291709347 0.0151506544 -5.522031e-03 -0.0127205087 0.007592483
## 10_25 -0.0053952376 -0.0143248388 3.210417e-02 0.0165331960 0.023127753
## 10_26 0.0160285438 0.0062040552 5.539105e-03 0.0148762267 -0.017140103
## 10_28 0.0490655709 0.0736778062 1.710901e-02 0.0487731300 0.027798762
## 10_29 0.0224659583 -0.0276796599 1.044735e-02 0.0177062591 -0.008180128

```

```

## 10_2 -0.0132926583 -0.0426091317 3.054214e-02 -0.0022720639 -0.005495394
## 10_30 -0.0063473480 0.0264489997 -6.918166e-02 -0.0156021875 0.025550218
## 10_33 0.0191642774 0.0433281373 -3.711019e-02 0.0276909239 -0.028711841
## 10_34 0.0208676789 0.0380963183 -6.555806e-03 -0.0044259308 0.006139046
## 10_35 -0.0002431682 0.0073764481 1.791834e-02 -0.0114731567 0.019547124
## 10_39 -0.0085009797 -0.0556262926 -8.182482e-03 0.0182993268 0.042656809
## 10_3 -0.0297347887 -0.0200209621 1.491705e-02 0.0042619225 -0.007129538
## 10_40 -0.0394796146 0.0027235833 9.348831e-05 -0.0087842746 -0.008592242
## 10_41 0.0235622432 0.0114619018 1.297490e-02 -0.0005971391 0.050160766
## 10_42 0.0012421474 0.0231653657 -1.049300e-02 0.0541101595 -0.070449047
## 10_43 0.0427387805 -0.0242951290 -1.808499e-02 -0.0289393268 -0.016840119
## 10_44 0.0606551208 0.0127737638 3.224241e-02 -0.0056792180 -0.001406506
## 10_48 -0.0354552737 -0.0245405344 -3.547521e-02 -0.0191851383 -0.032591013
## 10_49 -0.0283034539 -0.0043342431 3.973014e-03 -0.0207221770 -0.002271485
## 10_4 0.0340433029 0.0241037016 3.326168e-02 -0.0370987798 0.041115411
## 10_50 -0.0142709570 0.0397143177 9.078948e-03 0.0091215398 -0.011366700
## 10_51 -0.0406434065 -0.0082367258 4.854180e-02 0.0019427518 -0.005478996
## 10_52 0.0186749447 -0.0010092042 -4.004759e-03 0.0066344358 0.016866443
## 10_53 -0.0206531246 0.0053664562 1.538416e-02 0.0095355667 0.012589594
## 10_57 0.0246240401 0.0555241149 -2.138510e-02 -0.0580197741 -0.026472807
## 10_58 0.0108811199 -0.0083154637 1.510736e-02 -0.0158906776 0.020245389
## 10_59 -0.0002273590 0.0068159400 5.355898e-03 0.0135710514 0.022133620
## 10_60 -0.0157627233 0.0449227734 -2.497728e-02 0.0010700894 -0.002393428
## 10_63 0.0182596476 -0.0362177654 -1.531444e-02 0.0216192799 0.034966190
## 10_64 -0.0041609410 0.0263107747 1.411079e-02 -0.0130443757 -0.014421974
## 10_66 0.0287095543 0.0091862011 1.967777e-02 0.0093163075 -0.013865713
## 10_67 -0.0078048709 0.0080695220 2.242907e-02 0.0195889416 -0.031460886
## 10_68 -0.0033370966 -0.0153292706 -2.764336e-02 0.0478628629 -0.002834686
## 10_69 0.0055317475 -0.0221767253 1.947833e-02 0.0103128940 -0.012963863
## 10_7 0.0577428489 0.0344352222 1.270788e-02 -0.0142976201 0.033562410
## 10_8 -0.0191838070 0.0327123502 -5.930888e-02 0.0244926033 0.015124384
## 11_1 -0.0213118862 0.0130024949 -1.435847e-02 -0.0289346455 0.016227212
## 11_3 -0.0411242803 -0.0311433723 3.189178e-02 0.0108485925 -0.002197613
## 14_20 -0.0004261351 -0.0479306402 -5.376331e-02 -0.0088207979 -0.030739797
## 14_21 0.0137609118 -0.0421587666 4.419419e-02 -0.0125947320 -0.011671077
## 14_22 0.0106209303 -0.0051943963 2.300560e-02 -0.0247426326 -0.029409617
## 14_23 -0.0002295891 0.0013702779 3.025452e-03 0.0236306992 0.013995487
## 14_25 -0.0738085956 -0.0325518877 3.251976e-02 0.0066071428 -0.021205284
## 14_27 0.0175743513 -0.0205310271 -5.534157e-02 -0.0333662064 -0.020485640
## 14_29 -0.0143987523 -0.0284290321 -1.418045e-02 -0.0088149403 0.052684590
## 14_30 0.0101025061 -0.0112747129 6.910538e-02 -0.0054868261 0.016961461
## 14_33 -0.0044217813 -0.0229264236 2.439321e-03 0.0274569432 -0.021592625
## 14_34 0.0391032344 0.0150803592 2.215886e-02 0.0064510986 -0.013510496
## 14_35 0.0159026918 0.0329107110 -2.365755e-02 -0.0455610880 0.008289941
## 14_36 0.0443306811 -0.0393833474 -2.659021e-02 -0.0065288753 0.013166109
## 2_23 -0.0754269614 0.0282735308 -6.088887e-03 0.0279724498 0.022950889
## 2_24 -0.0027517832 -0.0484824575 -2.122751e-02 0.0104683809 -0.008115839
## 2_25 0.0035364405 -0.0005739397 -2.305979e-02 0.0193551763 0.026307493
## 2_26 0.0260759899 -0.0451584382 -3.802226e-03 -0.0052590268 -0.005847257
## 2_27 0.0074385730 -0.0012233488 -2.488951e-03 0.0097457825 -0.018830251
## 2_29 -0.0265329356 -0.0138020867 1.528448e-02 -0.0091571601 -0.004354987
## 2_36 0.0294956964 0.0337323792 -7.669033e-03 -0.0058156644 -0.011191500
## 4_65 0.0160051045 0.0107900154 1.017753e-02 0.0104348558 0.018175152
## 2_39 -0.0396235983 0.0380244492 7.328975e-03 -0.0285361587 0.032628151

```

## 2_40	0.0399120384	-0.0329314065	4.104718e-03	0.0101600876	-0.024247101
## 2_41	0.0429350859	-0.0031256175	6.455841e-03	0.0180077909	-0.045880302
## 2_42	0.0182548141	0.0293090139	1.698205e-02	-0.0206887491	-0.008976384
## 2_47	0.0186127368	-0.0685829858	-4.445624e-02	-0.0015462862	0.013528137
## 2_48	0.0077279068	0.0014863877	2.572074e-02	-0.0314883491	-0.012502377
## 2_49	-0.0229601957	-0.0050424066	3.427618e-02	-0.0219400148	-0.010266111
## 2_50	0.0009599282	0.0057318498	1.243672e-02	-0.0016769053	-0.003652949
## 2_51	-0.0351716385	0.0062030070	2.930627e-02	-0.0023633353	0.016605611
## 2_52	-0.0175715281	-0.0015871614	6.538040e-02	-0.0287871659	-0.008152387
## 2_56	-0.0207164682	0.0234069077	1.573559e-02	-0.0119729744	-0.013192977
## 2_57	-0.0260956114	-0.0247914189	-2.722321e-02	0.0071516499	0.018444667
## 2_58	-0.0429118997	-0.0099374529	-5.632516e-02	-0.0165363925	-0.004156209
## 2_59	0.0593244275	-0.0090936916	-1.521886e-02	0.0178377416	0.021265435
## 2_60	-0.0288060046	0.0160196517	-5.728549e-03	0.0360758218	0.001031091
## 2_61	-0.0135669280	0.0279509709	1.530252e-02	0.0526216280	0.028159873
## 4_36	-0.0252450986	0.0116691936	8.648079e-03	0.0613246872	-0.027026471
## 4_37	0.0015713601	0.0303308127	8.378404e-03	-0.0127138985	-0.017938080
## 4_38	-0.0247153623	0.0214894765	-2.224003e-02	0.0272735184	-0.010902069
## 4_39	0.0303688869	-0.0100074111	8.700475e-03	-0.0163342683	0.037713989
## 4_40	-0.0075492777	0.0042656844	1.573178e-02	-0.0187097011	0.006610374
## 4_41	-0.0048686426	-0.0068167563	-1.575680e-02	0.0295574864	-0.003297023
## 4_54	0.0345226937	0.0416202537	-8.587539e-03	0.0020897621	-0.040048992
## 4_55	-0.0164957698	0.0278670643	-1.589827e-02	0.0050293481	-0.003418128
## 4_56	-0.0064637004	0.0009720485	-1.486112e-02	-0.0278960679	-0.016580402
## 4_57	-0.0296433813	-0.0295992070	-2.054529e-02	-0.0056350443	0.008517097
## 5_39	-0.0497018222	0.0222007429	1.508908e-03	-0.0502983548	-0.006454551
## 5_40	0.0253309262	-0.0150250947	1.297514e-02	-0.0174362105	0.020019186
## 5_41	-0.0373370886	-0.0091369982	4.639527e-02	-0.0050419995	0.027836020
## 5_54	0.0026298607	-0.0268136752	-4.208151e-03	-0.0290065604	-0.027194142
## 5_55	0.0333571914	-0.0119230564	4.324503e-02	0.0035949600	0.004308583
## 5_59	0.0141510871	0.0006370473	-9.677128e-03	0.0010762386	0.001297618
## 6_36	-0.0012877559	-0.0114827394	-1.448818e-02	-0.0148607846	-0.031403013
## 6_37	-0.0310822602	0.0081899664	1.544824e-02	0.0366374947	0.003382540
## 6_38	0.0202727753	-0.0032697355	-2.005110e-02	0.0226067011	0.036689989
## 6_54	0.0515035132	-0.0105712970	-9.863383e-03	0.0288307956	0.028338054
## 6_55	-0.0123040934	0.0189751854	-1.020142e-02	0.0242438894	-0.021831614
## 6_56	0.0072201248	-0.0021201109	1.824303e-02	0.0138549003	0.003122171
## 6_57	0.0155174187	0.0153903331	2.866296e-03	0.0201169088	0.001041385
## 6_58	-0.0194016513	0.0542364994	-3.785797e-02	-0.0092962644	-0.016226241
## 9_16	-0.0167646092	-0.0398713161	-2.687496e-02	0.0117364479	-0.002312276
## 9_17	0.0011163426	0.0338881579	8.801942e-03	0.0165961256	0.013126055
## 9_18	-0.0195986064	-0.0035637812	5.401438e-03	0.0102956673	0.003793314
## 9_19	-0.0020418084	0.0006298147	-3.298545e-03	0.0390975950	0.017815502
## 9_21	-0.0140068245	0.0230606992	1.346106e-02	-0.0098944041	0.012225232
## 9_22	-0.0135429890	0.0120892459	2.690455e-02	-0.0432117815	-0.023662316
## 9_34	0.0010187533	-0.0198667832	9.215634e-03	-0.0253928390	-0.025236025
## 9_35	-0.0184265515	-0.0051158925	-2.759069e-02	0.0195102678	-0.023083049
## 9_36	-0.0190214924	0.0161449678	1.979275e-02	-0.0432259212	0.016694169
## 9_37	0.0496696262	0.0131344895	-8.361655e-03	0.0220642288	-0.051123944
## 9_38	0.0080675640	-0.0521413863	-2.125921e-02	0.0011391542	0.035073532
## 9_39	0.0301888093	-0.0526904016	5.665139e-03	-0.0258687219	-0.018214793
	Axis.36	Axis.37	Axis.38	Axis.39	Axis.40
## 10_10	8.986671e-03	-0.0367075169	-0.0048582724	1.725201e-02	-0.0359394593
## 10_11	4.089622e-02	0.0223304578	-0.0037453684	-9.648611e-03	-0.0050172937

```

## 10_12 -2.886887e-03 -0.0040882491 -0.0164616186 -1.004118e-02 0.0081759706
## 10_13 -1.000591e-02 0.0017286599 0.0076097295 -1.613319e-02 0.0323800218
## 10_14 1.489147e-02 0.0427426367 -0.0200796276 -3.077564e-02 0.0847503094
## 10_15 -2.059532e-02 0.0117785022 -0.0056190236 1.284468e-02 0.0002091321
## 10_19 -2.206183e-03 0.0300124610 0.0081972103 4.614363e-03 -0.0024606235
## 10_1 -1.082531e-02 -0.0281479753 0.0323226029 7.162523e-03 -0.0199041815
## 10_20 -1.366399e-02 -0.0028481945 -0.0362411866 -3.105456e-03 -0.0003025540
## 10_21 -1.162277e-02 -0.0030819366 0.0101029149 1.326689e-02 0.0184100877
## 10_22 3.177098e-03 -0.0269575372 -0.0189036494 2.568273e-04 -0.0106801814
## 10_25 -8.696261e-03 -0.0061455633 0.0216044574 -1.354576e-02 -0.0086950153
## 10_26 2.768719e-02 -0.0249046587 0.0224723535 4.478980e-03 0.0233306463
## 10_28 -3.599196e-02 0.0125329841 0.0001619634 -2.018870e-02 -0.0176171863
## 10_29 -3.925151e-04 0.0018082740 -0.0163890475 6.645721e-04 -0.0031137515
## 10_2 -1.427810e-02 0.0163996291 -0.0160652699 -4.297369e-03 -0.0114050522
## 10_30 -1.360795e-03 0.0239719624 0.0003515223 3.347106e-02 0.0271450148
## 10_33 5.346531e-02 -0.0084226641 -0.0092426970 2.095557e-02 0.0135603107
## 10_34 -1.648580e-02 -0.0208966833 0.0153782611 2.244612e-02 -0.0012868068
## 10_35 6.541661e-02 0.0244012386 -0.0161426234 -2.518018e-04 -0.0255472619
## 10_39 -1.939034e-02 -0.0474222430 0.0017250095 -9.579723e-03 0.0034210664
## 10_3 -9.696069e-03 0.0119384439 0.0126675722 4.549898e-03 -0.0055668205
## 10_40 -3.388978e-02 0.0197158629 -0.0177033406 1.777921e-02 -0.0421947245
## 10_41 1.293899e-02 0.0003149673 0.0049965660 -1.885469e-02 -0.0026769409
## 10_42 -1.066189e-03 -0.0289272689 -0.0012741413 8.634763e-03 0.0220218465
## 10_43 -5.836389e-03 -0.0290856692 -0.0252513090 2.036891e-03 -0.0356700761
## 10_44 -2.265904e-02 0.0303739094 0.0110428882 3.528218e-02 0.0147974100
## 10_48 6.042750e-03 -0.0403449527 -0.0247778583 1.628741e-02 0.0075202146
## 10_49 4.760637e-02 0.0329404517 0.0218862402 -2.461001e-02 0.0178440950
## 10_4 -4.143096e-02 0.0266845646 -0.0202281676 4.580046e-02 0.0386698916
## 10_50 -3.078518e-03 0.0183490563 -0.0117896035 1.718753e-02 -0.0165144493
## 10_51 -3.687218e-02 0.0179421428 -0.0116024316 3.817621e-02 0.0125301640
## 10_52 -3.009688e-02 0.0015930765 0.0107563965 -9.853156e-03 0.0139675123
## 10_53 -2.122940e-03 0.0128526179 0.0245154661 -5.143527e-03 -0.0415157623
## 10_57 -1.178551e-02 0.0126519006 0.0082897435 2.283571e-02 0.0163693244
## 10_58 1.453487e-02 0.0222645769 0.0008226774 -4.209203e-02 0.0104915522
## 10_59 1.048828e-02 -0.0040310103 -0.0142610315 -1.044345e-02 0.0092127976
## 10_60 8.657090e-03 -0.0015860405 -0.0527493989 -2.498445e-02 -0.0362543597
## 10_63 -1.878396e-02 -0.0104258596 0.0118911270 -1.199986e-02 0.0414436207
## 10_64 -1.488107e-03 0.0223495853 -0.0013171768 2.870439e-02 -0.0391104603
## 10_66 1.886227e-02 0.0079150352 0.0131599604 -1.132838e-02 -0.0184020217
## 10_67 -3.052774e-02 -0.0060505911 0.0423349460 1.749176e-02 -0.0016297364
## 10_68 2.089022e-02 0.0019703235 -0.0506255117 -2.703313e-02 0.0114519723
## 10_69 3.642343e-03 0.0065451953 0.0007839915 -1.289007e-02 -0.0345061531
## 10_7 -1.270836e-02 -0.0131340981 -0.0133149205 2.605940e-02 0.0189165327
## 10_8 -3.865413e-03 -0.0263874202 0.0291325073 4.202639e-03 0.0087761243
## 11_1 5.318480e-03 -0.0053659003 -0.0081888009 1.788969e-02 -0.0038766119
## 11_3 2.272185e-02 0.0012540734 0.0062965655 -1.430748e-03 0.0191442031
## 14_20 -5.602598e-04 0.0414122834 -0.0164373390 -2.597159e-02 -0.0030365730
## 14_21 -3.099039e-02 0.0181339464 -0.0427993597 -9.271042e-03 0.0258807948
## 14_22 -3.468284e-02 0.0299571399 0.0010039150 2.371908e-02 -0.0215355675
## 14_23 -4.517878e-03 -0.0090950530 -0.0103308360 -3.333901e-02 0.0267195397
## 14_25 -3.944376e-02 -0.0212486992 -0.0151789680 8.302357e-04 -0.0094639690
## 14_27 -2.182351e-03 0.0076489703 0.0314562528 -3.070981e-03 0.0174030591
## 14_29 -1.234034e-02 -0.0399846621 -0.0205396705 -2.135162e-02 -0.0004324508
## 14_30 -1.664911e-03 0.0013201168 -0.0178513130 5.269752e-03 -0.0541853794

```

```

## 14_33 -1.069702e-02 0.0037362951 -0.0014664743 -1.067387e-02 -0.0404027470
## 14_34 -2.543416e-03 -0.0324503250 -0.0078630969 -3.970729e-02 0.0014963999
## 14_35 -4.845810e-02 0.0054035130 -0.0055013440 -3.128149e-02 0.0246067020
## 14_36 -1.379323e-02 0.0408555860 0.0122130484 -3.737161e-02 -0.0262379623
## 2_23 -2.216465e-02 0.0041208136 0.0266105429 -9.261661e-03 -0.0067834677
## 2_24 -2.871560e-02 -0.0100380582 0.0159587595 -2.567922e-02 -0.0139284106
## 2_25 1.825772e-02 -0.0178028174 -0.0194334940 1.945511e-03 -0.0080376438
## 2_26 4.721422e-04 -0.0511672676 0.0065620386 1.099415e-02 -0.0295541469
## 2_27 4.770453e-02 -0.0336410409 0.0164610259 4.307128e-02 0.0145467335
## 2_29 2.697319e-03 -0.0367851262 0.0061954344 3.854240e-02 0.0268002801
## 2_36 3.992056e-05 0.0193056767 0.0128465085 -1.424161e-02 -0.0243380271
## 4_65 2.953208e-02 0.0211990914 0.0246889777 -8.775732e-03 -0.0234074874
## 2_39 7.646484e-03 0.0103163873 -0.0199016485 -3.271126e-02 -0.0108270644
## 2_40 -5.882114e-03 0.0280986362 0.0216269740 8.726892e-03 -0.0013699248
## 2_41 -1.775526e-02 0.0071718717 0.0097537683 -1.913287e-02 -0.0028443718
## 2_42 -1.460041e-02 -0.0136512164 0.0439411739 -5.742044e-03 -0.0042342309
## 2_47 3.197109e-02 -0.0025120501 0.0005380511 8.957501e-03 -0.0082914430
## 2_48 -7.245684e-03 -0.0295507604 -0.0314751600 -1.206920e-02 0.0200746955
## 2_49 -1.637496e-02 0.0051267113 0.0190281141 -3.785083e-03 -0.0002886786
## 2_50 1.525142e-02 0.0166044340 -0.0400065313 8.490837e-03 0.0075617770
## 2_51 3.866877e-02 0.0136510659 0.0502662515 -5.178364e-02 0.0123112384
## 2_52 3.581007e-03 0.0016440811 0.0357275395 2.571417e-02 0.0310275256
## 2_56 -7.655724e-03 0.0080747169 -0.0111553479 6.773303e-03 0.0144957023
## 2_57 -1.720736e-02 0.0106729710 0.0117683983 2.123818e-02 0.0058825017
## 2_58 -3.071512e-02 0.0134748830 -0.0330230570 1.152460e-02 -0.0036361434
## 2_59 -2.925498e-02 0.0017064387 0.0234853417 2.879282e-02 -0.0010324388
## 2_60 -1.755747e-02 0.0111311735 0.0044010515 1.289311e-02 0.0210414281
## 2_61 -7.455979e-03 0.0062339866 0.0183258144 -9.281734e-03 0.0038673492
## 4_36 -1.313265e-03 -0.0128643936 -0.0112098584 1.913431e-02 -0.0170058872
## 4_37 1.408113e-02 0.0305323637 -0.0106505031 1.634919e-03 -0.0052705786
## 4_38 1.037599e-03 0.0252740404 -0.0263018277 1.581873e-02 -0.0142856884
## 4_39 -1.747429e-03 -0.0219815842 0.0003118853 -1.273677e-02 0.0094239928
## 4_40 -3.7877779e-02 0.0028937684 -0.0155445785 -1.536216e-02 0.0180620299
## 4_41 4.623582e-02 -0.0121811527 -0.0180834622 2.319858e-02 -0.0001636984
## 4_54 -3.229029e-02 -0.0244379632 0.0283393938 -7.363795e-03 -0.0129876356
## 4_55 1.311018e-02 0.0094353889 0.0198275952 1.513692e-02 -0.01254558945
## 4_56 7.051621e-03 -0.0314001929 0.0527297690 -5.288795e-05 0.0126327224
## 4_57 6.397317e-03 0.0220566629 0.0179967043 -2.208963e-03 -0.0021449705
## 5_39 5.254881e-02 -0.0190795012 0.0073108362 -2.830940e-03 -0.0067625390
## 5_40 3.085517e-02 0.0256712296 -0.0243155474 -1.933878e-02 -0.0055478774
## 5_41 2.653811e-02 -0.0045031478 0.0116277776 2.911634e-02 0.0044367752
## 5_54 2.855356e-02 0.0165972534 0.0388910663 6.340866e-03 -0.0090225865
## 5_55 1.921952e-03 -0.0290344798 -0.0527163345 1.414717e-02 0.0046052362
## 5_59 -3.381074e-02 -0.0154880062 -0.0107548201 -9.602805e-03 0.0114616756
## 6_36 1.851558e-03 -0.0165430245 -0.0378187699 2.394860e-03 -0.0138493661
## 6_37 -6.085359e-03 -0.0025803891 0.0013045873 -1.127963e-02 0.0259234484
## 6_38 -2.346494e-02 0.0445262822 0.0109229417 2.930104e-02 -0.0014702714
## 6_54 4.969785e-02 -0.0360693527 -0.0069450179 8.700471e-03 -0.0034528736
## 6_55 3.847736e-02 -0.0028244143 0.0059038216 -1.656206e-02 -0.0107571693
## 6_56 7.109792e-03 -0.0263116193 0.0172610066 -9.957926e-03 0.0054512269
## 6_57 -1.045530e-02 -0.0255293874 -0.0243885364 1.151331e-02 0.0132991871
## 6_58 2.333462e-04 -0.0096407547 0.0202450190 -7.046983e-02 -0.0114578465
## 9_16 8.059381e-03 -0.0002025065 0.0351488570 1.341769e-02 0.0083662321
## 9_17 -9.513728e-03 0.0216610243 0.0023065462 -8.767127e-03 0.0063763082

```

	##	9_18	-1.124629e-02	-0.0281632577	0.0107921543	2.705824e-03	0.0043185335
	##	9_19	-6.836685e-03	0.0088485765	-0.0184821619	1.505797e-02	0.0143757576
	##	9_21	-1.143575e-02	-0.0031211593	0.0004013232	-2.348839e-02	0.0043642562
	##	9_22	-5.443066e-03	-0.0394687760	-0.0261873927	-5.865602e-03	-0.0037283583
	##	9_34	3.495642e-02	-0.0248968601	0.0228291081	2.819919e-03	-0.0243814455
	##	9_35	2.326903e-02	0.0586617528	-0.0204610482	5.494483e-03	0.0012133451
	##	9_36	-1.392411e-03	-0.0114912758	-0.0038824326	-7.993380e-03	0.0205846060
	##	9_37	1.501848e-03	0.0053785330	-0.0296407471	9.645088e-03	0.0009624612
	##	9_38	-1.622394e-02	0.0388571251	0.0164925428	5.266206e-02	-0.0218643594
	##	9_39	-9.881453e-03	-0.0020550722	0.0151741651	-2.944158e-02	-0.0036586841
			Axis.41	Axis.42	Axis.43	Axis.44	Axis.45
	##	10_10	-9.264133e-04	0.0078569517	-4.072830e-03	0.0304974749	-0.0025972617
	##	10_11	6.207454e-03	-0.0179885941	-9.923356e-03	-0.0256672385	-0.0015918648
	##	10_12	-1.723538e-03	-0.0089390069	2.548871e-02	-0.0021091836	-0.0128560810
	##	10_13	-4.663392e-02	0.0353867547	-1.406541e-02	-0.0271528963	0.0159789693
	##	10_14	-1.403505e-02	-0.0070183109	1.750220e-02	-0.0005424524	0.0259856579
	##	10_15	4.065140e-02	0.0258903806	4.715783e-03	-0.0056721619	-0.0013083887
	##	10_19	2.419398e-02	-0.0031539921	2.179378e-04	0.0425896744	-0.0086309385
	##	10_1	-1.081842e-02	0.0308964826	-1.352018e-02	-0.0056314212	-0.0201446326
	##	10_20	-9.190663e-03	-0.0136652612	3.208696e-02	-0.0160412117	0.0227342038
	##	10_21	-1.965260e-02	-0.0064149889	-1.941991e-02	0.0058924541	0.0142861743
	##	10_22	-1.627374e-02	0.0089181234	-2.444487e-03	0.0015770868	0.0068962041
	##	10_25	6.296621e-03	-0.0060358211	-9.617133e-03	-0.0339346240	-0.0112648140
	##	10_26	-5.224980e-03	0.0245899234	-5.020644e-03	-0.0058266732	0.0037086096
	##	10_28	2.117940e-02	-0.0341859387	3.964332e-02	-0.0094540466	0.0158544158
	##	10_29	1.140478e-02	-0.0026164998	-2.222013e-03	0.0072239351	0.0118998395
	##	10_2	5.061508e-03	-0.0225070853	1.580238e-02	-0.0231047831	-0.0052125690
	##	10_30	7.095601e-03	0.0057805980	-1.284993e-02	-0.0124778342	0.0075064834
	##	10_33	-1.412655e-03	-0.0329448413	-1.072382e-02	-0.0021112043	-0.0090296217
	##	10_34	-2.610343e-02	0.0007393283	3.303905e-02	0.0045957573	-0.0051506182
	##	10_35	-4.747465e-02	-0.0148585410	2.553830e-02	-0.0092562712	-0.0130297382
	##	10_39	-2.373611e-02	-0.0149850608	-5.327230e-03	0.0069056877	-0.0054755406
	##	10_3	4.521642e-03	-0.0150357654	-1.230324e-02	0.0075848172	0.0119875331
	##	10_40	2.299976e-02	-0.0389394376	-2.334636e-02	0.0053971692	0.0098295009
	##	10_41	-2.244831e-02	0.0152673366	1.846938e-02	-0.0244024291	-0.0029548259
	##	10_42	-1.032978e-02	-0.0295245069	2.613365e-02	-0.0158687956	0.0225083836
	##	10_43	1.206826e-02	-0.0280033810	8.730586e-03	0.0292628408	-0.0150039646
	##	10_44	1.402263e-02	-0.0052087228	-1.756359e-02	0.0050399659	-0.0028339447
	##	10_48	1.427554e-02	0.0358774871	3.184529e-02	-0.0115865170	-0.0253576909
	##	10_49	1.537125e-02	-0.0014801720	-2.496462e-02	0.0078734148	-0.0471765066
	##	10_4	3.507746e-02	-0.0146613465	-5.468710e-03	0.0003081549	0.0250536174
	##	10_50	-5.290822e-03	0.0187681687	-4.766384e-02	0.0168413536	0.0230047976
	##	10_51	6.265154e-03	0.0133114800	4.272268e-02	0.0038459743	0.0011615871
	##	10_52	-9.094953e-03	0.0030087687	8.537319e-03	-0.0065544252	0.0107189277
	##	10_53	-3.200111e-02	-0.0038018010	3.186140e-03	0.0124040213	0.0108365529
	##	10_57	-4.845452e-02	0.0070056423	-1.954424e-02	-0.0021625442	-0.0107558252
	##	10_58	3.686767e-02	-0.0348532088	-1.619173e-02	-0.0182721456	-0.0084810256
	##	10_59	1.419714e-02	0.0148746957	2.102050e-03	0.0050876014	-0.0028925586
	##	10_60	-1.884466e-02	0.0069798188	4.781276e-03	0.0260650109	0.0482555698
	##	10_63	4.488814e-03	-0.0007011281	9.753267e-03	0.0035875011	-0.0052848459
	##	10_64	5.482076e-05	0.0137720149	1.778442e-02	-0.0019949266	0.0133615548
	##	10_66	1.666825e-02	0.0201098265	4.048646e-03	-0.0036534507	-0.0029362444
	##	10_67	1.163994e-02	0.0106166153	-6.172334e-03	0.0176860129	-0.0015387546
	##	10_68	-1.161416e-02	-0.0075994023	-1.781530e-02	0.0237551373	-0.0109413938

```

## 10_69  2.156430e-02 -0.0175099458  9.768723e-03 -0.0325835628 -0.0217103912
## 10_7   -3.195341e-03 -0.0099825611  3.818244e-02  0.0234605657  0.0017730814
## 10_8   7.942706e-03  0.0296999414 -7.225823e-03  0.0109222813 -0.0173315603
## 11_1   -1.343542e-02  0.0063660611  7.186108e-03 -0.0026595824 -0.0140798152
## 11_3   -6.212697e-04  0.0067156142 -2.614155e-02  0.0091901406  0.0143714684
## 14_20   1.285476e-02  0.0149703864  2.398658e-03 -0.0024986955  0.0131995566
## 14_21   -7.550762e-03  0.0090895254  2.063407e-02  0.0318066359 -0.0319132061
## 14_22   -2.040149e-02 -0.0306046481  6.404640e-03 -0.0054047742 -0.0068373470
## 14_23   2.781550e-02 -0.0026476724 -1.351040e-02  0.0239768083 -0.0259929054
## 14_25   -4.361232e-03 -0.0056931044  5.247002e-03 -0.0305716398  0.0051328182
## 14_27   -9.255684e-03 -0.0515778949 -4.518643e-03  0.0078536548 -0.0249069887
## 14_29   -2.255560e-03  0.0160498494 -1.541110e-03  0.0077735117  0.0587292407
## 14_30   -5.080194e-03  0.0058683830 -1.907561e-02  0.0002008090  0.0082587201
## 14_33   -3.761142e-02  0.0312679490 -2.061682e-02 -0.0112533546 -0.0274672877
## 14_34   1.988589e-02 -0.0070802148 -1.552833e-02  0.0006953961 -0.0117273428
## 14_35   1.904625e-02  0.0054914201 -9.641170e-03 -0.0158262555  0.0045203524
## 14_36   1.524906e-02  0.0480594448 -6.666703e-03 -0.0131206240  0.0007235461
## 2_23    -5.902737e-03  0.0016709640 -2.218310e-02  0.0025078979 -0.0405283887
## 2_24    -6.504696e-03  0.0075102812  2.206912e-02  0.0033902748  0.0108713805
## 2_25    3.797459e-02  0.0145184861  2.928042e-02 -0.0270729095 -0.0067248242
## 2_26    -1.646387e-03  0.0096567771 -1.212809e-02 -0.0225359953  0.0017201466
## 2_27    2.525191e-02  0.0044250831 -5.742023e-04  0.0066790974 -0.0020857986
## 2_29    1.331103e-02 -0.0178338854  8.578995e-03 -0.0285707201 -0.0036444710
## 2_36    -2.287768e-02  0.0016642634 -2.199476e-02 -0.0190636373  0.0036805638
## 4_65    -1.035062e-02 -0.0311488694  1.567325e-03 -0.0097507479  0.0063612203
## 2_39    -1.316465e-02 -0.0095175671  2.071513e-03 -0.0131275495  0.0141235829
## 2_40    4.899277e-03 -0.0204446988 -8.961506e-03 -0.0057769532  0.0172505589
## 2_41    -9.321862e-03  0.0146241456 -6.368028e-03  0.0358834710 -0.0042894539
## 2_42    4.236647e-03  0.0059732509 -1.603625e-02  0.0244637717  0.0149551220
## 2_47    1.888858e-02 -0.0114435151 -2.048286e-02  0.0344475306  0.0106898931
## 2_48    2.246321e-02 -0.0006277448 -9.343175e-03 -0.0059516121 -0.0027195663
## 2_49    2.517411e-02  0.0144838884  1.218470e-02  0.0002479103  0.0199921119
## 2_50    7.829635e-03 -0.0174036698  9.458124e-03 -0.0220142487 -0.0406639473
## 2_51    -1.588278e-02  0.0307933688  4.726368e-02  0.0104038171 -0.0025664405
## 2_52    1.704759e-02  0.0143372330 -1.998434e-02 -0.0002533390  0.0098785007
## 2_56    -6.276098e-03  0.0398675731  2.540962e-03  0.0211791703 -0.0137323188
## 2_57    -1.315285e-02 -0.0051206387  1.019635e-02 -0.0118904352  0.0045268101
## 2_58    2.778216e-02 -0.0073318368 -3.892196e-02 -0.0109702133 -0.0039526641
## 2_59    -1.148228e-02 -0.0191079054  1.448017e-02  0.0226547350 -0.0255419982
## 2_60    -2.064933e-02 -0.0170268785 -2.388008e-02  0.0192620779  0.0047541838
## 2_61    -3.929579e-03 -0.0134744027 -1.891108e-02 -0.0114400753 -0.0040211082
## 4_36    2.212423e-02  0.0027401199  2.357197e-02  0.0208403938 -0.0060778043
## 4_37    7.176867e-04  0.0243470972 -1.605331e-03  0.0091744648 -0.0005913693
## 4_38    4.158104e-02  0.0098547360 -4.013091e-03 -0.0278546634 -0.0037267009
## 4_39    -2.353955e-02 -0.0215601783 -2.978685e-02 -0.0325333780 -0.0013152392
## 4_40    -1.029853e-02 -0.0019500401 -1.068679e-03  0.0013387661  0.0104146594
## 4_41    -2.533237e-02  0.0026406659 -7.648148e-03  0.0073206810  0.0054756039
## 4_54    3.221247e-02  0.0096047467  2.453558e-03  0.0059697014 -0.0057441529
## 4_55    -1.192044e-02  0.0085004375  2.724523e-03  0.0082241435  0.0007154705
## 4_56    1.106251e-03 -0.0151516157  1.480233e-02 -0.0028959736  0.0176145984
## 4_57    -1.878766e-02 -0.0208702904  2.801722e-02  0.0164371810  0.0046124797
## 5_39    1.108776e-02  0.0015768910  3.020599e-02  0.0134013492 -0.0057244456
## 5_40    1.600396e-03  0.0290245830 -7.080759e-03  0.0171141617 -0.0168312884
## 5_41    -2.446610e-03 -0.0220496206 -1.107388e-02  0.0122917823  0.0124884291

```

```

## 5_54 -1.460299e-02 -0.0078216658 8.987695e-03 -0.0199517352 0.0102330850
## 5_55 -2.256671e-02 0.0125785723 -1.449093e-02 -0.0023723667 -0.0170276529
## 5_59 -1.589244e-02 -0.0161125629 -2.350691e-02 -0.0222351782 0.0145471397
## 6_36 -7.636643e-04 -0.0218235286 9.596041e-03 0.0145581211 -0.0159907169
## 6_37 1.554941e-02 -0.0050869499 1.094211e-03 0.0015270330 -0.0057173362
## 6_38 4.028354e-03 0.0102890175 1.588775e-02 -0.0026242131 0.0198416930
## 6_54 -6.372743e-04 0.0433630426 -7.637763e-03 -0.0180752884 0.0164300196
## 6_55 1.624370e-02 -0.0261675461 1.350647e-03 0.0146239388 0.0216373145
## 6_56 4.078857e-02 0.0033859664 9.770076e-03 -0.0095671724 0.0095519040
## 6_57 2.038331e-02 0.0252880594 -3.051250e-02 -0.0259720935 0.0059340735
## 6_58 1.602958e-02 -0.0267415483 1.105524e-02 0.0108699958 -0.0139117358
## 9_16 1.141182e-03 0.0170083081 1.859410e-02 -0.0094835986 0.0021353307
## 9_17 1.144981e-02 0.0145667902 6.288657e-04 0.0016554974 -0.0065989241
## 9_18 -3.068680e-02 -0.0475050424 -1.701191e-02 -0.0047271359 -0.0061588634
## 9_19 -3.234451e-02 -0.0012530635 -1.066212e-03 0.0380102438 0.0028723498
## 9_21 -8.657146e-03 0.0128924800 -5.847366e-06 -0.0041016921 0.0016309441
## 9_22 9.205602e-03 -0.0079063982 -8.512073e-03 0.0275410295 -0.0007803777
## 9_34 1.108663e-02 -0.0130133476 -1.177717e-02 -0.0113743213 0.0158162451
## 9_35 -7.368378e-03 0.0169154035 -1.189958e-02 -0.0062440523 0.0125348891
## 9_36 -4.641927e-02 0.0067061431 -1.079817e-02 -0.0062597143 -0.0273603102
## 9_37 -3.139311e-02 0.0115140235 2.880964e-02 -0.0266390290 -0.0114693730
## 9_38 8.738983e-03 0.0138999663 9.267863e-03 -0.0092109324 -0.0237284390
## 9_39 4.950573e-03 0.0062625299 5.512207e-03 0.0260176125 0.0220745336
## Axis.46 Axis.47 Axis.48 Axis.49 Axis.50
## 10_10 -1.484388e-02 0.0116810620 -0.0042000780 0.0184746995 -0.0216825576
## 10_11 -1.348295e-02 0.0211617102 0.0090430637 0.0081854615 0.0048716449
## 10_12 1.071979e-02 -0.0099203340 -0.0184990868 -0.0180099122 -0.0074969796
## 10_13 6.289617e-03 -0.0038567970 -0.0150130019 0.0094881050 -0.0084120497
## 10_14 1.296066e-02 0.0114520337 -0.0134580754 0.0122743702 0.0051236318
## 10_15 3.191425e-02 -0.0054720000 0.0009752121 -0.0248845332 -0.0024330269
## 10_19 -6.269902e-03 0.0135445620 0.0059450393 0.0198385181 0.0237867779
## 10_1 -5.849524e-03 0.0118076827 -0.0338003727 0.0001629153 -0.0365909754
## 10_20 -9.707946e-03 0.0017943410 -0.0124931484 -0.0151309800 0.0006017609
## 10_21 -1.716273e-03 0.0156524350 -0.0010275342 -0.0084454105 -0.0137988828
## 10_22 3.137907e-02 -0.0133926595 -0.0119079014 0.0163124297 -0.0120846292
## 10_25 -1.099517e-02 0.0115213006 -0.0018824749 -0.0077234358 -0.0155128455
## 10_26 -4.338100e-03 -0.0228099968 0.0085300022 -0.0137926027 -0.0056411644
## 10_28 -4.358256e-02 0.0195207590 -0.0202861190 0.0016743272 -0.0175261067
## 10_29 2.856940e-03 -0.0044101131 -0.0043240465 0.0035855830 -0.0111121180
## 10_2 -4.172217e-03 0.0004130800 -0.0016345841 -0.0089505341 0.0154680853
## 10_30 -1.808141e-02 0.0133094699 0.0076758590 0.0090940799 0.0206030941
## 10_33 1.370191e-03 -0.0123843537 0.0107677267 0.0063563841 -0.0003217793
## 10_34 1.169620e-02 -0.0122557512 0.0129342763 -0.0065625936 0.0190162243
## 10_35 -1.235089e-02 -0.0034818099 0.0006764428 0.0213660348 -0.0095495172
## 10_39 1.962448e-02 0.0084693906 -0.0111048108 -0.0148084430 0.0038336544
## 10_3 -6.806070e-03 -0.0082574572 -0.0022587231 -0.0040005797 -0.0071498715
## 10_40 8.016782e-03 -0.0139365342 -0.0028892793 0.0105269162 -0.0045485634
## 10_41 2.164555e-02 -0.0081836023 -0.0070304045 -0.0160621261 0.0317935674
## 10_42 -2.611139e-02 -0.0097565830 0.0192147472 0.0037877972 -0.0262650150
## 10_43 -1.804738e-02 0.0006366237 0.0086628698 -0.0047905569 -0.0081790285
## 10_44 -2.542264e-02 0.0332465665 0.0181354520 -0.0036220883 -0.0008337754
## 10_48 -1.681357e-02 -0.0035535985 -0.0192822345 -0.0037896131 0.0248629144
## 10_49 3.604586e-02 0.0183498776 0.0175618210 -0.0391774956 -0.0091757776
## 10_4 9.031723e-03 -0.0201734091 -0.0077064439 0.0116888626 -0.0176238275

```

```

## 10_50 -2.083591e-02 0.0249365694 -0.0319940117 -0.0081383485 0.0251881265
## 10_51 1.800550e-02 -0.0154532031 -0.0004345498 0.0365538917 -0.0237187174
## 10_52 -9.860358e-03 -0.0027737190 -0.0009159213 -0.0024319350 0.0029977831
## 10_53 -3.372304e-03 -0.0335382895 -0.0085487677 0.0001740035 0.0061182621
## 10_57 2.768675e-02 0.0019510088 -0.0071763715 0.0007880734 0.0031545415
## 10_58 4.814453e-03 0.0042342843 -0.0048097777 -0.0059248266 -0.0176853944
## 10_59 -1.117578e-02 0.0097983500 0.0004260272 0.0098348709 0.0035114975
## 10_60 2.220385e-02 0.0046450958 0.0122887827 0.0292713322 0.0131618718
## 10_63 -5.068598e-03 0.0058059196 -0.0040400395 0.0044498141 -0.0093055493
## 10_64 -1.326805e-02 -0.0037528007 -0.0033255553 0.0084448743 0.0121748423
## 10_66 -2.149648e-02 -0.0100984347 -0.0066776295 -0.0113291879 0.0105457803
## 10_67 3.042970e-04 0.0111696278 0.0138533503 -0.0114536734 -0.0105432948
## 10_68 1.591373e-02 0.0175037582 -0.0081009026 0.0028474019 -0.0128981472
## 10_69 8.119564e-04 -0.0265895150 0.0078230068 0.0092441453 0.0209457402
## 10_7 1.326928e-02 0.0066517169 0.0357616078 -0.0146795551 0.0283369400
## 10_8 -2.386082e-02 -0.0080321554 0.0062479528 0.0207031277 0.0071157560
## 11_1 -1.321539e-03 0.0254342539 -0.0056931783 0.0009661390 -0.0169721416
## 11_3 -1.642767e-02 -0.0108772636 0.0040355825 0.0067544682 0.0174051723
## 14_20 -3.981534e-02 0.0230983997 0.0239120620 0.0048520160 0.0129248129
## 14_21 -7.521703e-03 0.0039239058 0.0173869763 -0.0016368043 -0.0025152685
## 14_22 6.371375e-03 0.0051928250 -0.0217854699 0.0008461897 0.0159797345
## 14_23 1.924417e-03 -0.0198250917 -0.0001112675 0.0333596553 0.0171480171
## 14_25 -4.815385e-03 0.0243273576 0.0254281562 0.0048921153 0.0153556488
## 14_27 1.000060e-02 -0.0051819502 -0.0247095944 0.0126585681 0.0079156448
## 14_29 2.387686e-02 -0.0116597863 0.0145553652 -0.0210546057 -0.0084598957
## 14_30 8.010562e-03 0.0062681253 -0.0137159707 -0.0099006415 -0.0103714362
## 14_33 8.932659e-03 0.0166075782 0.0094677078 -0.0077190626 -0.0072633831
## 14_34 -3.966388e-03 -0.0050252073 0.0004760730 0.0066320898 -0.0002686314
## 14_35 7.625027e-03 -0.0020259407 -0.0022671855 0.0094738037 -0.0189032689
## 14_36 -7.029312e-03 -0.0158946006 0.0102005688 0.0072704952 0.0045400958
## 2_23 -6.453755e-03 -0.0242263776 0.0197013377 0.0030292008 -0.0128918570
## 2_24 2.046171e-03 0.0003639479 -0.0130719921 0.0166169086 -0.0006207187
## 2_25 -3.572878e-04 0.0145187082 -0.0018395211 -0.0192121560 0.0041882871
## 2_26 3.856603e-03 0.0144523388 0.0172641933 0.0069300457 -0.0131366831
## 2_27 1.677740e-02 0.0027646285 -0.0062243620 -0.0004949324 0.0210348616
## 2_29 6.596959e-04 0.0117453685 -0.0162180426 0.0048094485 -0.0036827686
## 2_36 3.416534e-03 -0.0096632987 0.0181259127 -0.0074745312 0.0068057464
## 4_65 2.945930e-02 0.0040651879 0.0057884815 -0.0063105722 0.0081829094
## 2_39 -1.705905e-02 -0.0119442367 0.0093939220 -0.0078034768 -0.0019739905
## 2_40 6.934842e-03 -0.0076395136 0.0069630020 0.0043989366 -0.0033419832
## 2_41 -2.393169e-03 -0.0006389287 -0.0102560748 0.0010090603 -0.0129157689
## 2_42 8.229899e-03 0.0172283026 -0.0055089364 0.0004189333 0.0056226752
## 2_47 -2.706815e-03 -0.0382255729 -0.0325582025 -0.0106981294 0.0090964466
## 2_48 1.231617e-02 -0.0100258780 0.0014587595 0.0142304130 0.0047283797
## 2_49 4.006807e-03 -0.0057469837 0.0008653400 -0.0011091061 0.0030874501
## 2_50 2.356500e-02 0.0202847986 -0.0029737374 0.0069409462 -0.0142814559
## 2_51 -9.601260e-03 0.0068524684 -0.0167824434 0.0110428631 0.0012401705
## 2_52 -2.985851e-02 -0.0099584411 -0.0120806317 -0.0301712122 -0.0036884447
## 2_56 9.804787e-03 0.0045543275 0.0034391499 0.0166678389 -0.0158882940
## 2_57 -2.898094e-03 0.0034723135 -0.0026878290 0.0011912559 -0.0030079911
## 2_58 -2.330785e-02 0.0045389231 0.0084665328 -0.0100378804 0.0041418318
## 2_59 -2.249499e-02 -0.0254988499 0.0193142044 -0.0227847728 -0.0136469918
## 2_60 8.786573e-03 -0.0032796919 0.0455334680 -0.0010806057 0.0120846244
## 2_61 2.013391e-02 0.0055353409 0.0209459218 0.0085916711 0.0056372350

```

```

## 4_36  2.004966e-02 -0.0011199995 -0.0149880246 -0.0172458155  0.0047997777
## 4_37 -1.729912e-03  0.0072351034  0.0071622315 -0.0152555364  0.0029043957
## 4_38  1.416653e-02  0.0150147982 -0.0171612265 -0.0135615637 -0.0005992888
## 4_39 -1.279543e-02  0.0040309094  0.0033019875  0.0250849525 -0.0001190308
## 4_40 -1.435751e-02 -0.0255403690 -0.0056287709 -0.0198942113  0.0067739407
## 4_41 -1.948896e-04  0.0090982867 -0.0157319623 -0.0123433644  0.0187977624
## 4_54  2.785955e-02  0.0192741878 -0.0277251851  0.0118902928  0.0055941920
## 4_55  2.197426e-03 -0.0077914858  0.0038840197 -0.0035021201  0.0070013315
## 4_56 -8.714062e-03 -0.0241868550  0.0084277166  0.0033893311 -0.0056791005
## 4_57  8.637182e-03  0.0225116365  0.0108195755 -0.0123996779 -0.0209508024
## 5_39 -9.114083e-03  0.0096338556  0.0346713595  0.0020376069 -0.0136701822
## 5_40 -2.797177e-02 -0.0120058386  0.0015946298  0.0018721441 -0.0028620615
## 5_41 -1.504015e-02 -0.0027234368 -0.0089669674  0.0097765588  0.0043883646
## 5_54  1.448110e-02  0.0121551467  0.0109766912 -0.0044058657 -0.0051514290
## 5_55 -6.105422e-03 -0.0084215096 -0.0135787985 -0.0048625593  0.0128305154
## 5_59  1.310106e-02 -0.0154037110  0.0116291166 -0.0110195927 -0.0008215316
## 6_36 -1.461058e-02 -0.0019612893  0.0011491321 -0.0161101521 -0.0095294438
## 6_37  1.282850e-02 -0.0006710664 -0.0179342513  0.0013127309  0.0078839156
## 6_38 -5.118385e-03  0.0146287351 -0.0107950432 -0.0003615154  0.0003363944
## 6_54  2.340899e-03  0.0090090468  0.0189325734  0.0050909543 -0.0123153527
## 6_55  3.376187e-03  0.0248393500 -0.0036069268  0.0135519390 -0.0141379975
## 6_56 -2.011222e-03 -0.0095097909  0.0054623048 -0.0251166199  0.0079062828
## 6_57  9.946442e-05 -0.0054493473  0.0141891910  0.0183725908  0.0009323319
## 6_58 -1.797395e-03 -0.0155648925 -0.0071526779  0.0008053725  0.0158960820
## 9_16 -1.265384e-02  0.0056492431 -0.0003834258  0.0114398646 -0.0016008248
## 9_17  2.823506e-03 -0.0151825879 -0.0083943169  0.0045559571 -0.0014560672
## 9_18 -8.012234e-03  0.0020377296 -0.0220226242  0.0003534392  0.0137966667
## 9_19 -1.019001e-02  0.0236601463 -0.0079412889 -0.0169972386 -0.0065538957
## 9_21  1.281426e-02  0.0246727919 -0.0053570806  0.0009766856  0.0021969172
## 9_22 -1.414138e-03 -0.0195959514  0.0200690752  0.0002547308 -0.0122829979
## 9_34 -4.592475e-03  0.0062952548 -0.0010495512  0.0115289753  0.0073778237
## 9_35 -1.637846e-03 -0.0531474014 -0.0119442711 -0.0170416551 -0.0341391939
## 9_36 -3.208598e-02  0.0033420312 -0.0072882359  0.0029431449  0.0164157440
## 9_37  1.213580e-03 -0.0202062895  0.0050285174 -0.0155824703  0.0036312760
## 9_38  3.488367e-02 -0.0214415566  0.0105000462  0.0378237057  0.0063472347
## 9_39  2.792484e-02  0.0217695300  0.0139127894 -0.0148751794  0.0096465778
##          Axis.51      Axis.52      Axis.53      Axis.54      Axis.55
## 10_10  1.262027e-02 -2.518735e-02  0.0022185807 -0.0141557910  3.147184e-03
## 10_11  2.679330e-03 -3.065643e-04 -0.0078804366 -0.0023008515  3.109172e-03
## 10_12  2.797264e-02 -9.179848e-03  0.0104208436  0.0100353372  8.877242e-04
## 10_13 -7.893882e-03  9.165440e-03  0.0164733439  0.0022879085  1.535989e-02
## 10_14  3.188457e-04 -1.002129e-02  0.0022729490  0.0185419853 -1.064778e-02
## 10_15  6.733202e-03  1.463017e-03 -0.0075294265  0.0096837880 -1.260270e-02
## 10_19 -2.029235e-02  5.756430e-03 -0.0053484544 -0.0030537627  5.145930e-03
## 10_1   1.488318e-03 -2.077946e-03  0.0068557279 -0.0036264821  4.983989e-05
## 10_20  8.398812e-03  4.653944e-03 -0.0062003796 -0.0122438284  1.156248e-03
## 10_21 -1.935810e-02  3.806170e-03  0.0079867938  0.0023458954 -1.509886e-03
## 10_22 -2.026118e-03 -2.146135e-03  0.0012935836  0.0112534409 -3.534419e-03
## 10_25  7.165886e-03 -2.688596e-03  0.0049456074 -0.0043923296 -3.586484e-03
## 10_26  4.393543e-03 -1.012266e-02 -0.0034428248  0.0066394363  8.729955e-04
## 10_28 -1.005545e-02 -4.469551e-03  0.0080962831 -0.0220482484 -3.897654e-04
## 10_29 -8.444918e-05 -8.902422e-03 -0.0020093389  0.0084042744 -5.207904e-03
## 10_2   6.883881e-04 -1.147351e-03  0.0102678381  0.0032516135 -1.000529e-02
## 10_30  1.931902e-02  1.339283e-02  0.0181440875  0.0087919163 -5.103537e-03

```

```

## 10_33 -5.641982e-03 7.530812e-03 -0.0012846338 0.0162846656 3.717546e-03
## 10_34 4.543558e-03 3.289550e-03 -0.0075894157 -0.0146865216 1.715948e-03
## 10_35 -8.446193e-03 -7.289533e-03 -0.0026231553 0.0185853956 1.305449e-02
## 10_39 3.288563e-02 3.154208e-02 -0.0202610885 -0.0059802582 3.547565e-03
## 10_3 6.028186e-05 -6.615385e-03 -0.0032531312 -0.0058110123 6.427273e-03
## 10_40 -1.282729e-02 -2.227449e-02 0.0151838021 0.0056019363 1.839170e-03
## 10_41 -1.613333e-02 -6.231097e-03 0.0264199966 -0.0105092311 -1.956032e-03
## 10_42 -8.620306e-03 9.401788e-03 -0.0017886835 -0.0057302081 -1.204470e-02
## 10_43 1.019577e-02 -7.762187e-03 0.0043681261 -0.0059595739 2.494688e-02
## 10_44 1.564429e-02 -1.698573e-03 -0.0200914204 0.0135176549 1.911630e-03
## 10_48 -2.186306e-02 -1.397762e-03 0.0172363746 -0.0109691429 9.337902e-03
## 10_49 3.326363e-03 -8.624808e-03 -0.0102384376 -0.0121040648 1.302491e-03
## 10_4 -1.875645e-02 9.821900e-03 -0.0156991759 -0.0287867534 8.195386e-03
## 10_50 1.084580e-02 -8.244302e-04 0.0001967013 0.0131248837 1.805417e-02
## 10_51 2.373794e-02 -1.246520e-03 0.0242953370 0.0001670808 -3.225622e-03
## 10_52 7.148064e-03 -7.388700e-04 0.0063454026 0.0014505550 -4.379789e-03
## 10_53 9.630419e-03 1.505493e-02 -0.0270284080 0.0043177123 -1.635485e-02
## 10_57 1.023686e-02 -5.795245e-04 0.0038232156 -0.0052944288 1.713080e-02
## 10_58 -9.215101e-03 -8.329397e-03 0.0090086588 -0.0176991019 -3.901195e-03
## 10_59 -7.872061e-03 9.597507e-03 -0.0053949155 0.0045628783 -2.252279e-03
## 10_60 -5.548032e-03 -1.222321e-02 -0.0132728097 0.0040980128 -7.427682e-03
## 10_63 2.505691e-03 5.089474e-03 -0.0006057313 0.0047071903 -1.295560e-03
## 10_64 9.029352e-03 1.754116e-02 -0.0176900450 0.0015410944 6.138421e-04
## 10_66 -4.724661e-03 -5.485092e-03 0.0098380035 0.0095095608 1.229808e-03
## 10_67 -4.174209e-03 -1.181236e-04 -0.0008354056 0.0076155940 1.162078e-02
## 10_68 1.038011e-02 3.082973e-03 0.0181057513 -0.0013633778 -3.816002e-03
## 10_69 1.614694e-02 8.706899e-03 0.0054867439 -0.0056407918 9.218941e-03
## 10_7 -1.330993e-02 1.060926e-02 -0.0116370142 0.0065794810 -2.023722e-02
## 10_8 -5.471062e-03 -8.749630e-03 0.0053989645 0.0027887699 6.483473e-03
## 11_1 -3.528594e-03 -1.714309e-02 -0.0132479979 0.0107663663 -1.693225e-02
## 11_3 -1.065373e-02 4.593198e-03 0.0050272349 -0.0086428642 -7.983625e-04
## 14_20 1.002430e-02 -1.376466e-02 -0.0161487411 -0.0094683224 -4.704683e-03
## 14_21 -1.156612e-02 -1.423028e-03 -0.0030519361 0.0100199122 2.711197e-02
## 14_22 -7.041735e-03 1.885668e-03 0.0046996186 -0.0074833961 -4.609394e-03
## 14_23 3.842470e-03 -1.482530e-03 0.0146800357 0.0112973114 -9.217853e-03
## 14_25 1.626281e-02 -3.417127e-03 -0.0148454917 0.0027958224 -1.429118e-02
## 14_27 2.060779e-02 1.313661e-02 -0.0041845708 -0.0006638734 -1.110615e-02
## 14_29 -2.655665e-02 1.536923e-02 0.0084296253 -0.0114807216 1.986473e-02
## 14_30 -1.017047e-02 1.844341e-02 0.0281881110 0.0455556977 -6.598618e-03
## 14_33 -1.235841e-02 4.390329e-03 -0.0052077707 -0.0132344942 2.939823e-03
## 14_34 -2.401619e-03 5.280441e-03 -0.0025115446 -0.0020916332 2.404909e-03
## 14_35 -1.899574e-02 -3.351401e-03 -0.0029364185 0.0122227074 8.135560e-04
## 14_36 1.555692e-03 2.433440e-03 -0.0046627205 -0.0055394580 8.332499e-03
## 2_23 -1.202554e-02 2.652945e-03 -0.0243688348 0.0126890817 -6.989718e-04
## 2_24 -1.299482e-02 1.965781e-02 -0.0081495704 -0.0214152833 8.232220e-03
## 2_25 -2.200008e-04 1.848258e-02 -0.0008751849 0.0174241626 1.864726e-03
## 2_26 -1.185669e-02 -2.476177e-02 -0.0078623143 -0.0025791082 -3.990402e-02
## 2_27 -1.420255e-02 -2.655546e-02 0.0234127852 -0.0056646507 -1.316398e-02
## 2_29 -1.926989e-02 1.777823e-02 -0.0144650233 0.0153094249 2.010488e-02
## 2_36 -2.355187e-03 -6.529139e-03 0.0066677825 0.0014910973 1.052249e-02
## 4_65 -4.557463e-03 -9.914205e-03 -0.0007567393 -0.0110312971 4.906193e-03
## 2_39 2.623377e-02 -8.344547e-03 0.0161662253 -0.0033983037 4.684360e-03
## 2_40 8.541908e-03 2.459189e-02 0.0094705418 -0.0038787993 -1.971336e-03
## 2_41 -8.039000e-03 -1.090817e-02 -0.0070946734 -0.0082569609 -7.707973e-03

```

```

## 2_42  3.995381e-03  1.878403e-02  0.0080257599 -0.0113465509 -1.252435e-02
## 2_47  1.831097e-02  3.888250e-03  -0.0131455910  0.0153867227 -7.212640e-03
## 2_48  -4.309310e-03  -6.420188e-03  -0.0105999502  0.0014795957  6.574270e-03
## 2_49   2.606155e-03  -1.000796e-02  -0.0005068468 -0.0014414080 -1.725512e-03
## 2_50   7.475030e-03  4.310472e-03  -0.0020417793 -0.0048169570  4.476834e-03
## 2_51  -7.120165e-03  1.565941e-03  -0.0187955420  0.0043487560 -4.174811e-03
## 2_52  -1.486758e-02  1.331312e-02  -0.0101031878 -0.0034908619 -5.559315e-03
## 2_56   5.141168e-03  2.461449e-03  0.0020381359 -0.0142843946 -2.589553e-03
## 2_57   8.226279e-03  -3.376064e-03  -0.0030441614  0.0011371840  5.669554e-03
## 2_58  -2.455427e-02  -1.159116e-04  0.0021121932 -0.0007479324 -1.858929e-03
## 2_59   7.865311e-04  -9.731540e-03  0.0188680529  0.0027944930 -2.843831e-03
## 2_60   1.259768e-02  1.589673e-03  0.0192791241 -0.0210846442  9.871663e-03
## 2_61  -7.347006e-03  1.567688e-02  0.0109385557  0.0021021811 -4.823753e-03
## 4_36   4.364697e-03  4.722509e-03  0.0007152902 -0.0059615730  2.315780e-03
## 4_37  -4.395389e-03  3.977296e-03  0.0102653566 -0.0053219787 -6.859602e-03
## 4_38   1.097192e-05  1.607874e-02  0.0001731992  0.0011814896 -4.400917e-03
## 4_39   7.634971e-03  8.770307e-03  -0.0052005721 -0.0058212512 -2.511996e-03
## 4_40  -2.642126e-03  -3.002903e-02  -0.0067809579 -0.0041175551 -1.714197e-02
## 4_41  -6.825800e-03  -4.563384e-03  -0.0115392944 -0.0230788296 -5.350571e-03
## 4_54   1.218195e-02  4.541893e-03  -0.0104997402 -0.0064447541  3.216407e-03
## 4_55  -3.666142e-03  -5.932133e-03  0.0054812511  0.0076923641 -5.734831e-03
## 4_56   1.036365e-02  -3.507895e-03  0.0195922301  0.0062623771 -2.636709e-03
## 4_57  -3.082120e-02  9.708962e-03  0.0050187202  0.0031205610 -7.217804e-04
## 5_39   2.234049e-02  9.148747e-03  0.0019506188  0.0023701316  8.686491e-03
## 5_40   4.960272e-04  2.523302e-02  0.0122610643 -0.0003492993 -1.755140e-02
## 5_41   8.680061e-07  -3.145884e-03  -0.0113116407 -0.0035652671  3.937017e-03
## 5_54   8.203886e-03  -1.079697e-02  0.0023636049  0.0100618113 -6.111224e-03
## 5_55   1.388888e-02  -1.433747e-02  -0.0063887559 -0.0139605789  4.138562e-03
## 5_59  -3.050117e-04  -1.040785e-02  -0.0183939927  0.0092610279  1.511804e-02
## 6_36  -1.576874e-02  1.172061e-02  0.0077522784 -0.0035223089 -1.244452e-02
## 6_37   4.634306e-03  -3.161214e-02  -0.0051579781  0.0008107163 -3.635393e-04
## 6_38   6.032865e-03  -1.820835e-02  0.0007703942  0.0141119137  2.009135e-02
## 6_54  -2.149573e-03  -6.657153e-03  -0.0098321341  0.0103336717  8.646518e-03
## 6_55   9.205351e-03  -4.625520e-03  -0.0068414428 -0.0031082111  1.000293e-02
## 6_56   2.073672e-02  -1.968256e-02  -0.0051332662 -0.0039492040  1.667346e-02
## 6_57   1.896091e-02  1.260228e-02  0.0005307440 -0.0010090113 -1.106473e-02
## 6_58  -1.970584e-02  9.267734e-03  -0.0035813793  0.0140000242  6.272285e-03
## 9_16   6.380386e-03  -5.584901e-03  0.0084334683  0.0042692647  6.961927e-03
## 9_17  -1.145734e-03  8.174827e-05  -0.0150548862 -0.0032245218 -1.864643e-03
## 9_18  -2.196044e-02  -1.766412e-02  -0.0124878799  0.0139503957  3.677535e-03
## 9_19   5.765352e-03  1.006546e-02  0.0066605192  0.0007091694 -7.474727e-03
## 9_21   4.142693e-03  1.921446e-03  0.0027602493 -0.0117705508 -1.331074e-02
## 9_22   4.263719e-03  1.306850e-02  -0.0026483121  0.0082075404  1.194333e-04
## 9_34  -1.561904e-03  1.280696e-02  0.0214564820 -0.0083923992 -4.472968e-04
## 9_35   1.110635e-02  1.070622e-03  -0.0057070667 -0.0112434503 -5.955476e-03
## 9_36   2.408128e-03  2.074657e-05  0.0122633587 -0.0052398377  3.666391e-04
## 9_37  -6.763872e-03  1.716680e-03  -0.0144931895  0.0070672778  1.306307e-03
## 9_38  -2.116560e-02  -5.717513e-03  -0.0069437405 -0.0072441378  6.936034e-04
## 9_39   1.075750e-02  -1.015802e-02  0.0071718244  0.0158040808  5.766812e-03
##          Axis.56      Axis.57      Axis.58      Axis.59      Axis.60
## 10_10  1.179435e-02  -8.389646e-03  0.0082157918 -7.831629e-03 -1.585930e-03
## 10_11 -1.523071e-03  1.426412e-03  0.0036156529 -3.436364e-03  4.281062e-03
## 10_12  1.334469e-03  -1.696839e-02  0.0006428575 -4.719476e-03  2.692430e-04
## 10_13 -1.139645e-02  -9.985058e-04  0.0013920792  2.025132e-03  1.826867e-03

```

```

## 10_14 4.744683e-03 -1.197609e-02 -0.0056053050 -3.320249e-04 4.062050e-03
## 10_15 -9.481757e-04 -1.874669e-03 0.0080901453 -7.180281e-05 -6.261663e-03
## 10_19 -9.691133e-03 -6.179098e-03 -0.0012385657 -3.455458e-03 -2.493539e-04
## 10_1 1.159251e-02 -1.596094e-02 -0.0056116945 4.862169e-03 1.209444e-03
## 10_20 -1.037094e-02 -1.307199e-02 -0.0064204833 5.061149e-03 1.329259e-03
## 10_21 1.289541e-03 7.488773e-03 0.0072910003 5.614666e-03 4.249991e-03
## 10_22 -1.393680e-03 -1.283797e-03 0.0066047890 -5.091353e-03 3.188656e-03
## 10_25 9.369806e-03 1.446996e-02 -0.0007471996 3.646336e-03 -1.651134e-04
## 10_26 1.156344e-02 -6.186093e-04 0.0020216382 -1.841382e-03 -1.376856e-03
## 10_28 -3.274007e-03 -4.477295e-03 0.0036701879 -5.766910e-03 -5.491505e-04
## 10_29 -2.130200e-03 -2.017117e-03 -0.0031509762 -2.027478e-03 -1.047124e-03
## 10_2 2.982714e-03 3.538945e-03 0.0021524781 6.238827e-04 -3.190360e-03
## 10_30 9.778691e-03 1.302117e-02 0.0068857606 -3.055140e-03 -3.388150e-03
## 10_33 -6.549979e-03 3.055412e-03 -0.0016047731 6.296946e-04 -2.584833e-03
## 10_34 -3.953976e-03 1.753355e-03 -0.0015664898 4.003944e-03 1.694873e-03
## 10_35 1.237922e-02 -6.641301e-03 0.0134605925 1.085106e-02 -4.155933e-03
## 10_39 -3.120233e-03 2.342899e-03 0.0083105994 -3.972991e-03 -3.427426e-03
## 10_3 1.942666e-03 2.139871e-04 0.0004987269 -4.472354e-04 -6.563001e-04
## 10_40 -1.493076e-03 -8.346135e-03 0.0026720783 -9.374576e-04 -8.308155e-04
## 10_41 6.120763e-03 -4.765656e-03 -0.0066571084 -9.206162e-03 5.440629e-03
## 10_42 -1.102100e-02 6.665197e-03 -0.0011249195 1.207432e-03 1.300661e-03
## 10_43 -1.291001e-02 4.233756e-03 0.0042494074 1.072101e-02 -3.050184e-03
## 10_44 -9.654586e-03 -3.076641e-03 0.0037847500 1.037732e-02 1.788017e-03
## 10_48 2.647656e-03 7.639427e-03 -0.0120195623 1.211131e-02 -1.929272e-03
## 10_49 -1.532475e-02 1.964441e-03 0.0007711988 -1.154129e-03 -1.076889e-03
## 10_4 5.195264e-03 4.376624e-03 -0.0058699204 -1.260730e-03 6.841509e-03
## 10_50 -3.378381e-03 6.315877e-03 -0.0015354669 -1.028927e-02 2.796329e-03
## 10_51 1.278132e-02 7.367851e-03 0.0003068312 1.265187e-03 -9.950301e-04
## 10_52 -4.828429e-03 3.758184e-03 -0.0026173634 -2.889864e-03 3.820598e-03
## 10_53 6.908421e-03 -1.266068e-02 -0.0049849764 7.270793e-04 -5.296250e-03
## 10_57 1.267019e-03 -9.201902e-03 0.0144131067 -1.932444e-03 1.067386e-02
## 10_58 -7.045323e-03 -7.991792e-03 0.0101430212 3.683931e-03 -2.307141e-04
## 10_59 -2.610910e-03 4.443288e-03 -0.0021690495 -3.675788e-04 -5.415444e-04
## 10_60 8.882538e-04 1.115210e-03 -0.0092289232 9.696293e-05 5.032081e-04
## 10_63 9.982114e-03 5.958544e-05 -0.0168872119 7.478426e-04 4.253901e-03
## 10_64 2.533074e-04 3.022751e-03 0.0021491935 -4.483166e-03 -1.680518e-03
## 10_66 5.295136e-03 5.122175e-03 0.0005506438 -7.504288e-03 -8.189276e-03
## 10_67 3.126431e-03 -1.800255e-03 -0.0060371416 -4.573002e-03 3.263217e-05
## 10_68 -3.476136e-03 2.502230e-03 -0.0075720048 1.158721e-03 -4.826517e-03
## 10_69 1.047228e-02 1.647847e-02 0.0072601674 4.275257e-03 3.565252e-03
## 10_7 4.893307e-03 9.463562e-03 -0.0020681858 2.592570e-03 -4.215071e-03
## 10_8 -8.843844e-04 -3.273263e-03 0.0015367222 -3.213410e-03 7.177478e-04
## 11_1 -1.927723e-02 -3.802277e-03 -0.0029401488 -2.027673e-04 -6.293921e-03
## 11_3 7.057092e-03 -8.205593e-03 0.0057139903 2.206219e-03 -1.159380e-03
## 14_20 1.173117e-02 3.049602e-03 0.0121496535 -1.130155e-02 -7.321199e-03
## 14_21 1.674868e-02 -2.773575e-03 -0.0011847431 -8.780687e-04 1.662907e-04
## 14_22 -7.824267e-03 9.292196e-03 0.0021764334 -2.569937e-04 -1.741862e-03
## 14_23 -6.293535e-03 -1.253303e-02 0.0043945555 1.498855e-02 -3.517740e-03
## 14_25 1.052057e-02 -2.385374e-02 -0.0092270833 4.404417e-03 8.065160e-03
## 14_27 2.423282e-03 1.476017e-03 -0.0002519334 -3.854443e-03 3.707462e-03
## 14_29 1.537888e-03 5.699255e-03 0.0020272993 -5.893699e-05 -6.561905e-03
## 14_30 -1.434217e-02 1.502249e-02 -0.0071780007 2.992377e-03 5.024186e-03
## 14_33 -3.044693e-03 -4.768658e-03 -0.0036107811 -2.472880e-03 -4.946786e-03
## 14_34 -1.420676e-03 -1.070914e-03 -0.0065985733 -2.895518e-03 -1.914463e-03

```

```

## 14_35 6.856023e-03 5.645298e-03 0.0113787799 7.421002e-03 -7.033513e-03
## 14_36 -1.310436e-02 -2.255194e-03 -0.0104583143 -1.417618e-03 5.210022e-03
## 2_23 2.610627e-03 1.834636e-02 0.0004556322 3.317070e-03 4.873192e-03
## 2_24 -3.698208e-03 -5.358577e-03 0.0081609645 -1.604207e-03 -6.810324e-04
## 2_25 -9.609305e-03 -1.814502e-03 0.0104099688 2.536925e-03 2.690700e-03
## 2_26 4.813742e-03 5.745873e-03 0.0065329534 -2.822637e-03 1.079596e-02
## 2_27 3.267808e-03 -2.783218e-03 -0.0096511015 -5.963914e-03 -5.956785e-03
## 2_29 -1.485486e-02 -1.136478e-03 -0.0027195837 -1.348072e-02 -5.477383e-04
## 2_36 7.152003e-03 -1.145924e-03 0.0013733982 8.607114e-03 1.955374e-03
## 4_65 3.940916e-03 -7.976525e-03 -0.0065197349 2.973332e-03 -4.081143e-03
## 2_39 -7.885564e-03 7.691811e-03 -0.0079338982 4.841291e-03 2.101044e-03
## 2_40 8.144213e-03 -9.268387e-03 0.0060666365 8.120380e-03 -5.498045e-03
## 2_41 5.026612e-03 6.198190e-03 -0.0045727056 -6.356684e-03 2.954076e-03
## 2_42 2.891990e-03 6.179250e-03 -0.0049886982 5.720132e-03 -2.381742e-03
## 2_47 6.879439e-03 4.471514e-03 -0.0004125079 -1.282133e-03 2.496651e-03
## 2_48 1.678722e-03 5.651845e-03 0.0090495728 -5.858764e-03 4.680398e-04
## 2_49 -5.699162e-03 -1.557117e-03 0.0030109619 1.567265e-03 1.341606e-03
## 2_50 1.424859e-02 1.291938e-02 -0.0028185004 -7.519023e-03 8.813510e-05
## 2_51 -2.681128e-03 6.541629e-03 0.0049210072 5.435579e-03 -2.781682e-04
## 2_52 2.774310e-03 -1.716693e-02 -0.0003605779 8.025022e-03 -3.030784e-04
## 2_56 -1.533902e-02 9.041751e-03 0.0029787389 -3.951622e-03 -6.837365e-03
## 2_57 -2.182636e-05 4.468749e-03 -0.0004153626 1.423400e-03 1.919917e-03
## 2_58 7.236798e-03 -2.848806e-03 -0.0079318614 1.780246e-03 3.973456e-03
## 2_59 -3.147104e-03 -8.495987e-03 -0.0111898926 -9.739421e-04 -2.234662e-03
## 2_60 -5.481985e-03 1.314564e-03 -0.0057345467 1.362151e-03 2.747009e-03
## 2_61 5.879968e-03 -1.009948e-02 0.0112616514 -9.694821e-03 2.763300e-05
## 4_36 -1.802737e-03 -4.810518e-03 0.0098612665 -2.169163e-04 6.221566e-03
## 4_37 1.556890e-03 -8.056550e-03 0.0005331284 1.230243e-03 -1.636487e-03
## 4_38 7.999677e-03 -1.511646e-02 -0.0058492240 2.310651e-03 -5.272826e-03
## 4_39 -6.527317e-03 2.869209e-03 -0.0173570193 3.868570e-04 -5.135690e-03
## 4_40 -1.105533e-02 1.406091e-02 0.0049528447 4.176568e-04 1.581579e-03
## 4_41 -4.591974e-03 6.465630e-03 0.0076435250 1.259729e-02 4.323034e-03
## 4_54 1.276312e-02 9.735533e-03 -0.0101247171 8.964881e-03 -2.322656e-03
## 4_55 8.412793e-04 -3.833589e-03 0.0055513459 3.099327e-03 -4.619582e-04
## 4_56 -1.808242e-02 -5.209302e-03 0.0139380862 -7.345463e-03 -2.232871e-03
## 4_57 1.453707e-02 7.520238e-03 0.0026120307 -4.105315e-03 9.120484e-05
## 5_39 -1.912560e-03 -6.903780e-03 -0.0176425726 -4.447154e-03 2.711538e-03
## 5_40 -1.412919e-02 -6.581773e-03 0.0036214476 -8.811099e-03 4.450017e-03
## 5_41 9.001162e-04 9.028086e-03 -0.0065256099 7.417239e-04 -1.746814e-03
## 5_54 7.522731e-04 9.149992e-03 -0.0113798160 -2.187042e-03 -3.253827e-03
## 5_55 -6.340465e-03 -1.795017e-03 -0.0028250238 -3.019614e-04 -1.458936e-03
## 5_59 1.287699e-04 -7.734626e-04 -0.0055043127 -6.686326e-03 -6.478980e-03
## 6_36 6.686706e-03 4.417616e-03 0.0042478571 -1.195413e-03 6.194781e-03
## 6_37 -3.909660e-03 7.050167e-03 0.0038601590 4.633104e-03 1.869369e-03
## 6_38 1.053997e-03 -4.918087e-04 -0.0028545609 5.350313e-04 4.043390e-03
## 6_54 7.019554e-03 -2.127958e-03 0.0037600021 7.479392e-03 2.175020e-03
## 6_55 -8.899720e-03 -3.326244e-03 -0.0009730949 4.727882e-03 2.454894e-03
## 6_56 6.028125e-03 3.229238e-03 0.0062501061 -3.071805e-03 3.557603e-03
## 6_57 -5.110371e-03 -3.851905e-03 0.0029020981 2.014973e-03 -2.030774e-03
## 6_58 1.710668e-03 -1.036595e-03 -0.0075421404 -2.368243e-03 7.067651e-03
## 9_16 -1.094319e-02 7.233085e-03 -0.0025664759 5.578950e-04 -1.631311e-03
## 9_17 7.498866e-03 1.285763e-03 -0.0047705337 -3.393390e-03 6.360857e-04
## 9_18 2.279470e-03 -6.896356e-03 -0.0008978464 -1.002238e-03 -7.043707e-03
## 9_19 4.435283e-03 1.065592e-03 0.0069067974 3.337886e-03 1.783595e-03

```

```

## 9_21 -4.716911e-03 3.229907e-03 0.0000669300 -2.224066e-03 -8.828149e-04
## 9_22 6.475064e-03 -7.770887e-03 0.0022549895 -2.298480e-03 3.040765e-03
## 9_34 9.864731e-03 2.431009e-03 0.0028147708 -2.759210e-03 2.052761e-04
## 9_35 5.294198e-03 8.965051e-03 -0.0079503835 2.313832e-04 -2.551153e-03
## 9_36 9.665991e-03 1.831990e-03 0.0052849262 -1.925573e-03 -2.582241e-03
## 9_37 -3.365306e-03 -7.040619e-03 0.0028026832 -3.825351e-03 -2.033315e-03
## 9_38 -1.720675e-02 -6.500430e-03 0.0004281836 6.477760e-04 2.365672e-03
## 9_39 -3.670929e-04 2.447593e-03 0.0031344056 2.953968e-03 3.224218e-04
##           Axis.61      Axis.62
## 10_10 -3.676588e-03 1.198600e-03
## 10_11 -6.907604e-04 -1.035494e-03
## 10_12 -5.060819e-03 1.228952e-03
## 10_13 3.925319e-03 2.313621e-03
## 10_14 8.396379e-06 -1.020192e-03
## 10_15 -3.251324e-03 -4.206152e-04
## 10_19 -1.028383e-03 6.633585e-04
## 10_1 -2.109026e-03 -1.100613e-03
## 10_20 3.964895e-03 -3.646018e-04
## 10_21 -1.133150e-03 -2.646139e-04
## 10_22 2.023042e-03 -5.070069e-06
## 10_25 -2.236958e-03 1.520173e-03
## 10_26 -1.9911659e-03 -1.442346e-03
## 10_28 6.084421e-03 -3.334401e-04
## 10_29 -2.777322e-03 3.358545e-04
## 10_2 -1.831164e-04 -4.301719e-04
## 10_30 3.705032e-03 2.067428e-03
## 10_33 -4.175352e-03 6.659675e-04
## 10_34 2.083743e-03 -6.008968e-04
## 10_35 2.775323e-04 -1.351250e-04
## 10_39 4.734315e-03 1.801823e-03
## 10_3 -1.114706e-03 -8.012736e-05
## 10_40 -1.469196e-04 -2.445372e-04
## 10_41 -5.981762e-03 -4.780284e-04
## 10_42 -2.678988e-03 -3.034682e-03
## 10_43 -7.620657e-03 2.157876e-03
## 10_44 4.469041e-03 6.644335e-04
## 10_48 5.605766e-03 1.730591e-03
## 10_49 1.472931e-03 3.694845e-04
## 10_4 -3.059932e-03 1.344086e-03
## 10_50 -5.601848e-03 -2.681368e-03
## 10_51 -1.861853e-03 5.463187e-04
## 10_52 -1.476304e-03 -1.308050e-03
## 10_53 7.391294e-03 -1.166134e-03
## 10_57 5.174806e-03 -1.228681e-03
## 10_58 5.407509e-03 -4.687001e-04
## 10_59 -6.582474e-05 1.386901e-03
## 10_60 -1.002502e-03 1.602701e-03
## 10_63 -4.111450e-03 -5.951784e-04
## 10_64 -1.646912e-03 1.166211e-04
## 10_66 9.183850e-04 9.491706e-04
## 10_67 -1.885866e-03 -4.088255e-04
## 10_68 -2.218435e-03 7.702514e-04
## 10_69 -1.797049e-03 1.530850e-04
## 10_7 -8.257459e-04 1.033989e-03

```

```

## 10_8    2.674979e-03 -2.765380e-04
## 11_1    8.144237e-04 -1.226074e-03
## 11_3    8.771402e-04  9.851196e-04
## 14_20   -1.413160e-03 -2.739054e-03
## 14_21    2.168462e-03 -3.201204e-04
## 14_22    1.280016e-03 -3.731769e-04
## 14_23    3.221387e-03 -4.783552e-03
## 14_25   -9.037077e-04  1.119381e-03
## 14_27   -4.461776e-03  2.292168e-03
## 14_29    9.603220e-04 -1.998871e-03
## 14_30    6.089142e-03 -4.893846e-04
## 14_33   -3.221615e-03  1.000549e-03
## 14_34    4.288272e-04  8.187145e-04
## 14_35   -2.793578e-04  2.313200e-04
## 14_36   -7.645175e-04 -9.983270e-04
## 2_23     3.913460e-03 -4.854860e-04
## 2_24   -1.675499e-03 -1.812069e-04
## 2_25   -4.108546e-03  6.482667e-04
## 2_26    6.299402e-03  9.848095e-04
## 2_27    5.455690e-03  8.891600e-04
## 2_29   -1.354019e-03 -1.293226e-03
## 2_36   -4.306912e-03 -6.632000e-04
## 4_65     2.202433e-03  1.666068e-04
## 2_39   -6.396515e-03 -1.457422e-03
## 2_40     2.806949e-03  1.012102e-03
## 2_41     2.696052e-03  8.100478e-04
## 2_42     7.915777e-04 -1.192974e-03
## 2_47     2.430901e-04 -1.867955e-03
## 2_48     1.714410e-03  3.355898e-04
## 2_49   -8.240258e-04 -5.802029e-04
## 2_50     5.090857e-03 -2.896140e-03
## 2_51   -4.103199e-04  1.341800e-03
## 2_52   -7.553052e-03  6.950312e-04
## 2_56     5.067539e-04  7.391383e-04
## 2_57     2.469386e-04 -9.781088e-04
## 2_58   -4.496910e-04  1.071946e-03
## 2_59     2.442029e-03 -1.566506e-03
## 2_60     1.315997e-03  1.200314e-03
## 2_61   -6.700881e-03  1.351329e-04
## 4_36     2.510720e-05 -2.004709e-03
## 4_37   -5.579414e-04 -2.322993e-04
## 4_38     4.573629e-03  3.910272e-05
## 4_39   -6.197262e-04  2.261692e-04
## 4_40   -4.187603e-03  1.162546e-03
## 4_41     2.152324e-03 -2.127222e-03
## 4_54   -6.992394e-03 -6.837544e-04
## 4_55   -5.769615e-04  1.154359e-03
## 4_56     1.761753e-03  1.091212e-03
## 4_57   -6.709472e-04 -1.088509e-03
## 5_39     4.346200e-03 -1.113725e-03
## 5_40   -3.357906e-04  1.694684e-03
## 5_41     1.463934e-03 -1.082183e-03
## 5_54   -6.021193e-03  7.071847e-05
## 5_55     1.496211e-03 -5.372864e-04

```

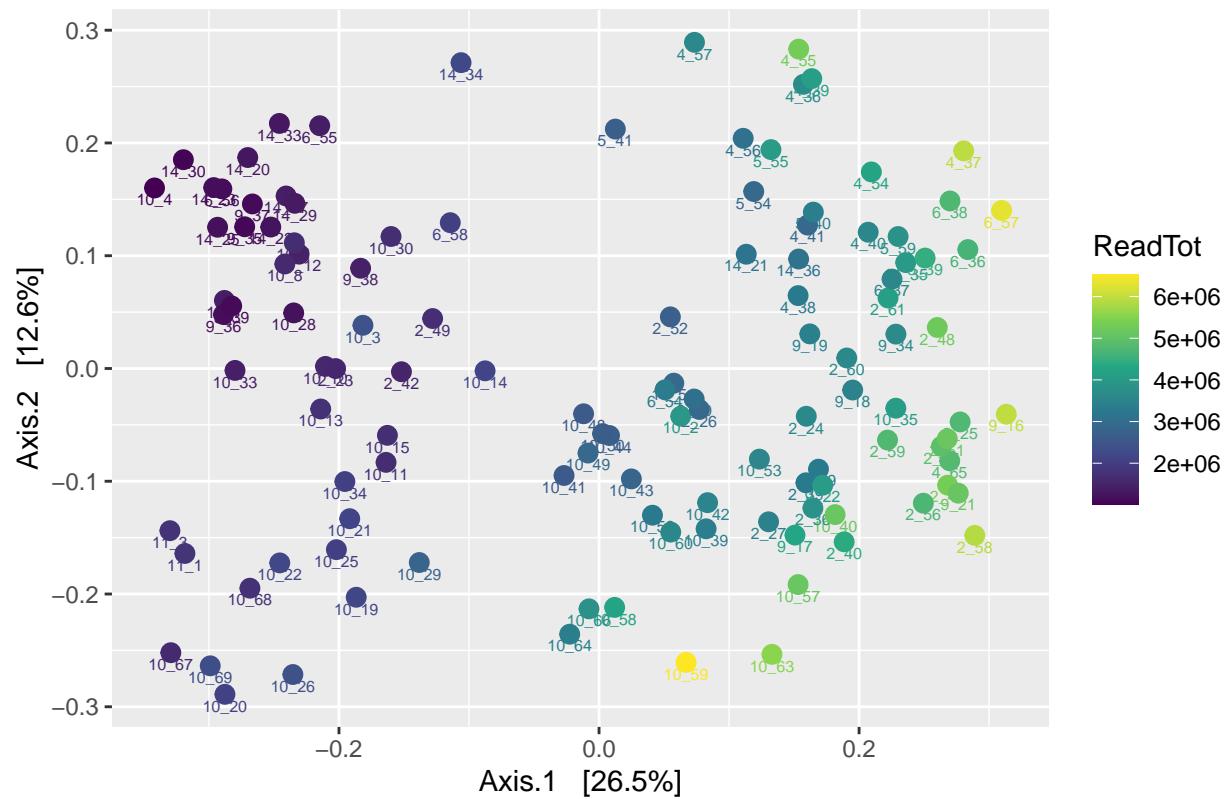
```

## 5_59  5.376563e-03 -3.937910e-04
## 6_36  -3.999251e-03 -5.170952e-04
## 6_37  -3.313328e-03  3.589929e-04
## 6_38   6.470736e-03  3.349355e-03
## 6_54  -1.695759e-03  1.435959e-04
## 6_55   3.905976e-04  1.154529e-03
## 6_56   5.760336e-03  6.195728e-04
## 6_57  -1.276814e-03  1.022611e-04
## 6_58   3.042403e-03  1.739552e-03
## 9_16  -3.505254e-04 -1.177361e-03
## 9_17   2.495145e-03 -1.030278e-04
## 9_18  -2.401754e-03  1.395386e-03
## 9_19  -2.097642e-04  9.178373e-04
## 9_21  -5.027623e-04 -2.151164e-04
## 9_22  -1.065475e-03  5.596011e-04
## 9_34   3.199152e-03 -2.696105e-04
## 9_35   2.790752e-03  1.471717e-03
## 9_36  -3.857625e-05 -1.187281e-03
## 9_37  -2.701022e-03  1.007638e-03
## 9_38  -4.518144e-03 -9.464999e-04
## 9_39   1.438932e-03 -4.628286e-04
##
## $trace
## [1] 19.46422
##
## attr(),"class")
## [1] "pcoa"

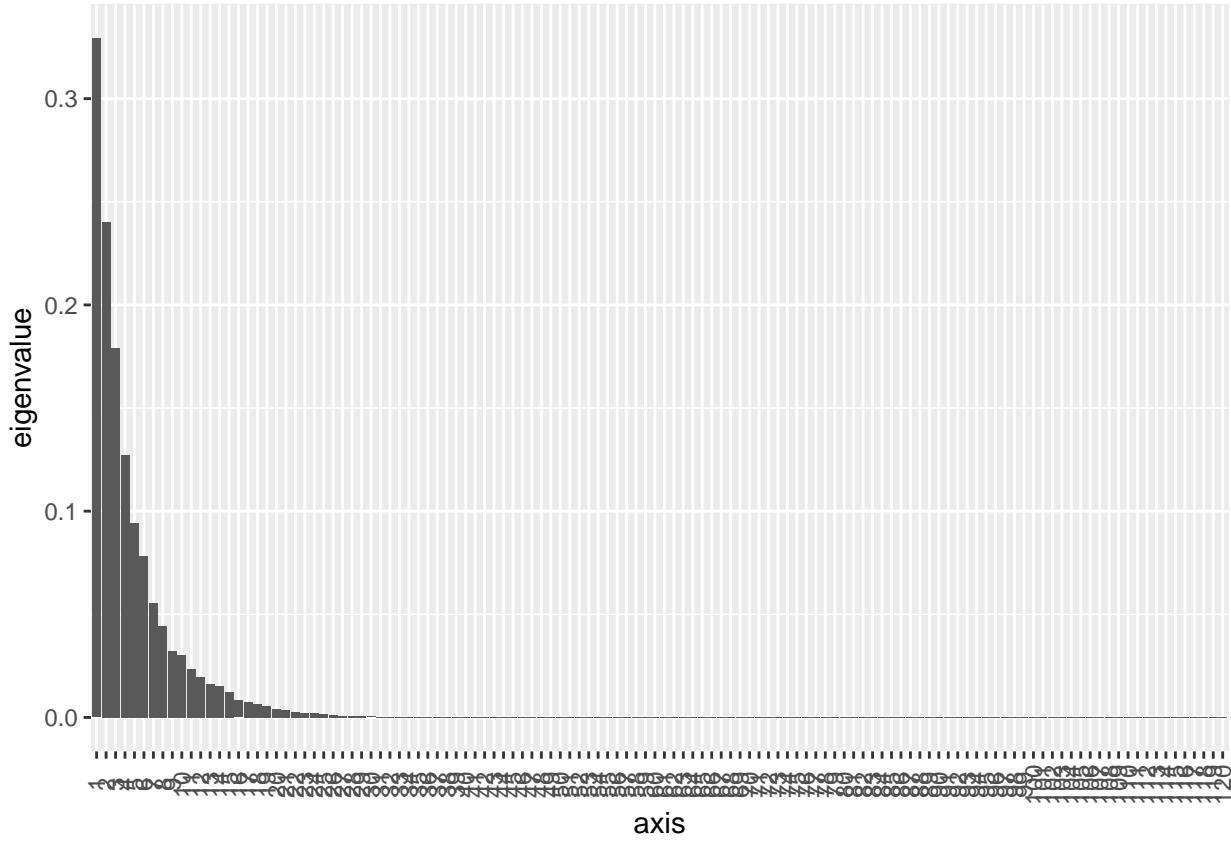
subsetMG %>% subset_samples(Sample_Unique != "2_57") %>% plot_ordination(pcoa_bc2, color = "ReadTot", 1
  geom_point(size = 3) + labs(title = "PCoA BC concentration",color = "ReadTot") +
  scale_colour_viridis_c()

```

PCoA BC concentration



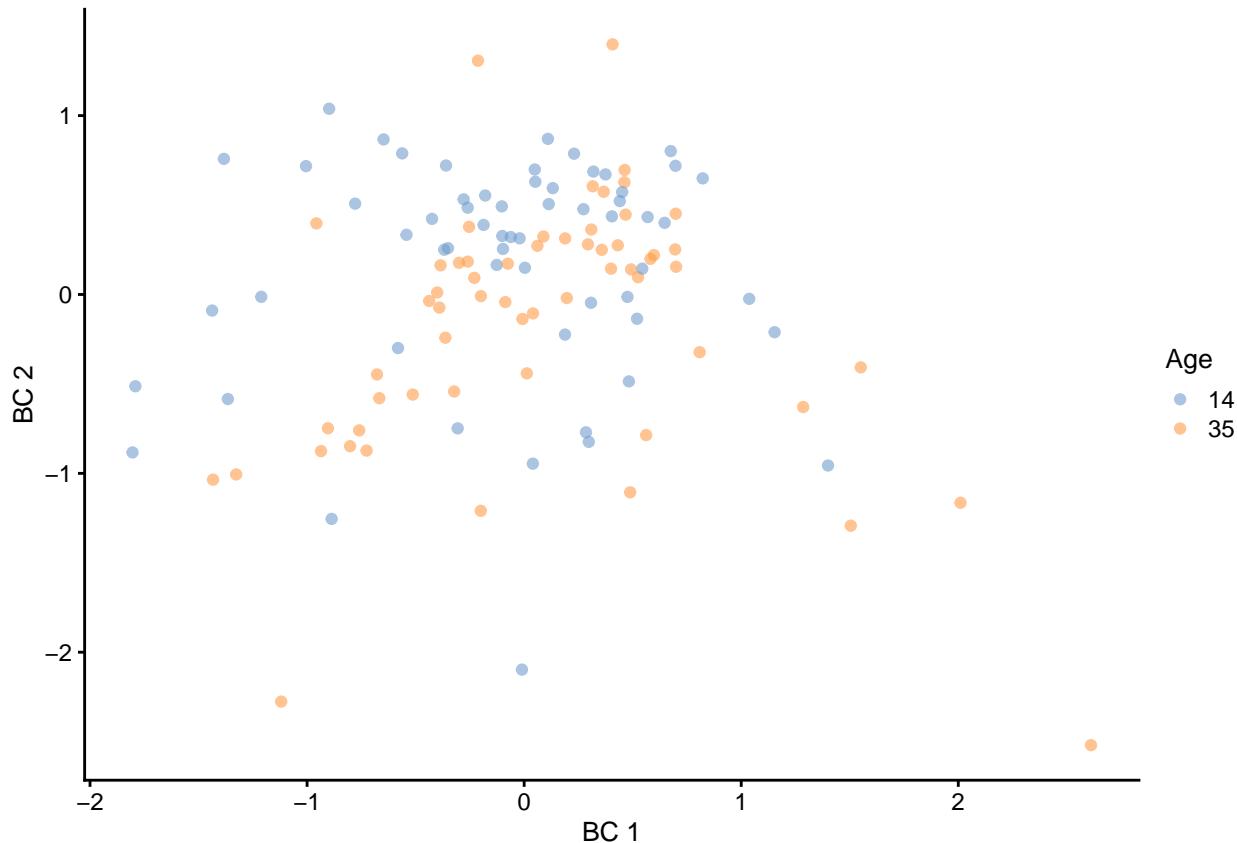
```
plot_scree(pcoa_jsd) #scree plots can be made for any of the PCoAs
```



different way of plotting with scater and tses, this specifically is NMDS BC

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)
tse <- transformCounts(tse, method = "relabundance")
tse <- runNMDS(tse, FUN = vegan::vegdist, name = "BC", nmdsFUN = "monoMDS",
                exprs_values = "relabundance",
                keep_dist = TRUE)

plotReducedDim(tse, "BC", colour_by = "Age")
```



PERMANOVAs

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)
tse <- transformCounts(tse, method = "relabundance")

adonis2(t(assay(tse, "relabundance")) ~ Age, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Age, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Age       1   0.7166 0.04537 5.6084  1e-04 ***
## Residual 118 15.0772 0.95463
## Total     119 15.7938 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(t(assay(tse, "relabundance")) ~ AB, data = colData(tse), permutations = 9999)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
```

```

## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ AB, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## AB         1   0.5354  0.0339 4.1401  3e-04 ***
## Residual  118  15.2584  0.9661
## Total     119  15.7938  1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Cox, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Cox, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox        3   1.9909  0.12606 5.5772  1e-04 ***
## Residual  116  13.8028  0.87394
## Total     119  15.7938  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Researcher, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Researcher, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher 4   1.1723  0.07422 2.305  3e-04 ***
## Residual   115  14.6215  0.92578
## Total      119  15.7938  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ FeedProducent, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ FeedProducent, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedProducent 2   1.4398  0.09116 5.8678  1e-04 ***
## Residual    117  14.3540  0.90884
## Total       119  15.7938  1.00000

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ LitterType, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ LitterType, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType   2   2.4121 0.15273 10.545 1e-04 ***
## Residual    117  13.3816 0.84727
## Total       119  15.7938 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ FeedType, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ FeedType, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedType    1   0.7166 0.04537 5.6084 1e-04 ***
## Residual   118  15.0772 0.95463
## Total      119  15.7938 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Gender, data = colData(tse), permutations = 9999) # NOT significant

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Gender, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender     1   0.0639 0.00405 0.4794 0.9254
## Residual  118  15.7298 0.99595
## Total     119  15.7938 1.00000

adonis2(t(assay(tse, "relabundance")) ~ Stables, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free

```

```

## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Stables, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables     9   5.2977  0.33543 6.169  1e-04 ***
## Residual 110  10.4960  0.66457
## Total     119  15.7938  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ FlockSize, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ FlockSize, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FlockSize    5   4.2411  0.26853 8.3701  1e-04 ***
## Residual 114  11.5527  0.73147
## Total     119  15.7938  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Farm2, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Farm2, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2       3   3.661  0.2318 11.667  1e-04 ***
## Residual 116  12.133  0.7682
## Total     119  15.794  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ AgeParentStock, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ AgeParentStock, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## AgeParentStock  4   4.1188  0.26079 10.143  1e-04 ***
## Residual     115  11.6749  0.73921
## Total       119  15.7938  1.00000

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ ReadPerc, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ ReadPerc, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## ReadPerc    1   2.1079 0.13346 18.174  1e-04 ***
## Residual 118   13.6859 0.86654
## Total     119   15.7938 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ ReadTot, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ ReadTot, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## ReadTot    1    0.751 0.04755  5.8913  1e-04 ***
## Residual 118   15.043 0.95245
## Total     119   15.794 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Variances: Age: 0.045, AB: 0.0339, agent: 0.126, researcher: 0.0742, FP = 0.091, LT = 0.153, FT = 0.0
# stable : 0.335, FS: 0.27, Farm: 0.232, APS: 0.261, readPerc: 0.133, readtot: 0.047
# Order: Stable>FS>APS>Farm>LT>ReadPerc>Cox>FP>LT>FP>Researcher>ReadTot>FT>Age>AB>Gender

# basically, composition seems to be different over every single variable, except for gender

# on genus level
tse_genus <- agglomerateByRank(tse, "Genus")
tse_genus <- transformCounts(tse_genus, method = "relabundance")

adonis2(t(assay(tse_genus, "relabundance")) ~ AB, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ AB, data = colData(tse_genus), permutations =
##          Df SumOfSqs      R2      F Pr(>F)

```

```

## AB      1  0.4263 0.03068 3.7345 0.0023 **
## Residual 118 13.4697 0.96932
## Total    119 13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ Cox, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ Cox, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox      3   1.8975 0.13655 6.115  1e-04 ***
## Residual 116 11.9985 0.86345
## Total    119 13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ Researcher, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ Researcher, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher 4   1.0993 0.07911 2.4697  9e-04 ***
## Residual   115 12.7967 0.92089
## Total     119 13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ FeedProducent, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ FeedProducent, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedProducent 2   1.4396 0.10359 6.7607  1e-04 ***
## Residual     117 12.4565 0.89641
## Total       119 13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

adonis2(t(assay(tse_genus, "relabundance")) ~ LitterType, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ LitterType, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType    2   1.7961 0.12925 8.6837  1e-04 ***
## Residual    117   12.0999 0.87075
## Total       119   13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ FeedType, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ FeedType, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedType     1   0.8182 0.05888 7.3824  1e-04 ***
## Residual   118   13.0778 0.94112
## Total       119   13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ Gender, data = colData(tse_genus), permutations = 9999) # 

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ Gender, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender      1   0.0726 0.00522 0.6195   0.74
## Residual   118   13.8234 0.99478
## Total       119   13.8960 1.00000

adonis2(t(assay(tse_genus, "relabundance")) ~ FarmRoundStable, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ FarmRoundStable, data = colData(tse_genus), permutations = 9999)

```

```

##               Df SumOfSqs      R2      F Pr(>F)
## FarmRoundStable   9  4.6147 0.33209 6.0769  1e-04 ***
## Residual        110  9.2813 0.66791
## Total          119 13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ FlockSize, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ FlockSize, data = colData(tse_genus), permutations = 9999)
##               Df SumOfSqs      R2      F Pr(>F)
## FlockSize      5  3.5985 0.25896 7.9675  1e-04 ***
## Residual     114 10.2975 0.74104
## Total        119 13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ Farm2, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ Farm2, data = colData(tse_genus), permutations = 9999)
##               Df SumOfSqs      R2      F Pr(>F)
## Farm2         3  3.0417 0.21889 10.836  1e-04 ***
## Residual    116 10.8543 0.78111
## Total        119 13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ AgeParentStock, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ AgeParentStock, data = colData(tse_genus), permutations = 9999)
##               Df SumOfSqs      R2      F Pr(>F)
## AgeParentStock  4  3.5111 0.25267 9.7202  1e-04 ***
## Residual       115 10.3849 0.74733
## Total          119 13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

adonis2(t(assay(tse_genus, "relabundance")) ~ ReadTot, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ ReadTot, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## ReadTot     1   0.6794 0.04889 6.0653  3e-04 ***
## Residual 118  13.2167 0.95111
## Total     119  13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_genus, "relabundance")) ~ ReadPerc, data = colData(tse_genus), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_genus, "relabundance")) ~ ReadPerc, data = colData(tse_genus), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## ReadPerc    1   2.2992 0.16546 23.395  1e-04 ***
## Residual 118  11.5968 0.83454
## Total     119  13.8960 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# same results on genus level (and on phylum level, though p values become higher)
# for different ordination methods
ps1.rel <- microbiome::transform(subsetMG, "compositional")
metadf <- data.frame(sample_data(ps1.rel))

# alternative calculations
#otu <- abundances(ps1.rel)
#meta <- meta(ps1.rel)
#adonis2(t(otu) ~ Age, data = meta, permutations=9999, method = "bray")

#permanova = adonis(t(otu) ~ Age, data = meta, permutations=9999, method = "bray")
#permanova$aov.tab

unifrac.dist <- UniFrac(ps1.rel)

adonis2(unifrac.dist ~ Age, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999

```

```

##  

## adonis2(formula = unifrac.dist ~ Age, data = metadf)  

##          Df SumOfSqs      R2      F Pr(>F)  

## Age       1   0.3846 0.03435 4.1977  0.001 ***  

## Residual 118 10.8123 0.96565  

## Total    119 11.1969 1.00000  

## ---  

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  

adonis2(unifrac.dist ~ AB, data = metadf)  

## Permutation test for adonis under reduced model  

## Terms added sequentially (first to last)  

## Permutation: free  

## Number of permutations: 999  

##  

## adonis2(formula = unifrac.dist ~ AB, data = metadf)  

##          Df SumOfSqs      R2      F Pr(>F)  

## AB        1   0.2048 0.01829 2.1989  0.001 ***  

## Residual 118 10.9921 0.98171  

## Total    119 11.1969 1.00000  

## ---  

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  

adonis2(unifrac.dist ~ Farm2, data = metadf)  

## Permutation test for adonis under reduced model  

## Terms added sequentially (first to last)  

## Permutation: free  

## Number of permutations: 999  

##  

## adonis2(formula = unifrac.dist ~ Farm2, data = metadf)  

##          Df SumOfSqs      R2      F Pr(>F)  

## Farm2     3   0.8443 0.0754 3.1533  0.001 ***  

## Residual 116 10.3527 0.9246  

## Total    119 11.1969 1.00000  

## ---  

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  

adonis2(unifrac.dist ~ Cox, data = metadf)  

## Permutation test for adonis under reduced model  

## Terms added sequentially (first to last)  

## Permutation: free  

## Number of permutations: 999  

##  

## adonis2(formula = unifrac.dist ~ Cox, data = metadf)  

##          Df SumOfSqs      R2      F Pr(>F)  

## Cox       3   0.7696 0.06874 2.854  0.001 ***  

## Residual 116 10.4273 0.93126  

## Total    119 11.1969 1.00000  

## ---  

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

adonis2(unifrac.dist ~ Researcher, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher    4   0.5923 0.0529 1.6057  0.001 ***
## Residual    115 10.6046 0.9471
## Total       119 11.1969 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(unifrac.dist ~ LitterType, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ LitterType, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType    2   0.6123 0.05468 3.3841  0.001 ***
## Residual    117 10.5846 0.94532
## Total       119 11.1969 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(unifrac.dist ~ Gender, data = metadf) # not sign

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Gender, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender       1   0.0876 0.00782 0.9304  0.726
## Residual   118 11.1093 0.99218
## Total       119 11.1969 1.00000

adonis2(unifrac.dist ~ Stables, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Stables, data = metadf)

```

```

##          Df SumOfSqs      R2      F Pr(>F)
## Stables     9   1.6260 0.14522 2.0765  0.001 ***
## Residual 110   9.5709 0.85478
## Total     119  11.1969 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

same patterns arise

```
wunifrac.dist <- UniFrac(ps1.rel,
                           weighted = TRUE)
```

```
adonis2(wunifrac.dist ~ Age, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Age, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Age        1   0.4663 0.04559 5.636  0.001 ***
## Residual 118   9.7633 0.95441
## Total     119  10.2296 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
adonis2(wunifrac.dist ~ AB, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB         1   0.2955 0.02889 3.5099  0.004 **
## Residual 118   9.9341 0.97111
## Total     119  10.2296 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
adonis2(wunifrac.dist ~ Farm2, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2      3   2.4355 0.23809 12.083  0.001 ***

```

```
## Residual 116  7.7941 0.76191
## Total     119  10.2296 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(wunifrac.dist ~ Cox, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox       3   1.3046 0.12753 5.6519  0.001 ***
## Residual 116   8.9251 0.87247
## Total     119  10.2296 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(wunifrac.dist ~ Researcher, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher 4   0.6943 0.06787 2.0935  0.003 **
## Residual   115  9.5353 0.93213
## Total      119  10.2296 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(wunifrac.dist ~ LitterType, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ LitterType, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType 2   1.6101 0.1574 10.928  0.001 ***
## Residual   117  8.6195 0.8426
## Total      119  10.2296 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(wunifrac.dist ~ Gender, data = metadf) # not sign
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Gender, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender     1   0.0413 0.00404 0.4787  0.902
## Residual 118 10.1883 0.99596
## Total     119 10.2296 1.00000
```

```
adonis2(wunifrac.dist ~ Stables, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Stables, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables    9   3.4452 0.33679 6.2066  0.001 ***
## Residual 110  6.7844 0.66321
## Total     119 10.2296 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# same patterns
```

```
jsd.dist <- phyloseq::distance(ps1.rel, "jsd")
```

```
adonis2(jsd.dist ~ Age, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Age, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Age        1   0.2259 0.05975 7.498  0.001 ***
## Residual 118  3.5548 0.94025
## Total     119  3.7807 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(jsd.dist ~ AB, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
```

```

## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB         1   0.1999 0.05286 6.586  0.001 ***
## Residual  118   3.5808 0.94714
## Total     119   3.7807 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jsd.dist ~ Farm2, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2      3   1.2936 0.34215 20.111  0.001 ***
## Residual  116   2.4871 0.65785
## Total     119   3.7807 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jsd.dist ~ Cox, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox        3   0.6795 0.17974 8.4727  0.001 ***
## Residual  116   3.1011 0.82026
## Total     119   3.7807 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jsd.dist ~ Researcher, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher  4   0.3554 0.094 2.9828  0.003 **
## Residual   115   3.4253 0.906

```

```
## Total      119  3.7807 1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(jsd.dist ~ LitterType, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ LitterType, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType    2   0.8587 0.22714 17.192  0.001 ***
## Residual    117   2.9219 0.77286
## Total       119   3.7807 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(jsd.dist ~ Gender, data = metadf) # not sign
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Gender, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender      1   0.0199 0.00527 0.6254  0.666
## Residual   118   3.7607 0.99473
## Total       119   3.7807 1.00000
```

```
adonis2(jsd.dist ~ Stables, data = metadf)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jsd.dist ~ Stables, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables     9   1.7634 0.46643 10.684  0.001 ***
## Residual  110   2.0172 0.53357
## Total      119   3.7807 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# same is true for JSD
```

```
bray.dist <- phyloseq::distance(ps1.rel, "bray")
```

```
adonis2(bray.dist ~ Age, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Age, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Age        1   0.7166 0.04537 5.6084  0.001 ***
## Residual  118  15.0772 0.95463
## Total     119  15.7938 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
adonis2(bray.dist ~ AB, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB        1   0.5354 0.0339 4.1401  0.001 ***
## Residual  118  15.2584 0.9661
## Total     119  15.7938 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
adonis2(bray.dist ~ Farm2, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2     3   3.661 0.2318 11.667  0.001 ***
## Residual  116  12.133 0.7682
## Total     119  15.794 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```
adonis2(bray.dist ~ Cox, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)

```

```

## Cox      3   1.9909 0.12606 5.5772  0.001 ***
## Residual 116  13.8028 0.87394
## Total    119  15.7938 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(bray.dist ~ Researcher, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher  4   1.1723 0.07422 2.305  0.001 ***
## Residual   115  14.6215 0.92578
## Total     119  15.7938 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(bray.dist ~ LitterType, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ LitterType, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType  2   2.4121 0.15273 10.545  0.001 ***
## Residual   117  13.3816 0.84727
## Total     119  15.7938 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(bray.dist ~ Gender, data = metadf) # not sign
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Gender, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender     1   0.0639 0.00405 0.4794   0.92
## Residual  118  15.7298 0.99595
## Total     119  15.7938 1.00000

```

```
adonis2(bray.dist ~ Stables, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = bray.dist ~ Stables, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables     9   5.2977 0.33543 6.169  0.001 ***
## Residual 110   10.4960 0.66457
## Total     119   15.7938 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# and BC

jaccard.dist <- phyloseq::distance(ps1.rel, "jaccard")

adonis2(jaccard.dist ~ Age, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Age, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Age        1   0.9791 0.03733 4.576  0.001 ***
## Residual 118   25.2484 0.96267
## Total     119   26.2276 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(jaccard.dist ~ AB, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB         1   0.7604 0.02899 3.5232  0.001 ***
## Residual 118   25.4672 0.97101
## Total     119   26.2276 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(jaccard.dist ~ Farm2, data = metadf)

```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free

```

```

## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2      3   4.8381 0.18446 8.7459  0.001 ***
## Residual  116  21.3895 0.81554
## Total     119  26.2276 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jaccard.dist ~ Cox, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox       3   2.6554 0.10124 4.3557  0.001 ***
## Residual  116  23.5722 0.89876
## Total     119  26.2276 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jaccard.dist ~ Researcher, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher 4   1.7617 0.06717 2.0702  0.001 ***
## Residual   115  24.4659 0.93283
## Total      119  26.2276 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
adonis2(jaccard.dist ~ LitterType, data = metadf)
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ LitterType, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType 2   3.1975 0.12191 8.1222  0.001 ***
## Residual   117  23.0301 0.87809
## Total      119  26.2276 1.00000

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(jaccard.dist ~ Gender, data = metadf) # not sign

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Gender, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender     1   0.1144 0.00436 0.517  0.978
## Residual  118  26.1132 0.99564
## Total     119  26.2276 1.00000

adonis2(jaccard.dist ~ Stables, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = jaccard.dist ~ Stables, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables    9   7.3682 0.28093 4.7751  0.001 ***
## Residual 110  18.8594 0.71907
## Total    119  26.2276 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# as well as jaccard

```

PERMANOVA plots - Age

```

permanova_age <- adonis(t(assay(tse, "relabundance")) ~ Age, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_age)[["Age1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

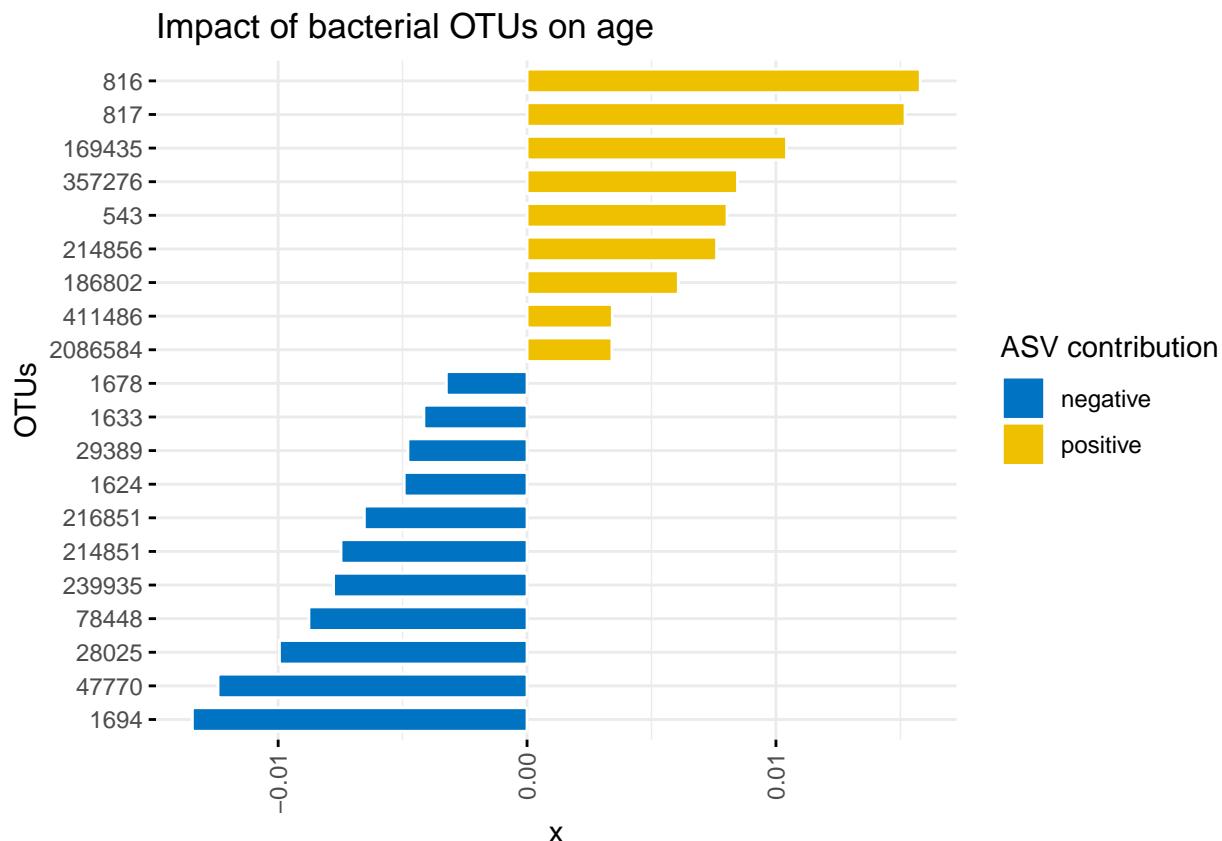
ggbarplot(df, x = "y", y = "x",
           fill = "contr",           # change fill color by mpg_level
           color = "white",          # Set bar border colors to white

```

```

palette = "jco",           # jco journal color palett. see ?ggpar
sort.val = "asc",          # Sort the value in ascending order
sort.by.groups = FALSE,    # Don't sort inside each group
x.text.angle = 90,         # Rotate vertically x axis texts
xlab = "OTUs",
legend.title = "ASV contribution",
title = "Impact of bacterial OTUs on age",
rotate = TRUE,
ggtheme = theme_minimal())

```



```

# AB

permanova_AB <- adonis(t(assay(tse, "relabundance")) ~ AB, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_AB)[["AB1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

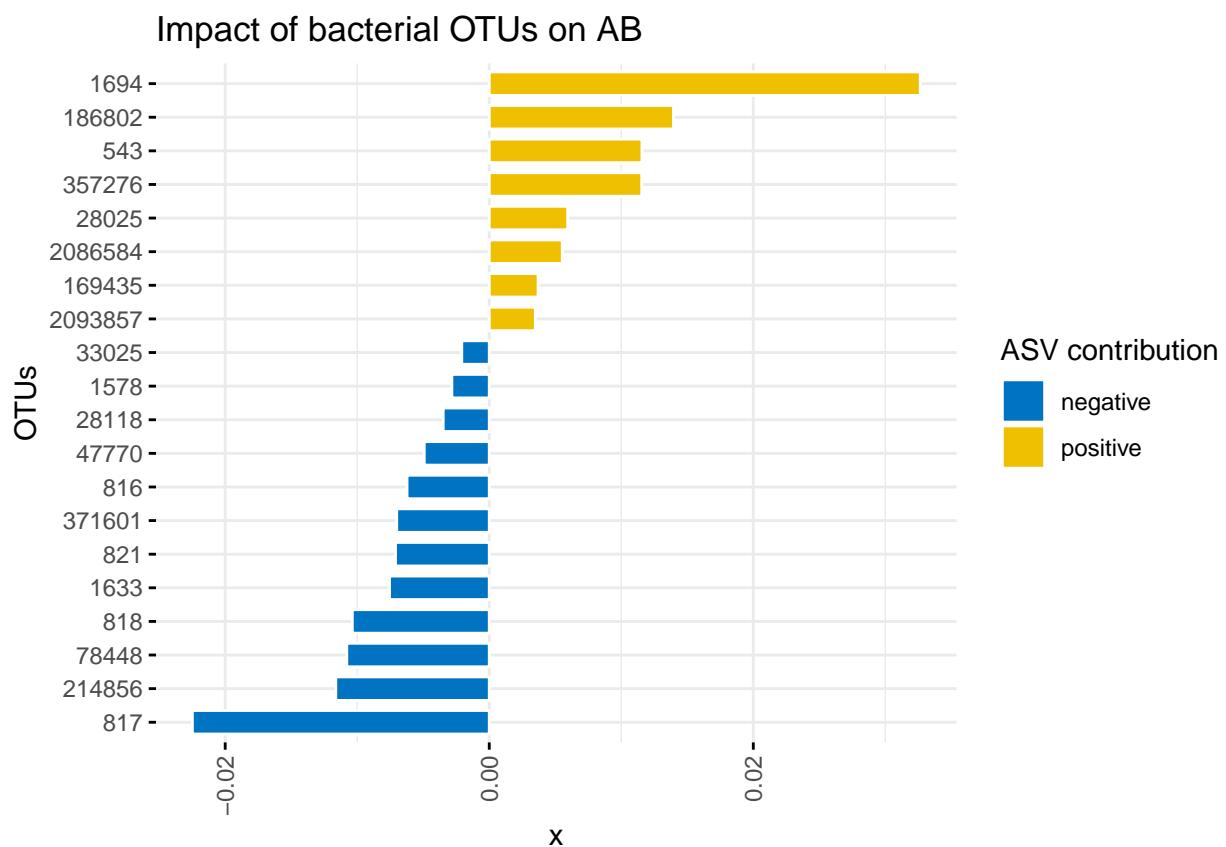
ggbarplot(df, x = "y", y = "x",

```

```

fill = "contr",           # change fill color by mpg_level
color = "white",          # Set bar border colors to white
palette = "jco",           # jco journal color palett. see ?ggpar
sort.val = "asc",          # Sort the value in ascending order
sort.by.groups = FALSE,    # Don't sort inside each group
x.text.angle = 90,         # Rotate vertically x axis texts
xlab = "OTUs",
legend.title = "ASV contribution",
title = "Impact of bacterial OTUs on AB",
rotate = TRUE,
ggtheme = theme_minimal())

```



```

# Stable

permanova_stable <- adonis(t(assay(tse, "relabundance")) ~ Stables, data = colData(tse), permutations = 1000)

coef <- coefficients(permanova_stable)[["Stables1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

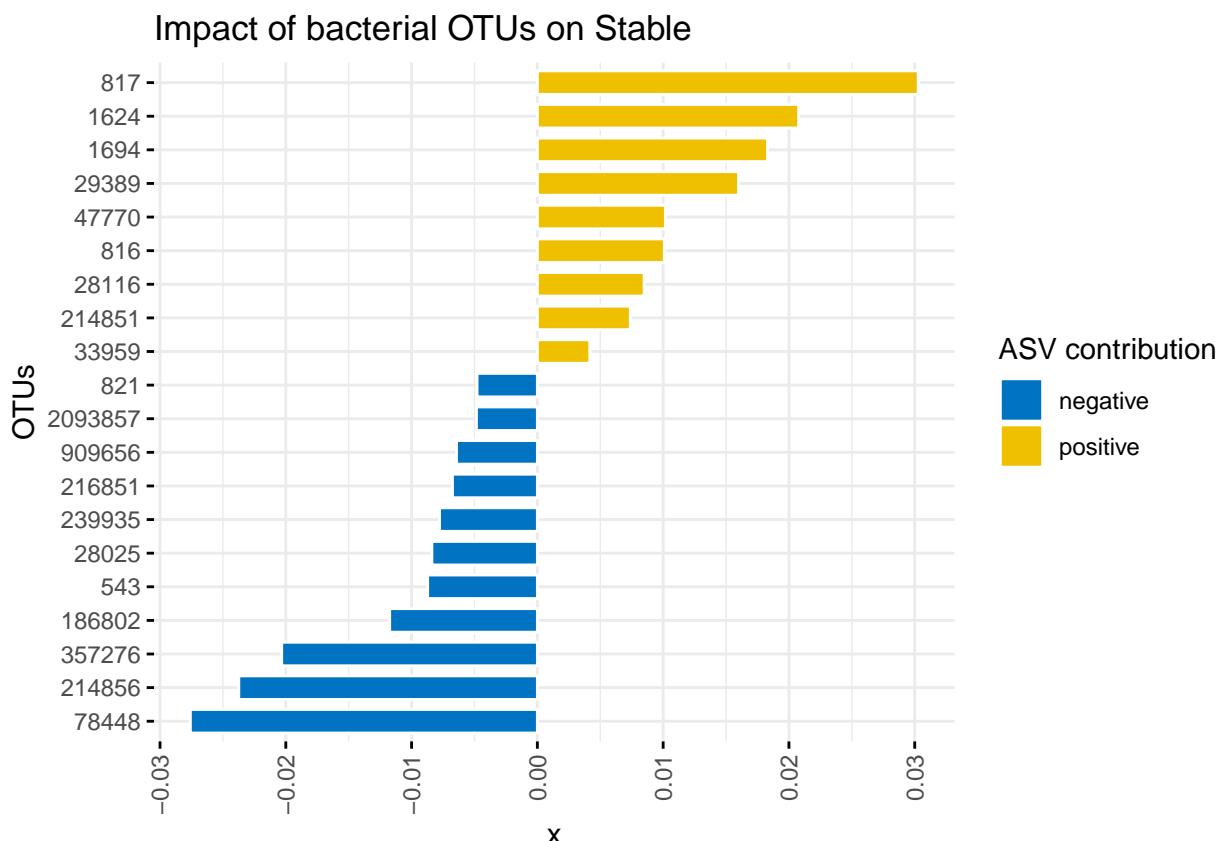
df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                    levels = c("negative", "positive"))

```

```

ggbbarplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",         # jco journal color palett. see ?ggpar
           sort.val = "asc",         # Sort the value in ascending order
           sort.by.groups = FALSE,   # Don't sort inside each group
           x.text.angle = 90,        # Rotate vertically x axis texts
           xlab = "OTUs",
           legend.title = "ASV contribution",
           title = "Impact of bacterial OTUs on Stable",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# Farm

permanova_farm <- adonis(t(assay(tse, "relabundance")) ~ Farm2, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_farm)[["Farm21",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef))),

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),

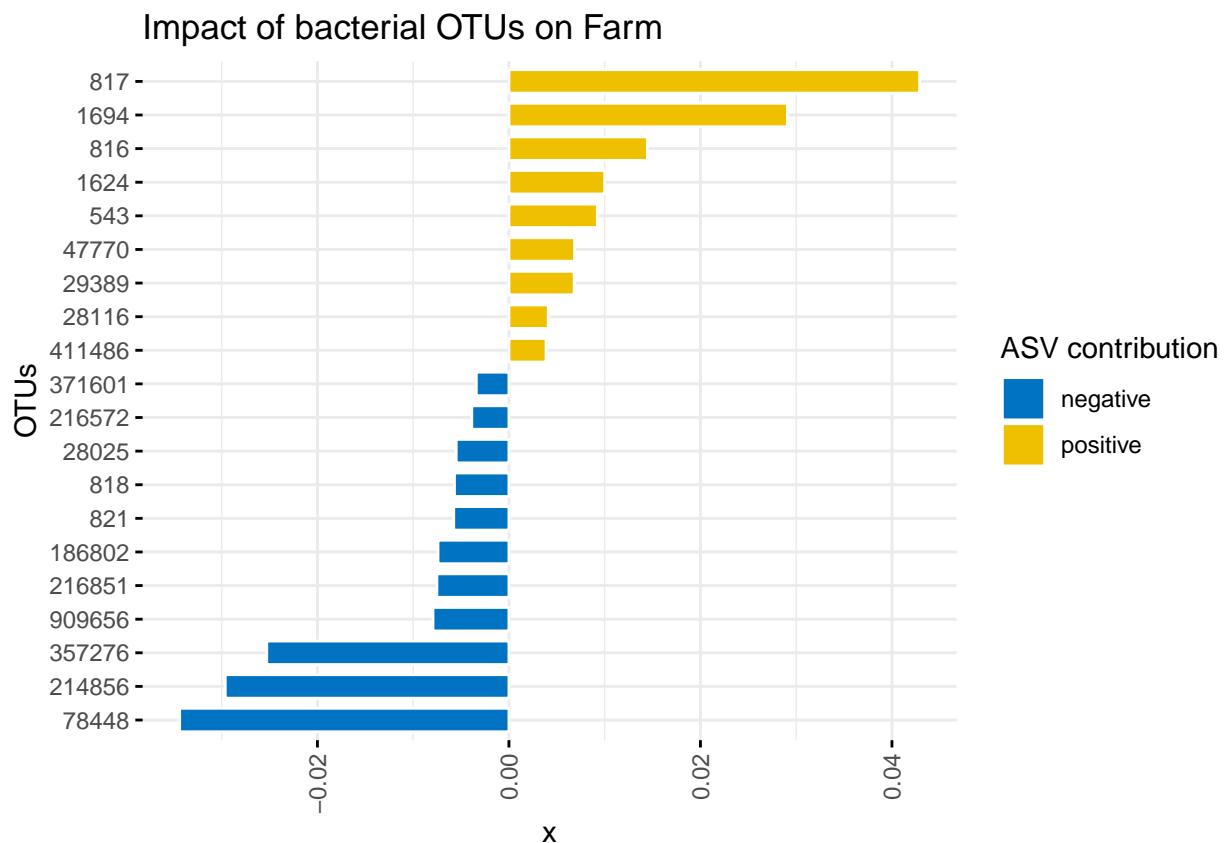
```

```

levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",          # jco journal color palett. see ?ggpar
           sort.val = "asc",          # Sort the value in ascending order
           sort.by.groups = FALSE,    # Don't sort inside each group
           x.text.angle = 90,          # Rotate vertically x axis texts
           xlab = "OTUs",
           legend.title = "ASV contribution",
           title = "Impact of bacterial OTUs on Farm",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# agent

permanova_agent <- adonis(t(assay(tse, "relabundance")) ~ Cox, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_agent)[["Cox1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef))))

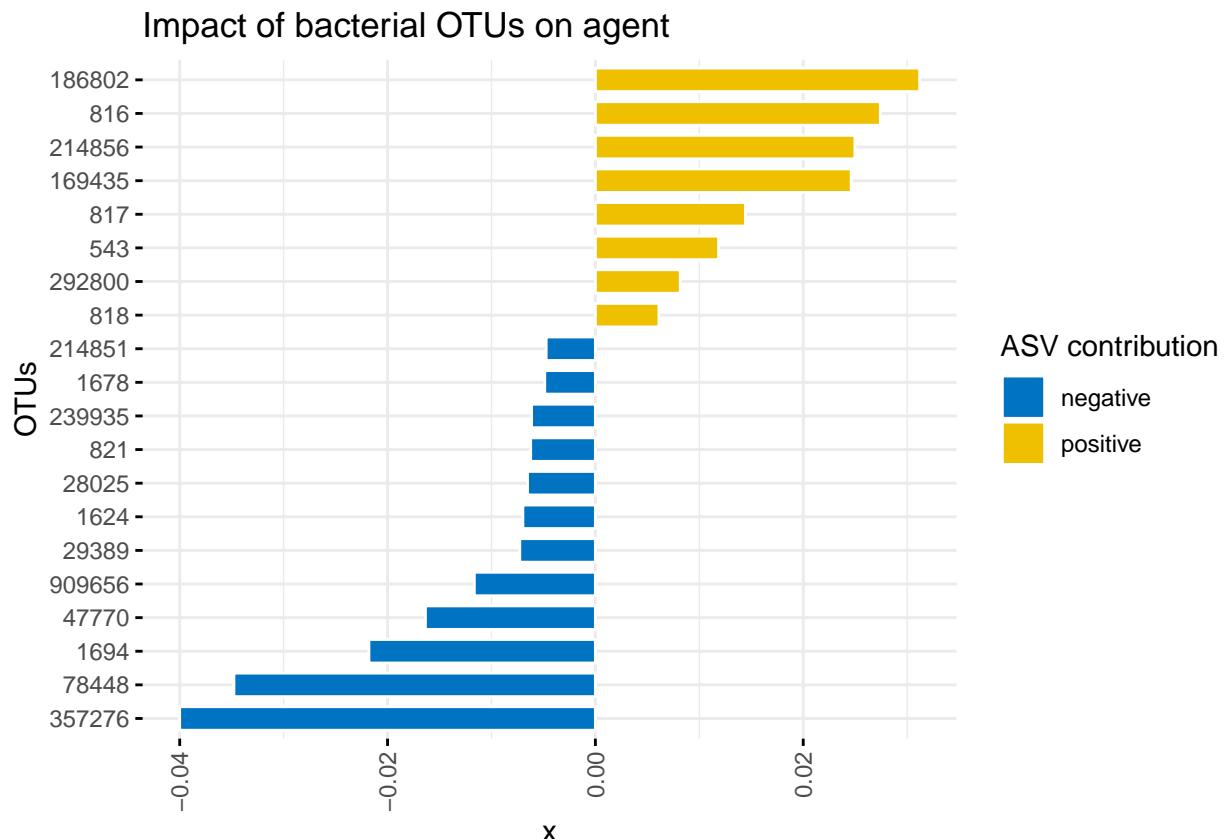
```

```

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",                      # change fill color by mpg_level
           color = "white",                      # Set bar border colors to white
           palette = "jco",                      # jco journal color palett. see ?ggpar
           sort.val = "asc",                      # Sort the value in ascending order
           sort.by.groups = FALSE,                # Don't sort inside each group
           x.text.angle = 90,                     # Rotate vertically x axis texts
           xlab = "OTUs",                        # x-axis label
           legend.title = "ASV contribution",    # legend title
           title = "Impact of bacterial OTUs on agent",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# same plots but for genera - age

permanova_age <- adonis(t(assay(tse_genus, "relabundance")) ~ Age, data = colData(tse_genus), permutations = 999)

coef <- coefficients(permanova_age)[["Age1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef))))

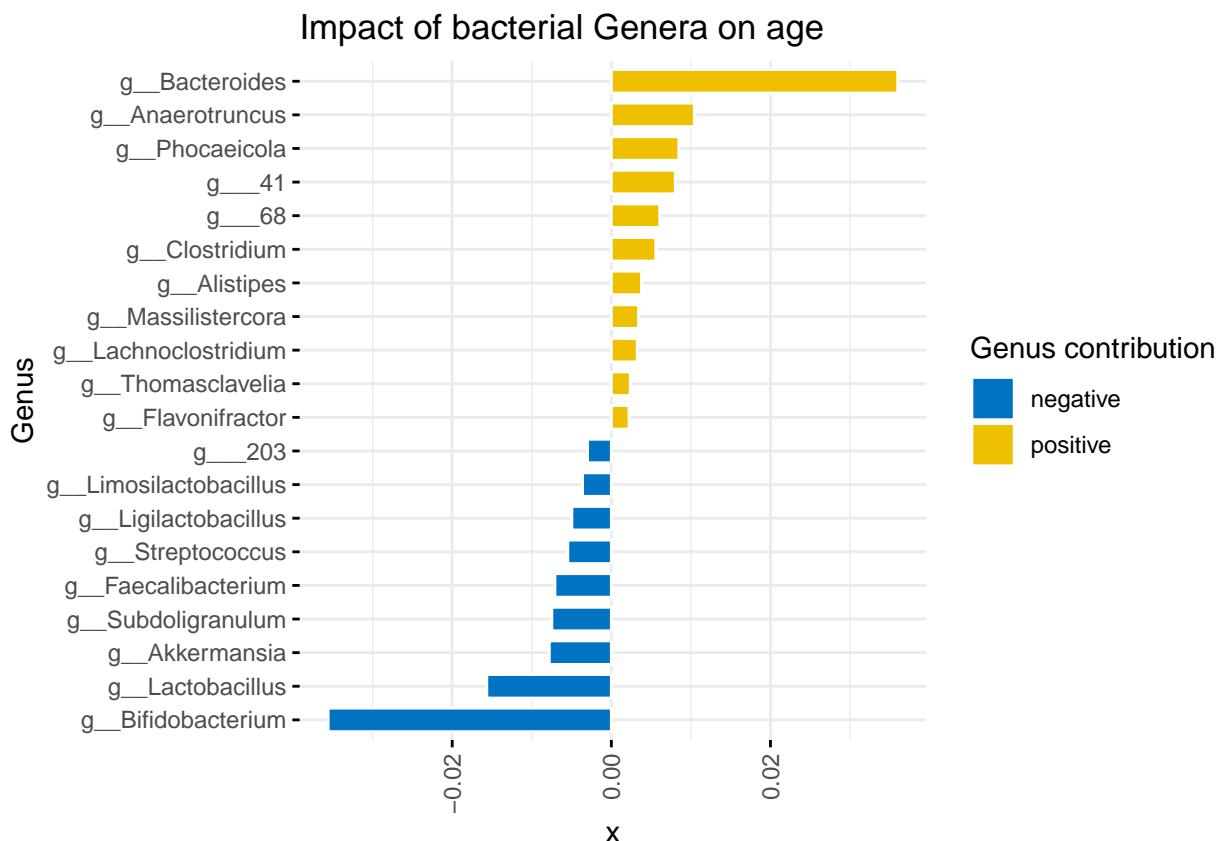
```

```

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",                      # change fill color by mpg_level
           color = "white",                      # Set bar border colors to white
           palette = "jco",                      # jco journal color palett. see ?ggpar
           sort.val = "asc",                      # Sort the value in ascending order
           sort.by.groups = FALSE,                # Don't sort inside each group
           x.text.angle = 90,                     # Rotate vertically x axis texts
           xlab = "Genus",
           legend.title = "Genus contribution",
           title = "Impact of bacterial Genera on age",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# AB

permanova_AB <- adonis(t(assay(tse_genus, "relabundance")) ~ AB, data = colData(tse_genus), permutations = 1000)

coef <- coefficients(permanova_AB)[["AB1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef)),
                 stringsAsFactors = FALSE)

```

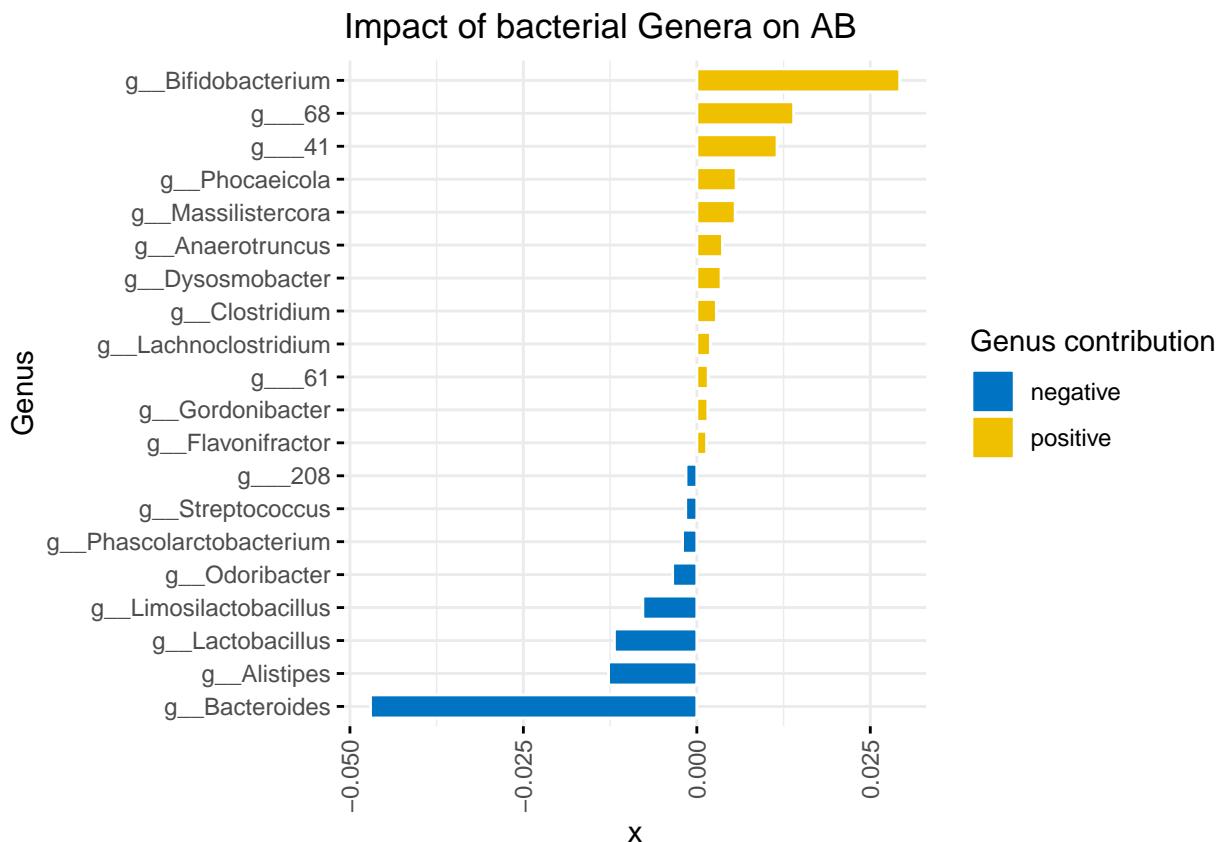
```

unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",                      # change fill color by mpg_level
           color = "white",                      # Set bar border colors to white
           palette = "jco",                      # jco journal color palett. see ?ggpar
           sort.val = "asc",                      # Sort the value in ascending order
           sort.by.groups = FALSE,                # Don't sort inside each group
           x.text.angle = 90,                     # Rotate vertically x axis texts
           xlab = "Genus",
           legend.title = "Genus contribution",
           title = "Impact of bacterial Genera on AB",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# Stable

permanova_stable <- adonis(t(assay(tse_genus, "relabundance")) ~ Stables, data = colData(tse_genus), per

coef <- coefficients(permanova_stable)[["Stables1",]
top.coef <- sort(head(coef[rev(order(abs(coef)))]), 20))

```

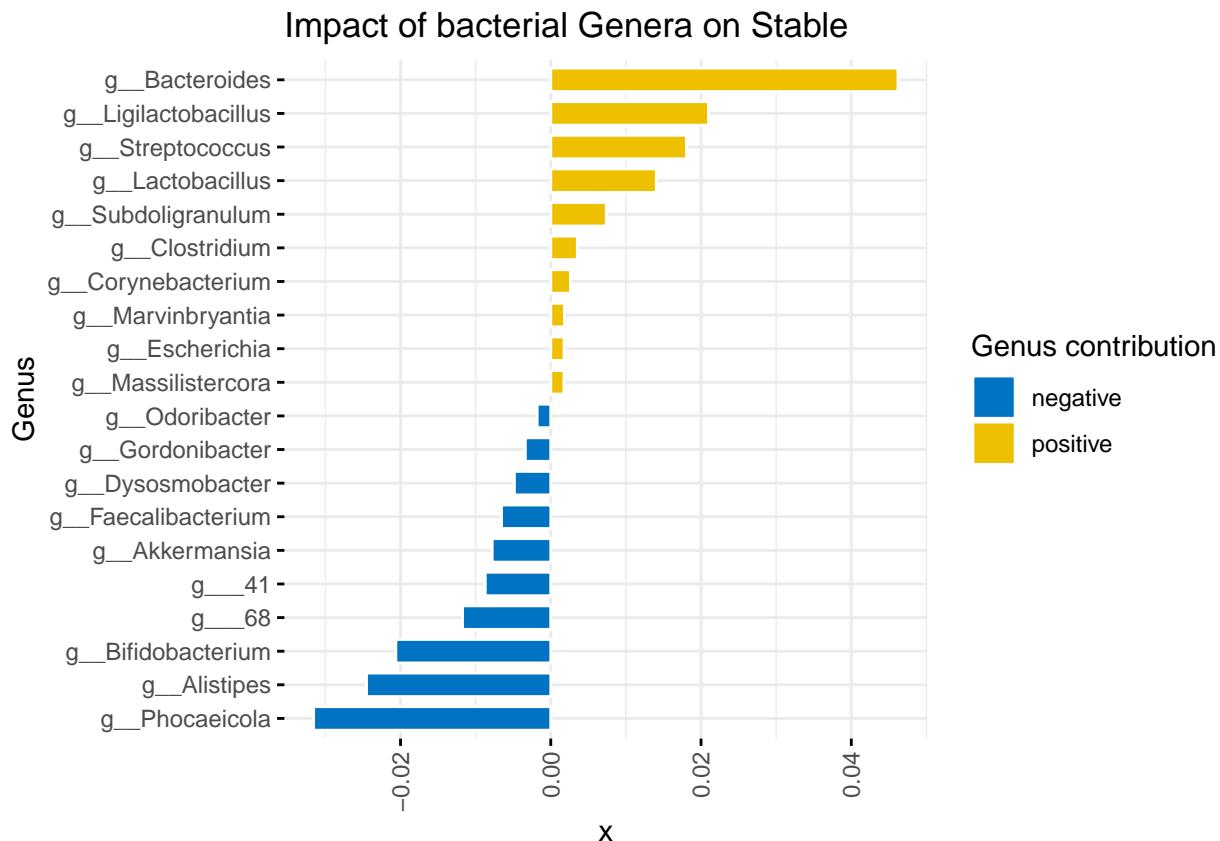
```

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                             unique(names(top.coef)))))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",          # jco journal color palett. see ?ggpar
           sort.val = "asc",         # Sort the value in ascending order
           sort.by.groups = FALSE,   # Don't sort inside each group
           x.text.angle = 90,        # Rotate vertically x axis texts
           xlab = "Genus",
           legend.title = "Genus contribution",
           title = "Impact of bacterial Genera on Stable",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# Farm

permanova_farm <- adonis(t(assay(tse_genus, "relabundance")) ~ Farm2, data = colData(tse_genus), permutations = 999)

coefficients(permanova_farm) ["Farm21",]

```

```

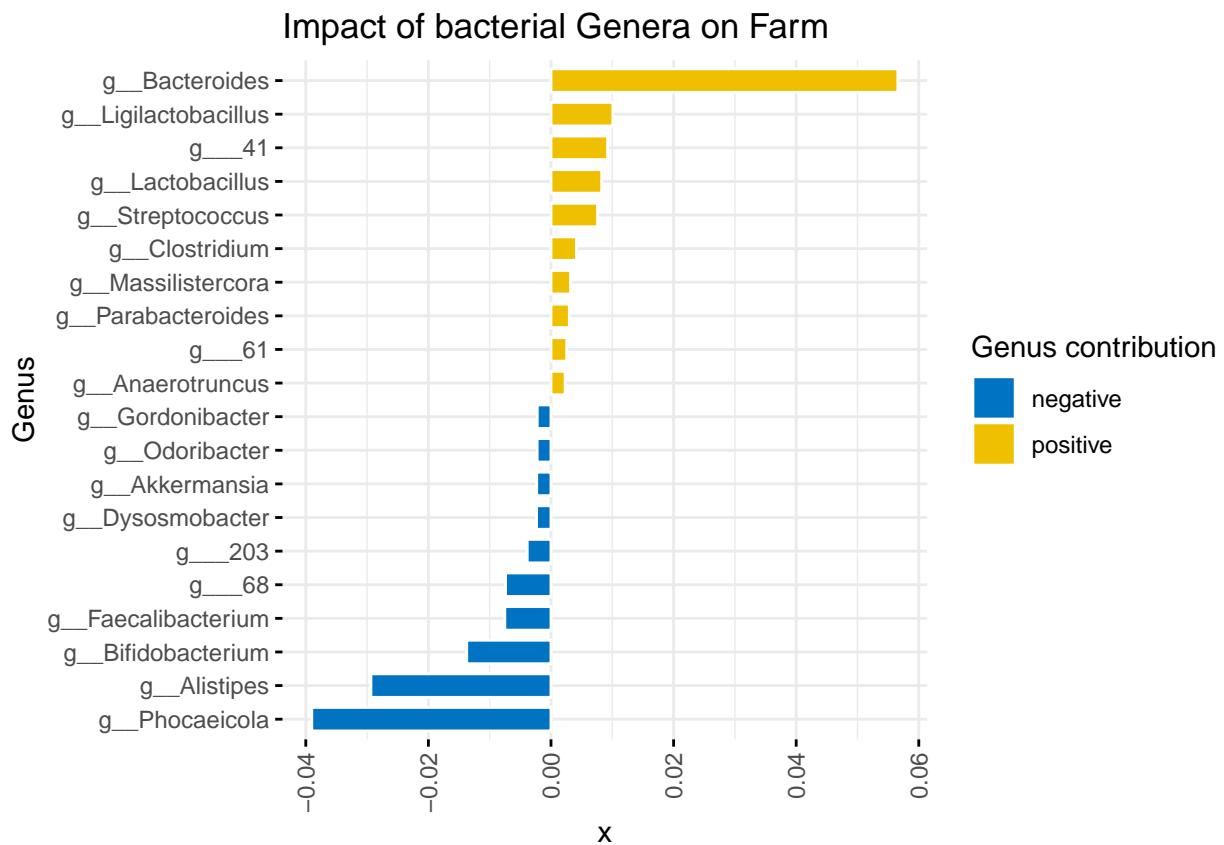
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                             unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",          # jco journal color palett. see ?ggpar
           sort.val = "asc",         # Sort the value in ascending order
           sort.by.groups = FALSE,   # Don't sort inside each group
           x.text.angle = 90,        # Rotate vertically x axis texts
           xlab = "Genus",
           legend.title = "Genus contribution",
           title = "Impact of bacterial Genera on Farm",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# Agent

permanova_agent <- adonis(t(assay(tse_genus, "relabundance")) ~ Cox, data = colData(tse_genus), permutations = 999)

```

```

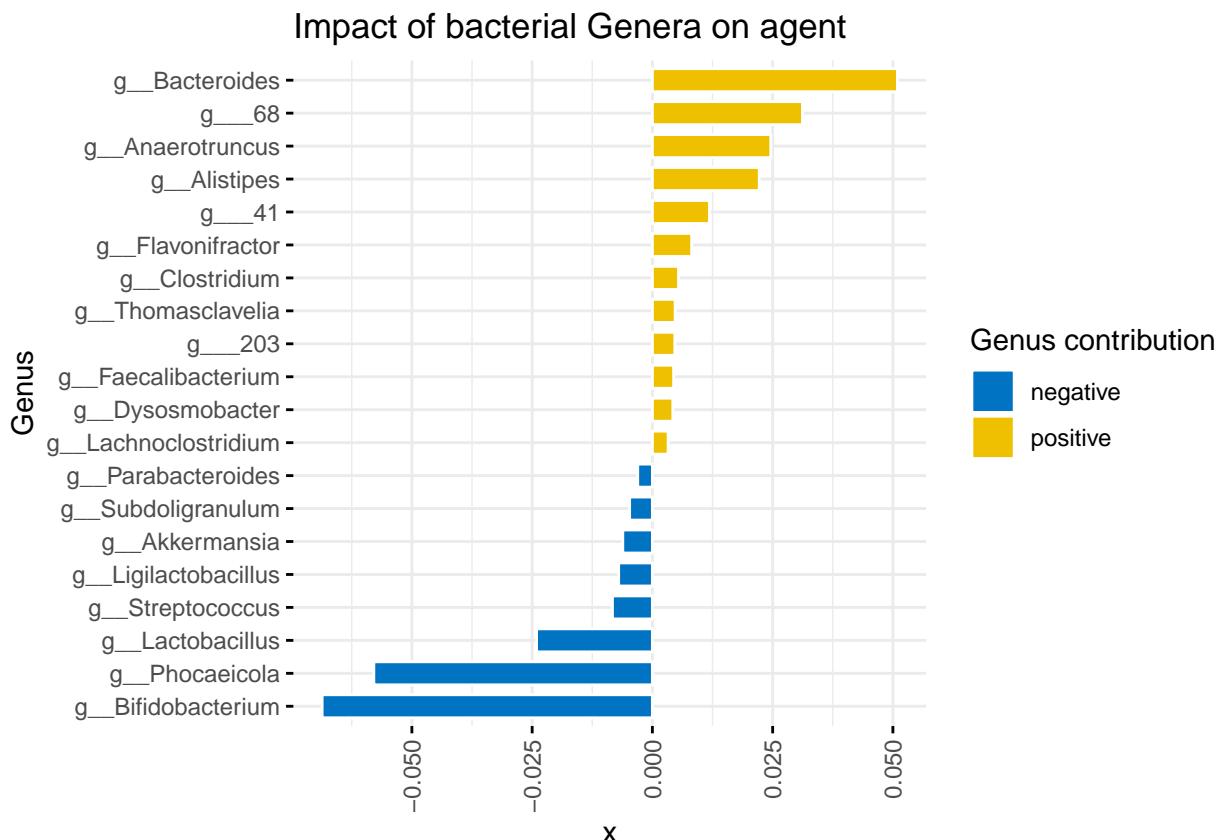
coef <- coefficients(permanova_agent)[ "Cox1", ]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                             unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",                      # change fill color by mpg_level
           color = "white",                     # Set bar border colors to white
           palette = "jco",                   # jco journal color palett. see ?ggpar
           sort.val = "asc",                  # Sort the value in ascending order
           sort.by.groups = FALSE,            # Don't sort inside each group
           x.text.angle = 90,                # Rotate vertically x axis texts
           xlab = "Genus",
           legend.title = "Genus contribution",
           title = "Impact of bacterial Genera on agent",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



checking homogeneity condition - bray ### ANOVAs are performed on betadispers of our rel abund data to test whether groups are more variable than others

```
# Bray
ps.rel = microbiome::transform(subsetMG, "compositional")
meta = meta(ps.rel)
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Age))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value Pr(>F)
## Groups      1 0.0018 0.0017998 0.1575 0.6922
## Residuals 118 1.3486 0.0114292
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$AB))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value Pr(>F)
## Groups      1 0.02684 0.026842 2.2436 0.1368
## Residuals 118 1.41171 0.011964
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Farm2))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value Pr(>F)
## Groups      3 0.02179 0.0072639 0.5719 0.6346
## Residuals 116 1.47333 0.0127011
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Stables))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value Pr(>F)
## Groups      9 0.09838 0.010931 1.057 0.4003
## Residuals 110 1.13750 0.010341
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Cox))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value Pr(>F)
## Groups      3 0.03972 0.013239 1.2876 0.282
## Residuals 116 1.19271 0.010282
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Researcher)) # not homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      4 0.22727 0.056817   5.135 0.000765 ***
## Residuals 115 1.27245 0.011065
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$LitterType)) # not homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      2 0.08141 0.040706   3.4736 0.03424 *
## Residuals 117 1.37109 0.011719
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Gender))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      1 0.00022 0.0002199   0.0191 0.8904
## Residuals 118 1.36149 0.0115381
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$FlockSize))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      5 0.05132 0.010263   0.8423 0.5225
## Residuals 114 1.38906 0.012185
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$AgeParentStock))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      4 0.02452 0.0061303   0.4984 0.737
## Residuals 115 1.41458 0.0123007
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$FeedProducent)) # not homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      2 0.14557 0.072786  5.7759 0.004053 **
## Residuals 117 1.47440 0.012602
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$FeedType))
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      1 0.0018 0.0017998  0.1575 0.6922
## Residuals 118 1.3486 0.0114292
```

ANOVAs do not work for ReadPerc, ReadToT, since they have different observations per group - continuous variables

We see that almost every variable is homogenously distributed, except for Researcher, LT and FP

Jaccard

```
anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Age))
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      1 0.00059 0.0005902   0.066 0.7977
## Residuals 118 1.05558 0.0089456
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$AB))
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      1 0.02772 0.0277242   2.9412 0.08897 .
## Residuals 118 1.11230 0.0094263
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Farm2))
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      3 0.0200 0.0066658   0.5905 0.6224
## Residuals 116 1.3094 0.0112876
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Stables)
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value Pr(>F)
## Groups      9 0.0783 0.0087000 0.9227 0.5085
## Residuals 110 1.0372 0.0094288
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Researcher)) # not homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Groups      4 0.24754 0.061884 7.2026 3.272e-05 ***
## Residuals 115 0.98807 0.008592
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$LitterType)) # not homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Groups      2 0.07866 0.039331 3.9483 0.02191 *
## Residuals 117 1.16552 0.009962
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Gender))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Groups      1 0.00038 0.0003807 0.0427 0.8367
## Residuals 118 1.05218 0.0089168
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$FlockSize))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df Sum Sq Mean Sq F value   Pr(>F)
## Groups      5 0.04708 0.009416 0.8455 0.5203
## Residuals 114 1.26962 0.011137
```

```

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$AgeParentStock))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      4 0.02168 0.0054212 0.4905 0.7427
## Residuals 115 1.27110 0.0110531

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$FeedProducent)) # not homogen

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      2 0.13238 0.066192 6.4706 0.002161 **
## Residuals 117 1.19687 0.010230
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$FeedType))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      1 0.00059 0.0005902 0.066 0.7977
## Residuals 118 1.05558 0.0089456

# same trends
# group variances are homogenous in most cases, so there are no differences in variances between groups

# Tukey tests can be performed to see if and which groups differ in relation to variance

TukeyHSD(betadisper(vegdist(t(abundances(ps.rel)))), meta$Farm2))

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
##
## $group
##          diff      lwr      upr      p adj
## Farm2-Farm1 0.018978801 -0.05446353 0.09242113 0.9068765
## Farm3-Farm1 0.003065054 -0.08173884 0.08786895 0.9996987
## Farm4-Farm1 0.037617091 -0.04718681 0.12242099 0.6555075
## Farm3-Farm2 -0.015913748 -0.08935608 0.05752858 0.9422838
## Farm4-Farm2 0.018638289 -0.05480404 0.09208062 0.9112724
## Farm4-Farm3 0.034552037 -0.05025186 0.11935593 0.7132117

```

```

# different way of calculating homogeneity, permutation tests, null = no difference in dispersion between groups
permutest(betadisper(vegdist(t(abundances(ps.rel))), meta$Age), pairwise = TRUE)

## 
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##          Df Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups      1 0.0018 0.0017998 0.1575    999   0.72
## Residuals 118 1.3486 0.0114292
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##          14     35
## 14        0.73
## 35  0.69221

permutest(betadisper(unifrac.dist, metadf$Age), pairwise = TRUE) # looks like unifrac distances are homogeneous

## 
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##          Df   Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups      1 0.000022 0.0000218 0.0487    999   0.824
## Residuals 118 0.052829 0.0004477
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##          14     35
## 14        0.822
## 35  0.82572

permutest(betadisper(unifrac.dist, metadf$AB), pairwise = TRUE) # not for AB though

## 
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##          Df   Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups      1 0.005927 0.0059266 15.638    999   0.001 ***
## Residuals 118 0.044720 0.0003790
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1
##
## Pairwise comparisons:

```

```

## (Observed p-value below diagonal, permuted p-value above diagonal)
##          no    yes
## no        0.001
## yes  0.00013123

permutest(betadisper(bray.dist, metadf$Age), pairwise = TRUE) # there are differences in P value with o

##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##              Df Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups       1 0.0018 0.0017998 0.1575     999   0.688
## Residuals 118 1.3486 0.0114292
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##          14     35
## 14      0.685
## 35  0.69221

```

SIMPER analyses

```

# We will automate simper with pre-existing scripts, sadly we cannot include all comparisons at once for
# convenience

#source("../Results/Scripts/Steinberger_scripts/simper_pretty.r")
#source("../Results/Scripts/Steinberger_scripts/R_krusk.r")

#Age
# again, functions are not repeated for rmarkdown but loaded in locally
#simper.pretty(otu_table(subsetMG), metrics = sample_data(subsetMG), interesting = c("Age"), perc_cutoff = 0.05)

#MG_age = data.frame(read.csv("MG_age_clean_simper.csv"))

#kruskal.pretty(otu_table(subsetMG), metrics = sample_data(subsetMG), csv = MG_age, interesting = c('Age'))

KW_MG_age = data.frame(read.csv("MG_Age_krusk_simper.csv"))
KW_MG_age = KW_MG_age[KW_MG_age$fdr_krusk_p.val < 0.05,] # filter out non-significant results, based on
# KW_MG_age = KW_MG_age[with(KW_MG_age, order(SIMPER, decreasing = TRUE)),]
KW_MG_age$OTU = as.factor(KW_MG_age$OTU)

KW_MG_age %>% mutate_if(is.numeric, format.pval, 2) %>% dplyr::select("SIMPER", "OTU", "fdr_krusk_p.val")
  rowwise() %>% mutate(Combined = paste("OTU =", OTU, ", SIMPER =", SIMPER, ", p-value =", fdr_krusk_p.
  dplyr::select(Combined)

## # A tibble: 13 x 1
## # Rowwise:
##   Combined

```

```

##      <chr>
## 1 OTU = 1694 , SIMPER = 0.100 , p-value = 0.04022
## 2 OTU = 47770 , SIMPER = 0.049 , p-value = 0.01784
## 3 OTU = 78448 , SIMPER = 0.045 , p-value = 0.02363
## 4 OTU = 214856 , SIMPER = 0.040 , p-value = 2.8e-05
## 5 OTU = 216851 , SIMPER = 0.034 , p-value = 0.00050
## 6 OTU = 357276 , SIMPER = 0.031 , p-value = 0.02363
## 7 OTU = 169435 , SIMPER = 0.031 , p-value = 5.4e-05
## 8 OTU = 214851 , SIMPER = 0.027 , p-value = 0.01457
## 9 OTU = 216572 , SIMPER = 0.023 , p-value = 0.04022
## 10 OTU = 28025 , SIMPER = 0.021 , p-value = 0.00160
## 11 OTU = 239935 , SIMPER = 0.014 , p-value = 0.00053
## 12 OTU = 1624 , SIMPER = 0.013 , p-value = 7.3e-07
## 13 OTU = 1678 , SIMPER = 0.010 , p-value = 0.00660

#AB
#simper.pretty(otu_table(subsetMG), metrics = sample_data(subsetMG), interesting = c("AB"), perc_cutoff

#MG_AB = data.frame(read.csv("MG_AB_clean_simper.csv"))

#kruskal.pretty(otu_table(subsetMG), metrics = sample_data(subsetMG), csv = MG_AB, interesting = c('AB'))

KW_MG_AB = data.frame(read.csv("MG_AB_krusk_simper.csv"))
KW_MG_AB = KW_MG_AB[KW_MG_AB$fdr_krusk_p.val < 0.05,] # filter out non-significant results, based on fdr
KW_MG_AB = KW_MG_AB[with(KW_MG_AB, order(SIMPER, decreasing = TRUE)),]
KW_MG_AB$OTU = as.factor(KW_MG_AB$OTU)

KW_MG_AB %>% mutate_if(is.numeric, format.pval, 2) %>% dplyr::select("SIMPER", "OTU", "fdr_krusk_p.val")
  rowwise() %>% mutate(Combined = paste("OTU =", OTU, ", SIMPER =", SIMPER, ", p-value =", fdr_krusk_p.val))
  dplyr::select(Combined)
}

## # A tibble: 11 x 1
## # Rowwise:
##   Combined
##   <chr>
## 1 OTU = 817 , SIMPER = 0.071 , p-value = 0.00780
## 2 OTU = 1694 , SIMPER = 0.068 , p-value = 0.00024
## 3 OTU = 214856 , SIMPER = 0.062 , p-value = 2.0e-05
## 4 OTU = 47770 , SIMPER = 0.050 , p-value = 0.02716
## 5 OTU = 816 , SIMPER = 0.049 , p-value = 0.01700
## 6 OTU = 357276 , SIMPER = 0.020 , p-value = 0.02716
## 7 OTU = 818 , SIMPER = 0.019 , p-value = 0.00031
## 8 OTU = 1633 , SIMPER = 0.018 , p-value = 0.00780
## 9 OTU = 821 , SIMPER = 0.018 , p-value = 2.0e-05
## 10 OTU = 28025 , SIMPER = 0.012 , p-value = 0.00369
## 11 OTU = 371601 , SIMPER = 0.011 , p-value = 2.4e-06

#Farms
#simper.pretty(otu_table(subsetMG), metrics = sample_data(subsetMG), interesting = c("Farm2"), perc_cutt

#G_Farm = data.frame(read.csv("MG_Farm_clean_simper.csv"))

#kruskal.pretty(otu_table(subsetMG), metrics = sample_data(subsetMG), csv = MG_Farm, interesting = c('Farm2'))

```

```

KW_MG_Farm = data.frame(read.csv("MG_Farm_krusk_simper.csv"))
KW_MG_Farm = KW_MG_Farm[KW_MG_Farm$fdr_krusk_p.val < 0.05,] # filter out non-significant results, based
KW_MG_Farm = KW_MG_Farm[with(KW_MG_Farm, order(SIMPER, decreasing = TRUE)),]
KW_MG_Farm$OTU = as.factor(KW_MG_Farm$OTU)

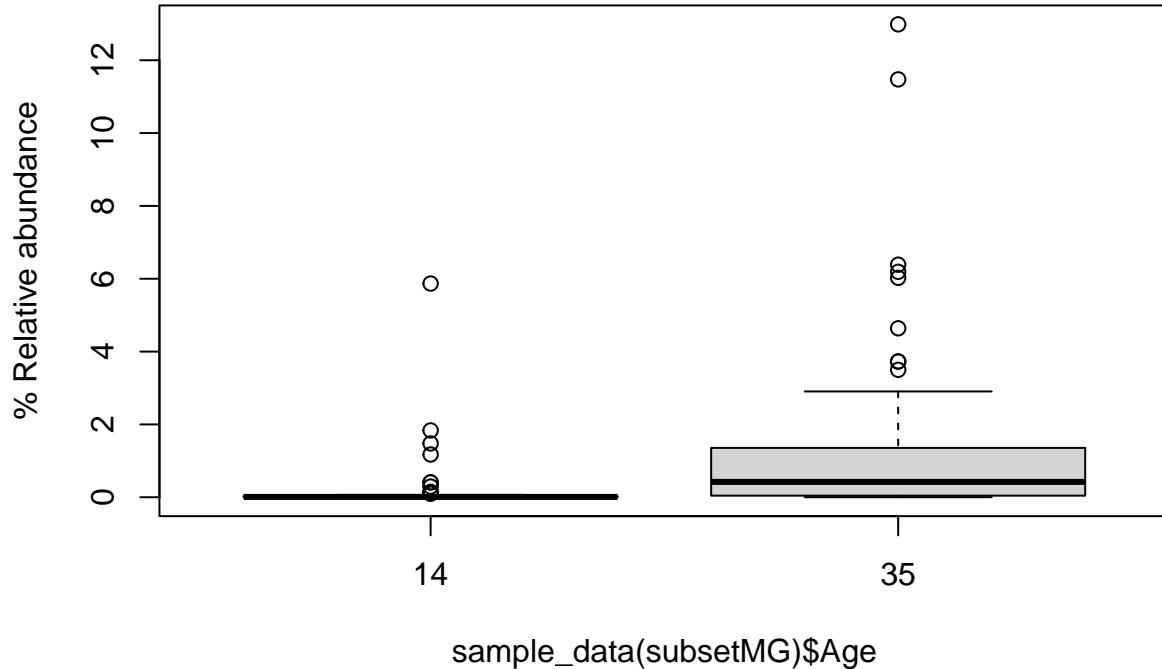
KW_MG_Farm %>% mutate_if(is.numeric, format.pval, 2) %>% dplyr::select("SIMPER", "OTU", "fdr_krusk_p.val")
  rowwise() %>% mutate(Combined = paste("OTU =", OTU, ", SIMPER =", SIMPER, ", p-value =", fdr_krusk_p.val))
  dplyr::select(Combined)

## # A tibble: 83 x 1
## # Rowwise:
##   Combined
##   <chr>
## 1 OTU = 78448 , SIMPER = 0.130 , p-value = 8.3e-08
## 2 OTU = 78448 , SIMPER = 0.113 , p-value = 8.2e-08
## 3 OTU = 1694 , SIMPER = 0.101 , p-value = 4.6e-07
## 4 OTU = 817 , SIMPER = 0.099 , p-value = 0.00562
## 5 OTU = 1694 , SIMPER = 0.098 , p-value = 0.00055
## 6 OTU = 78448 , SIMPER = 0.097 , p-value = 2.2e-06
## 7 OTU = 817 , SIMPER = 0.096 , p-value = 7.5e-08
## 8 OTU = 214856 , SIMPER = 0.095 , p-value = 0.02271
## 9 OTU = 186802 , SIMPER = 0.090 , p-value = 0.02370
## 10 OTU = 357276 , SIMPER = 0.089 , p-value = 4.3e-05
## # i 73 more rows

# plots to look at specific ASVs (age)
abund = otu_table(subsetMG)/rowSums(otu_table(subsetMG))*100
boxplot(unlist(data.frame(abund["1624"]))) ~ sample_data(subsetMG)$Age, ylab="% Relative abundance", main

```

OTU1



```

# specific test
kruskal.test(unlist(data.frame(otu_table(subsetMG) ["817"])), use.names = FALSE) ~ sample_data(subsetMG)$

##
## Kruskal-Wallis rank sum test
##
## data: unlist(data.frame(otu_table(subsetMG) ["817"])), use.names = FALSE by sample_data(subsetMG)$Age
## Kruskal-Wallis chi-squared = 0.98435, df = 1, p-value = 0.3211

# Trying out different distances, aggregation methods and indices for finding optimal number of clusters
tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)

tse <- transformCounts(tse, method = "relabundance")

assay <- t(assay(tse, "relabundance"))

diss_jaccard <- vegdist(assay, method = "jaccard")

# different aggregation methods and indices will grant different amount of clusters
NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "mcclain")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data

```

```

## Number_clusters      Value_Index
##                 2.0000      0.0542

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "frey")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##                 2.0000      7.9638

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "cindex")$Best.nc # 15 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##                 15.0000     0.3209

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##                 2.0000      0.2255

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "dunn")$Best.nc # 9 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##                 9.0000      0.502

NbClust(diss = diss_jaccard, distance = NULL, method = "ward.D2", index = "silhouette")$Best.nc # 3 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##                 3.0000      0.179

NbClust(diss = diss_jaccard, distance = NULL, method = "ward.D", index = "silhouette")$Best.nc # 14 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##                14.0000     0.1661

```

```

NbClust(diss = diss_jaccard, distance = NULL, method = "single", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.2207

NbClust(diss = diss_jaccard, distance = NULL, method = "average", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.2207

NbClust(diss = diss_jaccard, distance = NULL, method = "mcquitty", index = "silhouette")$Best.nc # 12 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                12.0000      0.1847

NbClust(diss = diss_jaccard, distance = NULL, method = "median", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.2858

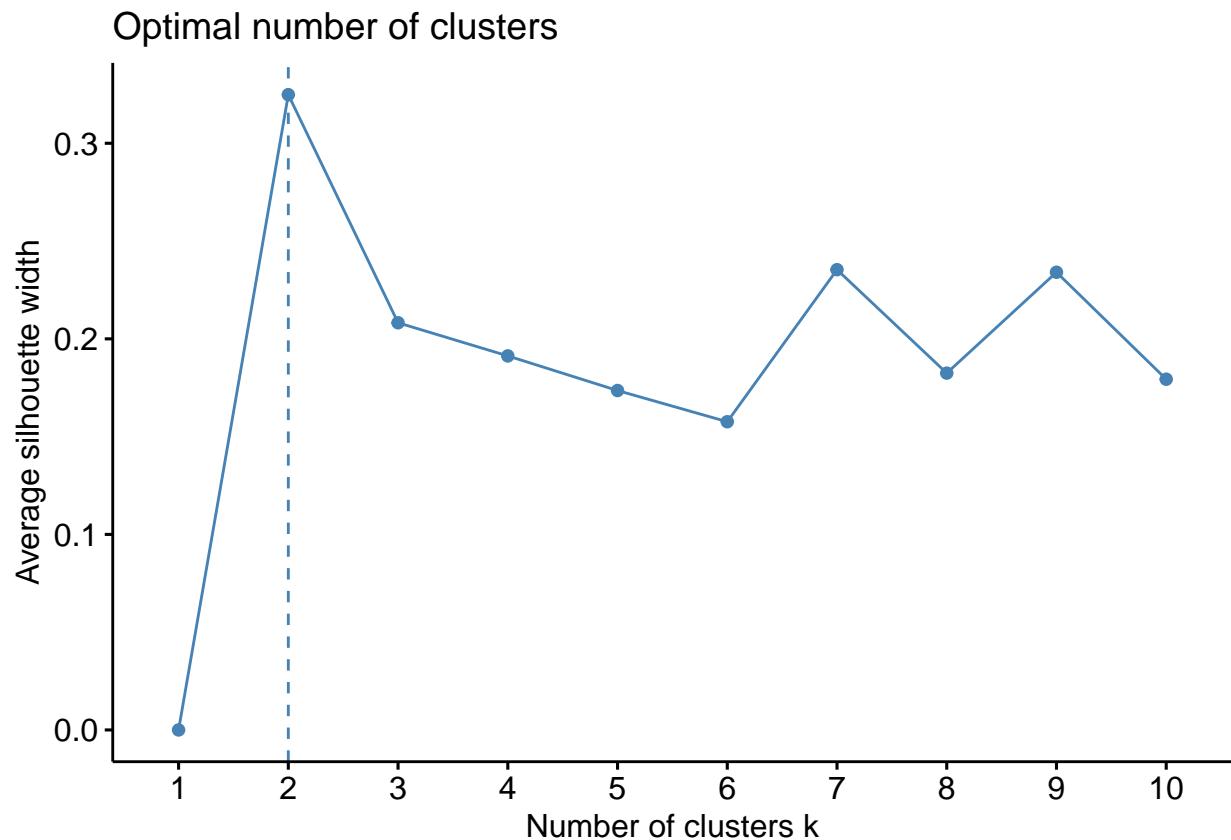
NbClust(diss = diss_jaccard, distance = NULL, method = "centroid", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

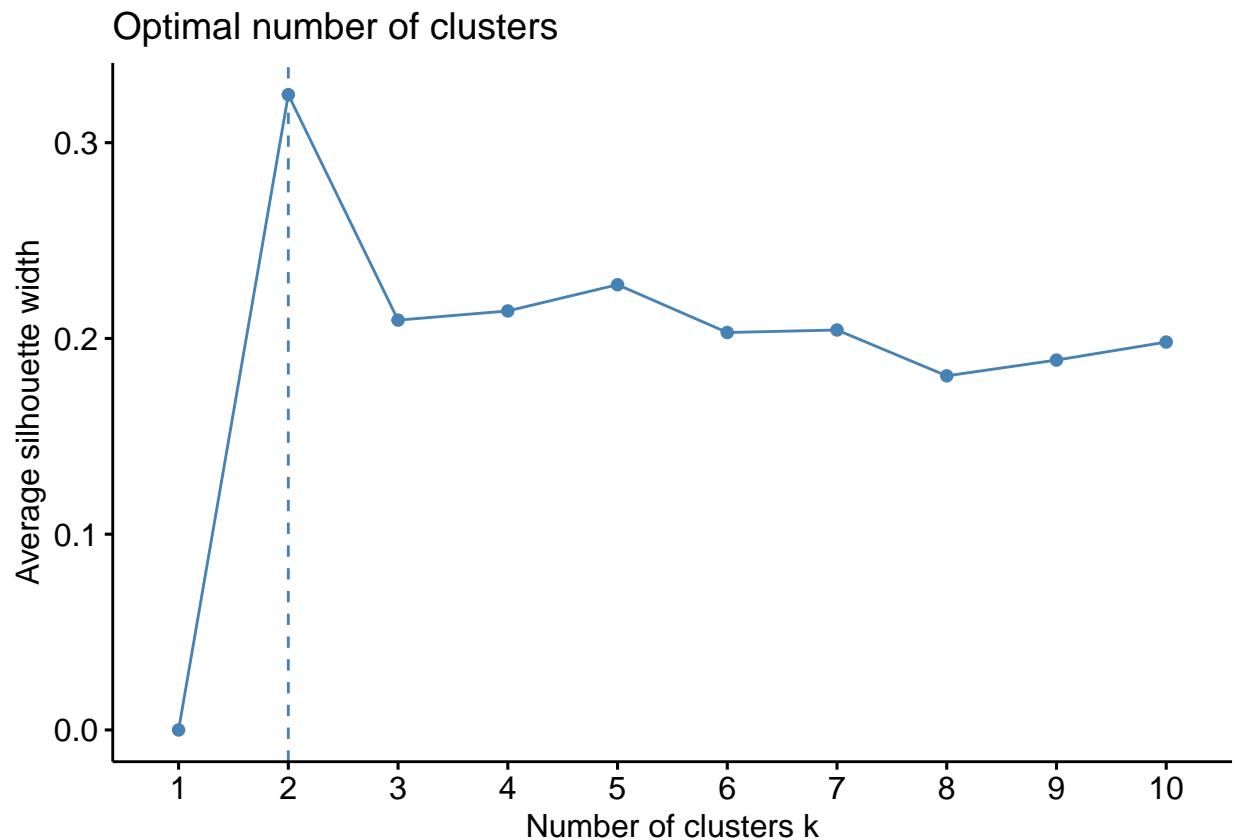
## Number_clusters      Value_Index
##                 2.0000      0.2207

# silhouette (ASW), different clustering methods
diss_jaccard <- as.matrix(diss_jaccard)
fviz_nbclust(diss_jaccard, kmeans, method = "silhouette") # 2 seems optimal for k-means

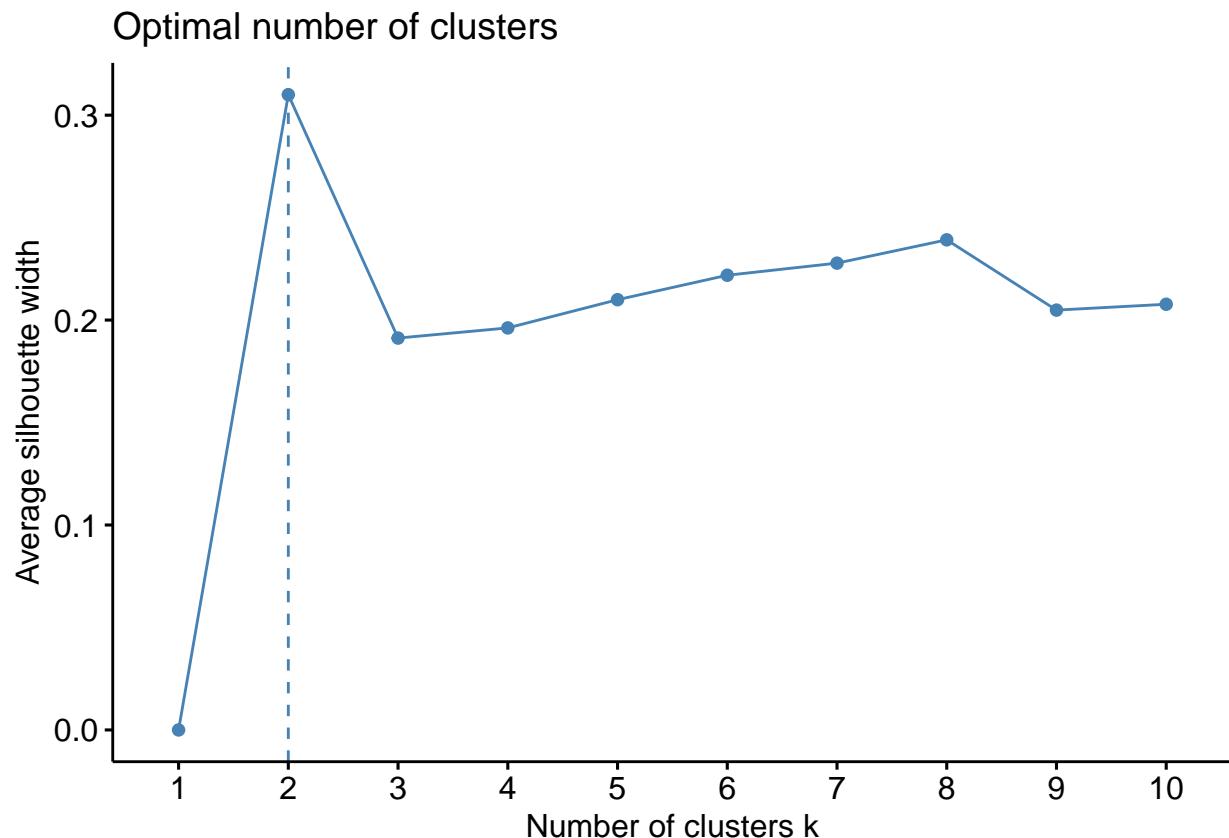
```



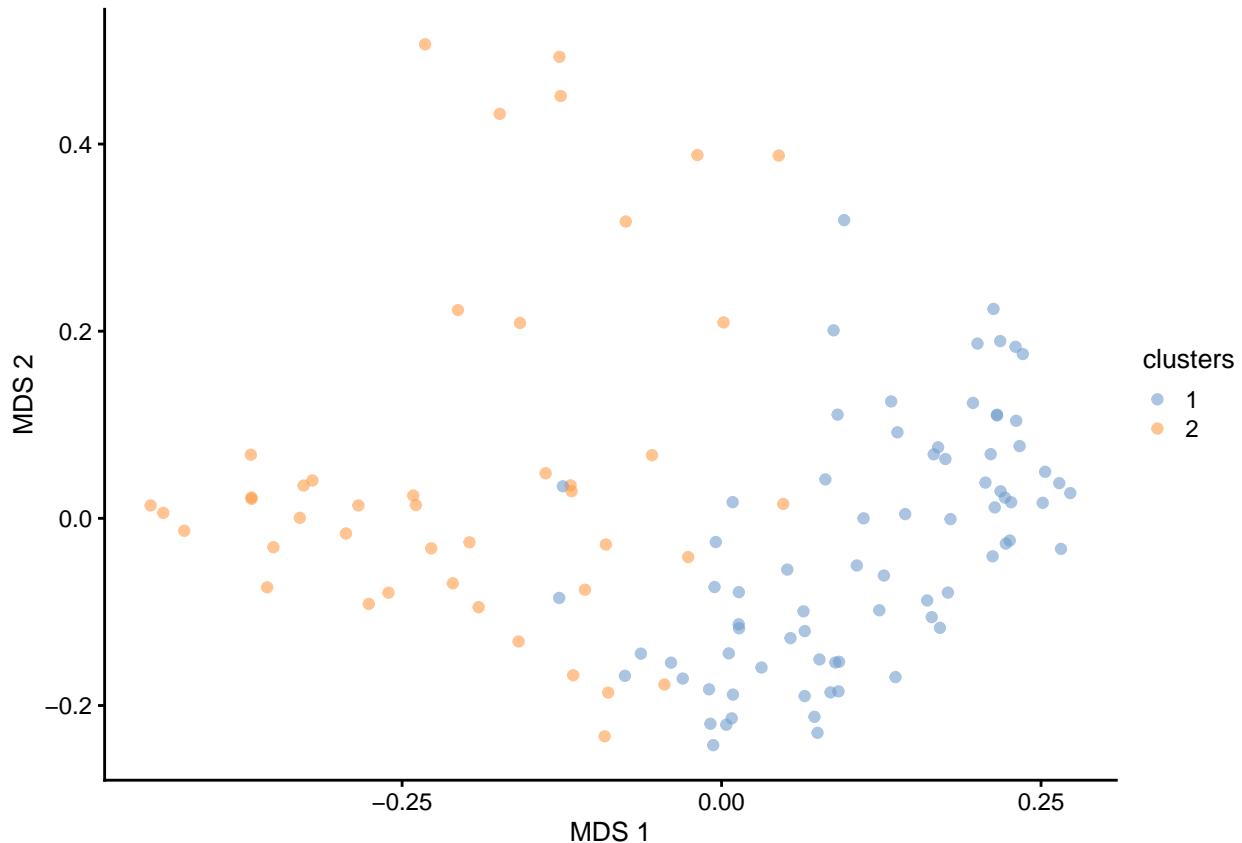
```
fviz_nbclust(diss_jaccard, cluster::pam, method = "silhouette") # 2 seems optimal for PAM
```



```
fviz_nbclust(diss_jaccard, hcut, method = "silhouette") # 2 seems optimal for hcut
```



```
# k-means
set.seed(1337)
km <- kmeans(diss_jaccard, 2, nstart = 25)
colData(tse)$clusters <- as.factor(km$cluster)
tse <- runMDS(tse, assay.type = "relabundance", FUN = vegan::vegdist, method = "jaccard")
plotReducedDim(tse, "MDS", colour_by = "clusters")
```



```

# now, let's repeat this for BC

diss_bray <- vegdist(assay, method = "bray")

#diss_euk <- vegdist(assay, method = "euclidian")

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "mcclain")$Best.nc # two clusters

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be in a data frame

## Number_clusters      Value_Index
##          2.0000        0.0475

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "frey")$Best.nc # 3 clusters

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be in a data frame

## Number_clusters      Value_Index
##          3.0000        1.1822

```

```

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "cindex")$Best.nc # 15 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##          15.0000        0.2162

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "silhouette")$Best.nc # two clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##          2.0000        0.3185

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "dunn")$Best.nc # 9 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##          9.0000        0.3901

NbClust(diss = diss_bray, distance = NULL, method = "ward.D2", index = "silhouette")$Best.nc # 3 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##          3.0000        0.2425

NbClust(diss = diss_bray, distance = NULL, method = "ward.D", index = "silhouette")$Best.nc # 3 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##          3.0000        0.2425

NbClust(diss = diss_bray, distance = NULL, method = "single", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##          2.0000        0.3056

```

```

NbClust(diss = diss_bray, distance = NULL, method = "average", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.3056

NbClust(diss = diss_bray, distance = NULL, method = "mcquitty", index = "silhouette")$Best.nc # 6 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 6.0000      0.2408

NbClust(diss = diss_bray, distance = NULL, method = "median", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.4058

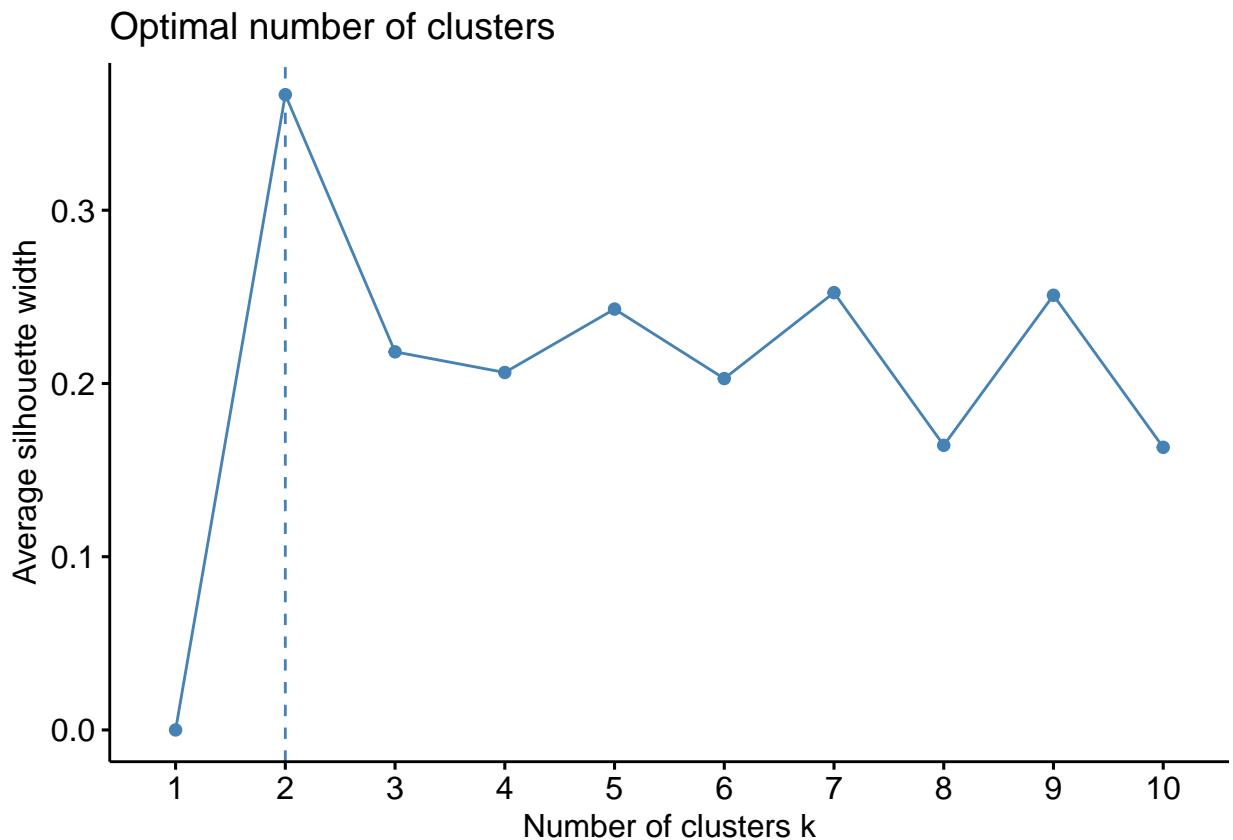
NbClust(diss = diss_bray, distance = NULL, method = "centroid", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

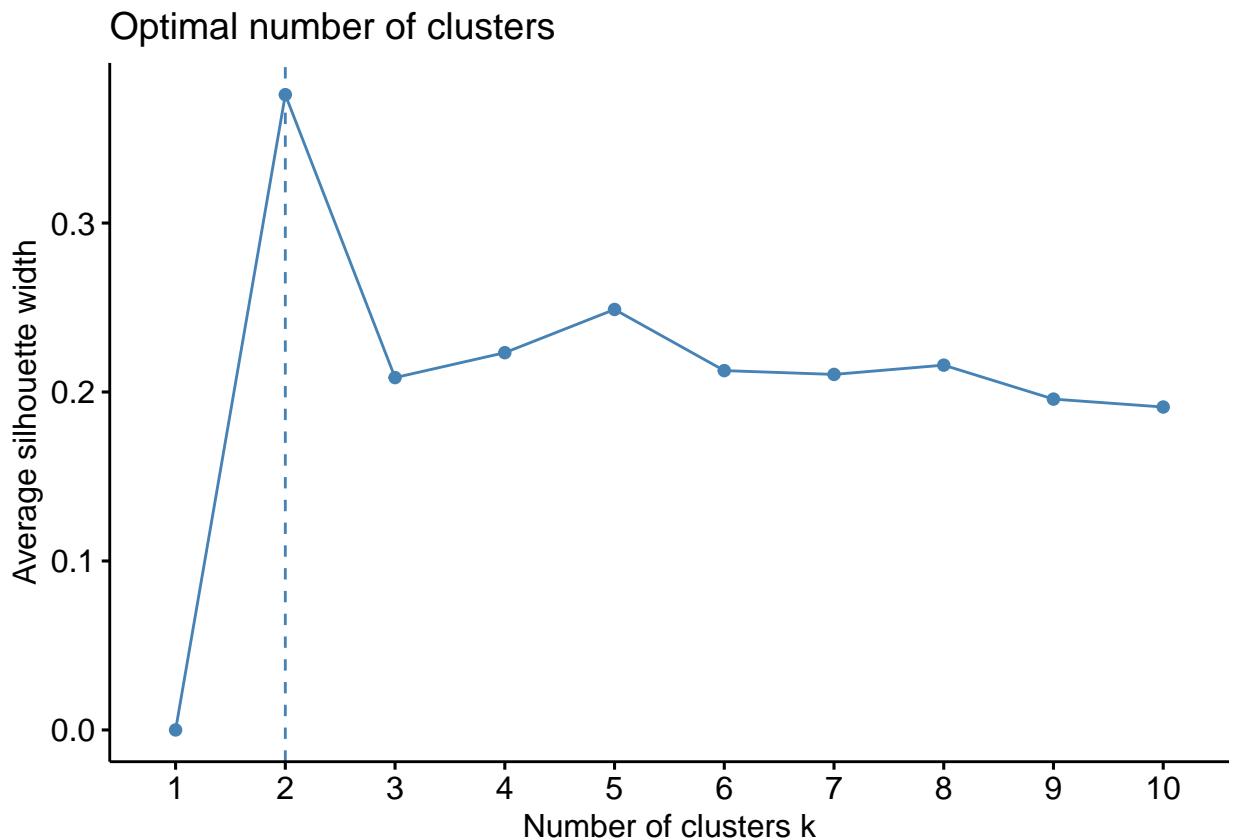
## Number_clusters      Value_Index
##                 2.0000      0.4058

# silhouette (ASW)
diss_bray <- as.matrix(diss_bray)
fviz_nbclust(diss_bray, kmeans, method = "silhouette") # 2 seems optimal

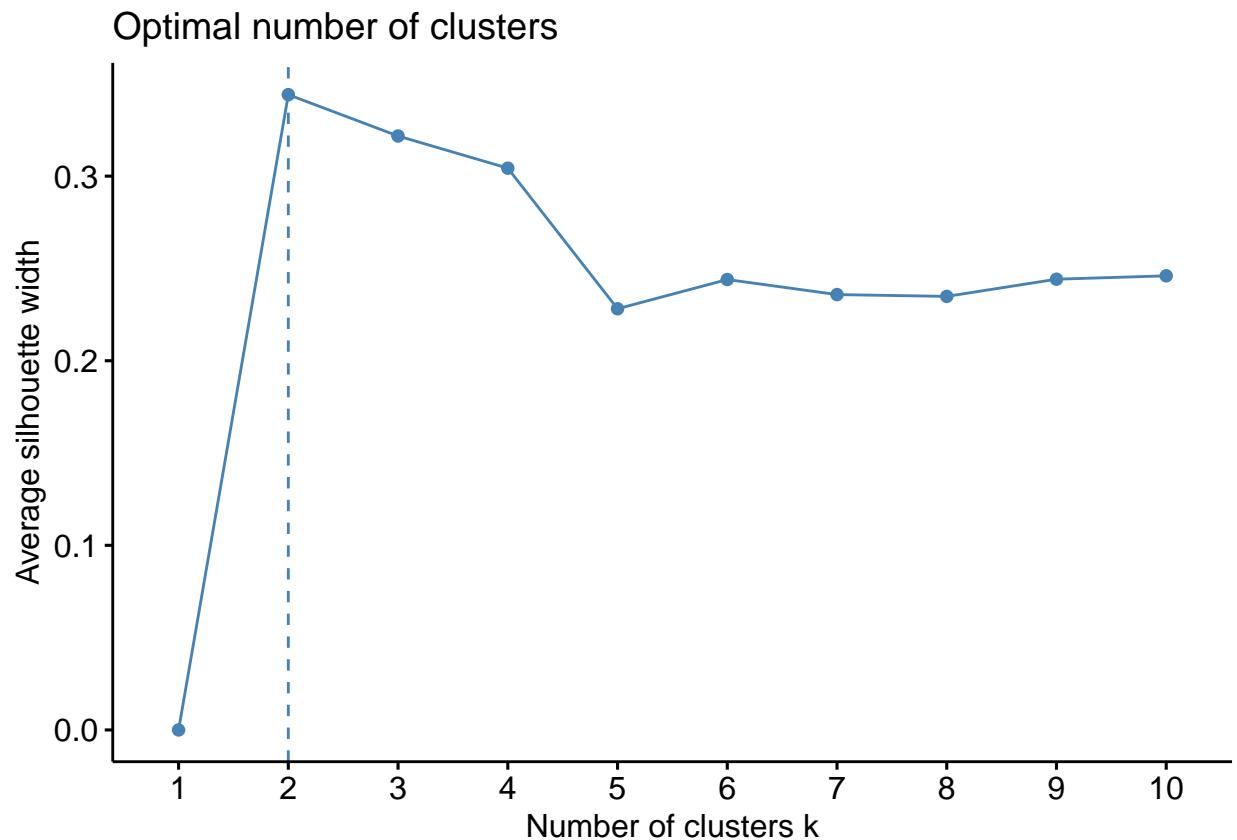
```



```
fviz_nbclust(diss_bray, cluster::pam, method = "silhouette") # 2 seems optimal for PAM
```

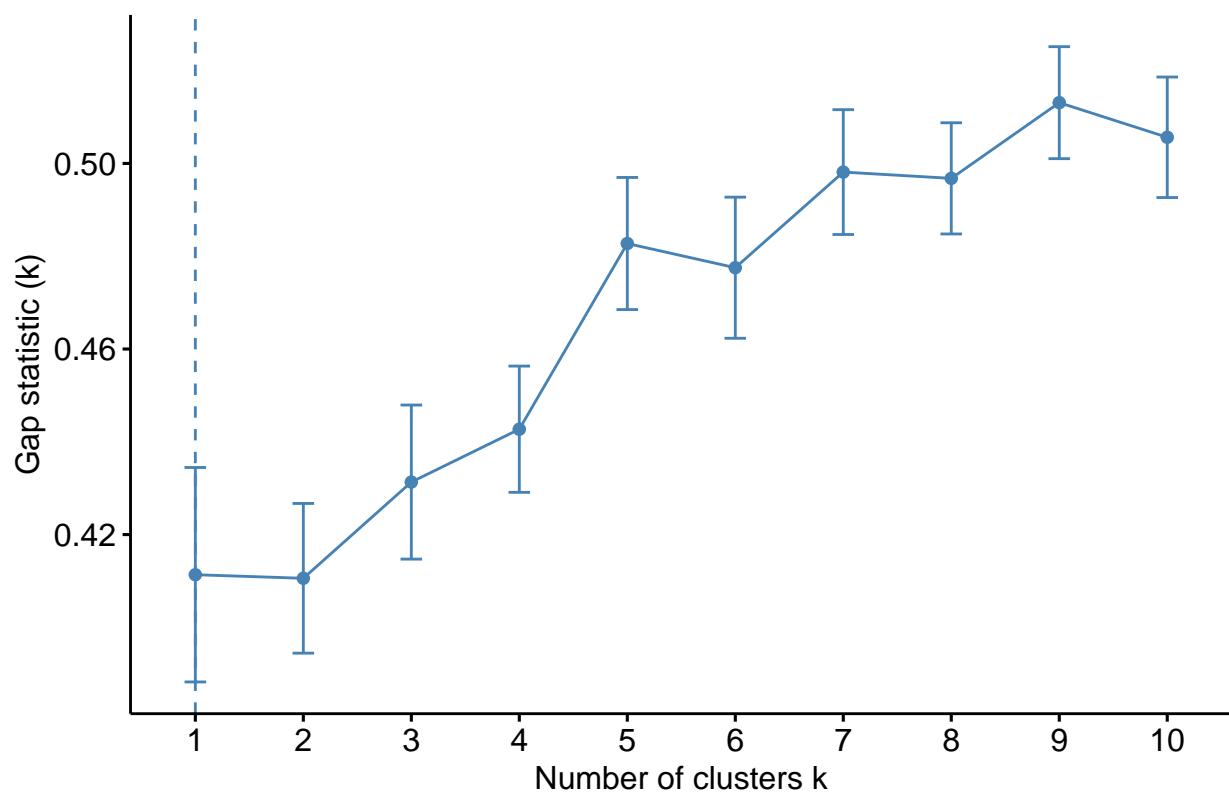


```
fviz_nbclust(diss_bray, hcut, method = "silhouette") # 2 seems optimal for hcut
```



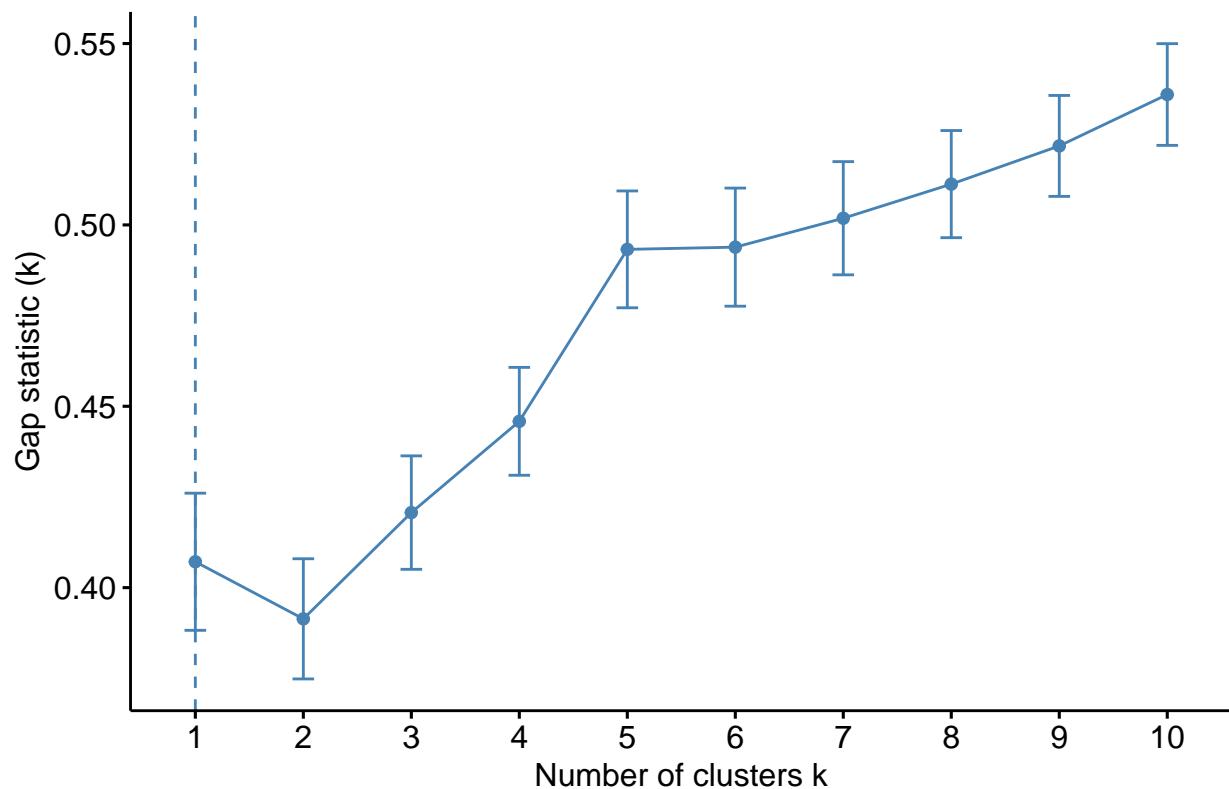
```
fviz_nbclust(diss_bray, kmeans, method = "gap_stat") # 1 seems optimal for k-means gap stat
```

Optimal number of clusters

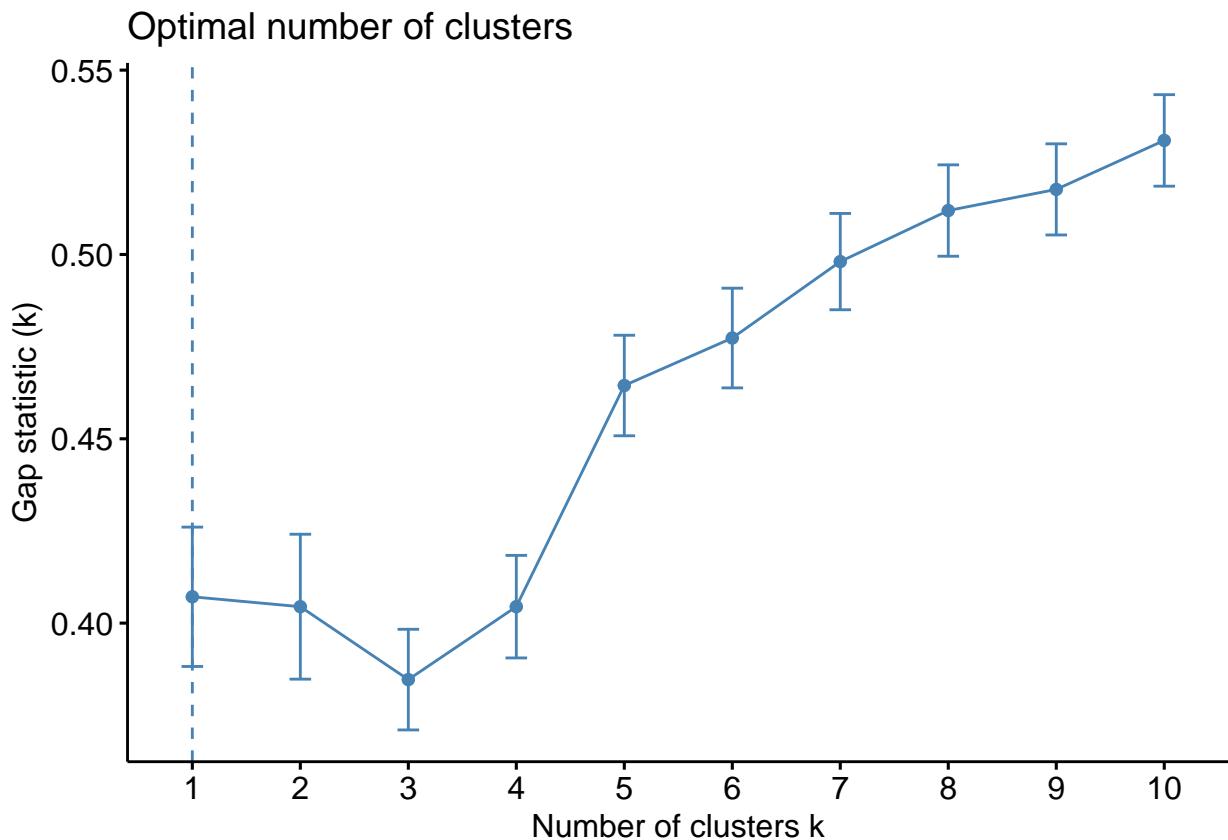


```
fviz_nbclust(diss_bray, cluster::pam, method = "gap_stat") # 1 seems optimal for PAM gap stat
```

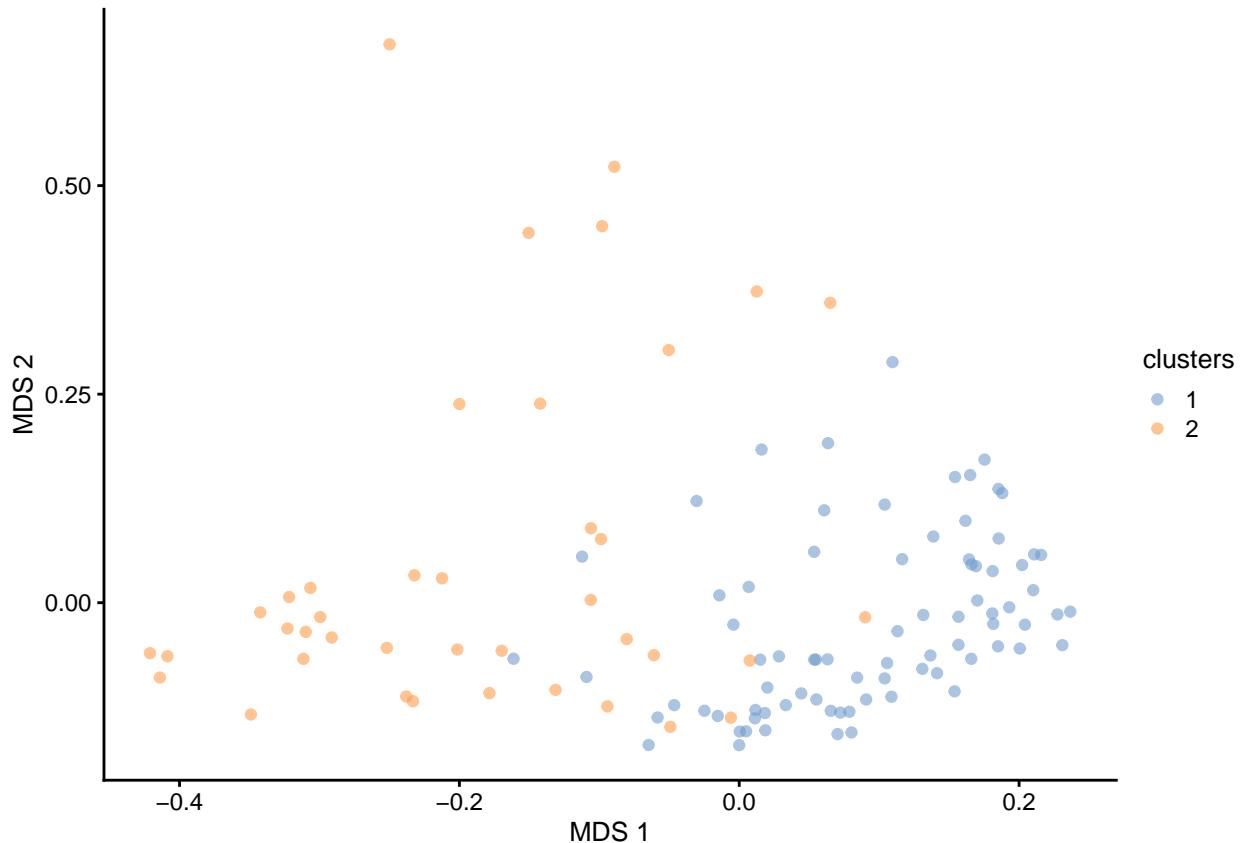
Optimal number of clusters



```
fviz_nbclust(diss_bray, hc, method = "gap_stat") # 1 seems optimal for hc cut gap stat
```



```
# k-means
set.seed(1337)
km <- kmeans(diss_bray, 2, nstart = 25)
colData(tse)$clusters <- as.factor(km$cluster)
tse <- runMDS(tse, assay.type = "relabundance", FUN = vegan::vegdist, method = "bray")
plotReducedDim(tse, "MDS", colour_by = "clusters")
```



```
# DMM (Laplace approximation) - ASV level (extremely time consuming so we will skip this for the rmark)
#tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)
#tse_dmn <- mia::runDMN(tse, name = "DMN", k = 1:7) # calculate most likely number of clusters from 1 to 7
#tse_dmn
#getDMN(tse_dmn)
#miaViz::plotDMNFit(tse_dmn, type = "laplace")
#getBestDMNFit(tse_dmn, type = "laplace") # 1 cluster seems optimal
```

genus level

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)
tse_genus <- agglomerateByRank(tse, rank = "Genus", agglomerateTree = TRUE)
tse_dmn <- mia::runDMN(tse_genus, name = "DMN", k = 1:7) # calculate most likely number of clusters from 1 to 7
tse_dmn
```

```
## class: TreeSummarizedExperiment
## dim: 1879 120
## metadata(2): agglomerated_by_rank DMN
## assays(1): counts
## rownames(1879): Genus:g__Bradyrhizobium Genus:g__Janthinobacterium ...
##   Genus:g__Amphibacillus Genus:g__Dehalobacterium
## rowData names(7): Domain Phylum ... Genus Species
## colnames(120): 10_10 10_11 ... 9_38 9_39
```

```

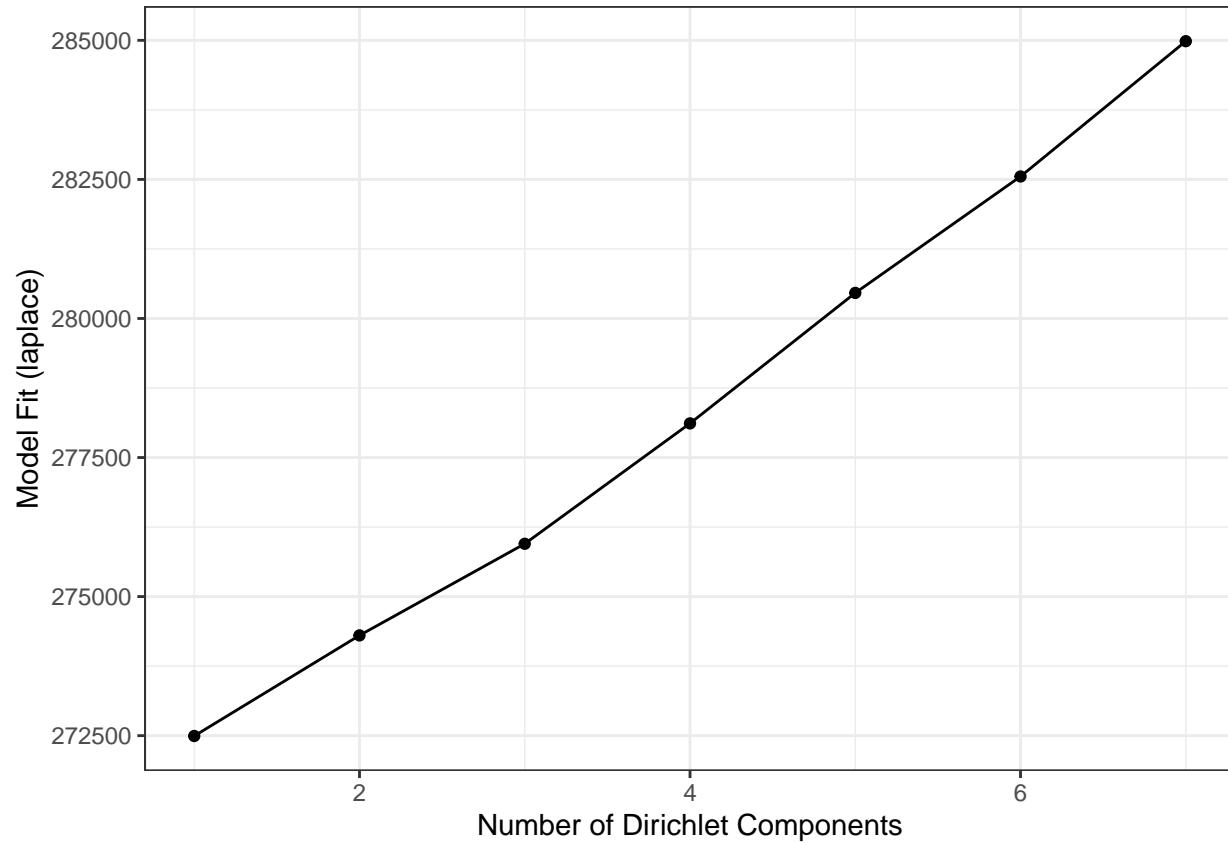
## colData names(35): Id SampleIdentifier ... Conc...ng..pl..1 Stables
## reducedDimNames(0):
## mainExpName: NULL
## altExpNames(0):
## rowLinks: a LinkDataFrame (1879 rows)
## rowTree: 1 phylo tree(s) (1879 leaves)
## colLinks: NULL
## colTree: NULL

getDMN(tse_dmn)

## [[1]]
## class: DMN
## k: 1
## samples x taxa: 120 x 1879
## Laplace: 272493.7 BIC: 276716.8 AIC: 274097.9
##
## [[2]]
## class: DMN
## k: 2
## samples x taxa: 120 x 1879
## Laplace: 274302.3 BIC: 284610.8 AIC: 279371.7
##
## [[3]]
## class: DMN
## k: 3
## samples x taxa: 120 x 1879
## Laplace: 275949.4 BIC: 292357.5 AIC: 284498.2
##
## [[4]]
## class: DMN
## k: 4
## samples x taxa: 120 x 1879
## Laplace: 278113.8 BIC: 300678.1 AIC: 290198.6
##
## [[5]]
## class: DMN
## k: 5
## samples x taxa: 120 x 1879
## Laplace: 280459.2 BIC: 309997.1 AIC: 296897.3
##
## [[6]]
## class: DMN
## k: 6
## samples x taxa: 120 x 1879
## Laplace: 282553 BIC: 318328.6 AIC: 302608.5
##
## [[7]]
## class: DMN
## k: 7
## samples x taxa: 120 x 1879
## Laplace: 284984.9 BIC: 327320.6 AIC: 308980.3

```

```
miaViz::plotDMNFit(tse_dmn, type = "laplace")
```



```
getBestDMNFit(tse_dmn, type = "laplace") # Gives 1 as best fit for genus level data
```

```
## class: DMN
## k: 1
## samples x taxa: 120 x 1879
## Laplace: 272493.7 BIC: 276716.8 AIC: 274097.9
```

```
# phylum level
```

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)
tse_phylum <- agglomerateByRank(tse, rank = "Phylum", agglomerateTree = TRUE)
tse_dmn <- mia::runDMN(tse_phylum, name = "DMN", k = 1:7) # calculate most likely number of clusters fr
tse_dmn
```

```
## class: TreeSummarizedExperiment
## dim: 43 120
## metadata(2): agglomerated_by_rank DMN
## assays(1): counts
## rownames(43): Phylum:p__Pseudomonadota Phylum:p__Bacteroidota ...
##   Phylum:p__Fibrobacterota Phylum:p__Balneolota
## rowData names(7): Domain Phylum ... Genus Species
## colnames(120): 10_10 10_11 ... 9_38 9_39
```

```

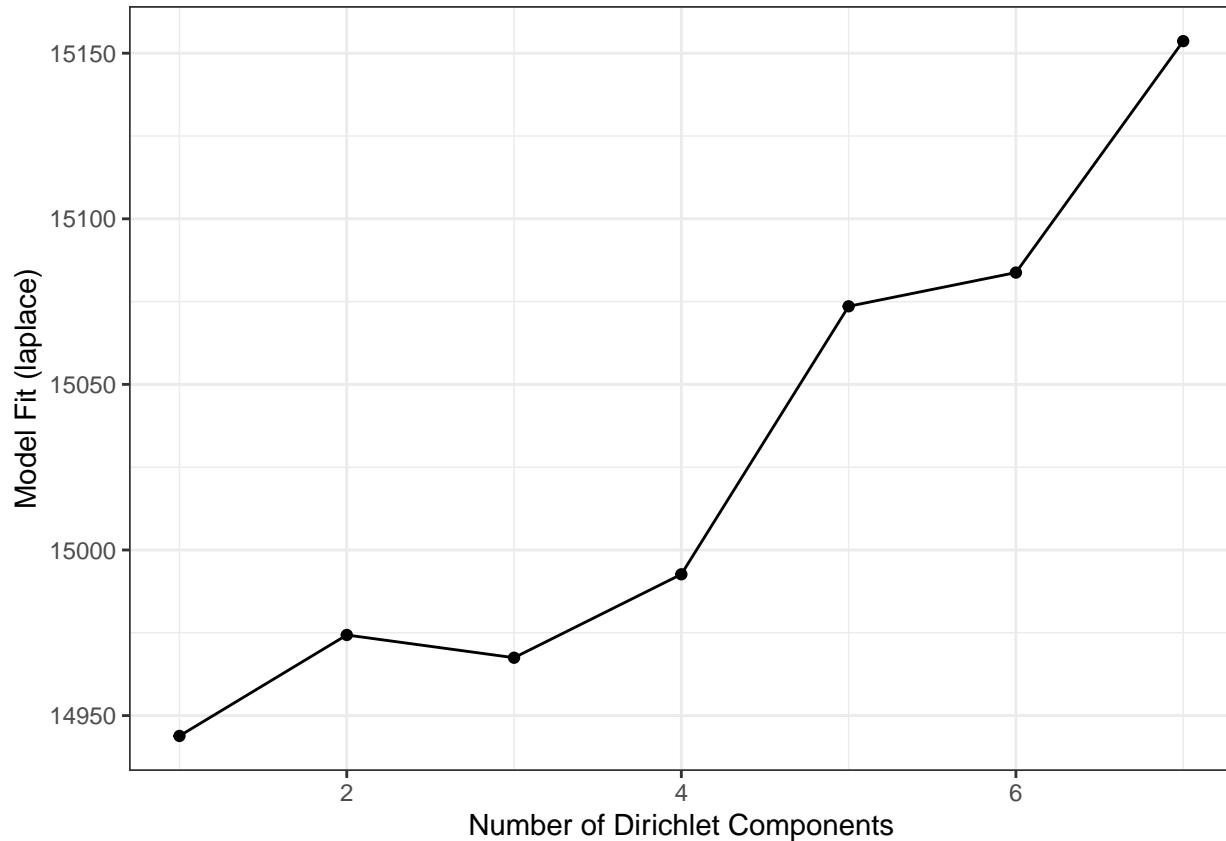
## colData names(35): Id SampleIdentifier ... Conc...ng..pl..1 Stables
## reducedDimNames(0):
## mainExpName: NULL
## altExpNames(0):
## rowLinks: a LinkDataFrame (43 rows)
## rowTree: 1 phylo tree(s) (43 leaves)
## colLinks: NULL
## colTree: NULL

getDMN(tse_dmn)

## [[1]]
## class: DMN
## k: 1
## samples x taxa: 120 x 43
## Laplace: 14943.86 BIC: 15017.08 AIC: 14957.15
##
## [[2]]
## class: DMN
## k: 2
## samples x taxa: 120 x 43
## Laplace: 14974.34 BIC: 15150.48 AIC: 15029.22
##
## [[3]]
## class: DMN
## k: 3
## samples x taxa: 120 x 43
## Laplace: 14967.47 BIC: 15256.23 AIC: 15073.65
##
## [[4]]
## class: DMN
## k: 4
## samples x taxa: 120 x 43
## Laplace: 14992.66 BIC: 15386.85 AIC: 15142.94
##
## [[5]]
## class: DMN
## k: 5
## samples x taxa: 120 x 43
## Laplace: 15073.58 BIC: 15589.73 AIC: 15284.5
##
## [[6]]
## class: DMN
## k: 6
## samples x taxa: 120 x 43
## Laplace: 15083.76 BIC: 15714.69 AIC: 15348.13
##
## [[7]]
## class: DMN
## k: 7
## samples x taxa: 120 x 43
## Laplace: 15153.65 BIC: 15906.3 AIC: 15478.42

```

```
miaViz::plotDMNFit(tse_dmn, type = "laplace")
```



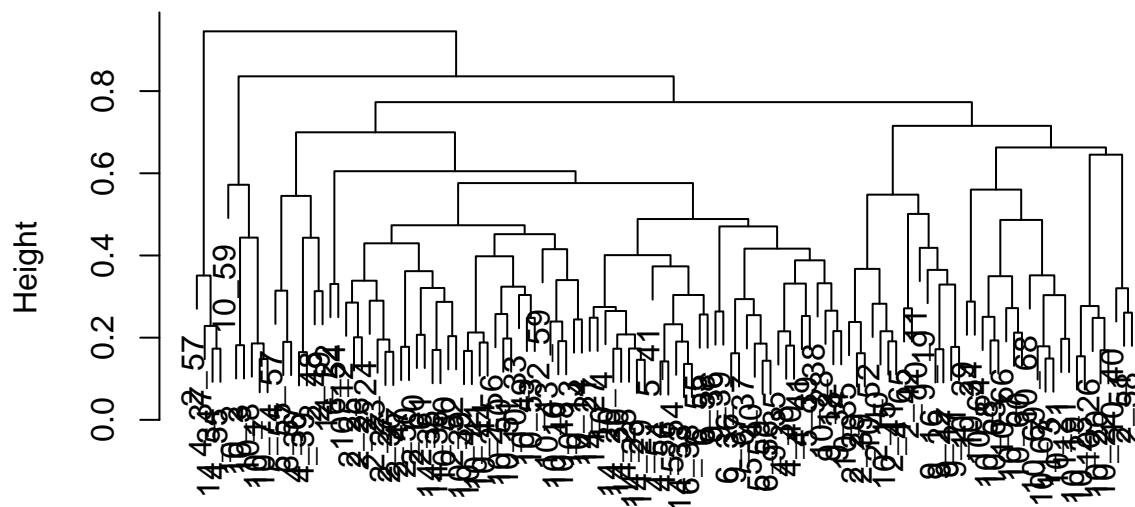
```
getBestDMNFit(tse_dmn, type = "laplace") # Gives 1 as best fit for phylum level data
```

```
## class: DMN
## k: 1
## samples x taxa: 120 x 43
## Laplace: 14943.86 BIC: 15017.08 AIC: 14957.15
```

Hierachal clustering BC asv

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)
tse <- transformCounts(tse, method = "relabundance")
tse <- runMDS(tse,
               assay.type = "relabundance",
               FUN = vegan::vegdist,
               method = "bray")
)
hc_bray <- hclust(vegdist(t(assay(tse, "relabundance"))), method = "bray"), method = "complete")
plot(hc_bray)
```

Cluster Dendrogram



```
vegdist(t(assay(tse, "relabundance")), method = "bray")
        hclust (*, "complete")
```

```

hcd = as.dendrogram(hc_bray)

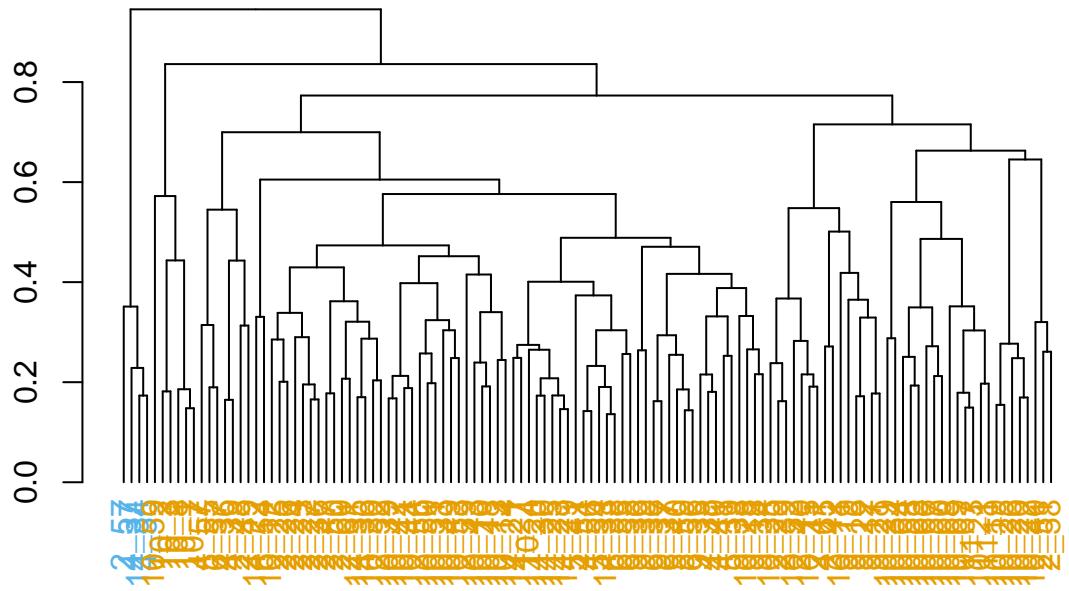
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442",
               "#0072B2", "#D55E00", "#CC79A7")

colorCode <- c(Control=cbPalette[2], CRC = cbPalette[3])

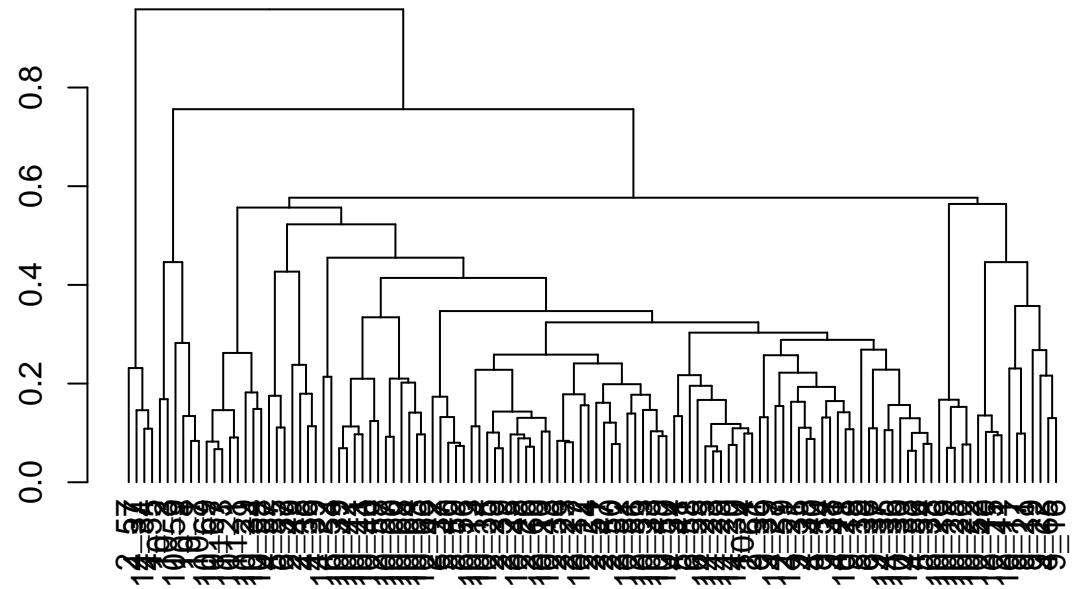
grouping = cutree(hc_bray, k = 2) # all methods gave 2 clusters, based on cuttree

labels_colors(hcd) <- colorCode[grouping][order.dendrogram(hcd)]
plot(hcd)

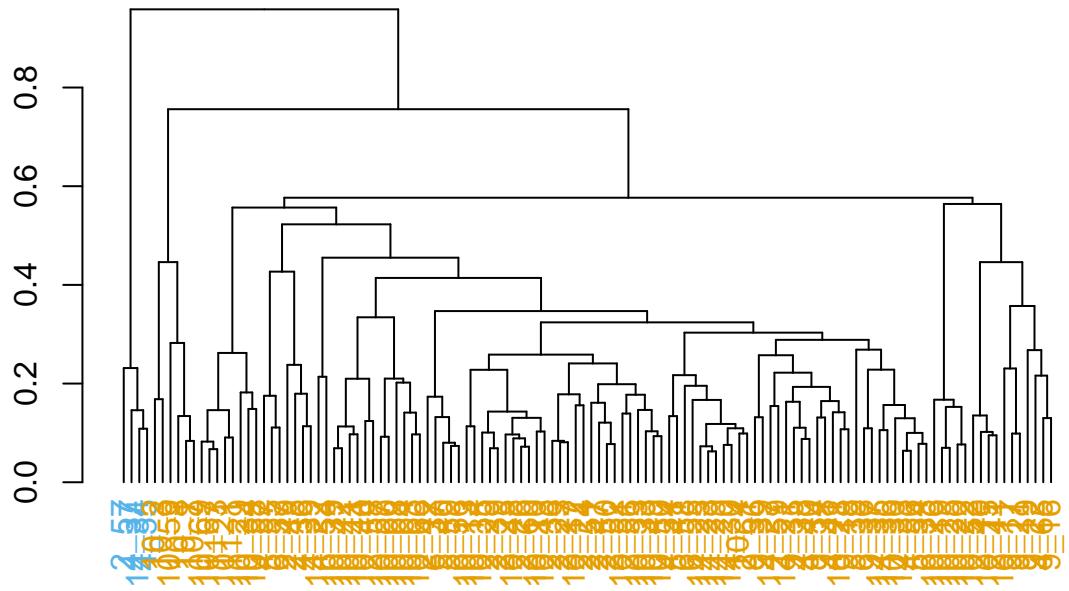
```



```
hclust.out <- clusterRows(assay, HclustParam(method = "complete"), full = TRUE) # cutting based on comp
colData(tse)$clusters <- hclust.out$clusters
dendro <- as.dendrogram(hclust.out$objects$hclust)
plot(dendro)
```



```
labels_colors(dendro) <- colorCode[grouping][order.dendrogram(dendro)]
plot(dendro)
```

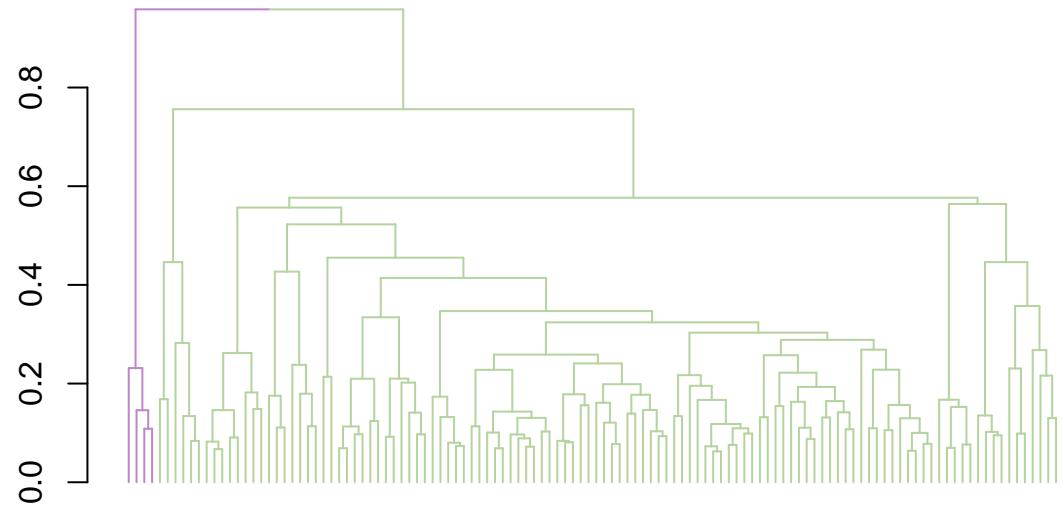


```

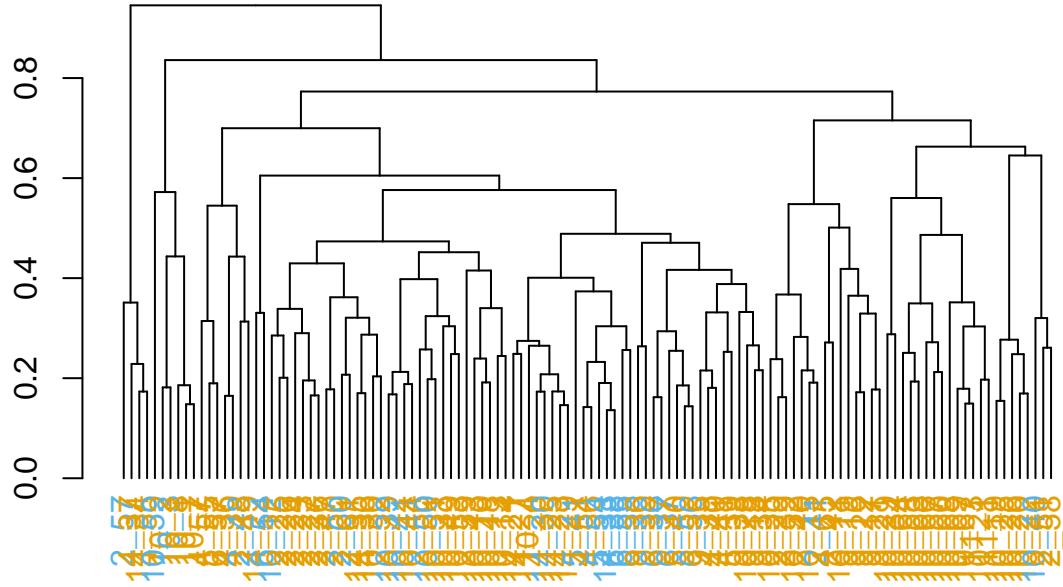
col_val_map <- randomcoloR::distinctColorPalette("2") %>%
  as.list() %>%
  setNames(paste0("clust_", seq("2")))

dend <- color_branches(dendro, k = 2, col = unlist(col_val_map))
labels(dend) <- NULL
plot(dend) # very similar to 16S scripts: based on two visualisations, only a few samples are clustered

```



```
# based on splitting at the root, which is not informative. this particular plot splits it more down the
labels_colors(hcd) <- colorCode[pam.out][order.dendrogram(hcd)]
plot(hcd)
```



```
tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)

tse <- transformCounts(tse, method = "relabundance")

pam.out <- clusterCells(tse,
                         assay.type = "relabundance",
                         BLUSPARAM = PamParam(centers = 2)
)

pam.out
```

```

## 10_10 10_11 10_12 10_13 10_14 10_15 10_19 10_1 10_20 10_21 10_22 10_25 10_26
## 1 1 1 1 1 1 2 1 2 2 2 2 2
## 10_28 10_29 10_2 10_30 10_33 10_34 10_35 10_39 10_3 10_40 10_41 10_42 10_43
## 1 2 1 1 1 2 1 1 1 1 1 1 1
## 10_44 10_48 10_49 10_4 10_50 10_51 10_52 10_53 10_57 10_58 10_59 10_60 10_63
## 1 1 1 1 1 1 1 1 1 2 1 1 2
## 10_64 10_66 10_67 10_68 10_69 10_7 10_8 11_1 11_3 14_20 14_21 14_22 14_23
## 2 2 2 2 2 1 1 2 2 1 1 1 1
## 14_25 14_27 14_29 14_30 14_33 14_34 14_35 14_36 2_23 2_24 2_25 2_26 2_27
## 1 1 1 1 1 1 1 1 1 1 1 1 1
## 2_29 2_36 4_65 2_39 2_40 2_41 2_42 2_47 2_48 2_49 2_50 2_51 2_52
## 1 1 1 1 1 1 1 1 1 1 1 1 1
## 2_56 2_57 2_58 2_59 2_60 2_61 4_36 4_37 4_38 4_39 4_40 4_41 4_54
## 1 1 1 1 1 1 1 1 1 1 1 1 1

```

```

## 4_55 4_56 4_57 5_39 5_40 5_41 5_54 5_55 5_59 6_36 6_37 6_38 6_54
##   1     1     1     1     1     1     1     1     1     1     1     1     1
## 6_55 6_56 6_57 6_58 9_16 9_17 9_18 9_19 9_21 9_22 9_34 9_35 9_36
##   1     1     1     1     1     1     1     1     1     1     1     1     1
## 9_37 9_38 9_39
##   1     1     1
## Levels: 1 2

n_iterations <- 1000
previous_cluster_assignment <- NULL
cluster_assignments <- list()

# loop that runs PAM clusterings X times and stores the results in a list, additionally checks if any changes occur
for (i in 1:n_iterations) {
  result <- clusterCells(tse, assay.type = "relabundance", BLUSPARAM = PamParam(centers = 2))
  cluster_assignments[[i]] <- result

  # Check if cluster assignments have changed
  if (!is.null(previous_cluster_assignment)) {
    samples_changed <- which(result != previous_cluster_assignment)
    if (length(samples_changed) > 0) {
      cat(sprintf("In iteration %d, the following samples changed clusters: %s\n", i, paste(samples_changed)))
    }
  }
  previous_cluster_assignment <- result
}

# To see if all of the clusters are the same or not
if (all(sapply(cluster_assignments, identical, cluster_assignments[[1]]))) {
  cat("All cluster assignments are the same across iterations.\n")
} else {
  cat("Cluster assignments vary across iterations.\n")
}

## All cluster assignments are the same across iterations.

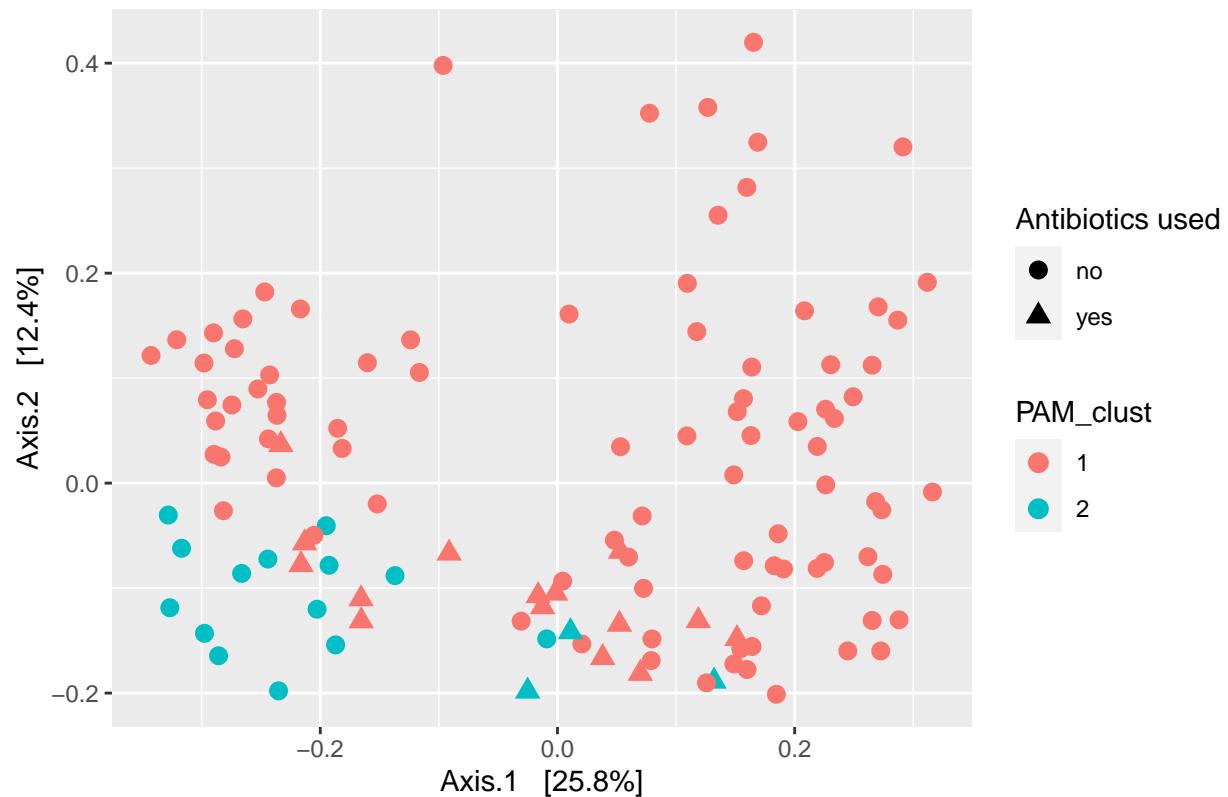
# There are no differences in clusters when run 1000 times

# save to metadata and make original PCoA plot
subsetMG@sam_data$PAM_clust = pam.out
sample_data(subsetMG)$PAM_clust = as.factor(sample_data(subsetMG)$PAM_clust)
pcoa_bc = ordinate(subsetMG, "PCoA", "bray")

plot_pcoa_ordination(subsetMG, pcoa_bc, "PAM_clust", "PCoA Bray Curtis")

```

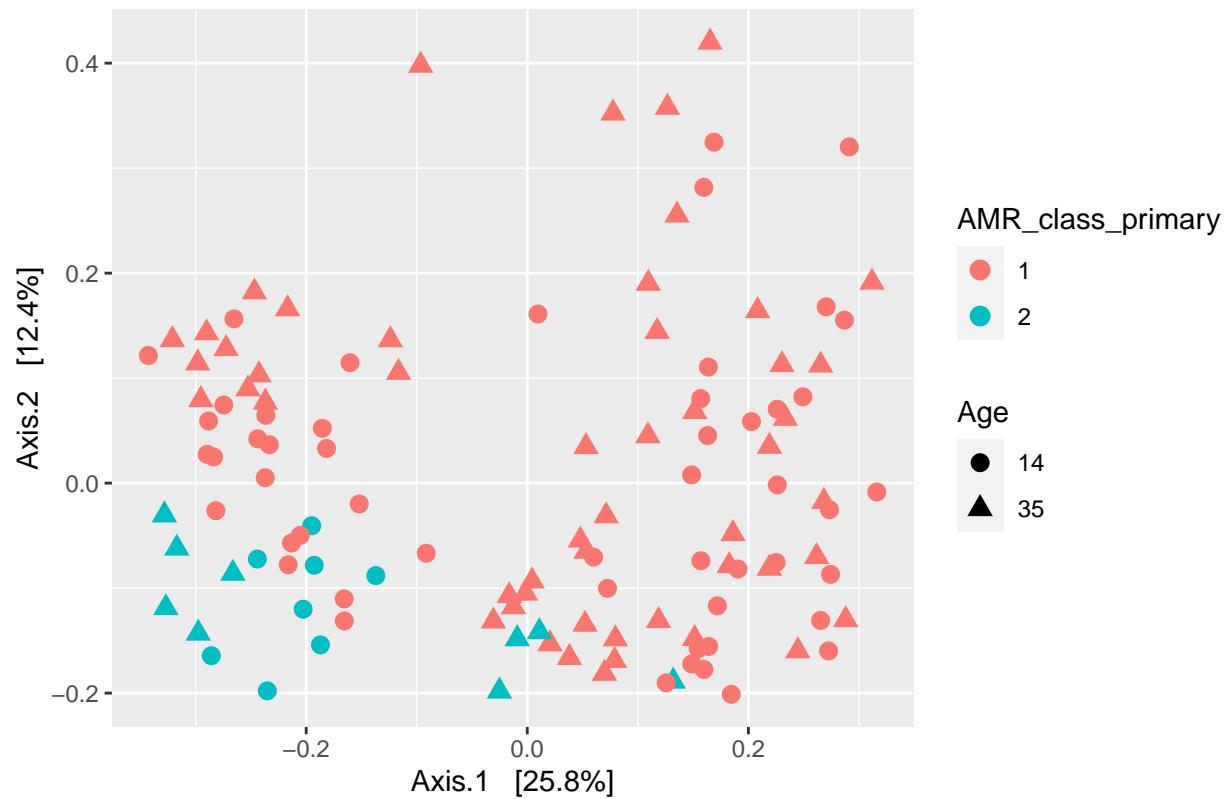
PCoA Bray Curtis



```
#plot_pcoa_ordination(subsetMG, pcoa_bc, "Cluster", "PCoA Bray Curtis")

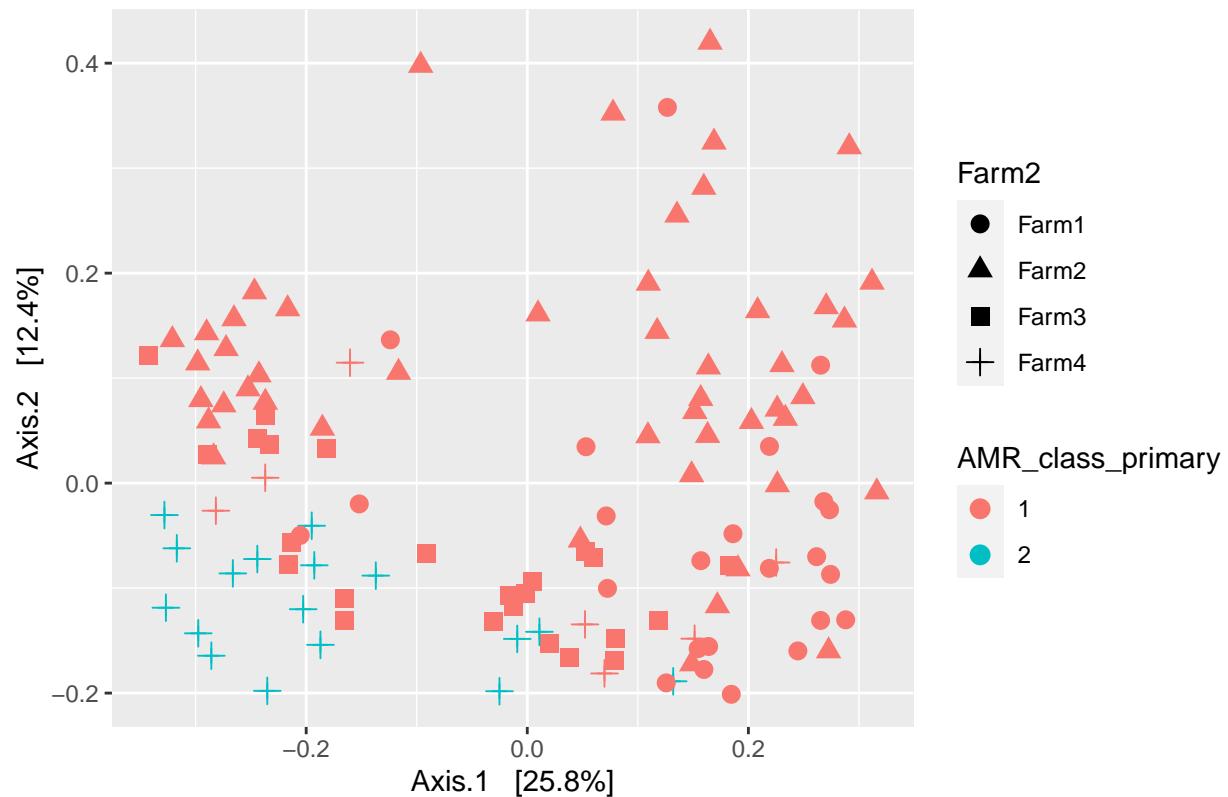
# change shape to different variables, age
plot_ordination(subsetMG, pcoa_bc, color = "PAM_clust", shape = "Age") +
  geom_point(size = 3) + labs(title = "PCoA Bray curtis", color = "AMR_class_primary")
```

PCoA Bray curtis



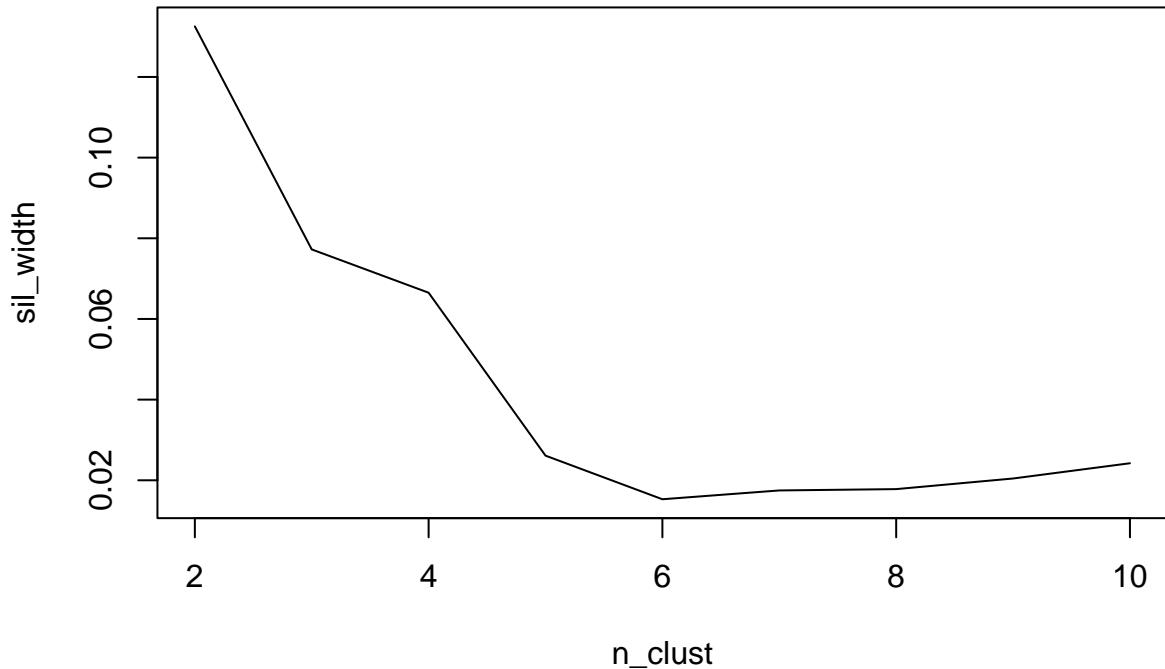
```
# change shape to different variables, farm
plot_ordination(subsetMG, pcoa_bc, color = "PAM_clust", shape = "Farm2") +
  geom_point(size = 3) + labs(title = "PCoA Bray curtis", color = "AMR_class_primary")
```

PCoA Bray curtis



```
# Create PAM PCoA - from 2 to 10 clusters
phy_rel <- transform_sample_counts(subsetMG, function(x) log10(x+1/sum(x+1)))
UF <- UniFrac(phy_rel, weighted = TRUE)
n_clust <- 2:10
pam_list <- lapply(n_clust, function(x) pam(UF, k = x))

sil_width <- lapply(pam_list, function(x) mean(x$silinfo$widths[, "sil_width"]))
plot(n_clust, sil_width, type="l")
```



```

pcoa_data <- cmdscale(UF, eig = TRUE)
pcoa_df <- data.frame(PC1 = c(pcoa_data$points[,1]),
                       PC2 = c(pcoa_data$points[,2]),
                       Sample = rownames(pcoa_data$points))

# Add sample data
Samp <- data.frame(sample_data(subsetMG))
Samp$Sample <- sample_names(subsetMG)

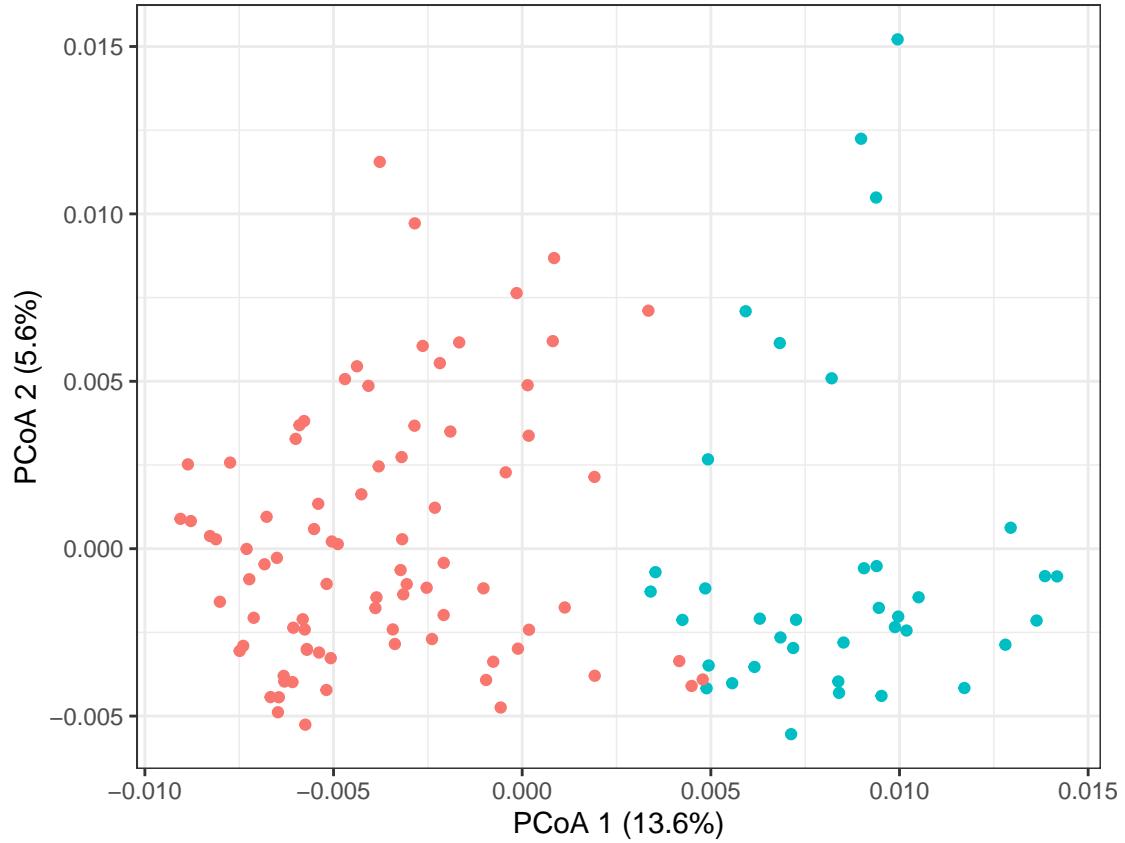
pcoa_df <- merge(pcoa_df, Samp, by = "Sample")

# Add cluster information
clusters <- factor(pam_list[[which.max(sil_width)]]$clustering)
pcoa_df <- merge(pcoa_df, clusters, by.x = "Sample", by.y = "row.names")
colnames(pcoa_df)[ncol(pcoa_df)] <- "PAM"

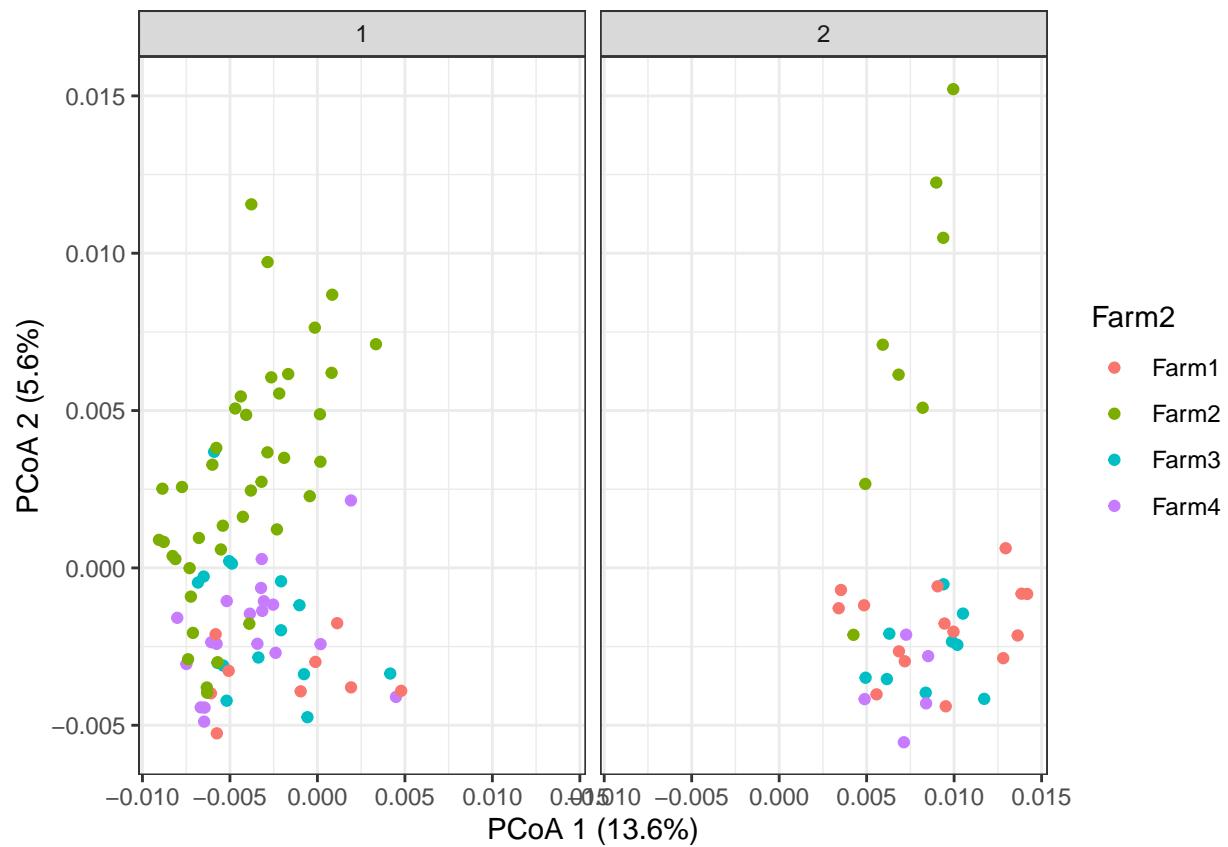
# Variance explained
ve <- pcoa_data$eig/sum(pcoa_data$eig)

# Plot
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = PAM)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)"))

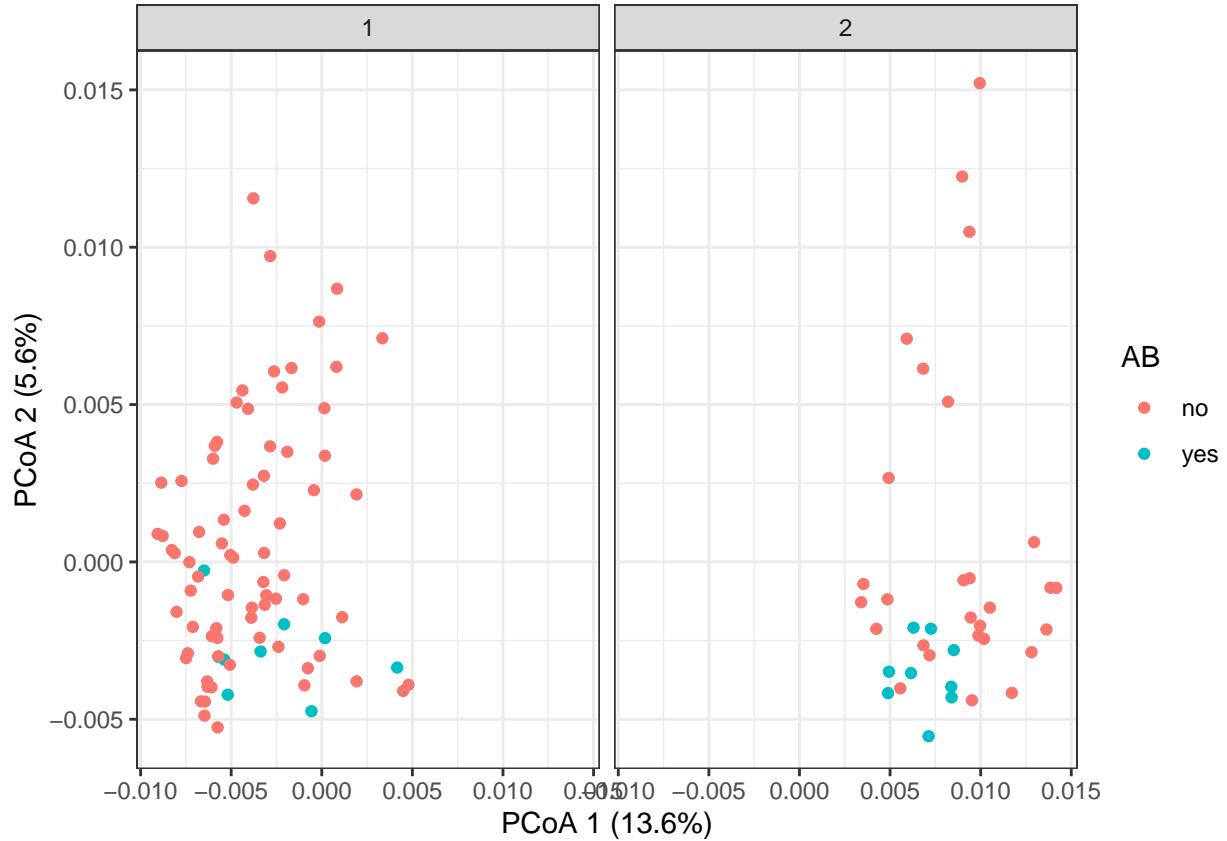
```



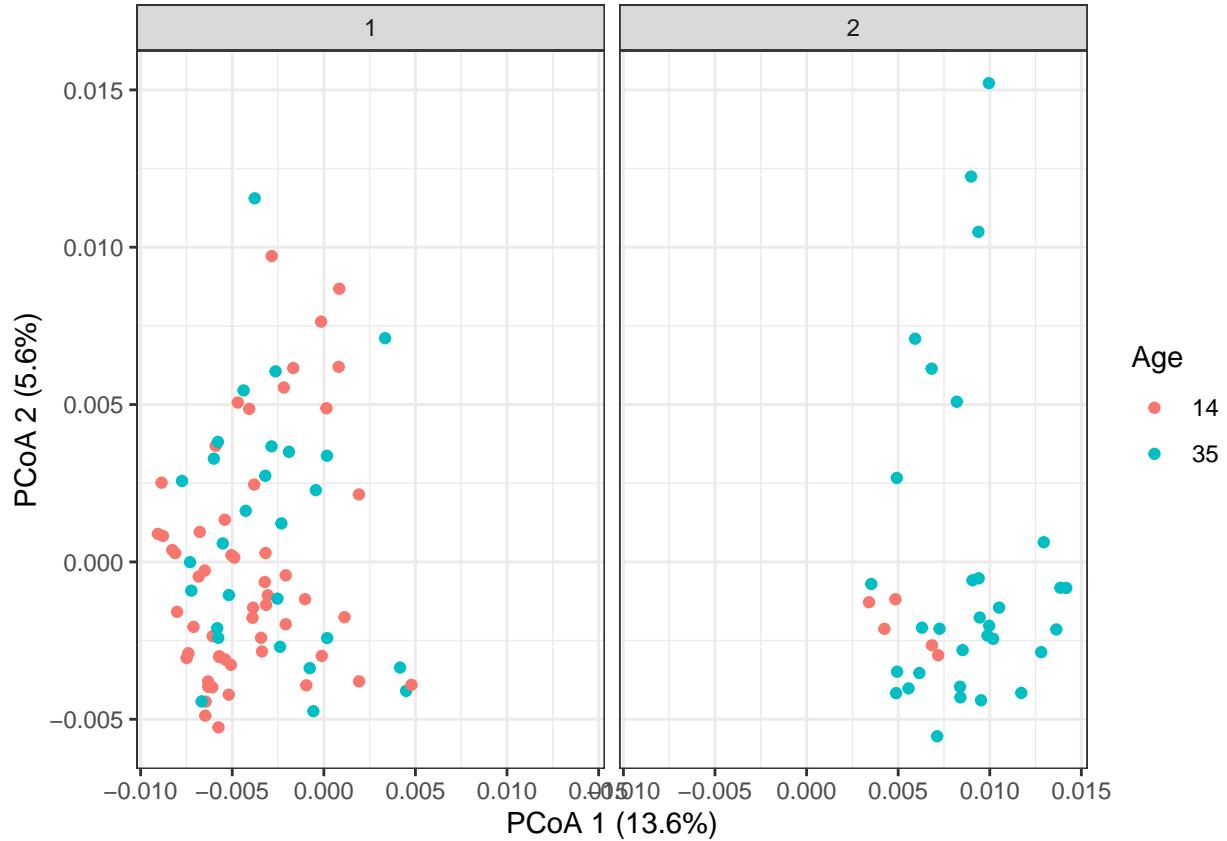
```
# facet by clusters and colour by farm
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = Farm2)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM)
```



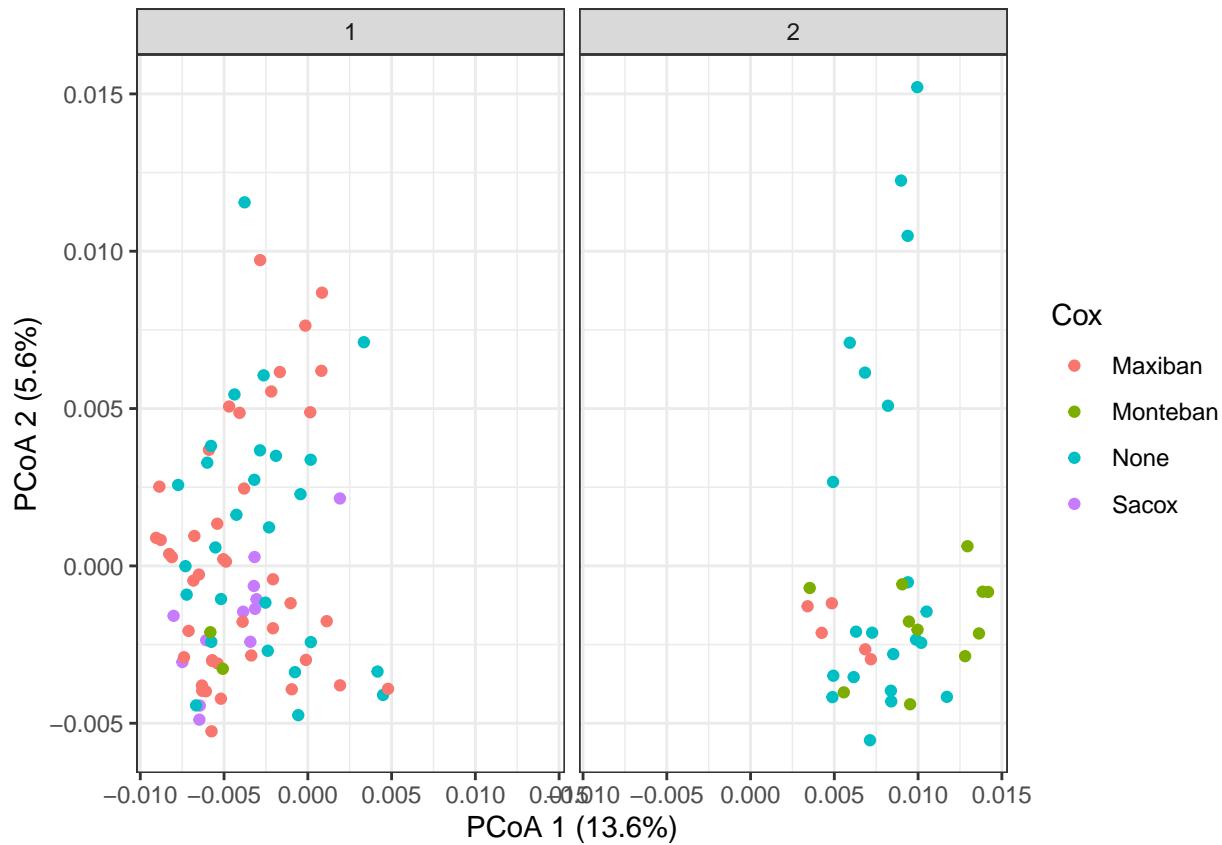
```
# facet by clusters and colour by AB
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = AB)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100, 1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100, 1), "%)")) +
  facet_wrap(~PAM)
```



```
# facet by clusters and colour by Age
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = Age)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM)
```



```
# facet by clusters and colour by Agent
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = Cox)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM)
```



```
# PCoA for Phylum level data, BC with DMM, euclidian ( can changetse_dmn to another taxonomic level)

dmn_group <- calculateDMNgroup(tse_dmn,
                                variable = "Age", assay.type = "counts",
                                k = 2, seed = .Machine$integer.max)

dmn_group <- calculateDMNgroup(tse_dmn,
                                variable = "Farm2", assay.type = "counts",
                                k = 2, seed = .Machine$integer.max)
dmn_group <- calculateDMNgroup(tse_dmn,
                                variable = "AB", assay.type = "counts",
                                k = 2, seed = .Machine$integer.max)
dmn_group

## class: DMNGroup
## summary:
##      k samples taxa      NLE      LogDet Laplace      BIC      AIC
## yes  2      18   43  2513.978  53.13916  2460.6  2639.709  2600.978
## no   2     102   43 12617.821 212.84500 12644.3 12819.007 12704.821

DirichletMultinomial::mixturewt(getBestDMNFit(tse_dmn)) # measure weights

##    pi      theta
## 1  1 11.29043
```

```

head(DirichletMultinomial::mixture(getBestDMNFit(tse_dmn))) # sample-cluster assignment probabilities

##          [,1]
## 10_10      1
## 10_11      1
## 10_12      1
## 10_13      1
## 10_14      1
## 10_15      1

head(DirichletMultinomial::fitted(getBestDMNFit(tse_dmn))) # taxa contribution

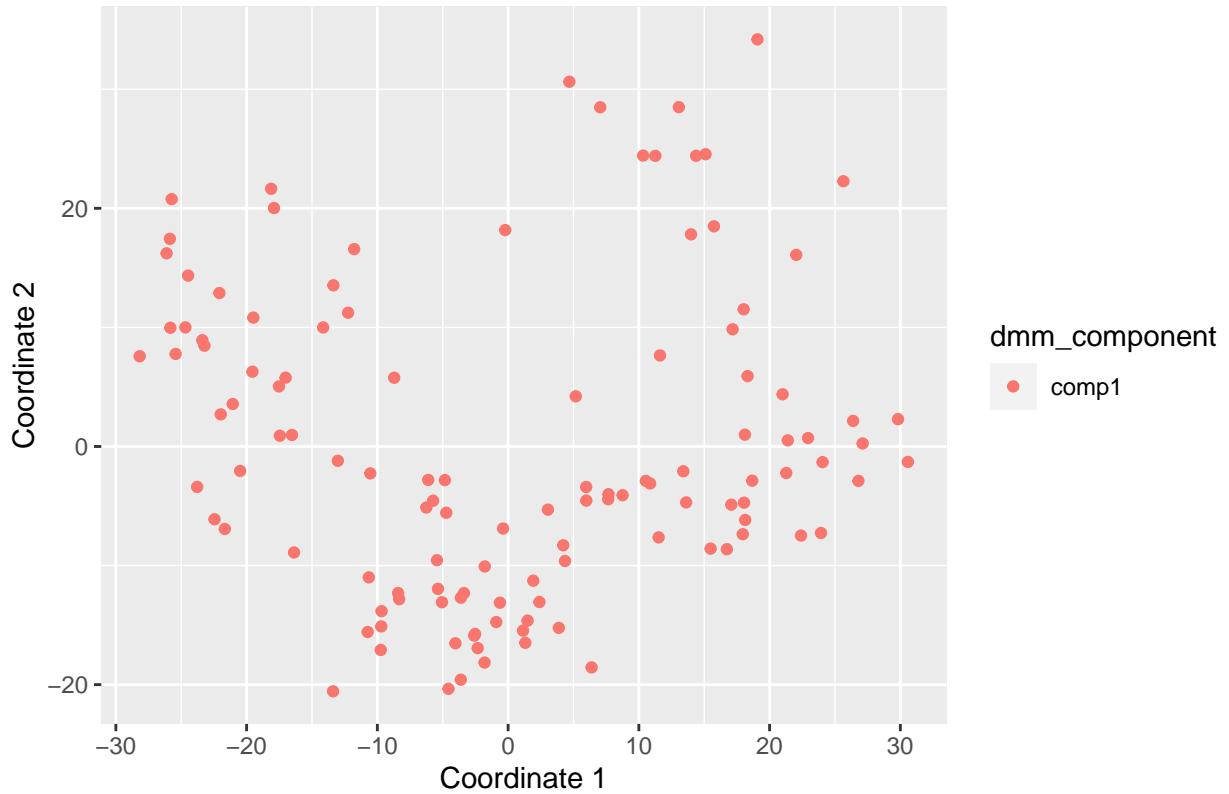
##          [,1]
## Phylum:p__Pseudomonadota 0.58218315
## Phylum:p__Bacteroidota   0.90665579
## Phylum:p__Bacillota       7.38970716
## Phylum:p__Cyanobacteriota 0.08617192
## Phylum:p__Actinomycetota  0.88611170
## Phylum:p__Mycoplasmatota 0.06512775

prob <- DirichletMultinomial::mixture(getBestDMNFit(tse_dmn))
colnames(prob) <- c("comp1")
vec <- colnames(prob)[max.col(prob, ties.method = "first")]
assay(tse, "pseudo") <- assay(tse, "counts") + 1
tse <- transformCounts(tse, assay.type = "pseudo", method = "relabundance")
tse <- transformCounts(tse, "relabundance", method = "clr")
df <- calculateMDS(tse, assay.type = "clr", method = "euclidean")
euclidean_pcoa_df <- data.frame(
  pcoa1 = df[, 1],
  pcoa2 = df[, 2])
euclidean_dmm_pcoa_df <- cbind(euclidean_pcoa_df,
                                   dmm_component = vec)

ggplot(
  data = euclidean_dmm_pcoa_df,
  aes(
    x = pcoa1, y = pcoa2,
    color = dmm_component
  )
) +
  geom_point() +
  labs(
    x = "Coordinate 1",
    y = "Coordinate 2",
    title = "PCoA with Aitchison distances")

```

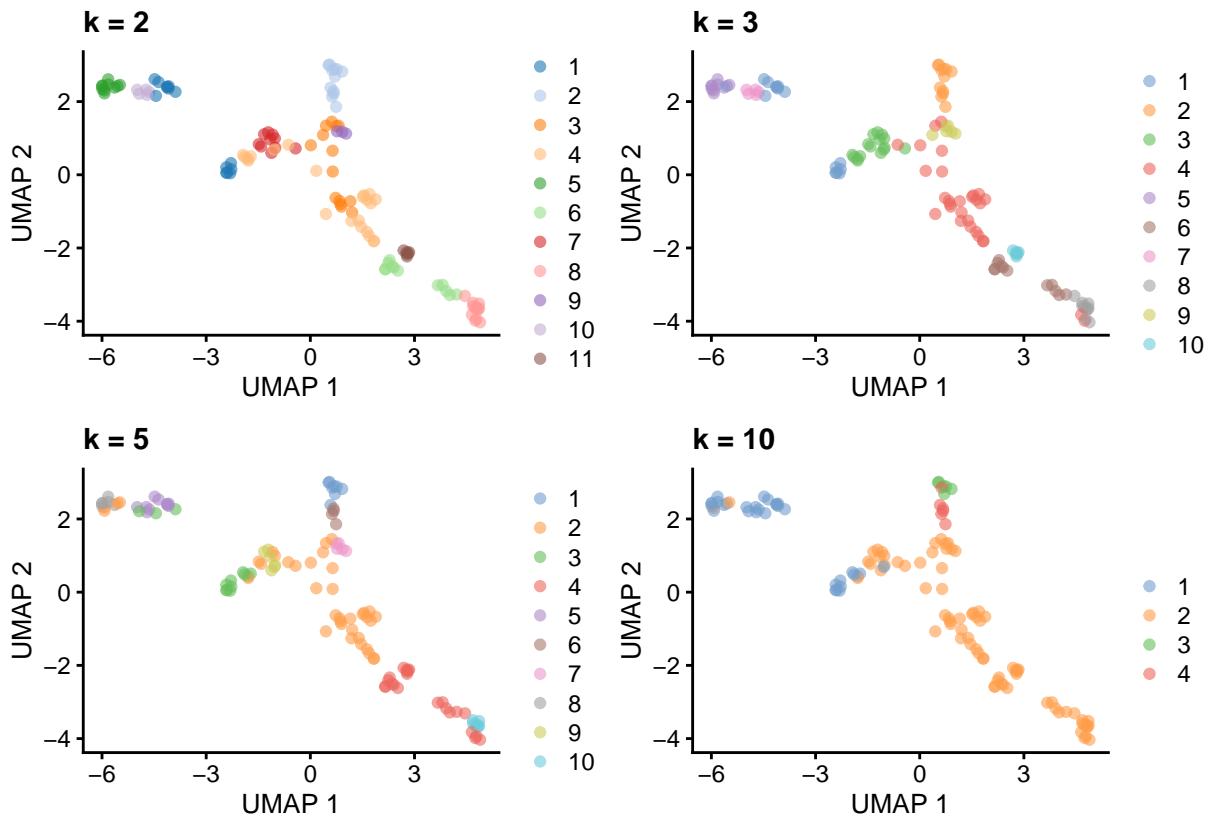
PCoA with Aitchison distances



UMAP with different ks

```
tse = makeTreeSummarizedExperimentFromPhyloseq(subsetMG)
tse <- transformCounts(tse, method = "rclr")
tse <- runUMAP(tse, name = "UMAP", assay.type = "rclr")
k <- c(2, 3, 5, 10)
ClustAndPlot <- function(x) {
  # Creating the graph and running the short random walks algorithm
  graph_clusters <- clusterRows(t(assays(tse)$rclr), NNGraphParam(k = x))

  # Results of the clustering as a color for each sample
  plotUMAP(tse, colour_by = I(graph_clusters)) +
    labs(title = paste0("k = ", x))
}
plots <- lapply(k, ClustAndPlot)
(plots[[1]] + plots[[2]]) / (plots[[3]] + plots[[4]])
```



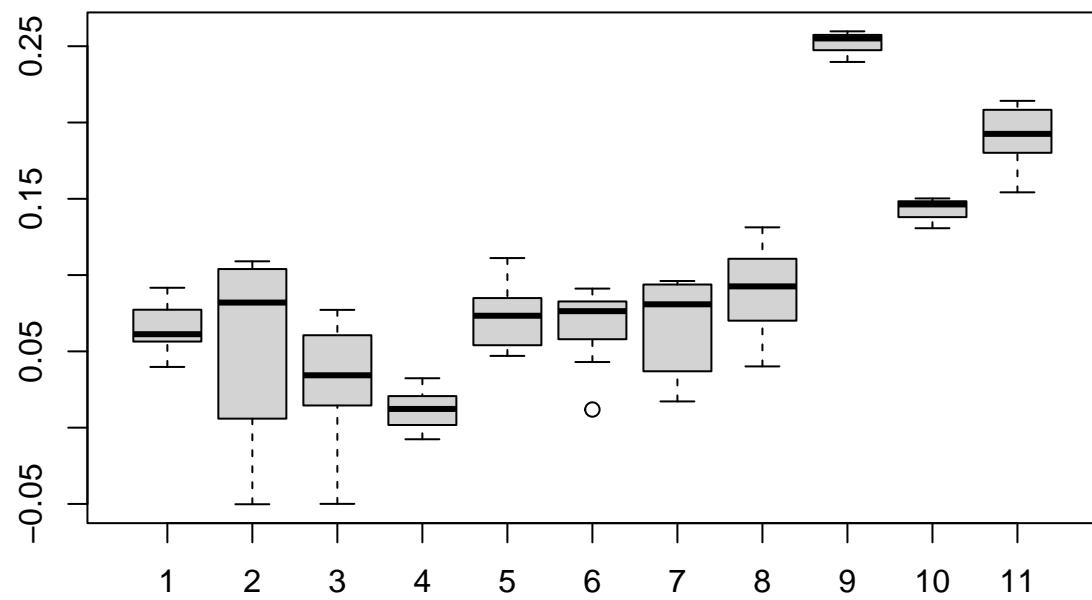
```
# boxplots
ClustDiagPlot <- function(x) {
  # Getting the clustering results
  graph_clusters <- clusterRows(t(assays(tse)$rclr), NNGraphParam(k = x))

  # Computing the diagnostic info
  sil <- approxSilhouette(t(assays(tse)$rclr), graph_clusters)

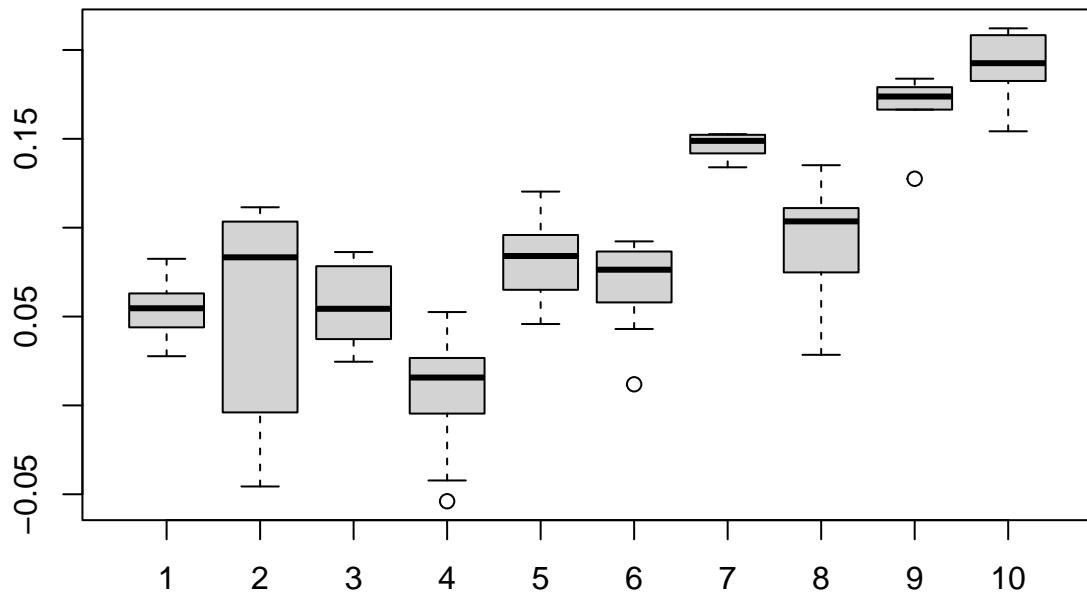
  # Plotting as a boxplot to observe cluster separation
  boxplot(split(sil$width, graph_clusters), main = paste0("k = ", x))
}

# Applying the function for different k values
res <- lapply(k, ClustDiagPlot)
```

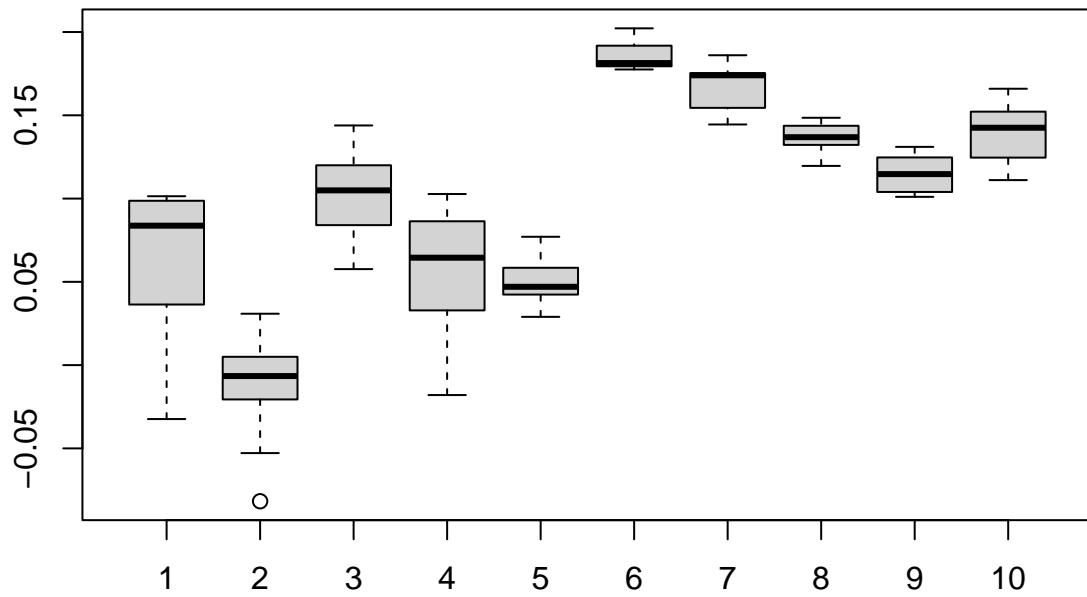
k = 2



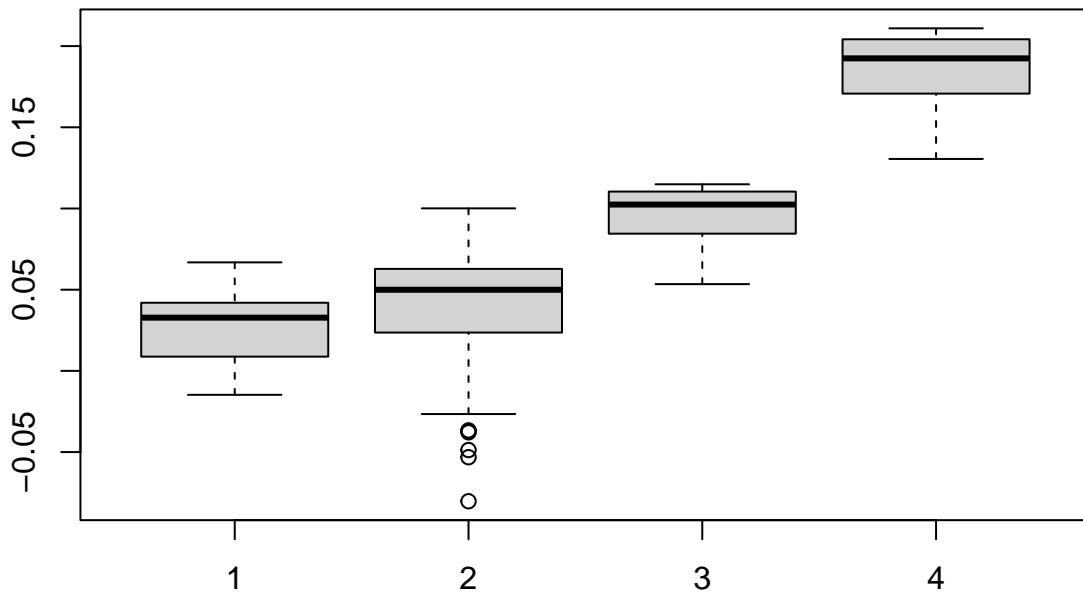
k = 3



k = 5



k = 10



Resistome data

loading data

```
### loading a subset of metagenomic data into phyloseq format
Rps= readRDS("Phyloseq_k2") # this reads a pre-existing phyloseq object containing OTU and tax tables, ...
Rps_mp= readRDS("Phyloseq") # reads in the data with count data corrected with metaphlan bacterial counts
Rps_tpm = readRDS("Phyloseq_tpm") # also read in TPM data instead of FPKM

#We rewrite the sample names to a format filtering out Firm and firm and the first underscore so that it matches the sample names in the metadata
sample_names(Rps) = sapply(regmatches(sample_names(Rps), regexpr("_", sample_names(Rps))), invert = TRUE)
sample_names(Rps_mp) = sapply(regmatches(sample_names(Rps_mp), regexpr("_", sample_names(Rps_mp))), invert = TRUE)
sample_names(Rps_tpm) = sapply(regmatches(sample_names(Rps_tpm), regexpr("_", sample_names(Rps_tpm))), invert = TRUE)

# Because the names in both metadata sets do not completely overlap, we need to manually edit one of them
sample_names(Rps)[68] = "4_65"
sample_names(Rps_mp)[68] = "4_65"
sample_names(Rps_tpm)[68] = "4_65"

# reading in and combining metadata from 16S and metagenomic origins, adding missing underscores
firm_names = read_excel("./Metagenomic/FIRM_MetaData.xlsx")
firm_names = firm_names[,-2] # Remove wrongful Raw_data_name column, to avoid confusion
```

```

meta_data = read.csv("MetaData.csv", header = TRUE, sep = ",")
meta_data_R = dplyr::right_join(firm_names, meta_data, by="SampleID")

# using Sample_Unique as rownames so we can match the two sets in phyloseq
rownames(meta_data_R) = meta_data_R$Sample_Unique

# now we'll also add in microbial load
microbial_load = read.table("bacterial_load_kraken2.tab", sep = "\t", header = TRUE)
microbial_load$Sample_Unique = sapply(regmatches(microbial_load$Sample_Unique, regexpr("_", microbial_load$Sample_Unique)), function(x) x[1])
microbial_load$Sample_Unique[68] = "4_65"
meta_data_R = dplyr::right_join(meta_data_R, microbial_load, by="Sample_Unique")

# creating tree and making phyloseq components, adding tree and sample data components to phyloseq
set.seed("877") # setting seed for reproducibility purposes
random_tree = rtree(ntaxa(Rps), rooted=TRUE, tip.label=taxa_names(Rps))
meta_data_R = sample_data(meta_data_R)
rownames(meta_data_R) = meta_data_R$Sample_Unique
Rps = merge_phyloseq(Rps, meta_data_R, random_tree)

# repeat for mp
set.seed("878") # setting seed for reproducibility purposes
random_tree2 = rtree(ntaxa(Rps_mp), rooted=TRUE, tip.label=taxa_names(Rps_mp))
Rps_mp = merge_phyloseq(Rps_mp, meta_data_R, random_tree2)

# repeat for tpm
set.seed("879") # setting seed for reproducibility purposes
random_tree3 = rtree(ntaxa(Rps_tpm), rooted=TRUE, tip.label=taxa_names(Rps_tpm))
Rps_tpm = merge_phyloseq(Rps_tpm, meta_data_R, random_tree3)

# overview data
datatable(tax_table(Rps))

```

```
rank_names(Rps) # Shows classes and ARGs
```

```
## [1] "AMR_class_primary"    "AMR_class_secondary" "ARGCluster90"
## [4] "ID_ClustRefSequence"
```

```
sort(get_taxa_unique(Rps, "AMR_class_primary")) # Shows primary AMR classes
```

```
## [1] "Aminoglycoside"   "Beta-lactam"      "Fosfomycin"       "Glycopeptide"
## [5] "Lincosamide"       "Macrolide"        "Not determined"   "Oxazolidinone"
## [9] "Phenicol"          "Quinolone"        "Streptogramin"    "Sulphonamide"
## [13] "Tetracycline"      "Trimethoprim"
```

```
sort(sample_sums(Rps)) # Amount of unique "taxa" per sample, the min is 1365.913 and max 44483.138, which is Rps
```

```
##      4_55      4_57      4_36      4_39      4_37      14_34      5_55      5_41
## 1365.915 1746.186 2162.499 2332.954 2466.228 2718.590 3066.586 3172.468
##      10_26     10_22     10_20     14_22     10_67     2_57      14_21      4_56
## 3673.885 3685.018 3729.530 3765.937 3823.575 3839.225 3978.278 4001.996
##      9_17      6_37     11_1     10_59     10_14     10_29     10_12     10_15
```

```

##   4093.433  4180.804  4345.176  4424.335  4498.027  4651.786  4689.540  4755.647
##   10_19     10_4      6_55      10_7      14_30     10_69     2_48      14_36
##  4869.370  4877.468  4903.775  4918.347  4936.114  5041.872  5072.382  5106.032
##   4_54      10_8      6_57      10_40     5_40      14_35     9_22      14_20
##  5129.624  5167.848  5190.200  5214.607  5270.470  5336.436  5395.266  5410.307
##   5_54      10_68     14_33     10_34     10_57     2_41      9_16      5_59
##  5581.891  5584.014  5595.311  5619.167  5671.777  5753.916  5758.922  5795.072
##   10_11     4_65      4_40      10_58     9_21      14_25     10_10     14_29
##  5816.317  5826.670  5838.197  5885.278  5977.274  6024.628  6044.767  6065.843
##   10_44     10_63     2_49      6_36      2_58      14_23     11_3      2_40
##  6105.070  6124.063  6206.665  6234.746  6254.150  6267.295  6268.648  6326.603
##   10_30     10_35     10_13     6_58      6_54      6_56      5_39      10_21
##  6375.700  6392.360  6431.761  6452.046  6506.618  6516.612  6522.346  6581.651
##   6_38      10_25     2_26      2_36     10_51     10_64     4_38      2_56
##  6643.052  6768.292  6862.483  6869.818  6924.205  6937.682  6956.992  6984.465
##   10_52     2_25      10_39     2_29     2_51      2_59      9_19      14_27
##  7060.573  7067.078  7091.217  7169.829  7383.892  7463.646  7528.052  7542.397
##   2_24      4_41      2_52      9_37     10_42     9_38      10_48     10_53
##  7543.974  7558.754  7639.814  7741.943  7759.352  7851.401  7890.425  7894.041
##   10_66     2_50      10_41     9_36     10_33     2_27      10_60     2_39
##  8025.357  8083.126  8104.861  8142.604  8192.787  8365.139  8412.984  8511.399
##   2_23      10_28     9_34      10_43     9_39      2_42      9_35      2_47
##  8613.113  8811.783  8881.043  8927.401  9038.993  9095.792  9217.621  9321.329
##   2_61      10_49     9_18      2_60     10_50     10_2      10_3      10_1
##  9503.493  9672.041  9749.846  9753.045  10461.847  23616.052  27684.887  44483.138

```

```
summary(sample_sums(Rps)) # summary of the sampling depths
```

```

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##    1366    5065   6244    6827    7579   44483

```

```
summary(sample_sums(Rps_mp)) # there are big differences between kraken2 and metaphlan counts data, with
```

```

##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
##   125.5  6816.4 11282.9 16409.8 26186.9 93167.6

```

```
sample_variables(Rps) # metadata variables
```

```

## [1] "SampleIdentifier" "ResCap"          "Conc...ng..pl."   "SampleID"
## [5] "LibraryNumber"    "Sample_Unique"    "LibraryName"     "Farm"
## [9] "Farm2"            "Stable"          "FarmRoundStable" "Days"
## [13] "Age"              "Sname"           "WeightAnimal"   "Gender"
## [17] "AgeParentStock"  "Hatchery"        "Researcher"     "AB"
## [21] "Abday"            "FlockSize"       "FeedF"          "FeedType"
## [25] "FeedProducent"   "Cox"             "OPG"            "Cluster"
## [29] "LitterType"       "Metagenomics"    "ReadPerc"       "ReadTot"

```

```
#taxa_names(Rps) # ARGs (662) (will not print in rmarkdown)
```

```
# Stable "Farm2R1S1" has the five lowest sampling depths of the dataset, some other samples are also v
Rps %>% ps_filter(FarmRoundStable == c("Farm2R1S1")) %>% sample_sums() %>% sort()
```

```

##      4_55      4_57      4_36      4_39      4_37      5_55      4_56      4_54
## 1365.915 1746.186 2162.499 2332.954 2466.228 3066.586 4001.996 5129.624
##      5_54      4_40      4_38      4_41
## 5581.891 5838.197 6956.992 7558.754

# For metaphlan data, Stable "Farm2R1S1 has 11/12 lowest sampling depths of the dataset, and there are
Rps_mp %>% ps_filter(FarmRoundStable == c("Farm2R1S1")) %>% sample_sums() %>% sort()

##      4_57      4_55      4_36      4_39      4_56      4_37      4_54      5_55
## 125.4769 190.1299 198.9479 211.1739 305.2175 345.8083 394.6092 495.9399
##      5_54      4_40      4_38      4_41
## 521.6233 572.6645 635.2437 822.9736

# factorizing variables as not to create problems with visualisation later down the line
sample_data(Rps)$Cluster = as.factor(sample_data(Rps)$Cluster)
sample_data(Rps)$FlockSize = as.factor(sample_data(Rps)$FlockSize)
sample_data(Rps)$AgeParentStock = as.factor(sample_data(Rps)$AgeParentStock)
sample_data(Rps)$Age = as.factor(sample_data(Rps)$Age)
sample_data(Rps)$LibraryNumber = as.factor(sample_data(Rps)$LibraryNumber)

# repeat for MP
sample_data(Rps_mp)$Cluster = as.factor(sample_data(Rps_mp)$Cluster)
sample_data(Rps_mp)$FlockSize = as.factor(sample_data(Rps_mp)$FlockSize)
sample_data(Rps_mp)$AgeParentStock = as.factor(sample_data(Rps_mp)$AgeParentStock)
sample_data(Rps_mp)$Age = as.factor(sample_data(Rps_mp)$Age)
sample_data(Rps_mp)$LibraryNumber = as.factor(sample_data(Rps_mp)$LibraryNumber)

# repeat for TPM
sample_data(Rps_tpm)$Cluster = as.factor(sample_data(Rps_tpm)$Cluster)
sample_data(Rps_tpm)$FlockSize = as.factor(sample_data(Rps_tpm)$FlockSize)
sample_data(Rps_tpm)$AgeParentStock = as.factor(sample_data(Rps_tpm)$AgeParentStock)
sample_data(Rps_tpm)$Age = as.factor(sample_data(Rps_tpm)$Age)
sample_data(Rps_tpm)$LibraryNumber = as.factor(sample_data(Rps_tpm)$LibraryNumber)

# add stable column with shorter names
sample_data(Rps)$FarmRoundStable = as.factor(sample_data(Rps)$FarmRoundStable)
Rps@sam_data$Stables = revalue(sample_data(Rps)$FarmRoundStable, c("Farm1R1S1"="Stable1", "Farm1R1S2"="Stable2",
"Farm2R1S1"="Stable5", "Farm2R1S2"="Stable6", "Farm2R2S1"="Stable7", "Farm2R2S2"="Stable8",
"Farm4R1S1"="Stable9", "Farm4R1S2"="Stable10"))

# Shortening agent names
Rps@sam_data$Cox[Rps@sam_data$Cox == "narasinandnicarbazin(maxiban)"] = "Maxiban"
Rps@sam_data$Cox[Rps@sam_data$Cox == "narasin(monteban)"] = "Monteban"
Rps@sam_data$Cox[Rps@sam_data$Cox == "salinomycin(Sacox120microGranulate)"] = "Sacox"

# repeat
sample_data(Rps_mp)$FarmRoundStable = as.factor(sample_data(Rps_mp)$FarmRoundStable)
Rps_mp@sam_data$Stables = revalue(sample_data(Rps_mp)$FarmRoundStable, c("Farm1R1S1"="Stable1", "Farm1R1S2"="Stable2",
"Farm2R1S1"="Stable5", "Farm2R1S2"="Stable6", "Farm2R2S1"="Stable7", "Farm2R2S2"="Stable8",
"Farm4R1S1"="Stable9", "Farm4R1S2"="Stable10"))

Rps_mp@sam_data$Cox[Rps_mp@sam_data$Cox == "narasinandnicarbazin(maxiban)"] = "Maxiban"
Rps_mp@sam_data$Cox[Rps_mp@sam_data$Cox == "narasin(monteban)"] = "Monteban"
Rps_mp@sam_data$Cox[Rps_mp@sam_data$Cox == "salinomycin(Sacox120microGranulate)"] = "Sacox"

```

```

sample_data(Rps_tpm)$FarmRoundStable = as.factor(sample_data(Rps_tpm)$FarmRoundStable)
Rps_tpm@sam_data$Stables = revalue(sample_data(Rps_tpm)$FarmRoundStable, c("Farm1R1S1"="Stable1", "Farm2R1S1"="Stable3", "Farm2R2S1"="Stable5", "Farm2R3S1"="Stable7", "Farm3R1S1"="Stable9", "Farm3R2S1"="Stable11", "Farm4R1S1"="Stable13", "Farm4R2S1"="Stable15"))
Rps_tpm@sam_data$Cox[Rps_tpm@sam_data$Cox == "narasinandnicarbazin(maxiban)"] = "Maxiban"
Rps_tpm@sam_data$Cox[Rps_tpm@sam_data$Cox == "narasin(monteban)"] = "Monteban"
Rps_tpm@sam_data$Cox[Rps_tpm@sam_data$Cox == "salinomycin(Sacox120microGranulate)"] = "Sacox"

# In order to create taxa prevalence plots, and many more functions, we need to change our "taxa" level.
# We'll make a copy to fulfill these purposes
# Phylum = AMR_class_primary, Class = AMR_class_secondary, Order = ARGCluster90, Family = ID_Clust_Refseq
Rps_copy = Rps
colnames(Rps_copy@tax_table) = c("Phylum", "Class", "Order", "Family")
Rps_mp_copy = Rps_mp
colnames(Rps_mp_copy@tax_table) = c("Phylum", "Class", "Order", "Family")

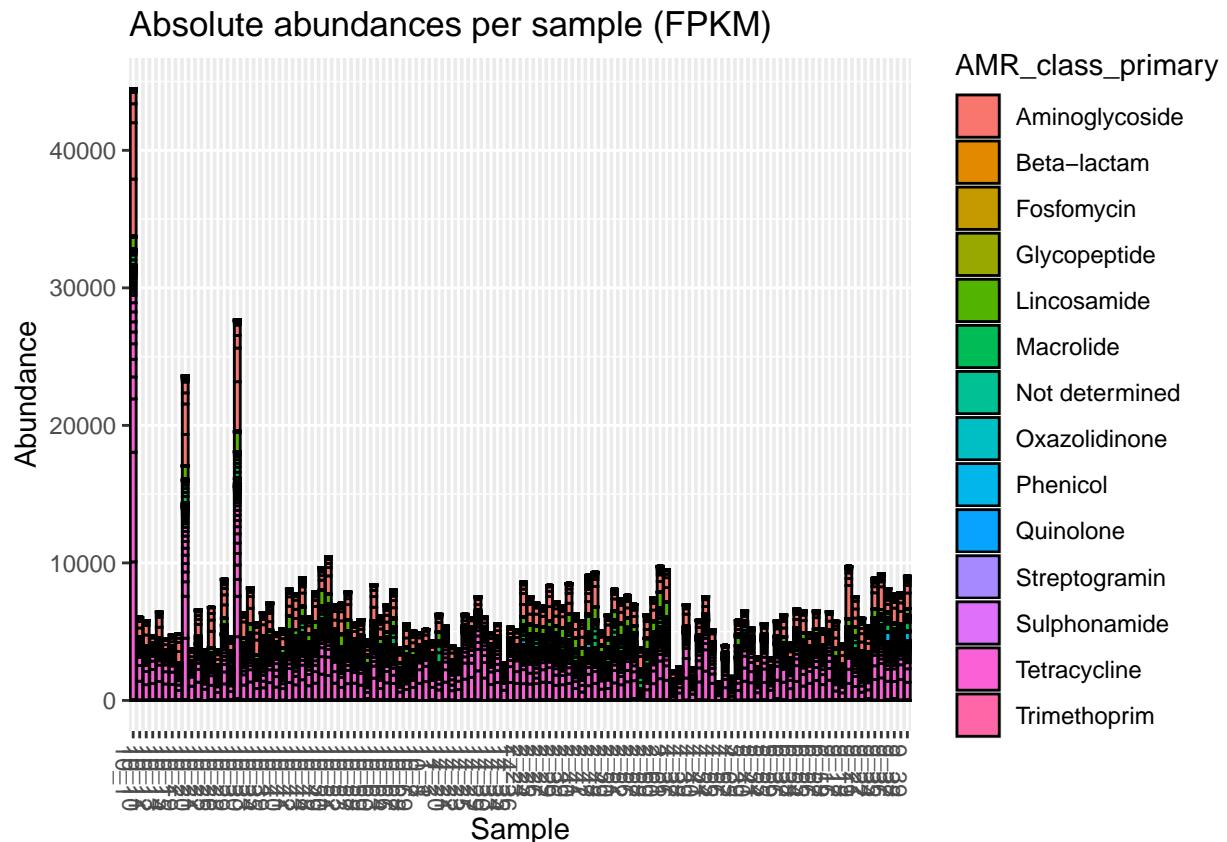
```

abundances and heatmaps

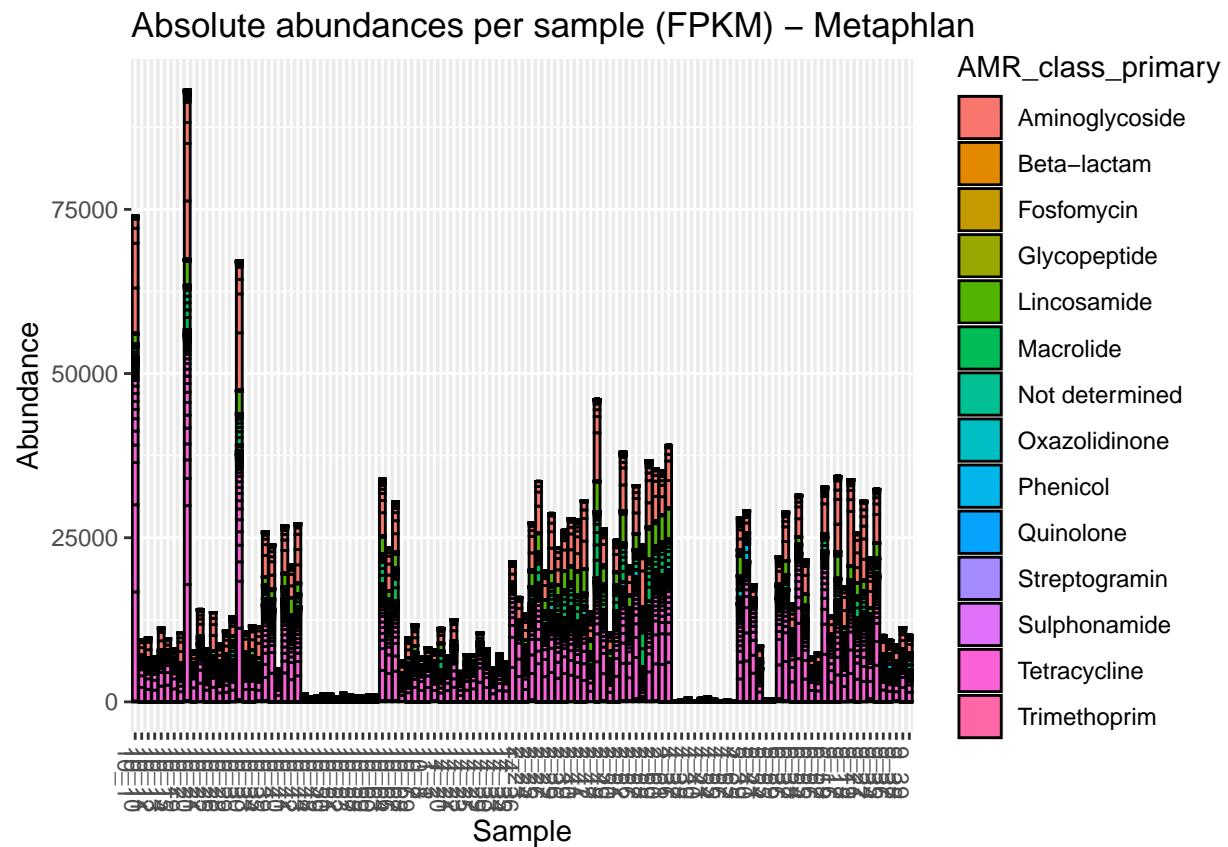
```

# absolute abundances
plot_bar(Rps, fill="AMR_class_primary", title = "Absolute abundances per sample (FPKM)")

```

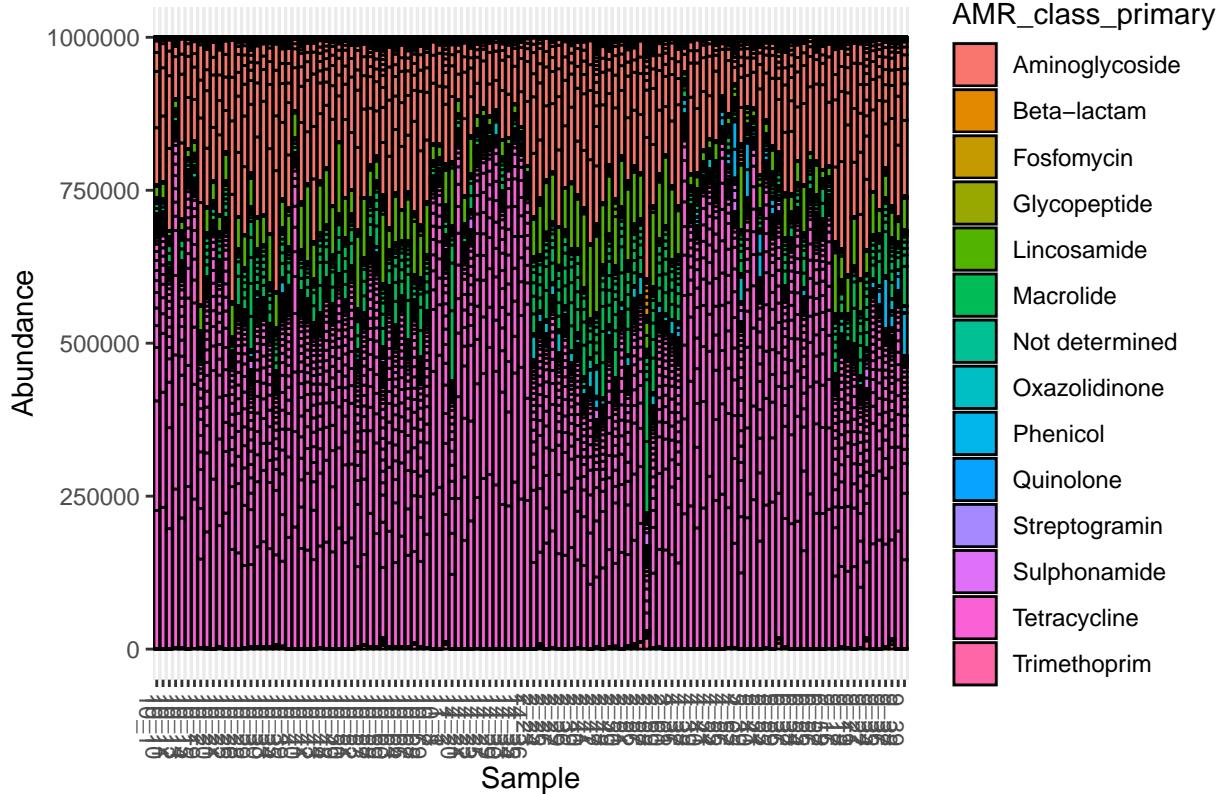


```
plot_bar(Rps_mp, fill="AMR_class_primary", title = "Absolute abundances per sample (FPKM) - Metaphlan")
```



```
plot_bar(Rps_tpm, fill="AMR_class_primary", title = "Absolute abundances per sample (TPM)")
```

Absolute abundances per sample (TPM)



```
# on average, for the three outliers, abundance is
mean(sample_sums(Rps)[c("10_1", "10_2", "10_3")]) # 31928.03
```

```
## [1] 31928.03
```

```
# on average, the samples without the three outliers, abundance is
mean(sample_sums(Rps)[!sample_names(Rps) %in% c("10_1", "10_2", "10_3")]) # 6183.18
```

```
## [1] 6183.18
```

```
# so there are 5,1636x as much abundance in these samples
```

```
# MP:
# on average, for the three outliers, abundance is
mean(sample_sums(Rps_mp)[c("10_1", "10_2", "10_3")]) # 78089.08
```

```
## [1] 78089.08
```

```
# on average, the samples without the three outliers, abundance is
mean(sample_sums(Rps_mp)[!sample_names(Rps_mp) %in% c("10_1", "10_2", "10_3")]) # 14828.26
```

```
## [1] 14828.26
```

```

# 5,266x as much abundance

# Amount of different AMR classes present.
sort(table(tax_table(Rps)[, "AMR_class_primary"]))

##          Fosfomycin Oxazolidinone Streptogramin Lincosamide Quinolone
##                1            3            9           12            14
##      Phenicol Glycopeptide Sulphonamide Not determined Trimethoprim
##                18           20           30           37            57
##      Macrolide Tetracycline Aminoglycoside Beta-lactam
##                63            92           97          209

sort(table(tax_table(Rps)[, "ARGCluster90"]))

```

##	aac(2')-IIa	aac(3)-I_aac(3)-Ia
##	1	1
##	aac(3)-Ib	aac(6')-Iaj
##	1	1
##	aac(6')-Iak	aac(6')-Ian
##	1	1
##	aac(6')-Ib_aadA_ant(3')_clust	aac(6')-Ii
##	1	1
##	aac(6')-Iid	aac(6')-Iih
##	1	1
##	aac(6')-Im	aac(6')-Iw
##	1	1
##	ant(6)-Ib	ant(9)-Ia-1
##	1	1
##	ant(9)-Ia-2	aph(2')-Ig
##	1	1
##	aph(3')-VIIa	aph(4)-Ia
##	1	1
##	aph(6)-Ic	ARR_clust
##	1	1
##	blaACT-9	blaCGB
##	1	1
##	blaIND-5	blaOXA-308
##	1	1
##	blaOXA-449_clust	blaOXA-9
##	1	1
##	blaOXA_clust5	blaOXA_clust8
##	1	1
##	blaPEDO-2	blaSED1
##	1	1
##	cat(pC194)	cat(pC221)
##	1	1
##	cat_2	cat_3
##	1	1
##	catA1	catB
##	1	1

```

##          catP          catS
##          1              1
##          cmx             dfrA29
##          1                  1
##          dfrA6_dfrA31      dfrA8
##          1                  1
##          dfrD             dfrG
##          1                  1
##          dfrK_clust        erm(33)
##          1                  1
##          erm(34)           erm(35)
##          1                  1
##          erm(36)           erm(43)
##          1                  1
##          erm(A)_1_X03216   erm(Q)
##          1                  1
##          erm(T)_4_AJ488494  fexB
##          1                  1
##          fmr0              fosD
##          1                  1
##          lnu(A)            lnu(C)
##          1                  1
##          lnu(G)            lnu(P)
##          1                  1
##          lsa(B)            lsa(E)
##          1                  1
##          mdf(A)            mecA1
##          1                  1
##          mef(A)-3          mef(B)
##          1                  1
##          mef(C)            mph(B)
##          1                  1
##          mph(G)            npmA
##          1                  1
##          oqxB              poxtA
##          1                  1
##          rmtE              sul3_2_AJ459418
##          1                  1
##          tet(G)             tet(J)
##          1                  1
##          tet(L)_1_HM235948  tet(X)_3_AB097942
##          1                  1
##          tet(X3)            tet(Y)
##          1                  1
##          tet(Z)             tetB(P)
##          1                  1
##          VanG2XY           VanGXY
##          1                  1
##          VanHAX             VanHDX_clust
##          1                  1
##          VanLXY             vanXmurFvanKWI-2
##          1                  1
##          vat(D)             vgb(B)
##          1                  1

```

```

##          aac(3)-Iva          aac(3)-VIIa
##          2                      2
##          aadA13                 aadA4_aadA5
##          2                      2
##          aadA7                 aadD
##          2                      2
##          ant(6)-Ia_clust1      ant(6)-Ia_clust2
##          2                      2
##          aph(2'')-Id_aph(2'')-Ie    aph(2'')-If
##          2                      2
##          aph(3')-IIa            blaIND_clust3
##          2                      2
##          blaZ-8-10             cat_clust
##          2                      2
##          cfr(C)                erm(G)_clust
##          2                      2
##          erm(T)_clust          floR_clust
##          2                      2
##          lnu(B)                lsa(A)
##          2                      2
##          mph(A)                mph(C)
##          2                      2
##          msr(A)                msr(C)
##          2                      2
##          msr(D)                str
##          2                      2
##          tet(32)                tet(33)
##          2                      2
##          tet(40)                tet(44)
##          2                      2
##          tet(K)                 tet(0/32/0)
##          2                      2
##          VanA-vanH-vanHAX_clust VanC1XY
##          2                      2
##          VanHBX                 aac(6')-aph(2'')_clust
##          2                      3
##          aph(2'')-Ib            blaOXA-464_clust
##          3                      3
##          blaOXA-493_clust      cfr(B)
##          3                      3
##          cml_clust              dfrA5_dfrA30
##          3                      3
##          erm(D)_clust          erm(F)_clust
##          3                      3
##          lnu(F)                 mef(A)_clust
##          3                      3
##          tet(B)                 tet(X)_clust
##          3                      3
##          tetA(P)                VanHDX
##          3                      3
##          aph(3')-III             blaOXA_clust13
##          4                      4
##          dfrA15_clust           dfrA16_clust
##          4                      4

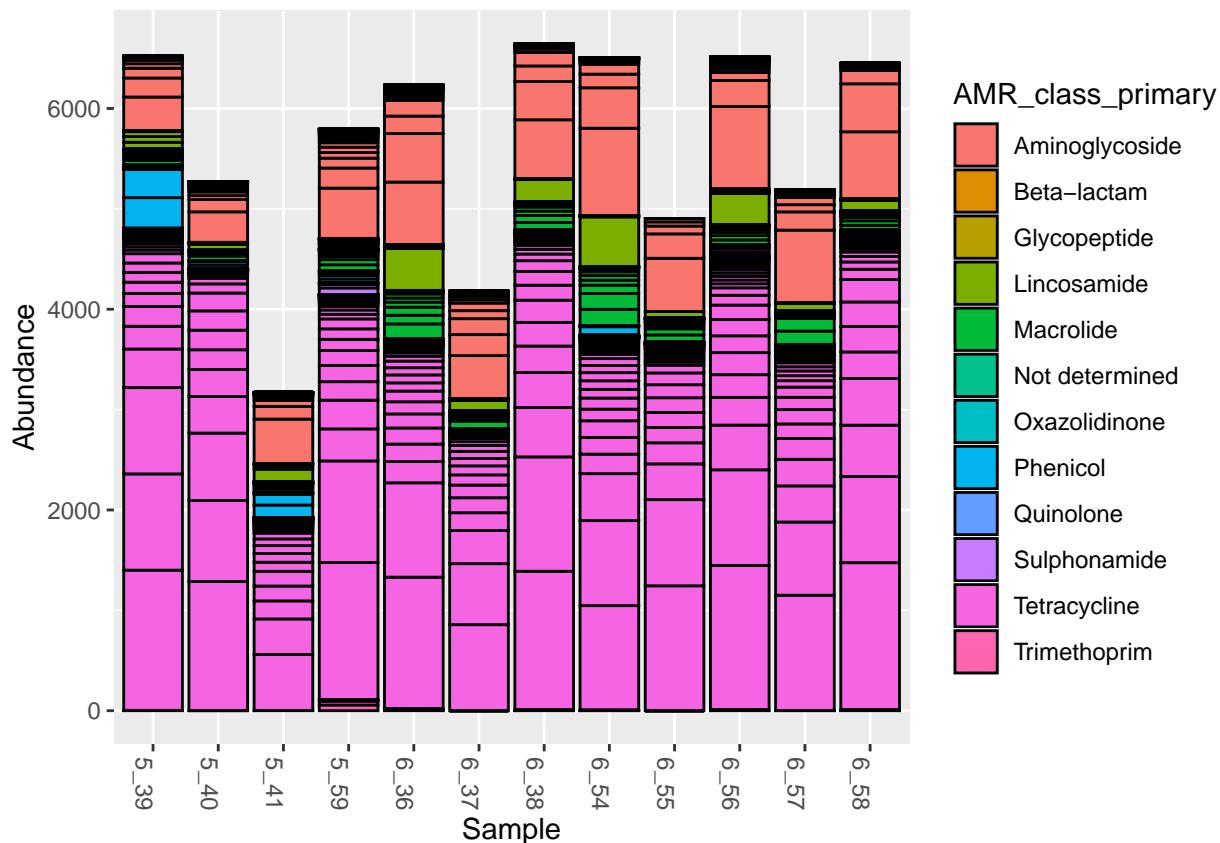
```

```

##          erm(X)_clust      tet(O/W/O)-1
##                      4                  4
##          tet(Q)            tet(S)        4
##                      4                  4
##          aph(3'')-Ib       aph(6)-Id   5
##                      5
##          blaOXA_clust11    blaOXA_clust20
##                      5                  5
##          blaOXA_clust4     blaOXA_clust9  5
##                      5                  5
##          blaSHV_clust      cfxA_clust   5
##                      5                  5
##          dfrA14             mecC_clust  5
##                      5
##          tet(L)_clust1     VanC2-3-4_clust
##                      5                  5
##          cepA_clust         qnrS_clust1  6
##                      6                  6
##          tet(A)              tet(O)       6
##                      6
##          tet(O/W/32/O/W/O)  aadA_clust1 7
##                      6
##          qnrB_clust1         vat(E)      7
##                      7
##          blaOXA-450_clust   cfiA_clust  9
##                      9
##          erm(C)_clust      meca_clust  9
##                      9
##          aph(3')-Ia_aph(3')-Ic
##                      10
##          dfrA7_dfrA17        dfrA_clust 10
##                      11
##          sul1                aadA_ant(3'')-Ia_clust
##                      14
##          tet(W)              tet(M)      14
##                      14
##          tet(W)              erm(B)_clust 15
##                      14
##          sul2                blaCTX-M_clust1
##                      15
##          dfrA1_clust         blaZ_clust 17
##                      17
##          blaSHV_blaLen_clust blaTEM_clust 18
##                      25
##                                     84

# for plotting abundances of specific stables
Rps %>% ps_filter(FarmRoundStable == c("Farm2R1S2")) %>% plot_bar(fill="AMR_class_primary")

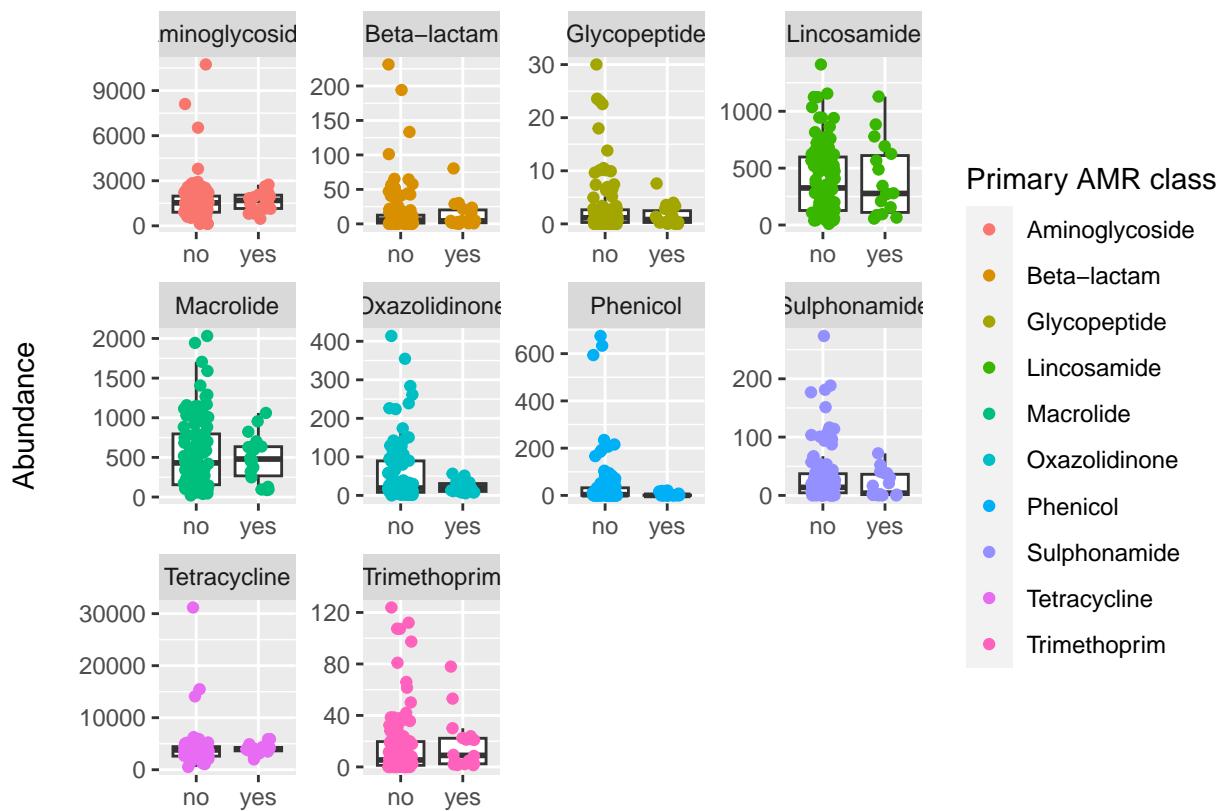
```



```
# visualisation on AB at AMR primary class level, more data for samples which have not been treated with antibiotics

ps_prim <- phyloseq:::tax_glom(Rps_copy, "Phylum")
taxa_names(ps_prim) <- phyloseq:::tax_table(ps_prim)[, "Phylum"]

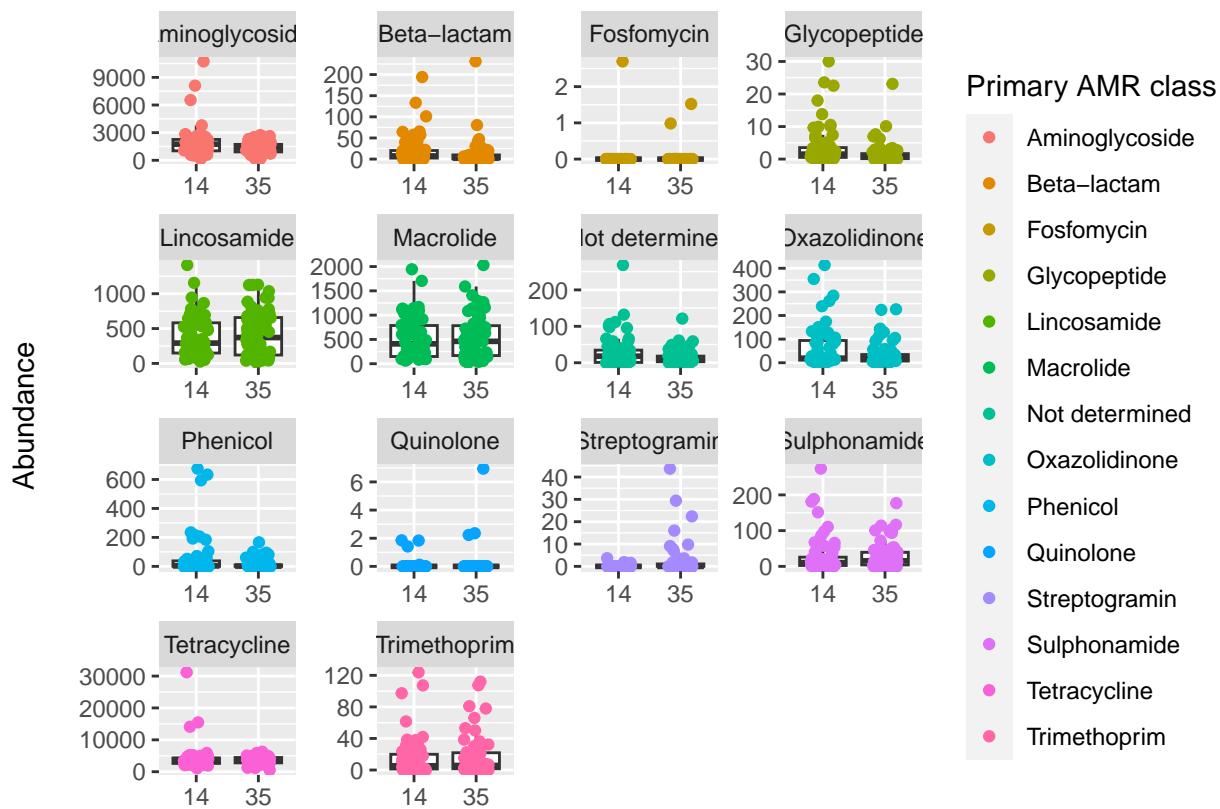
psmelt(ps_prim %>% subset_taxa(!Phylum %in% c("Not determined", "Fosfomycin", "Quinolone", "Streptogramin", "Vancomycin", "Beta-lactamase inhibitor", "Aminoglycoside", "Glycopeptide", "Lincosamide", "Macrolide", "Sulphonamide", "Tetracycline", "Trimethoprim", "Oxazolidinone", "Phenicol", "Quinolone", "Sulphonamide", "Tetracycline", "Trimethoprim", "Not determined"), "Phylum"))
ggplot(data = ., aes(x = AB, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  facet_wrap(~ Phylum, scales = "free") +
  labs(x = "", y = "Abundance\n", color = "Primary AMR class")
```



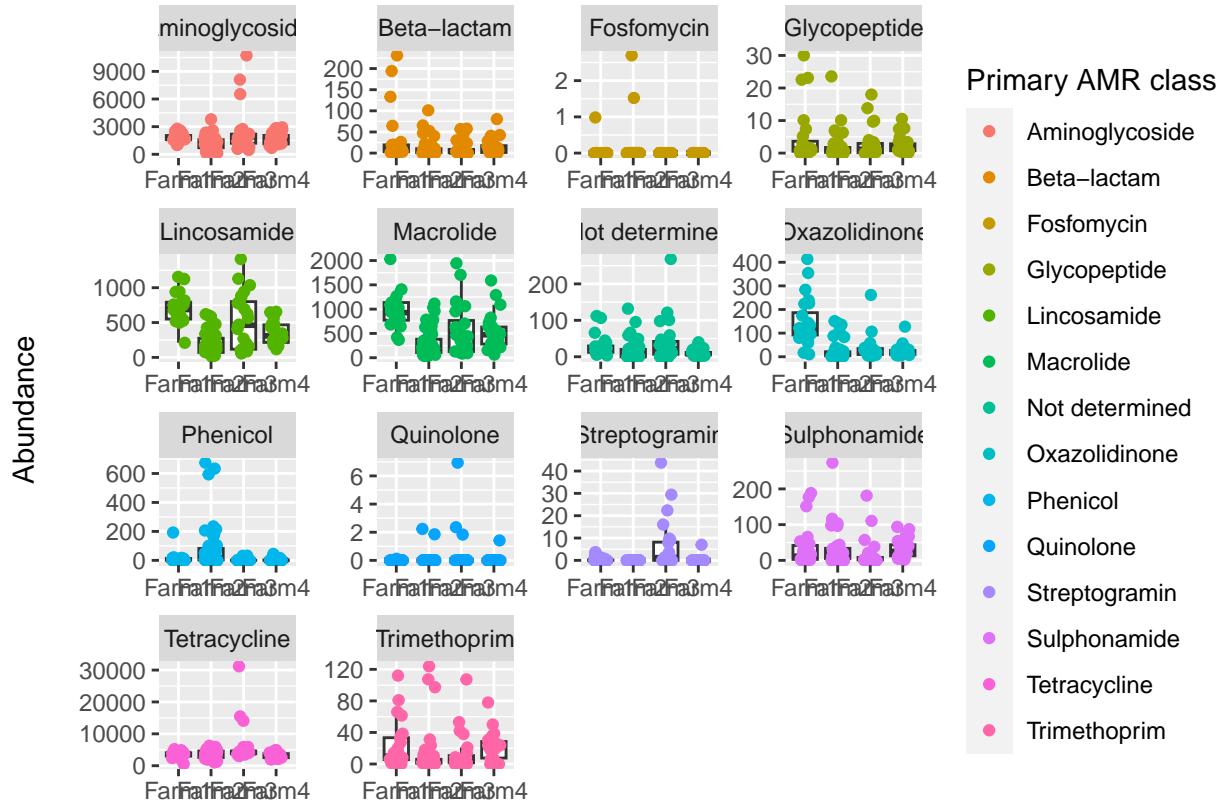
```

psmelt(ps_prim) %>% # Age
ggplot(data = ., aes(x = Age, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  facet_wrap(~ Phylum, scales = "free") +
  labs(x = "", y = "Abundance\n", color = "Primary AMR class")

```



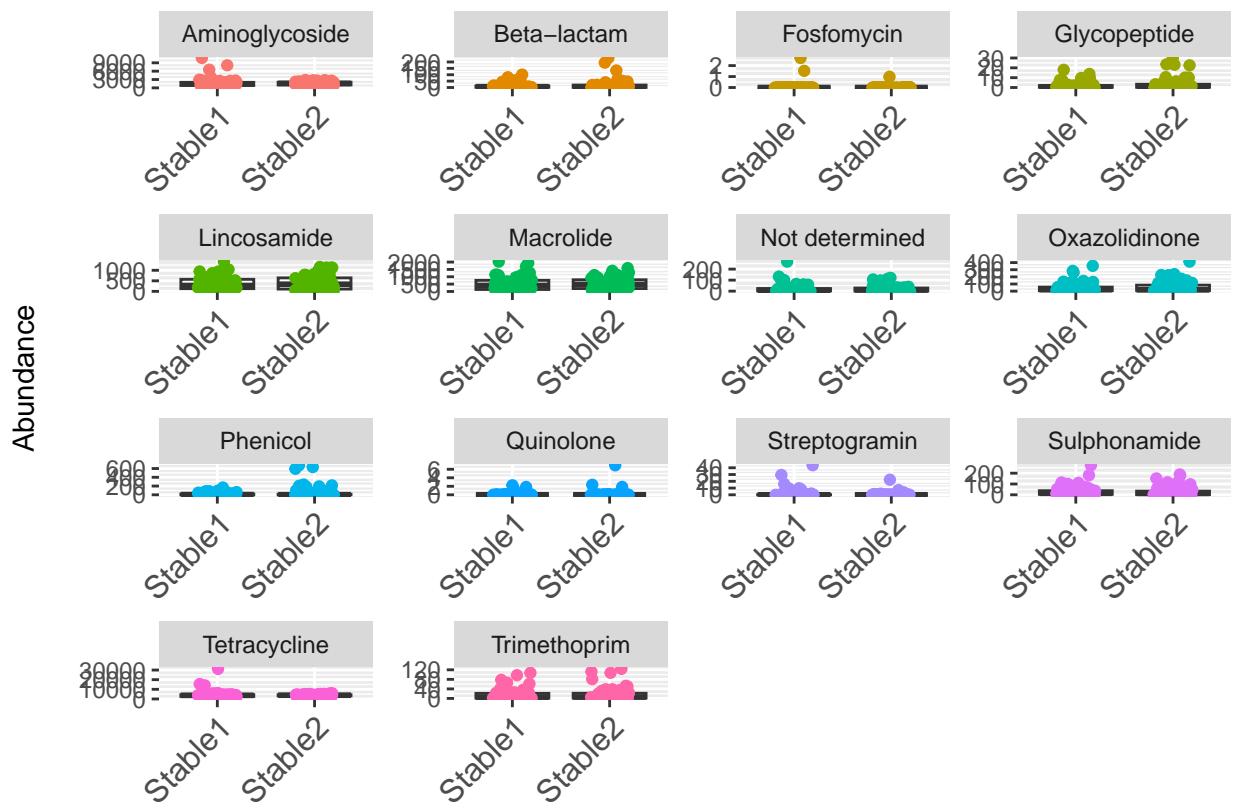
```
psmelt(ps_prim) %>% # Farm
ggplot(data = ., aes(x = Farm2, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  facet_wrap(~ Phylum, scales = "free") +
  labs(x = "", y = "Abundance\n", color = "Primary AMR class")
```



```

psmelt(ps_prim) %>% # Stable
ggplot(data = ., aes(x = Stable, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  facet_wrap(~ Phylum, scales = "free") +
  labs(x = "", y = "Abundance\n", color = "Primary AMR class") +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12))

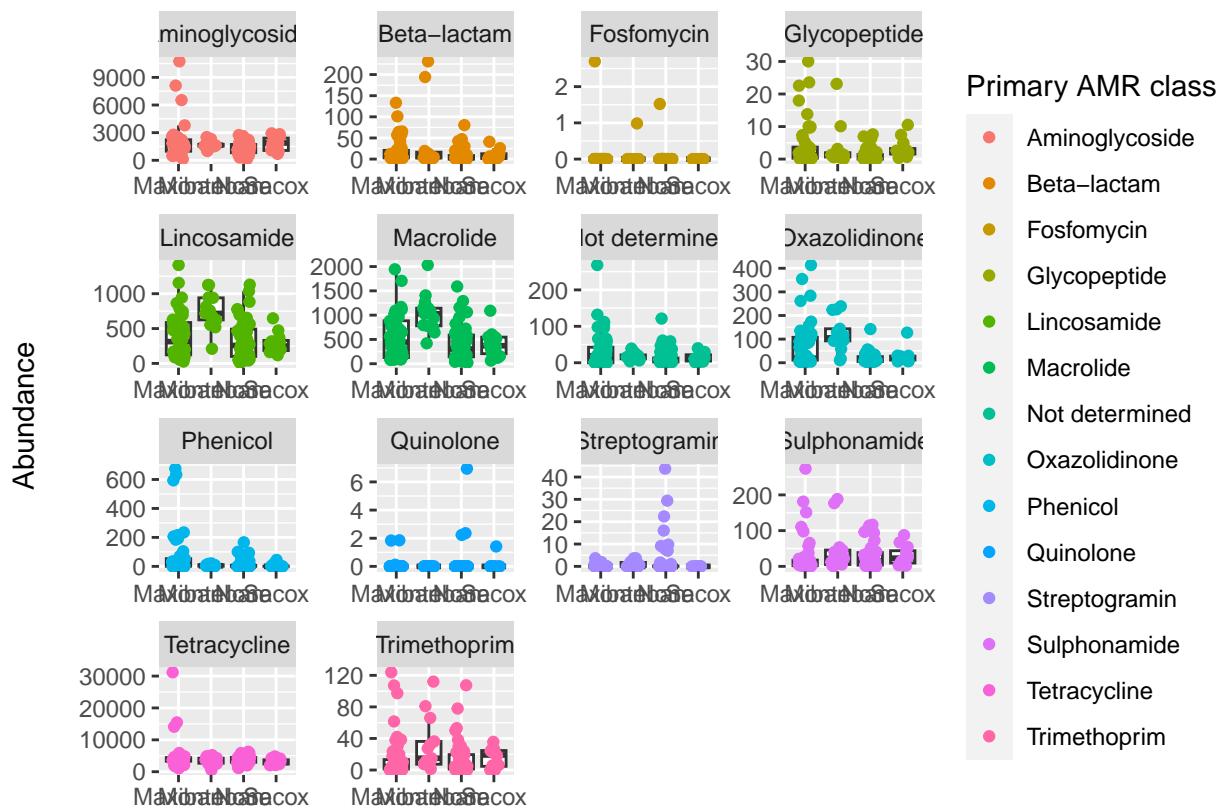
```



```

psmelt(ps_prim) %>% # Agent
ggplot(data = ., aes(x = Cox, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Phylum), height = 0, width = .2) +
  facet_wrap(~ Phylum, scales = "free") +
  labs(x = "", y = "Abundance\n", color = "Primary AMR class")

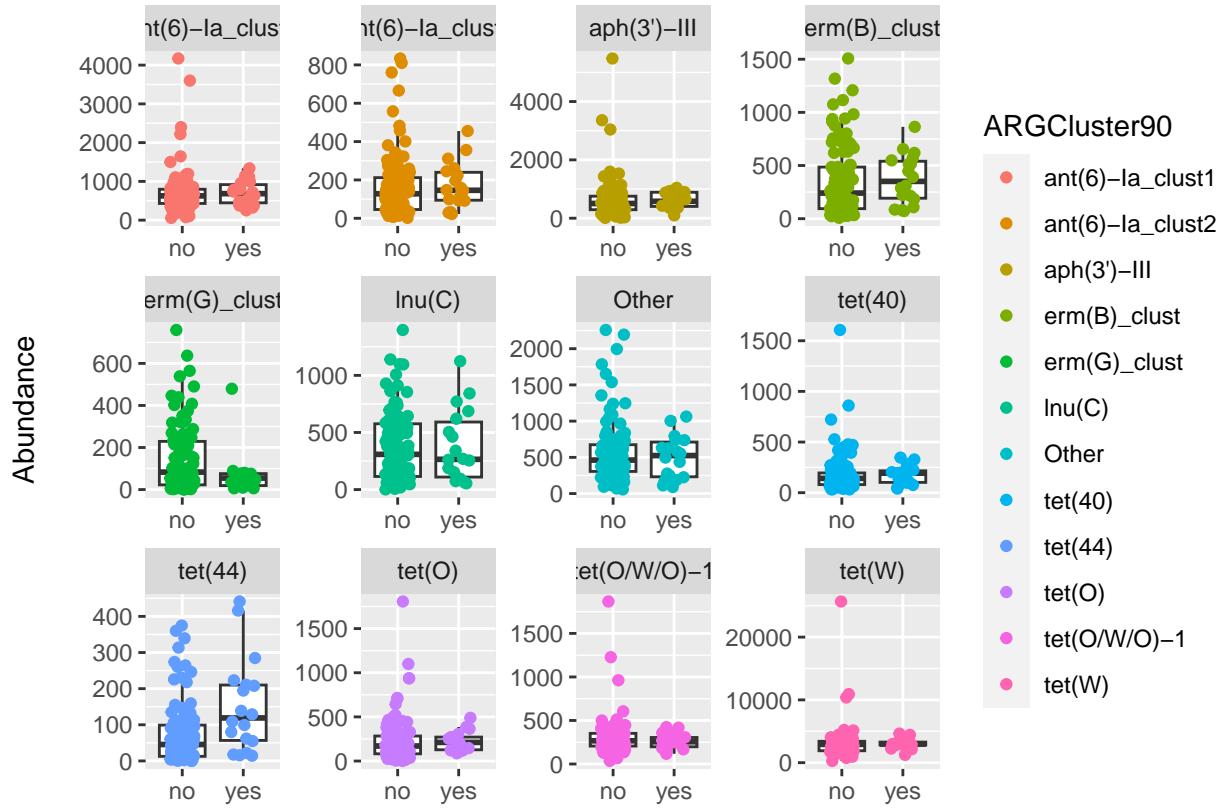
```



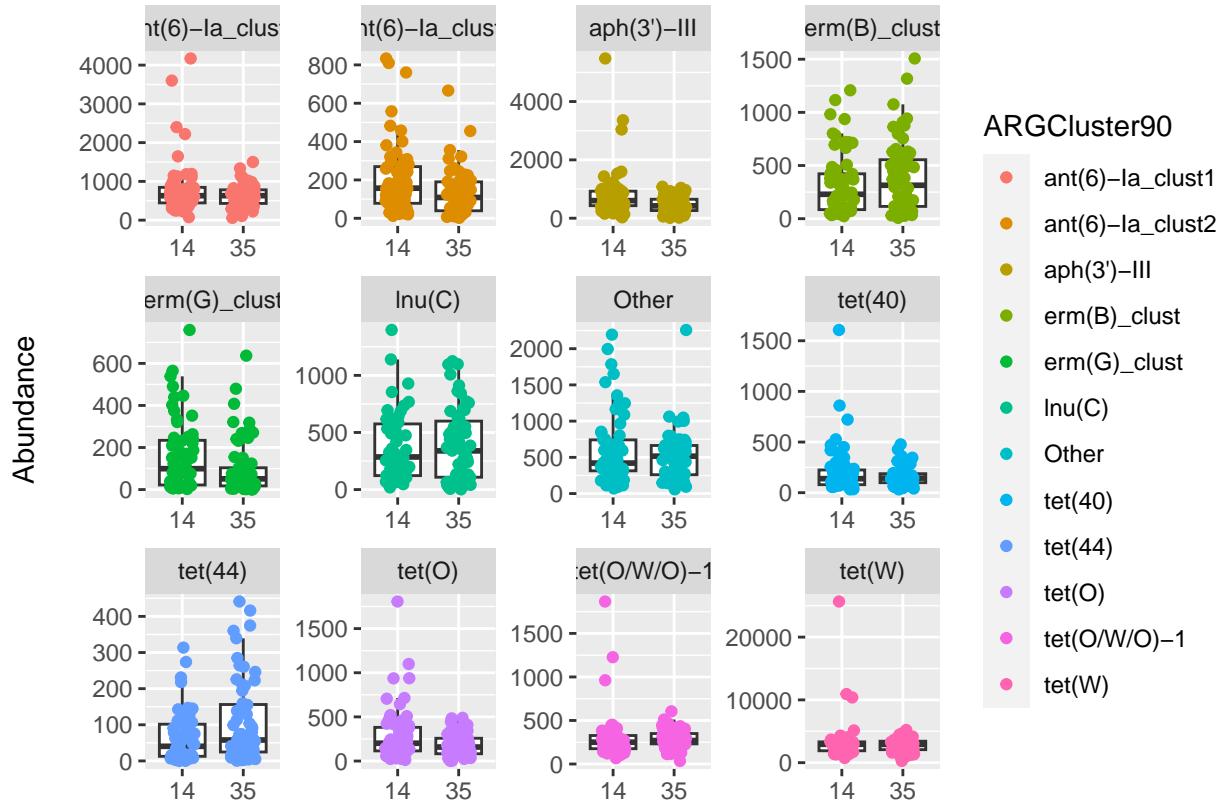
```
# visualisation on AB at ARGclust90 level, more data for samples which have not been treated with AB, b

ps_prim <- Rps_copy %>% aggregate_top_taxa2("Order", top = 11) %>% phyloseq::tax_glom("Order")
taxa_names(ps_prim) <- phyloseq::tax_table(ps_prim)[, "Order"]

psmelt(ps_prim) %>% # AB
  ggplot(data = ., aes(x = AB, y = Abundance)) +
    geom_boxplot(outlier.shape = NA) +
    geom_jitter(aes(color = Order), height = 0, width = .2) +
    facet_wrap(~ Order, scales = "free") +
    labs(x = "", y = "Abundance\n", color = "ARGCluster90")
```



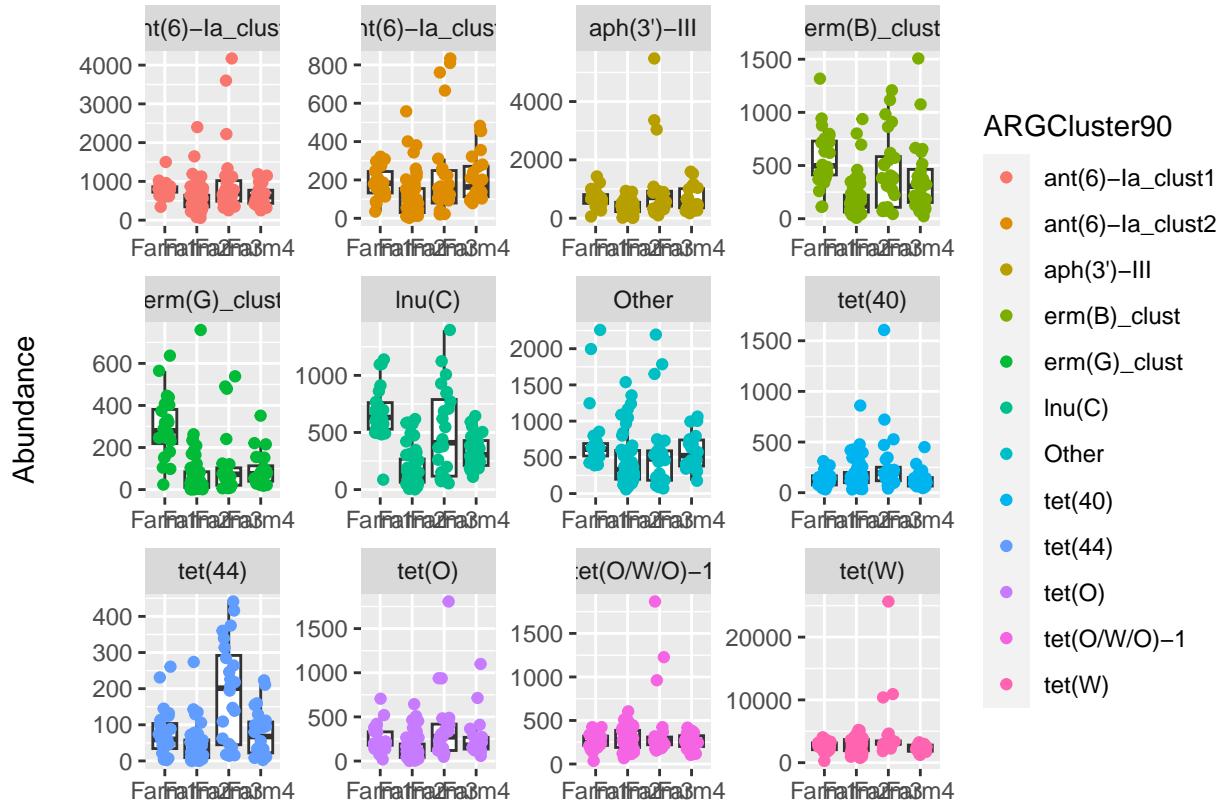
```
psmelt(ps_prim) %>% # Age
ggplot(data = ., aes(x = Age, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Order), height = 0, width = .2) +
  facet_wrap(~ Order, scales = "free") +
  labs(x = "", y = "Abundance\n", color = "ARGCluster90")
```



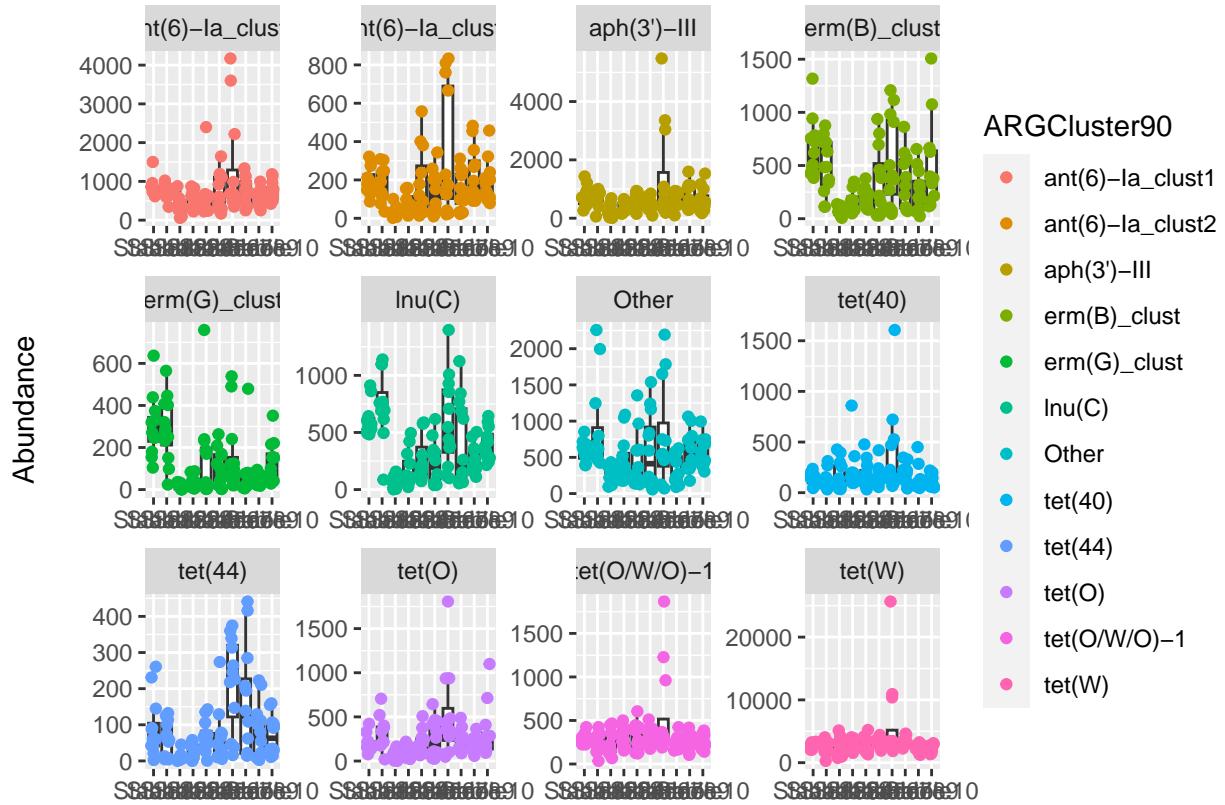
```

psmelt(ps_prim) %>% # Farm
ggplot(data = ., aes(x = Farm2, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Order), height = 0, width = .2) +
  facet_wrap(~ Order, scales = "free") +
  labs(x = "", y = "Abundance\n", color = "ARGCluster90")

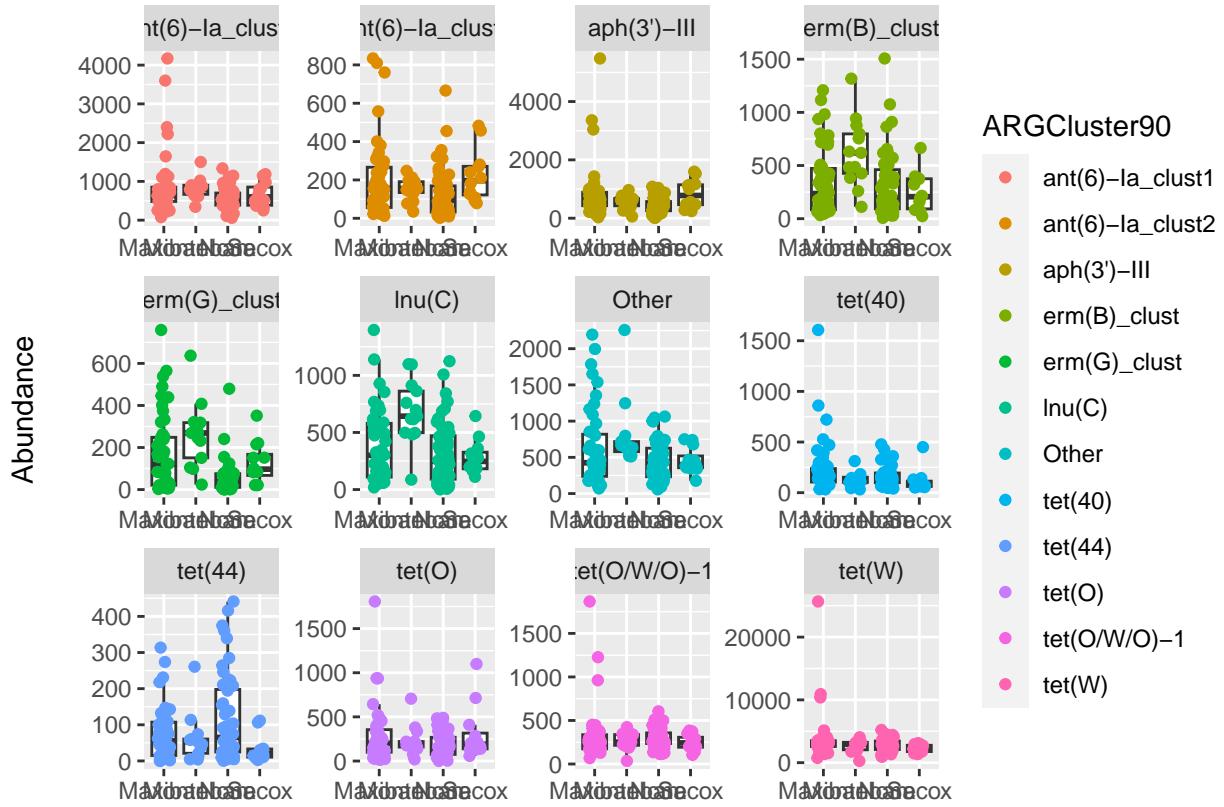
```



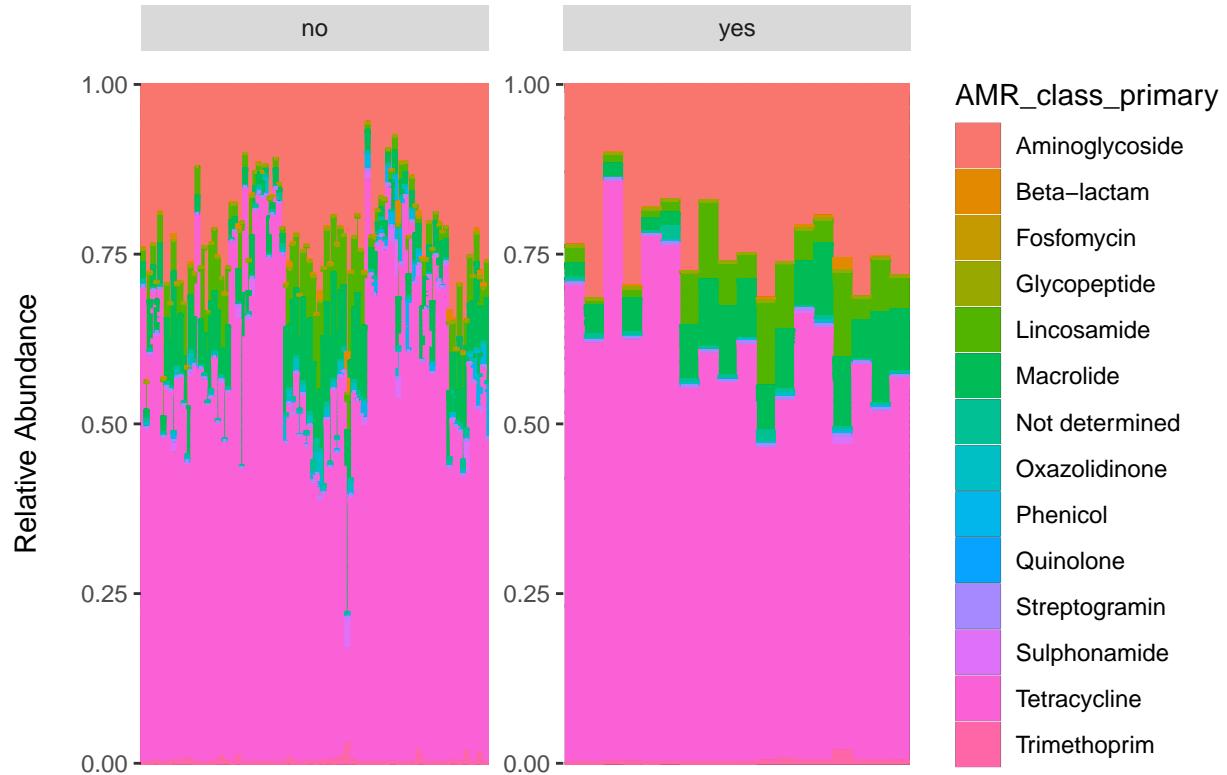
```
psmelt(ps_prim) %>% # Stable
ggplot(data = ., aes(x = Stables, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Order), height = 0, width = .2) +
  facet_wrap(~ Order, scales = "free") +
  labs(x = "", y = "Abundance\n", color = "ARGCluster90")
```



```
psmelt(ps_prim) %>% # Agent
ggplot(data = ., aes(x = Cox, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = Order), height = 0, width = .2) +
  facet_wrap(~ Order, scales = "free") +
  labs(x = "", y = "Abundance\n", color = "ARGCluster90")
```



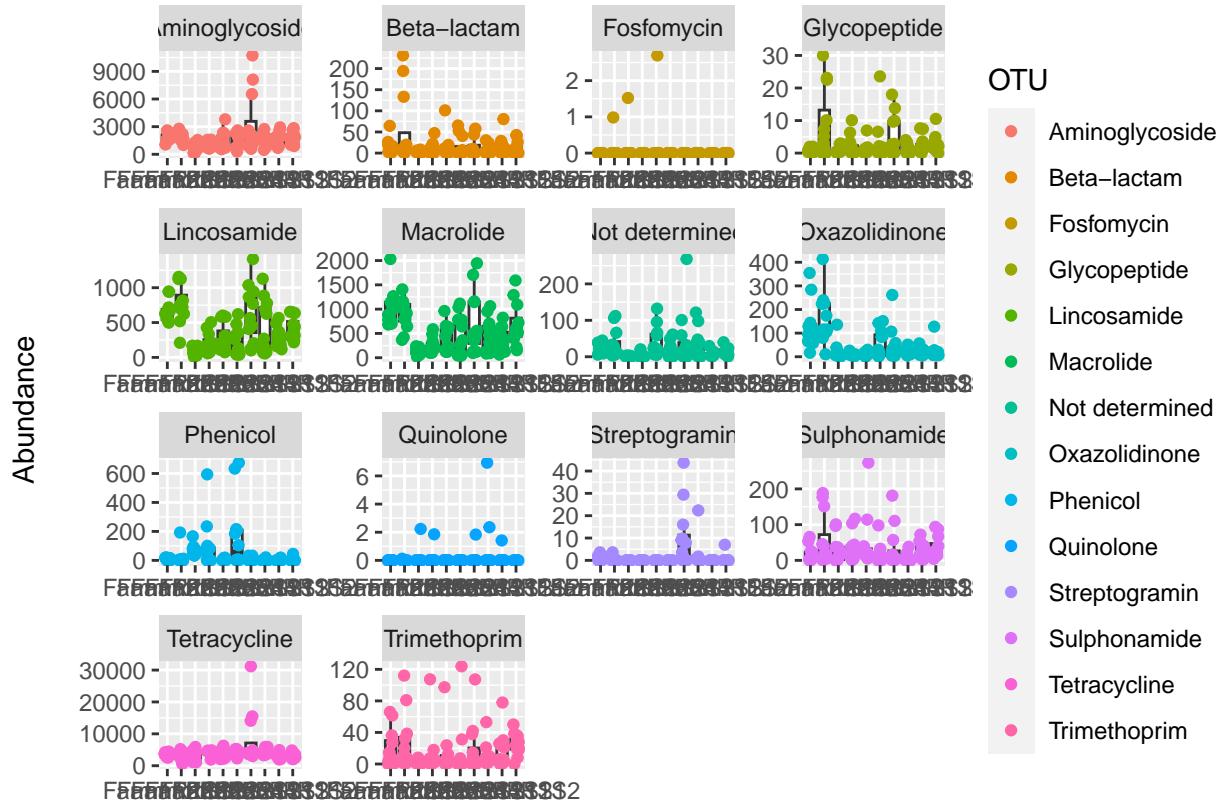
```
# relative abundances of primary AMR, split by AB
ps_rel_abund = transform_sample_counts(Rps, function(x){x / sum(x)})
plot_bar(ps_rel_abund, fill = "AMR_class_primary") +
  geom_bar(aes(color = AMR_class_primary, fill = AMR_class_primary), stat = "identity", position = "stack")
  labs(x = "", y = "Relative Abundance\n") +
  facet_wrap(~ AB, scales = "free") +
  theme(panel.background = element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```



```
# visualisation on primary AMR classes, more data for samples which have not been treated with AB, but

ps_prim <- phyloseq:::tax_glom(Rps, "AMR_class_primary")
taxa_names(ps_prim) <- phyloseq:::tax_table(ps_prim)[, "AMR_class_primary"]

psmelt(ps_prim) %>%
  ggplot(data = ., aes(x = FarmRoundStable, y = Abundance)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(color = OTU), height = 0, width = .2) +
  labs(x = "", y = "Abundance\n") +
  facet_wrap(~ OTU, scales = "free")
```



```
# Check the amount of unique ARGs in samples which have and have not been treated with antibiotics
Rps %>% ps_filter(AB == "no") %>% get_taxa_unique("ARGCluster90") # 163 different genes for non AB trea
```

```
## [1] "aac(6')-Iih"                                "blaSED1"
## [3] "aadA4_aadA5"                               "tet(M)"
## [5] "tet(B)"                                    "dfrA14"
## [7] "aac(6')-Iw"                                 "lسا(E)"
## [9] "dfrA5_dfrA30"                               "tet(44)"
## [11] "dfrA1_clust"                               "aph(2')-Ib"
## [13] "lnu(A)"                                    "cepA_clust"
## [15] "aph(3')-Ia_aph(3')-Ic"                    "blaTEM_clust"
## [17] "blaZ_clust"                                "blaOXA-450_clust"
## [19] "tet(O)"                                    "erm(B)_clust"
## [21] "ant(9)-Ia-2"                               "aac(3)-Iva"
## [23] "cat_clust"                                 "VanC2-3-4_clust"
## [25] "tet(X)_clust"                             "tet(G)"
## [27] "tet(Z)"                                    "erm(D)_clust"
## [29] "cfr(B)"                                    "erm(Q)"
## [31] "tet(W)"                                    "lnu(F)"
## [33] "tet(O/W/32/O/W/O)"                         "aph(3')-Ib"
## [35] "dfrA_clust"                                "sul2"
## [37] "cfrA_clust"                                "msr(A)"
## [39] "lnu(C)"                                    "mph(C)"
## [41] "blaCTX-M_clust1"                           "tet(X3)"
## [43] "tet(32)"                                   "VanG2XY"
## [45] "cat(pC221)"                                "cmx"
```

```

## [47] "mph(G)"
## [49] "tetA(P)"
## [51] "cfr(C)"
## [53] "dfrA7_dfrA17"
## [55] "msr(C)"
## [57] "cfxA_clust"
## [59] "vat(E)"
## [61] "tet(X)_3_AB097942"
## [63] "aadA_clust1"
## [65] "lnu(B)"
## [67] "erm(A)_1_X03216"
## [69] "aph(2')-Ig"
## [71] "blaOXA-308"
## [73] "msr(D)"
## [75] "lnu(P)"
## [77] "aph(3')-III"
## [79] "vanXmurFvanKWI-2"
## [81] "VanHDX"
## [83] "erm(36)"
## [85] "tet(Q)"
## [87] "tet(A)"
## [89] "lsa(A)"
## [91] "erm(C)_clust"
## [93] "aac(3)-VIa"
## [95] "aac(6')-Iid"
## [97] "tet(0/32/0)"
## [99] "dfrK_clust"
## [101] "erm(X)_clust"
## [103] "mef(B)"
## [105] "qnrS_clust1"
## [107] "tet(S)"
## [109] "rmtE"
## [111] "tet(0/W/0)-1"
## [113] "dfrA6_dfrA31"
## [115] "blaOXA_clust8"
## [117] "tet(Y)"
## [119] "aac(2')-IIa"
## [121] "blaOXA-464_clust"
## [123] "mef(A)_clust"
## [125] "blaOXA_clust13"
## [127] "erm(35)"
## [129] "blaOXA_clust4"
## [131] "aac(6')-Ian"
## [133] "VanGXY"
## [135] "fmr0"
## [137] "blaOXA-449_clust"
## [139] "lnu(G)"
## [141] "tet(33)"
## [143] "erm(43)"
## [145] "aac(6')-Iaj"
## [147] "aadD"
## [149] "mecA_clust"
## [151] "erm(G)_clust"
## [153] "aac(6')-Ib_aadA_ant(3')_clust" "sul1"
## [49] "tet(K)"
## [51] "tet(40)"
## [53] "mph(B)"
## [55] "aac(6')-aph(2')_clust"
## [57] "erm(34)"
## [59] "aadA13"
## [61] "poxtA"
## [63] "VanC1XY"
## [65] "catP"
## [67] "aadA_ant(3')-Ia_clust"
## [69] "lsa(B)"
## [71] "catA1"
## [73] "dfrA29"
## [75] "oqxB"
## [77] "aac(3)-I_aac(3)-Ia"
## [79] "VanHDX_clust"
## [81] "erm(F)_clust"
## [83] "mdf(A)"
## [85] "qnrB_clust1"
## [87] "ant(6)-Ib"
## [89] "fexB"
## [91] "dfrA15_clust"
## [93] "aph(2')-If"
## [95] "aph(2')-Id_aph(2')-Ie"
## [97] "dfrD"
## [99] "tetB(P)"
## [101] "floR_clust"
## [103] "tet(J)"
## [105] "ant(6)-Ia_clust1"
## [107] "mph(A)"
## [109] "sul3_2_AJ459418"
## [111] "erm(T)_4_AJ488494"
## [113] "blaSHV_blaLen_clust"
## [115] "dfrG"
## [117] "ant(6)-Ia_clust2"
## [119] "str"
## [121] "tet(L)_clust1"
## [123] "aac(6')-Im"
## [125] "dfrA16_clust"
## [127] "cat_2"
## [129] "fosD"
## [131] "aph(4)-Ia"
## [133] "cat(pC194)"
## [135] "aph(6)-Id"
## [137] "cml_clust"
## [139] "VanHBX"
## [141] "mecA1"
## [143] "aac(6')-Iak"
## [145] "blaZ-8-10"
## [147] "aac(3)-Ib"
## [149] "aac(6')-Ii"
## [151] "catB"
## [153] "blaIND-5"

```

```

## [155] "mecC_clust"                                "ant(9)-Ia-1"
## [157] "aph(3')-IIa"                               "VanLXY"
## [159] "aph(3')-VIIa"                             "cat_3"
## [161] "npmA"                                     "blaACT-9"
## [163] "erm(33)"                                  "catS"
## [165] "aadA7"                                    "aph(6)-Ic"
## [167] "mef(A)-3"                                 "VanA-vanH-vanHAX_clust"

Rps %>% ps_filter(AB == "yes") %>% get_taxa_unique("ARGCluster90") # 81 different genes for AB treated

## [1] "aadA4_aadA5"                                "tet(M)"
## [3] "tet(B)"                                     "dfrA14"
## [5] "lsa(E)"                                    "dfrA5_dfrA30"
## [7] "tet(44)"                                   "dfrA1_clust"
## [9] "aph(2')-Ib"                                "lnu(A)"
## [11] "cepA_clust"                               "aph(3')-Ia_aph(3')-Ic"
## [13] "tet(0)"                                    "erm(B)_clust"
## [15] "tet(X)_clust"                            "erm(T)_clust"
## [17] "blaOXA_clust9"                           "tet(Z)"
## [19] "tet(W)"                                    "tet(0/W/32/0/W/0)"
## [21] "aph(3')-Ib"                                "dfrA8"
## [23] "sul2"                                      "cfiA_clust"
## [25] "msr(A)"                                    "blaTEM_clust"
## [27] "lnu(C)"                                    "mef(C)"
## [29] "tet(32)"                                  "VanG2XY"
## [31] "sul1"                                       "tet(K)"
## [33] "cfr(C)"                                   "tet(40)"
## [35] "mph(B)"                                   "aac(6')-aph(2')_clust"
## [37] "cfxA_clust"                             "blaCGB"
## [39] "vgb(B)"                                   "blaOXA-493_clust"
## [41] "vat(E)"                                    "aadA13"
## [43] "blaOXA_clust20"                           "poxtA"
## [45] "aadA_clust1"                            "lnu(B)"
## [47] "catP"                                     "aadA_ant(3')-Ia_clust"
## [49] "aph(2')-Ig"                                "catA1"
## [51] "msr(D)"                                    "lnu(P)"
## [53] "aph(3')-III"                               "mdf(A)"
## [55] "erm(F)_clust"                            "tet(Q)"
## [57] "qnrB_clust1"                            "tet(A)"
## [59] "ant(6)-Ib"                                "lsa(A)"
## [61] "aac(3)-VIa"                               "aph(2')-Id_aph(2')-Ie"
## [63] "tet(0/32/0)"                             "dfrD"
## [65] "dfrK_clust"                             "tetB(P)"
## [67] "blaOXA_clust11"                           "mef(B)"
## [69] "ant(6)-Ia_clust1"                         "sul3_2_AJ459418"
## [71] "erm(T)_4_AJ488494"                        "blaCTX-M_clust1"
## [73] "dfrG"                                     "lnu(F)"
## [75] "ant(6)-Ia_clust2"                         "tet(0/W/0)-1"
## [77] "blaOXA-9"                                 "tet(L)_clust1"
## [79] "mef(A)_clust"                            "aac(6')-Im"
## [81] "cat_2"                                    "VanGXY"
## [83] "cml_clust"                               "lnu(G)"
## [85] "aph(6)-Id"                                "erm(43)"
## [87] "mph(C)"                                 "meca_clust"

```

```

## [89] "erm(G)_clust"                      "VanHBX"
## [91] "aac(6')-Ib_aadA_ant(3'')_clust"   "vat(D)"
## [93] "ant(9)-Ia-1"                        "aph(2'')-If"
## [95] "cat_3"                                "npmA"
## [97] "mef(A)-3"

Rps %>% get_taxa_unique("ARGCluster90") # 188 different genes in total, meaning 25 do not overlap

## [1] "blaTEM_clust"                      "blaSHV_clust"
## [3] "aac(6')-Iih"                        "blaSED1"
## [5] "aadA4_aadA5"                        "blaCTX-M_clust1"
## [7] "tet(M)"                            "tet(B)"
## [9] "dfrA14"                             "aac(6')-Iw"
## [11] "lsa(E)"                           "dfrA5_dfrA30"
## [13] "tet(44)"                           "dfrA1_clust"
## [15] "aph(2'')-Ib"                        "lnu(A)"
## [17] "cepA_clust"                        "aph(3')-Ia_aph(3')-Ic"
## [19] "mecC_clust"                        "vat(E)"
## [21] "blaZ_clust"                        "blaOXA-450_clust"
## [23] "tet(0)"                            "erm(B)_clust"
## [25] "ant(9)-Ia-2"                        "aac(3)-Iva"
## [27] "blaOXA_clust4"                     "cat_clust"
## [29] "blaSHV_blaLen_clust"               "VanC2-3-4_clust"
## [31] "tet(X)_clust"                       "tet(G)"
## [33] "sul2"                               "erm(T)_clust"
## [35] "blaOXA_clust9"                     "tet(Z)"
## [37] "tetA(P)"                           "erm(D)_clust"
## [39] "cfr(B)"                            "erm(Q)"
## [41] "blaOXA_clust11"                    "tet(W)"
## [43] "lnu(F)"                            "tet(O/W/32/O/W/O)"
## [45] "aph(3'')-Ib"                        "dfrA8"
## [47] "mecA_clust"                        "dfrA_clust"
## [49] "cfiA_clust"                        "dfrA7_dfrA17"
## [51] "qnrS_clust1"                       "blaPEDO-2"
## [53] "msr(A)"                            "aadA13"
## [55] "lnu(C)"                            "sul1"
## [57] "mph(C)"                            "tet(X3)"
## [59] "mef(C)"                            "tet(32)"
## [61] "qnrB_clust1"                       "VanG2XY"
## [63] "cat(pC221)"                         "dfrA16_clust"
## [65] "cmx"                                "mph(G)"
## [67] "tet(K)"                            "cfr(C)"
## [69] "VanHDX"                            "tet(40)"
## [71] "mph(B)"                            "msr(C)"
## [73] "aac(6')-aph(2'')_clust"            "cfxA_clust"
## [75] "blaCGB"                            "vgb(B)"
## [77] "aadA_clust1"                       "erm(34)"
## [79] "blaOXA-493_clust"                  "blaOXA_clust20"
## [81] "tet(X)_3_AB097942"                 "poxtA"
## [83] "cml_clust"                          "tet(L)_1_HM235948"
## [85] "VanC1XY"                            "lnu(B)"
## [87] "catP"                                "erm(A)_1_X03216"
## [89] "erm(C)_clust"                       "aadA_ant(3'')-Ia_clust"
## [91] "aph(2'')-Ig"                         "lsa(B)"

```

```

## [93] "dfrA15_clust"                                "blaOXA-308"
## [95] "blaZ-8-10"                                   "catA1"
## [97] "msr(D)"                                     "tet(A)"
## [99] "dfrA29"                                      "lnu(P)"
## [101] "oqxB"                                       "aph(3')-III"
## [103] "blaOXA-464_clust"                            "aac(3)-I_aac(3)-Ia"
## [105] "vanXmurFvanKWI-2"                            "aadA7"
## [107] "VanHDX_clust"                               "erm(F)_clust"
## [109] "erm(36)"                                    "mdf(A)"
## [111] "tet(Q)"                                     "ant(6)-Ib"
## [113] "lsa(A)"                                     "blaIND_clust3"
## [115] "fexB"                                       "aac(3)-VIa"
## [117] "aph(2')-If"                                 "aac(6')-Iid"
## [119] "blaOXA_clust13"                             "aph(2')-Id_aph(2')-Ie"
## [121] "tet(0/32/0)"                                "VanHAX"
## [123] "dfrD"                                       "VanA-vanH-vanHAX_clust"
## [125] "dfrK_clust"                                 "tetB(P)"
## [127] "tet(L)_clust1"                             "erm(X)_clust"
## [129] "floR_clust"                                "mef(B)"
## [131] "tet(J)"                                    "ant(6)-Ia_clust1"
## [133] "tet(S)"                                    "ant(6)-Ia_clust2"
## [135] "mph(A)"                                    "rmtE"
## [137] "ARR_clust"                                 "sul3_2_AJ459418"
## [139] "tet(0/W/0)-1"                               "erm(T)_4_AJ488494"
## [141] "dfrA6_dfrA31"                               "blaOXA_clust8"
## [143] "aph(6)-Id"                                 "dfrG"
## [145] "tet(Y)"                                    "aac(2')-IIa"
## [147] "str"                                       "blaOXA-9"
## [149] "mef(A)_clust"                            "aac(6')-Im"
## [151] "erm(35)"                                    "cat_2"
## [153] "fosD"                                       "aac(6')-Ian"
## [155] "aph(4)-Ia"                                 "VanGXY"
## [157] "cat(pC194)"                                "fmr0"
## [159] "blaOXA-449_clust"                           "lnu(G)"
## [161] "VanHBX"                                     "tet(33)"
## [163] "mecA1"                                      "erm(43)"
## [165] "aac(6')-Iak"                               "aac(6')-Iaj"
## [167] "aadD"                                       "blaOXA_clust5"
## [169] "aac(3)-Ib"                                "aac(6')-Ii"
## [171] "erm(G)_clust"                            "catB"
## [173] "aac(6')-Ib_aadA_ant(3')_clust"           "vat(D)"
## [175] "blaIND-5"                                   "aph(3')-IIa"
## [177] "ant(9)-Ia-1"                                "VanLXY"
## [179] "aph(3')-VIIa"                              "cat_3"
## [181] "npmA"                                       "blaACT-9"
## [183] "erm(33)"                                    "catS"
## [185] "aph(6)-Ic"                                 "mef(A)-3"

```

```
Rps TPM %>% ps_filter(AB == "no") %>% get_taxa_unique("ARGCluster90") # 168 different genes for non AB
```

```

## [1] "tet(M)"                                     "blaTEM_clust"
## [3] "sul3_2_AJ459418"                            "VanG2XY"
## [5] "tet(0/W/0)-1"                               "erm(X)_clust"
## [7] "dfrA_clust"                                 "msr(D)"

```

```

## [9] "blaOXA_clust8"
## [11] "rmtE"
## [13] "sul2"
## [15] "lnu(P)"
## [17] "erm(35)"
## [19] "blaZ_clust"
## [21] "tet(A)"
## [23] "aac(6')-Iid"
## [25] "aadD"
## [27] "tet(Q)"
## [29] "blaCTX-M_clust1"
## [31] "dfrA29"
## [33] "tet(40)"
## [35] "cfr(C)"
## [37] "ant(6)-Ib"
## [39] "vanXmurFvanKWI-2"
## [41] "dfrA7_dfrA17"
## [43] "mph(A)"
## [45] "tet(32)"
## [47] "sul1"
## [49] "tet(44)"
## [51] "msr(C)"
## [53] "lsa(E)"
## [55] "erm(F)_clust"
## [57] "erm(G)_clust"
## [59] "VanC2-3-4_clust"
## [61] "erm(T)_4_AJ488494"
## [63] "aac(6')-Ii"
## [65] "mecC_clust"
## [67] "erm(33)"
## [69] "aac(6')-Iaj"
## [71] "aac(6')-Im"
## [73] "cfr(B)"
## [75] "tet(0)"
## [77] "lnu(B)"
## [79] "cml_clust"
## [81] "aph(2')-Id_aph(2')-Ie"
## [83] "aph(2')-Ig"
## [85] "aac(6')-Iih"
## [87] "mef(A)_clust"
## [89] "tet(X)_3_AB097942"
## [91] "dfrA16_clust"
## [93] "cfxA_clust"
## [95] "cat_2"
## [97] "catP"
## [99] "aph(6)-Ic"
## [101] "tet(0/W/32/0/W/0)"
## [103] "aac(6')-Iak"
## [105] "aadA7"
## [107] "mef(A)-3"
## [109] "lnu(A)"
## [111] "tet(J)"
## [113] "tetB(P)"
## [115] "aac(3)-Ib"
## [9] "erm(43)"
## [11] "dfrA15_clust"
## [13] "aac(2')-IIa"
## [15] "aadA_clust1"
## [17] "tet(W)"
## [19] "dfrA14"
## [21] "tet(B)"
## [23] "tetA(P)"
## [25] "blaSHV_blaLen_clust"
## [27] "tet(Y)"
## [29] "mph(C)"
## [31] "aph(2')-If"
## [33] "cepA_clust"
## [35] "tet(X3)"
## [37] "aadA_ant(3')-Ia_clust"
## [39] "vat(E)"
## [41] "ant(9)-Ia-2"
## [43] "npmA"
## [45] "aph(3')-Ib"
## [47] "tet(0/32/0)"
## [49] "aph(4)-Ia"
## [51] "blaOXA-450_clust"
## [53] "cfiA_clust"
## [55] "dfrA1_clust"
## [57] "qnrB_clust1"
## [59] "lnu(F)"
## [61] "VanC1XY"
## [63] "mph(G)"
## [65] "tet(K)"
## [67] "aph(3')-Ia_aph(3')-Ic"
## [69] "erm(B)_clust"
## [71] "aph(6)-Id"
## [73] "tet(33)"
## [75] "aph(3')-IIa"
## [77] "catS"
## [79] "dfrK_clust"
## [81] "blaOXA-308"
## [83] "tet(X)_clust"
## [85] "poxtA"
## [87] "cat_3"
## [89] "lsa(B)"
## [91] "VanHBX"
## [93] "cat_clust"
## [95] "tet(Z)"
## [97] "aph(3')-III"
## [99] "blaSED1"
## [101] "aac(6')-Ib_aadA_ant(3')_clust"
## [103] "erm(36)"
## [105] "msr(A)"
## [107] "mph(B)"
## [109] "ant(6)-Ia_clust1"
## [111] "dfrG"
## [113] "qnrS_clust1"
## [115] "tet(L)_clust1"

```

```

## [117] "aph(2')-Ib"
## [119] "blaZ-8-10"
## [121] "ant(9)-Ia-1"
## [123] "cat(pC194)"
## [125] "fexB"
## [127] "dfrA5_dfrA30"
## [129] "erm(D)_clust"
## [131] "blaACT-9"
## [133] "mecA1"
## [135] "VanHDX_clust"
## [137] "cmx"
## [139] "dfrA6_dfrA31"
## [141] "aac(3)-VIIa"
## [143] "ant(6)-Ia_clust2"
## [145] "blaOXA_clust4"
## [147] "catA1"
## [149] "lsa(A)"
## [151] "aadA13"
## [153] "meca_clust"
## [155] "cat(pC221)"
## [157] "fosD"
## [159] "blaOXA_clust13"
## [161] "VanGXY"
## [163] "aac(6')-Ia"
## [165] "VanA-vanH-vanHAX_clust"
## [167] "catB"

```

```
Rps_tpm %>% ps_filter(AB == "yes") %>% get_taxa_unique("ARGCluster90") # 97 different genes for AB trea
```

```

## [1] "tet(M)"
## [3] "VanG2XY"
## [5] "msr(D)"
## [7] "sul2"
## [9] "blaOXA-9"
## [11] "tet(O/W/O)-1"
## [13] "tet(A)"
## [15] "mph(C)"
## [17] "tet(40)"
## [19] "cfr(C)"
## [21] "blaCTX-M_clust1"
## [23] "vat(E)"
## [25] "tet(32)"
## [27] "tet(0/32/0)"
## [29] "lsa(E)"
## [31] "dfrA1_clust"
## [33] "qnrB_clust1"
## [35] "erm(T)_4_AJ488494"
## [37] "blaOXA-493_clust"
## [39] "aph(3')-Ia_aph(3')-Ic"
## [41] "aac(6')-Im"
## [43] "tet(O)"
## [45] "lnu(B)"
## [47] "dfrK_clust"
## [49] "mef(C)"

```

```

## [51] "tet(X)_clust"                      "poxtA"
## [53] "mef(A)_clust"                      "cat_3"
## [55] "cfxA_clust"                        "cat_2"
## [57] "tet(Z)"                            "catP"
## [59] "aph(3')-III"                       "tet(0/W/32/0/W/0)"
## [61] "aac(6')-Ib_aadA_ant(3')_clust"    "sul1"
## [63] "mef(A)-3"                          "mph(B)"
## [65] "lnu(A)"                           "ant(6)-Ia_clust1"
## [67] "dfrG"                             "tetB(P)"
## [69] "tet(L)_clust1"                     "aph(2')-Ib"
## [71] "erm(T)_clust"                      "msr(A)"
## [73] "lnu(C)"                           "ant(9)-Ia-1"
## [75] "lnu(G)"                           "mef(B)"
## [77] "dfrA5_dfrA30"                      "vat(D)"
## [79] "mdf(A)"                           "aac(3)-VIa"
## [81] "ant(6)-Ia_clust2"                  "vgb(B)"
## [83] "VanHBX"                           "dfrA8"
## [85] "aadA_clust1"                      "blaOXA_clust20"
## [87] "catA1"                            "lsa(A)"
## [89] "dfrD"                             "aadA13"
## [91] "mecA_clust"                       "aac(6')-aph(2')_clust"
## [93] "aadA4_aadA5"                      "erm(F)_clust"
## [95] "lnu(F)"                           "VanGXY"
## [97] "blaCGB"

```

Rps TPM %>% get_taxa_unique("ARGCluster90") # 186 different genes in total, meaning 18 do not overlap

```

## [1] "tet(M)"                           "blaTEM_clust"
## [3] "tet(L)_clust1"                   "tet(A)"
## [5] "sul3_2_AJ459418"                 "VanG2XY"
## [7] "tet(0/W/0)-1"                    "erm(X)_clust"
## [9] "blaSHV_clust"                   "vat(E)"
## [11] "dfrA14"                          "dfrA_clust"
## [13] "aadA_ant(3')-Ia_clust"          "msr(D)"
## [15] "blaOXA_clust8"                  "cfxA_clust"
## [17] "erm(43)"                         "rmtE"
## [19] "aph(6)-Id"                      "dfrA1_clust"
## [21] "sul2"                            "mecC_clust"
## [23] "dfrA15_clust"                  "aac(2')-IIa"
## [25] "lnu(P)"                          "aadA_clust1"
## [27] "blaOXA-9"                        "erm(35)"
## [29] "blaOXA_clust4"                  "tet(W)"
## [31] "blaZ_clust"                      "blaSHV_blaLen_clust"
## [33] "tet(B)"                           "aac(6')-Iid"
## [35] "blaOXA_clust13"                 "tetA(P)"
## [37] "aadD"                            "tet(0/W/32/0/W/0)"
## [39] "tet(Q)"                           "aadA7"
## [41] "tet(Y)"                           "blaCTX-M_clust1"
## [43] "mph(C)"                          "blaOXA_clust9"
## [45] "dfrA29"                           "aph(2')-If"
## [47] "dfrA7_dfrA17"                   "erm(C)_clust"
## [49] "tet(40)"                          "cepA_clust"
## [51] "cfr(C)"                           "tet(X3)"
## [53] "ant(6)-Ib"                        "qnrS_clust1"

```

```

## [55] "sul1"
## [57] "ant(9)-Ia-2"
## [59] "mph(A)"
## [61] "tet(32)"
## [63] "tet(0/32/0)"
## [65] "aph(4)-Ia"
## [67] "VanHDX"
## [69] "lsa(E)"
## [71] "cfiA_clust"
## [73] "blaOXA_clust20"
## [75] "erm(G)_clust"
## [77] "lnu(F)"
## [79] "VanC1XY"
## [81] "mph(G)"
## [83] "erm(33)"
## [85] "aph(3')-Ia_aph(3')-Ic"
## [87] "erm(B)_clust"
## [89] "blaOXA-464_clust"
## [91] "tet(33)"
## [93] "aph(3')-IIa"
## [95] "erm(D)_clust"
## [97] "catS"
## [99] "aph(2')-Id_aph(2')-Ie"
## [101] "aadA13"
## [103] "aph(2')-Ig"
## [105] "aac(6')-Iih"
## [107] "mef(A)_clust"
## [109] "tet(X)_3_AB097942"
## [111] "dfrA16_clust"
## [113] "cat_clust"
## [115] "tet(Z)"
## [117] "dfrA5_dfrA30"
## [119] "aph(3')-III"
## [121] "aph(6)-Ic"
## [123] "aac(6')-Ib_aadA_ant(3')_clust"
## [125] "aac(6')-Iak"
## [127] "msr(A)"
## [129] "aac(3)-Iva"
## [131] "lnu(A)"
## [133] "tet(J)"
## [135] "tetB(P)"
## [137] "aadA4_aadA5"
## [139] "tet(S)"
## [141] "VanLXY"
## [143] "VanHAX"
## [145] "ant(9)-Ia-1"
## [147] "blaIND_clust3"
## [149] "blaOXA_clust5"
## [151] "fexB"
## [153] "erm(A)_1_X03216"
## [155] "lsa(A)"
## [157] "blaOXA-449_clust"
## [159] "tet(G)"
## [161] "cmx"
"vanXmurFvanKWI-2"
"qnrB_clust1"
"npmA"
"aph(3')-Ib"
"tet(44)"
"blaOXA-450_clust"
"msr(C)"
"blaOXA_clust11"
"erm(F)_clust"
"mecA_clust"
"VanC2-3-4_clust"
"erm(T)_4_AJ488494"
"aac(6')-Ii"
"tet(K)"
"blaOXA-493_clust"
"aac(6')-Iaj"
"aac(6')-Im"
"cfr(B)"
"tet(0)"
"lnu(B)"
"cml_clust"
"dfrK_clust"
"blaOXA-308"
"mef(C)"
"tet(X)_clust"
"poxtA"
"cat_3"
"lsa(B)"
"VanHBX"
"cat_2"
"catP"
"ARR_clust"
"VanA-vanH-vanHAX_clust"
"blaSED1"
"erm(T)_clust"
"erm(36)"
"mef(A)-3"
"mph(B)"
"ant(6)-Ia_clust1"
"dfrG"
"aac(3)-Ib"
"tet(L)_1_HM235948"
"aph(2')-Ib"
"blaZ-8-10"
"lnu(C)"
"aac(3)-I_aac(3)-Ia"
"cat(pC194)"
"lnu(G)"
"mef(B)"
"blaACT-9"
"mecA1"
"VanHDX_clust"
"vat(D)"
"blaPEDO-2"

```

```

## [163] "mdf(A)"                      "dfrA6_dfrA31"
## [165] "erm(34)"                      "aac(3)-VIa"
## [167] "aac(6')-aph(2')_clust"        "aac(6')-Iw"
## [169] "ant(6)-Ia_clust2"              "str"
## [171] "vgb(B)"                        "floR_clust"
## [173] "fmr0"                           "dfrA8"
## [175] "catA1"                          "blaIND-5"
## [177] "dfrD"                            "oqxB"
## [179] "cat(pC221)"                    "fosD"
## [181] "erm(Q)"                         "VanGXY"
## [183] "blaCGB"                         "aac(6')-Ian"
## [185] "catB"                            "aph(3')-VIIa"

# Plots of relative abundances, fixing some genes that are clustered in the data twice, showing top 12

# TPM and FPKM in the same plot, comparing non AB
dataset1 = ps_filter(Rps)
dataset2 = ps_filter(Rps_tpm)

dataset1 %<>% ps_mutate(dataset = "FPKM")
dataset2 %<>% ps_mutate(dataset = "TPM")

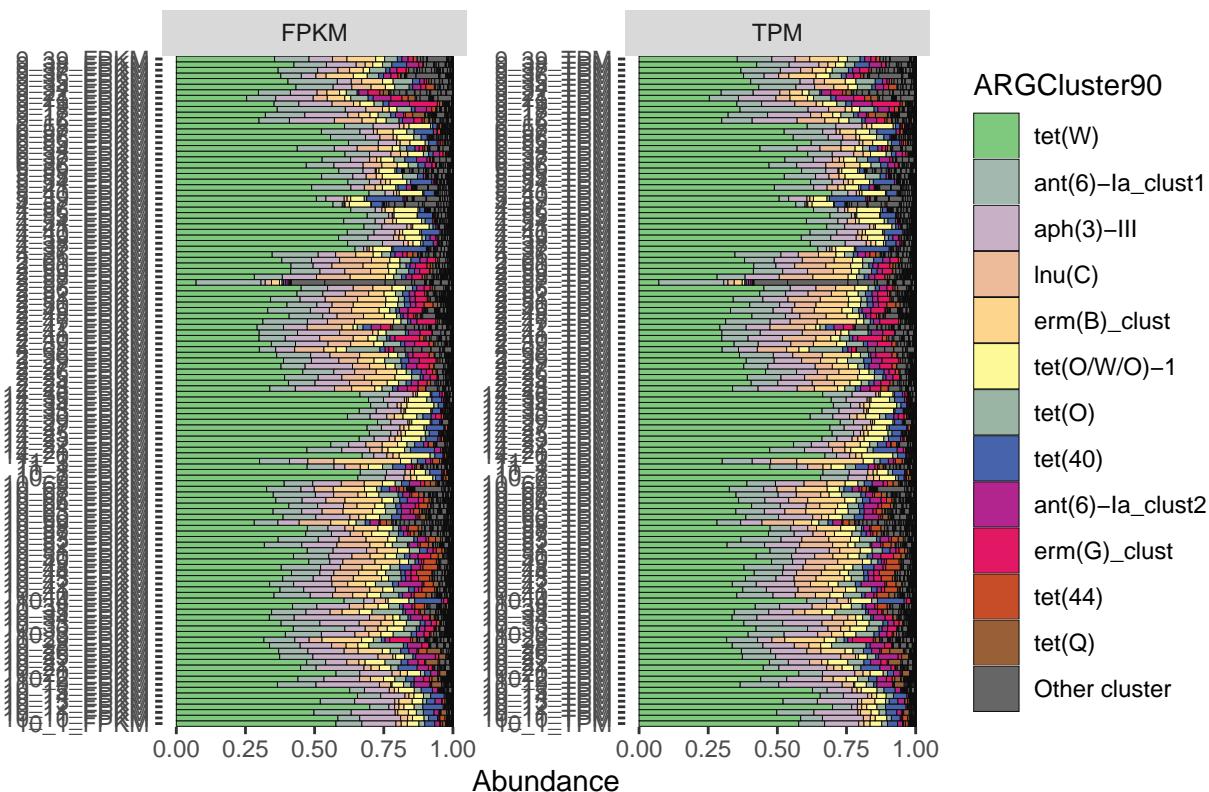
sample_names(dataset1) <- paste(sample_names(dataset1), "FPKM", sep="_")
sample_names(dataset2) <- paste(sample_names(dataset2), "TPM", sep="_")

combined <- phyloseq::merge_phyloseq(
  dataset1 %>% tax_fix(unknowns = c("blaOXA-493_clust", "cfr(B)", "dfrA16_clust", "dfrA7_dfrA17", "lnu(B"))
  dataset2 %>% tax_fix(unknowns = c("blaOXA-493_clust", "cfr(B)", "dfrA16_clust", "dfrA7_dfrA17", "lnu(B"))
)

combined %>% tax_fix(unknowns = c("blaOXA-493_clust", "cfr(B)", "dfrA16_clust", "dfrA7_dfrA17", "lnu(B"))
  comp_barplot("ARGCluster90", facet_by = "dataset", n_taxa = 12, palette = colorRampPalette(brewer.pal
    other_name = "Other cluster", merge_other = F, sample_order = "asis") +
  coord_flip() + ggtitle("FPKM vs TPM relative abundance of ARG clusters at 90% sequence identity")

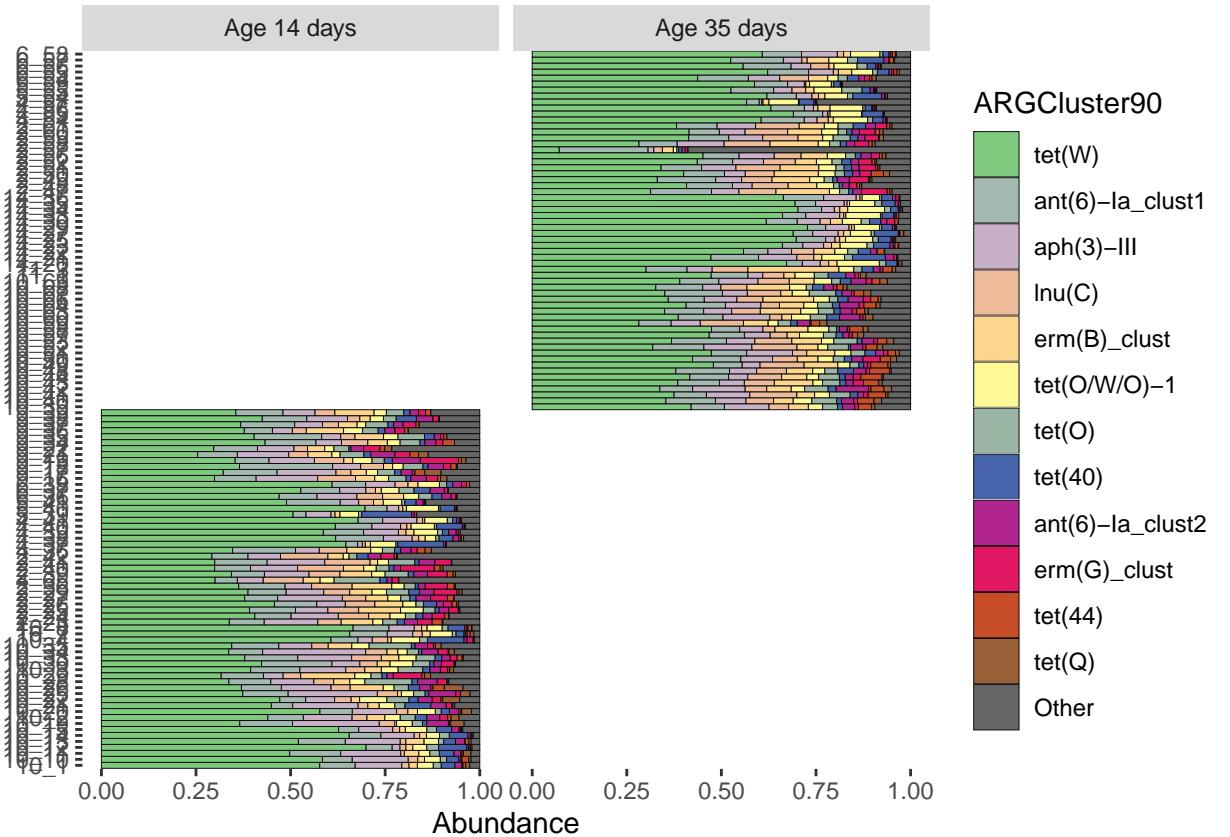
```

FPKM vs TPM relative abundance of ARG clusters at 90% sequence identity



```
# uitprobeersel
```

```
Rps_tpm %>% tax_fix(unknowns = c("lnu(B)", "cfr(B)", "blaOXA-493_clust", "dfrA16_clust", "dfrA7_dfrA17"))
  comp_barplot(tax_level = "ARGCluster90", n_taxa = 12, sample_order = "asis", palette = colorRampPalette
  facet_wrap(
    facets = vars(Age), labeller = as_labeller(~ paste("Age", ., "days"))),
    scales = "fixed") +
  coord_flip()
```



```
# Kraken 2 vs MetaPhlAn plot
```

```
dataset1 = ps_filter(Rps)
dataset2 = ps_filter(Rps_mp)

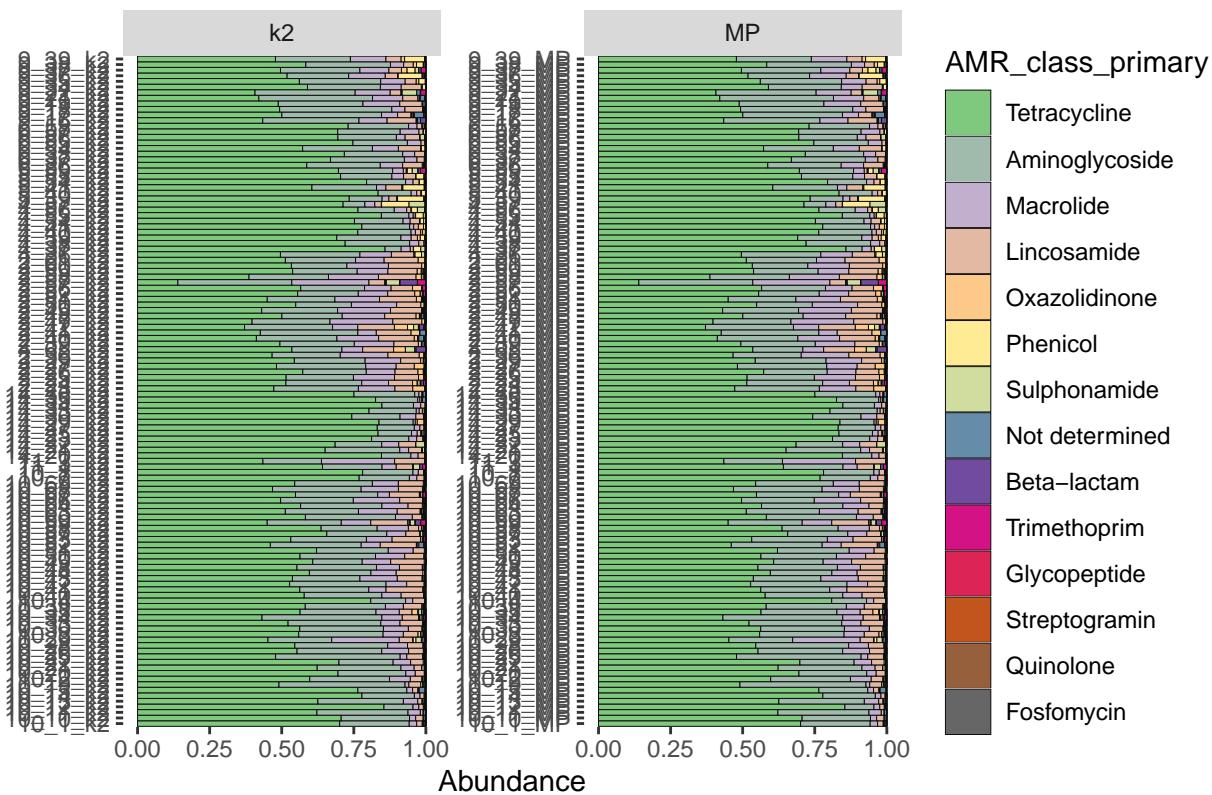
dataset1 %<>% ps_mutate(dataset = "k2")
dataset2 %<>% ps_mutate(dataset = "MP")

sample_names(dataset1) <- paste(sample_names(dataset1), "k2", sep = "_")
sample_names(dataset2) <- paste(sample_names(dataset2), "MP", sep = "_")

combined <- phyloseq::merge_phyloseq(
  dataset1 %>% tax_fix(unknowns = c("blaOXA-493_clust", "cfr(B)", "dfrA16_clust", "dfrA7_dfrA17", "lnu(B"))
  dataset2 %>% tax_fix(unknowns = c("blaOXA-493_clust", "cfr(B)", "dfrA16_clust", "dfrA7_dfrA17", "lnu(B"))
)

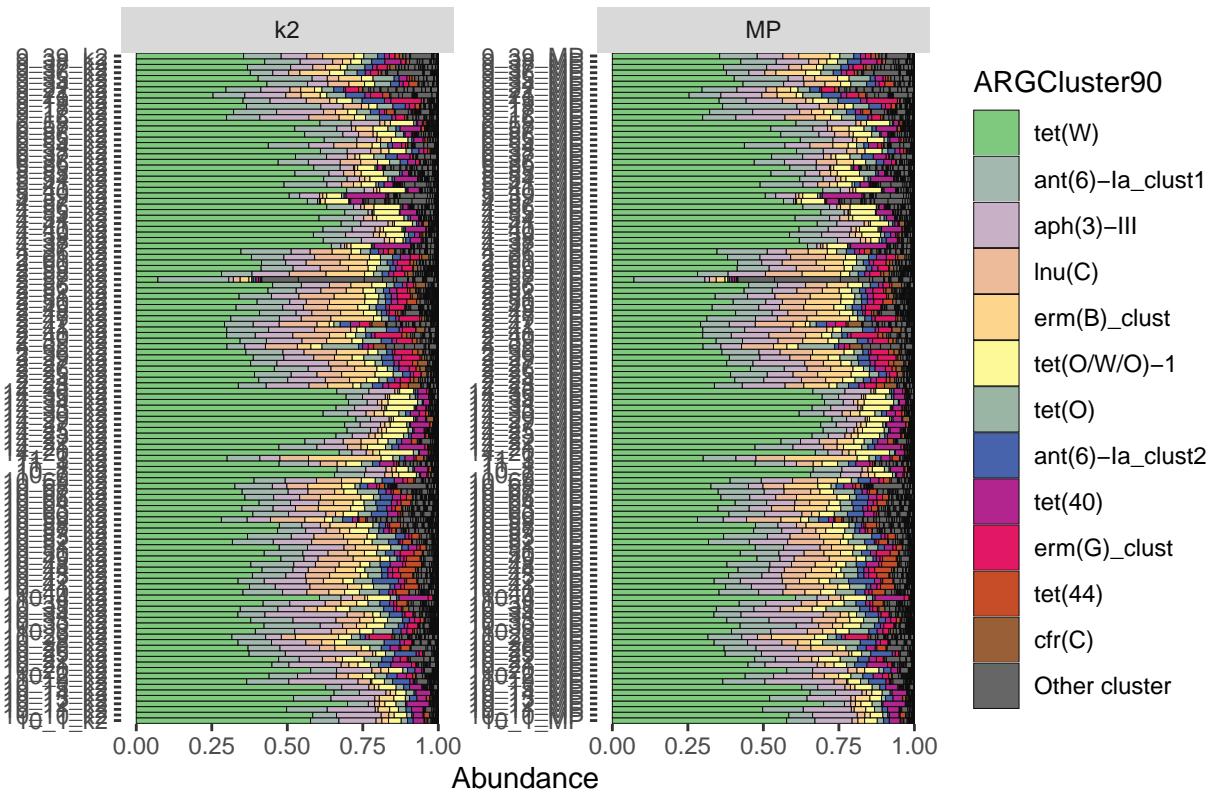
# primary class
combined %>% tax_fix(unknowns = c("blaOXA-493_clust", "cfr(B)", "dfrA16_clust", "dfrA7_dfrA17", "lnu(B"))
  comp_barplot("AMR_class_primary", facet_by = "dataset", n_taxa = 14, palette = colorRampPalette(brewer.pal(12, "Set1")))
  sample_order = "asis") +
  coord_flip() + ggtitle("Kraken2 vs Metaphlan relative abundance of primary AMR classes")
```

Kraken2 vs Metaphlan relative abundance of primary AMR classes



```
#Argclust90
combined %>% tax_fix(unknowns = c("blaOXA-493_clust", "cfr(B)", "dfrA16_clust", "dfrA7_dfrA17", "lnu(B")
  comp_barplot("ARGCluster90", facet_by = "dataset", n_taxa = 12, palette = colorRampPalette(brewer.pal
    other_name = "Other cluster", merge_other = F, sample_order = "asis") +
  coord_flip() + ggtitle("Kraken2 vs Metaphlan relative abundance of ARG clusters at 90% sequence ident
```

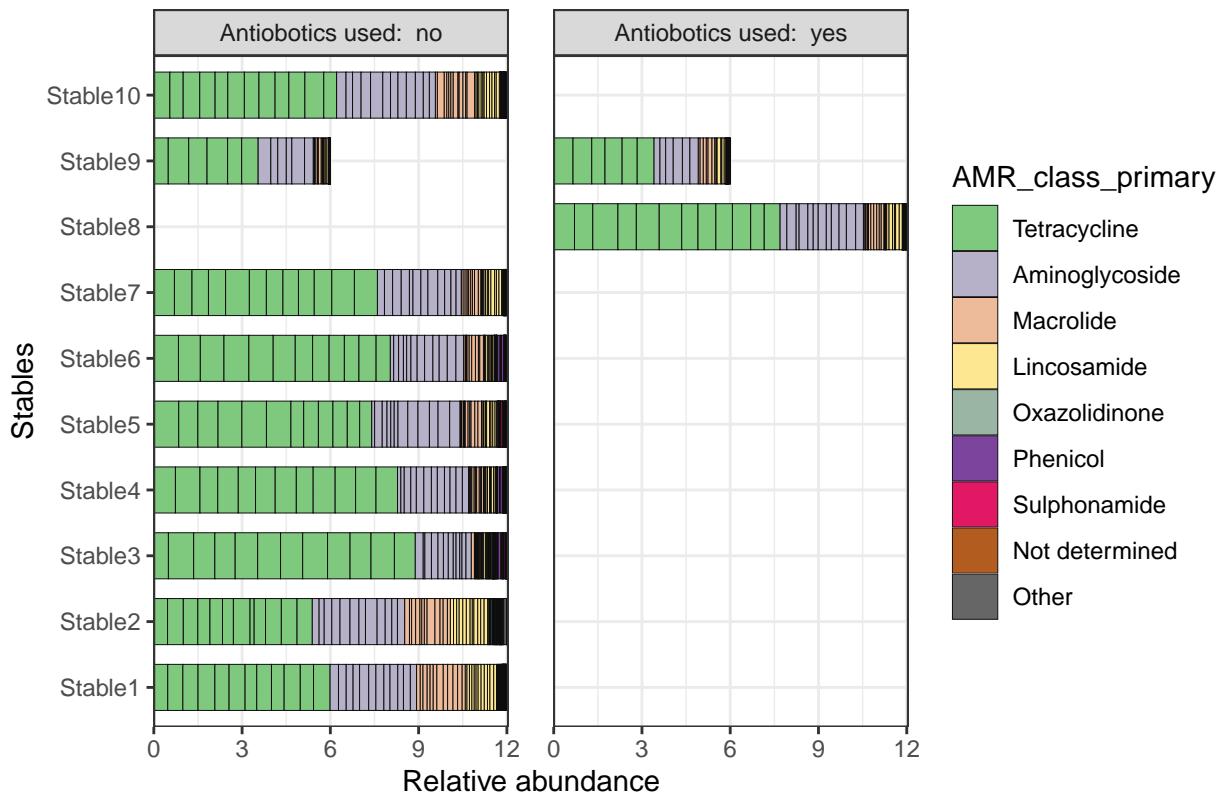
Kraken2 vs Metaphlan relative abundance of ARG clusters at 90% sequence



```
# Relative abundance for both stable and antibiotics used
```

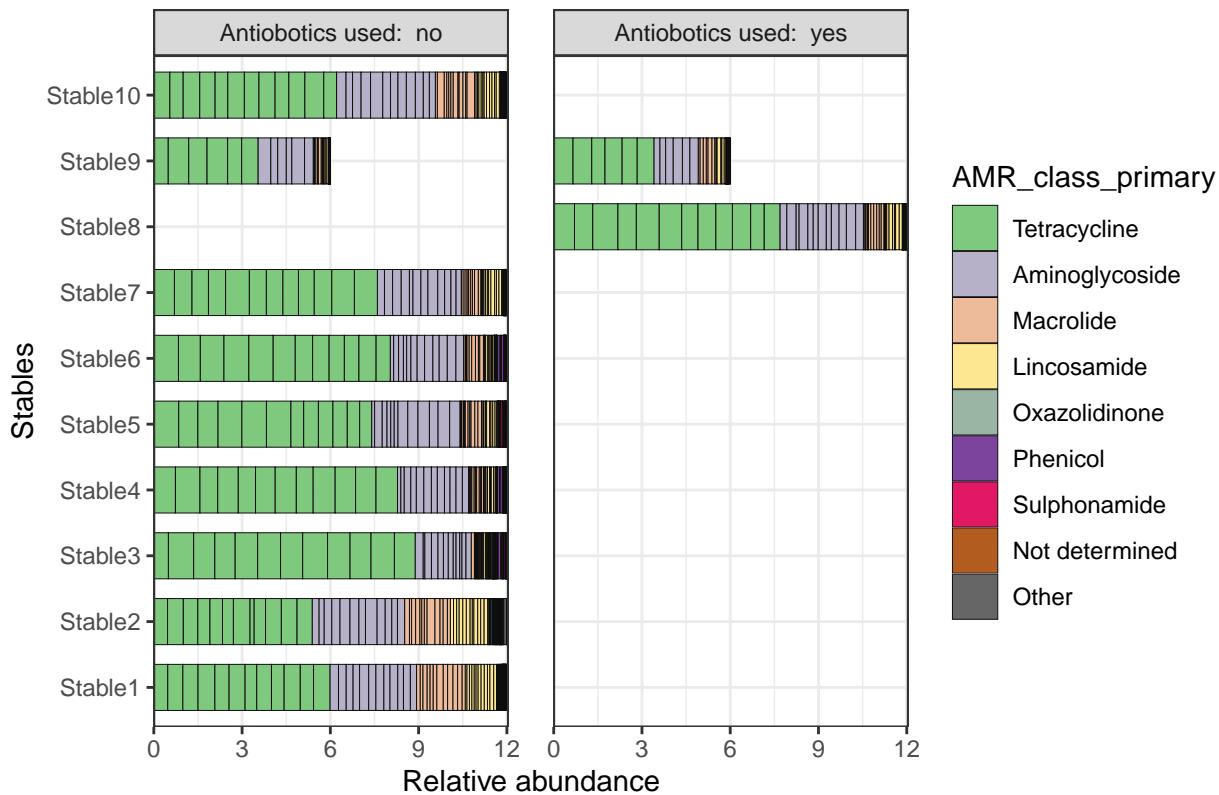
```
Rps %>% tax_fix(unknowns = c("lnu(B)", "cfr(B)", "blaOXA-493_clust", "dfrA16_clust", "dfrA7_dfrA17")) %>%
  ps_arrange(FarmRoundStable) %>%
  ps_mutate(
    FarmRoundStable = factor(FarmRoundStable, rev(unique(FarmRoundStable))))
  ) %>%
  comp_barplot(
    tax_level = "AMR_class_primary", bar_width = 0.7, sample_order = "asis",
    palette = colorRampPalette(brewer.pal(8, "Accent"))(9),
    x = "Stables") +
  facet_wrap(
    facets = vars(AB), labeller = as_labeller(~ paste("Antibiotics used: ", .)),
    scales = "fixed"
  ) +
  coord_flip() +
  labs(x = "Stables", y = "Relative abundance") +
  scale_y_continuous(expand = expansion(add = c(0, 0.05))) +
  theme_bw() +
  theme(panel.spacing.x = unit(6, "mm")) +
  ggtitle("Relative abundance of primary antimicrobial class by stable and antibiotics used (FPKM)")
```

Relative abundance of primary antimicrobial class by stable and antibiotic used



```
Rps_tpm %>% tax_fix(unknowns = c("lnu(B)", "cfr(B)", "blaOXA-493_clust", "dfrA16_clust", "dfrA7_dfrA17"))
  ps_arrange(FarmRoundStable) %>%
  ps_mutate(
    FarmRoundStable = factor(FarmRoundStable, rev(unique(FarmRoundStable)))
  ) %>%
  comp_barplot(
    tax_level = "AMR_class_primary", bar_width = 0.7, sample_order = "asis",
    palette = colorRampPalette(brewer.pal(8, "Accent"))(9),
    x = "Stables" +
  facet_wrap(
    facets = vars(AB), labeller = as_labeller(~ paste("Antibiotics used: ", .)),
    scales = "fixed"
  ) +
  coord_flip() +
  labs(x = "Stables", y = "Relative abundance") +
  scale_y_continuous(expand = expansion(add = c(0, 0.05))) +
  theme_bw() +
  theme(panel.spacing.x = unit(6, "mm")) +
  ggtitle("Relative abundance of primary antimicrobial class by stable and antibiotics used (TPM)"))
```

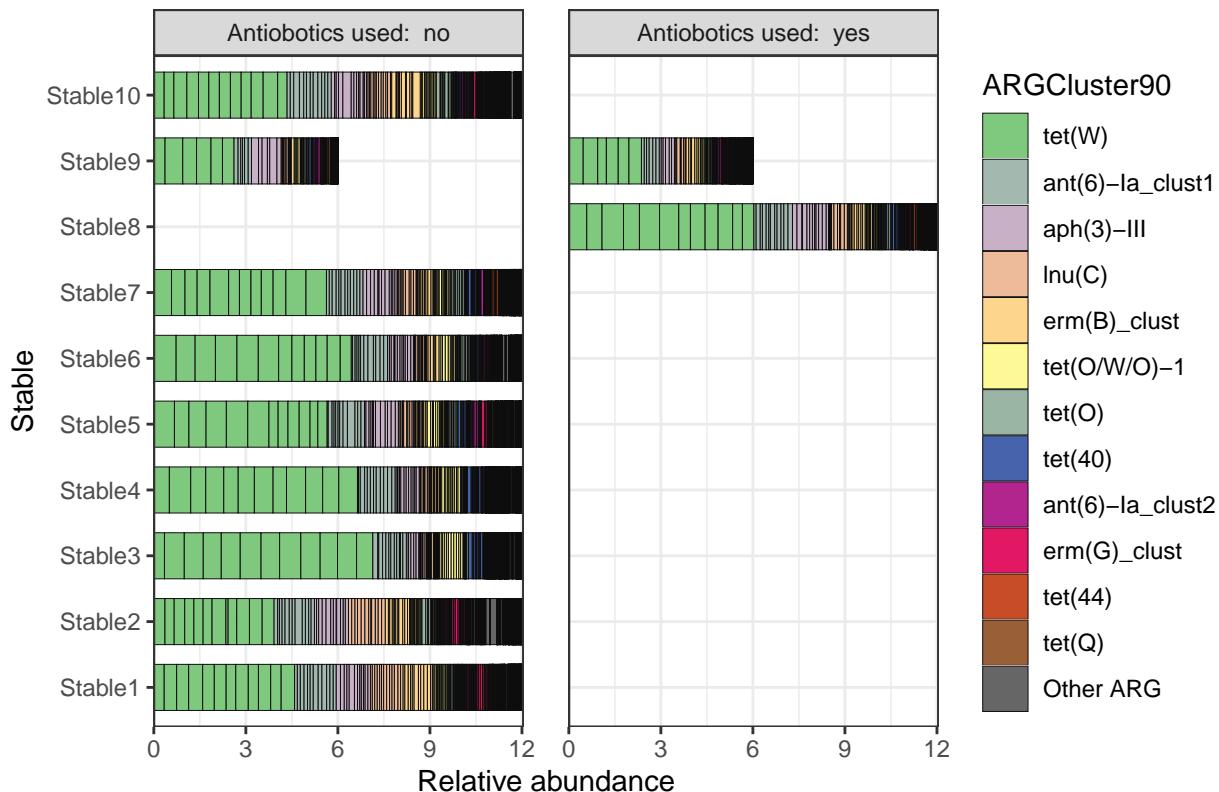
Relative abundance of primary antimicrobial class by stable and antibiotic used



Same plots but at the ARG level

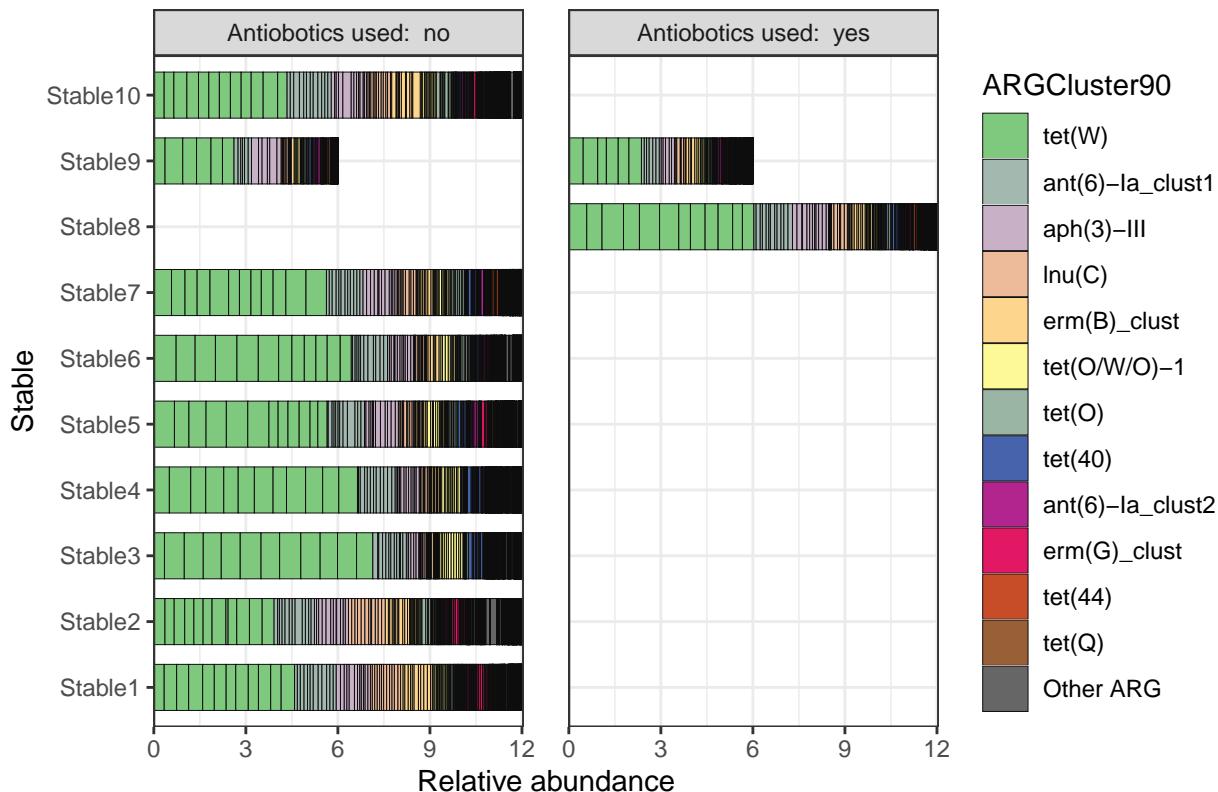
```
Rps %>% tax_fix(unknowns = c("lnu(B)", "cfr(B)", "blaOXA-493_clust", "dfrA16_clust", "dfrA7_dfrA17")) %>%
  ps_arrange(FarmRoundStable) %>%
  ps_mutate(
    FarmRoundStable = factor(FarmRoundStable, rev(unique(FarmRoundStable)))
  ) %>%
  comp_barplot(
    tax_level = "ARGCluster90", bar_width = 0.7, sample_order = "asis",
    palette = colorRampPalette(brewer.pal(8, "Accent"))(13),
    x = "Stables",
    n_taxa = 12, other_name = "Other ARG", merge_other = F) +
  facet_wrap(
    facets = vars(AB), labeller = as_labeller(~ paste("Antibiotics used: ", .)),
    scales = "fixed"
  ) +
  coord_flip() +
  labs(x = "Stable", y = "Relative abundance") +
  scale_y_continuous(expand = expansion(add = c(0, 0.05))) +
  theme_bw() +
  theme(panel.spacing.x = unit(6, "mm")) +
  ggtitle("Relative abundance of ARG by stable and antibiotics used (FPKM)")
```

Relative abundance of ARG by stable and antibiotics used (FPKM)



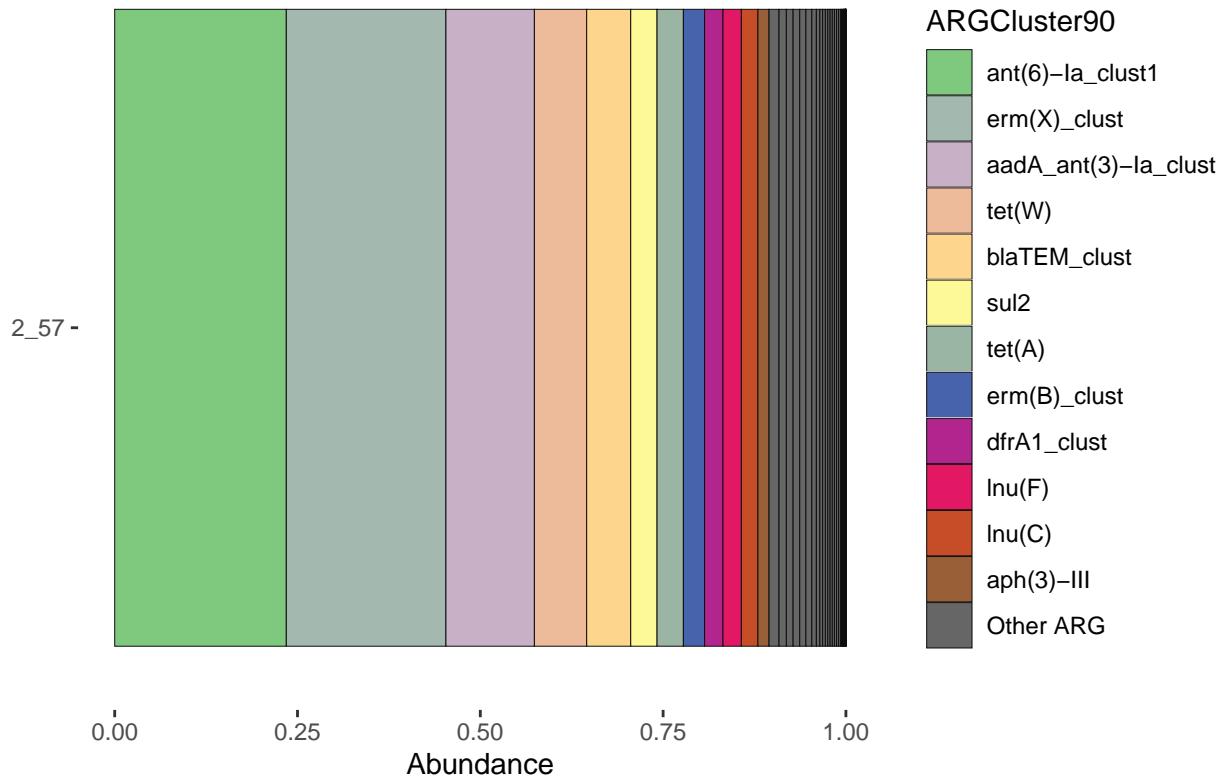
```
Rps_tpm %>% tax_fix(unknowns = c("lnu(B)", "cfr(B)", "blaOXA-493_clust", "dfrA16_clust", "dfrA7_dfrA17"))
  ps_arrange(FarmRoundStable) %>%
  ps_mutate(
    FarmRoundStable = factor(FarmRoundStable, rev(unique(FarmRoundStable)))
  ) %>%
  comp_barplot(
    tax_level = "ARGCluster90", bar_width = 0.7, sample_order = "asis",
    palette = colorRampPalette(brewer.pal(8, "Accent"))(13),
    x = "Stables",
    n_taxa = 12, other_name = "Other ARG", merge_other = F) +
  facet_wrap(
    facets = vars(AB), labeller = as_labeller(~ paste("Antibiotics used: ", .)),
    scales = "fixed"
  ) +
  coord_flip() +
  labs(x = "Stable", y = "Relative abundance") +
  scale_y_continuous(expand = expansion(add = c(0, 0.05))) +
  theme_bw() +
  theme(panel.spacing.x = unit(6, "mm")) +
  ggtitle("Relative abundance of ARG by stable and antibiotics used (TPM)")
```

Relative abundance of ARG by stable and antibiotics used (TPM)



```
# 2_57 is a big outlier, for some reason barely has tet(W) and other common ARG clusters, and therefore
```

```
Rps %>% tax_fix(unknowns = c("lnu(B)", "cfr(B)")) %>% ps_filter(SampleID == "7.F1S2.21.09") %>% ps_calc_o  
comp_barplot(tax_level = "ARGCluster90", n_taxa = 12, palette = colorRampPalette(brewer.pal(8, "Accent"),  
other_name = "Other ARG", merge_other = F) +  
coord_flip()
```



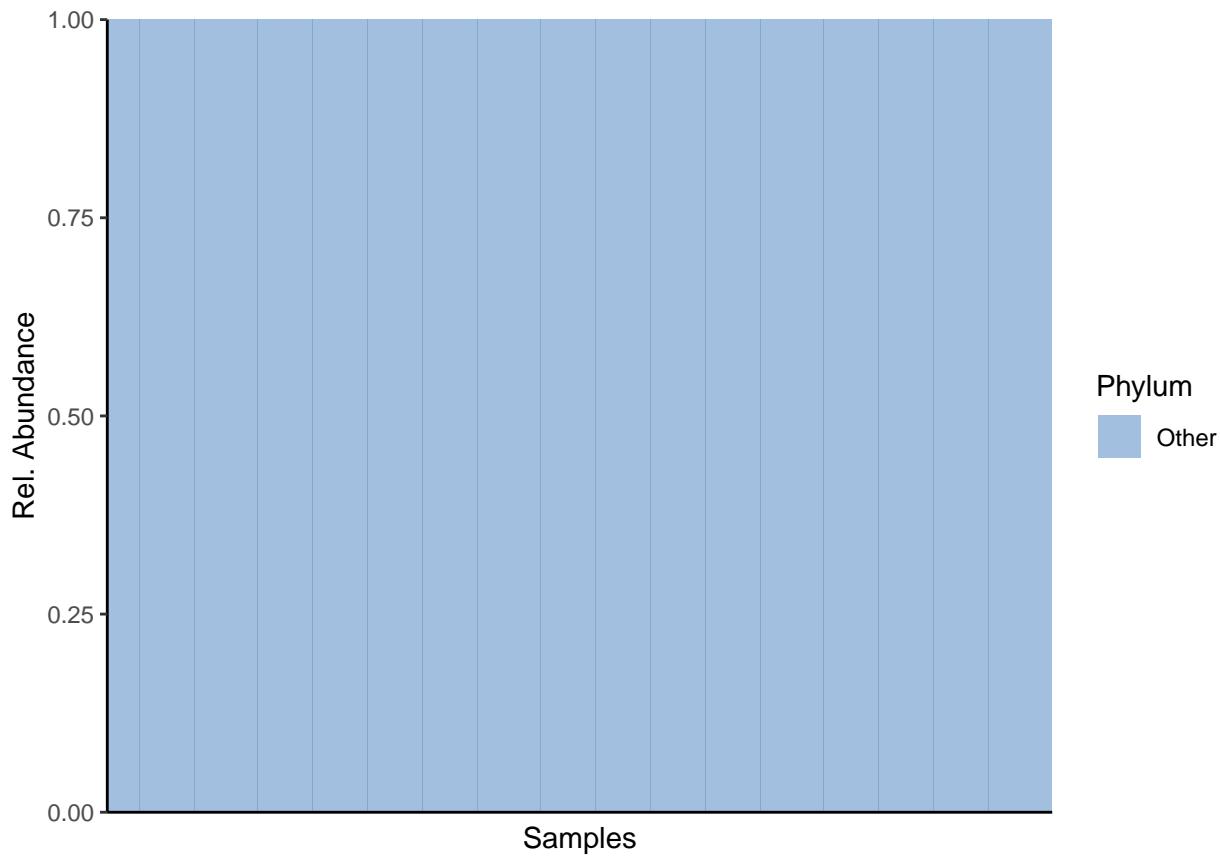
```
# rel abundance on primary AMR class level

tse = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)

tse <- transformCounts(tse, method = "relabundance")

tse_AMRClass <- agglomerateByRank(tse,
                                      rank = "Phylum",
                                      onRankOnly = TRUE)
tse_AMRClass <- transformCounts(tse,
                                 assay.type = "counts",
                                 method = "relabundance")
top_taxa <- getTopTaxa(tse_AMRClass,
                        top = 10,
                        assay.type = "relabundance")
AMRClass_renames <- lapply(rowData(tse_AMRClass)$Phylum,
                            function(x){if (x %in% top_taxa) {x} else {"Other"}})
rowData(tse_AMRClass)$Phylum <- as.character(AMRClass_renames)

# rel abundance figures, can order by specific taxa
miaViz::plotAbundance(tse_AMRClass,
                       assay.type = "relabundance",
                       rank = "Phylum",
                       order_rank_by = "abund")
```



```
tse_AMRClass$Farm2 = as.factor(tse_AMRClass$Farm2)
tse_AMRClass$AB = as.factor(tse_AMRClass$AB)

# Add AB plot on top

plots <- miaViz:::plotAbundance(tse_AMRClass,
                                    assay.type = "relabundance",
                                    rank = "Phylum",
                                    order_rank_by = "abund",
                                    #                                         order_sample_by = "o_Clostridiales",
                                    order_sample_by = "AB",
                                    features = "AB")

plots[[1]] <- plots[[1]] +
  theme(legend.key.size = unit(0.3, 'cm'),
        legend.text = element_text(size = 6),
        legend.title = element_text(size = 8))
plots[[2]] <- plots[[2]] +
  theme(legend.key.height = unit(0.3, 'cm'),
        legend.key.width = unit(0.3, 'cm'),
        legend.text = element_text(size = 6),
        legend.title = element_text(size = 8),
        legend.direction = "vertical")

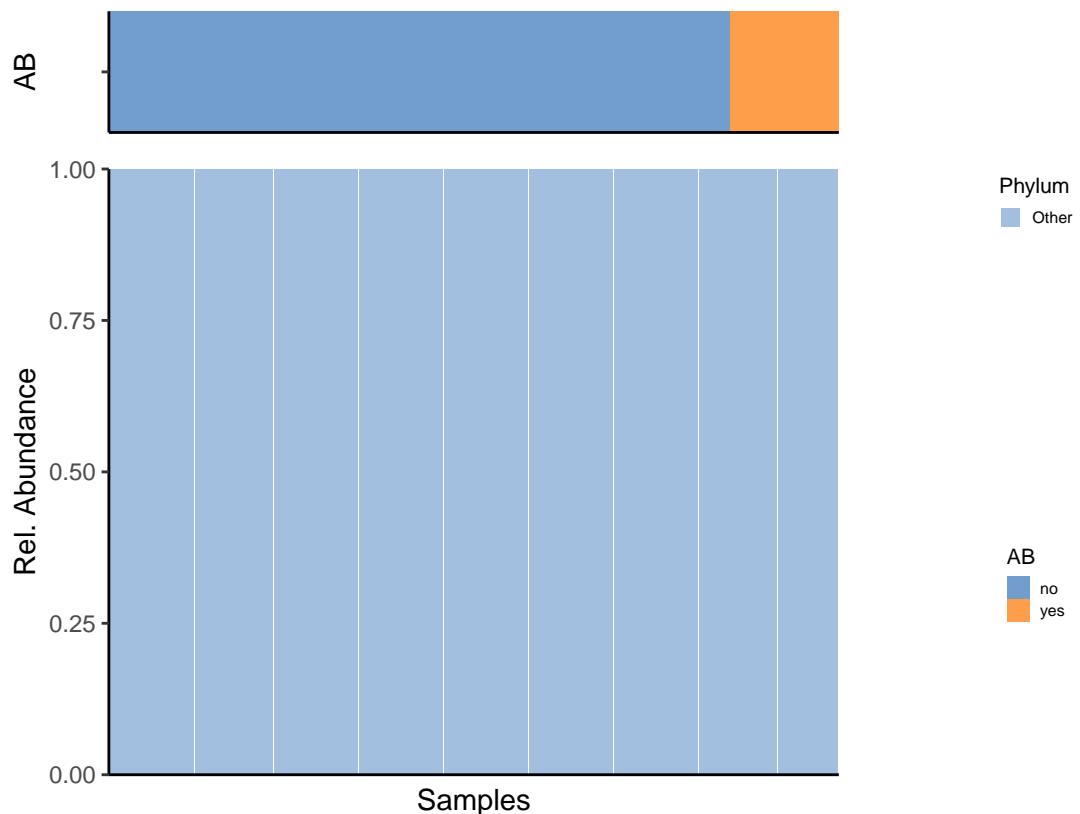
legend <- wrap_plots(as_ggplot(get_legend(plots[[1]])), as_ggplot(get_legend(plots[[2]])), ncol = 1)
plots[[1]] <- plots[[1]] + theme(legend.position = "none")
```

```

plots[[2]] <- plots[[2]] + theme(legend.position = "none", axis.title.x=element_blank())

plot <- wrap_plots(plots[[2]], plots[[1]], ncol = 1, heights = c(2, 10))
wrap_plots(plot, legend, nrow = 1, widths = c(2, 1))

```



heatmaps on phylum level

```

tse_AMRClass <- agglomerateByRank(tse,
                                    rank = "Phylum",
                                    onRankOnly = TRUE)

tse_AMRClass <- transformCounts(tse_AMRClass, MARGIN = "samples", method = "clr", assay.type = "counts"
tse_AMRClass <- transformCounts(tse_AMRClass, assay.type = "clr",
                                 MARGIN = "features",
                                 method = "z", name = "clr_z")

top_taxa <- getTopTaxa(tse_AMRClass, top = 11)
tse_AMRClass <- tse_AMRClass[top_taxa, ]

# Phylum AB heatmap
tse_AMRClass@metadata$anno_colors$AB = c(yes = "darkred", no = "darkblue")

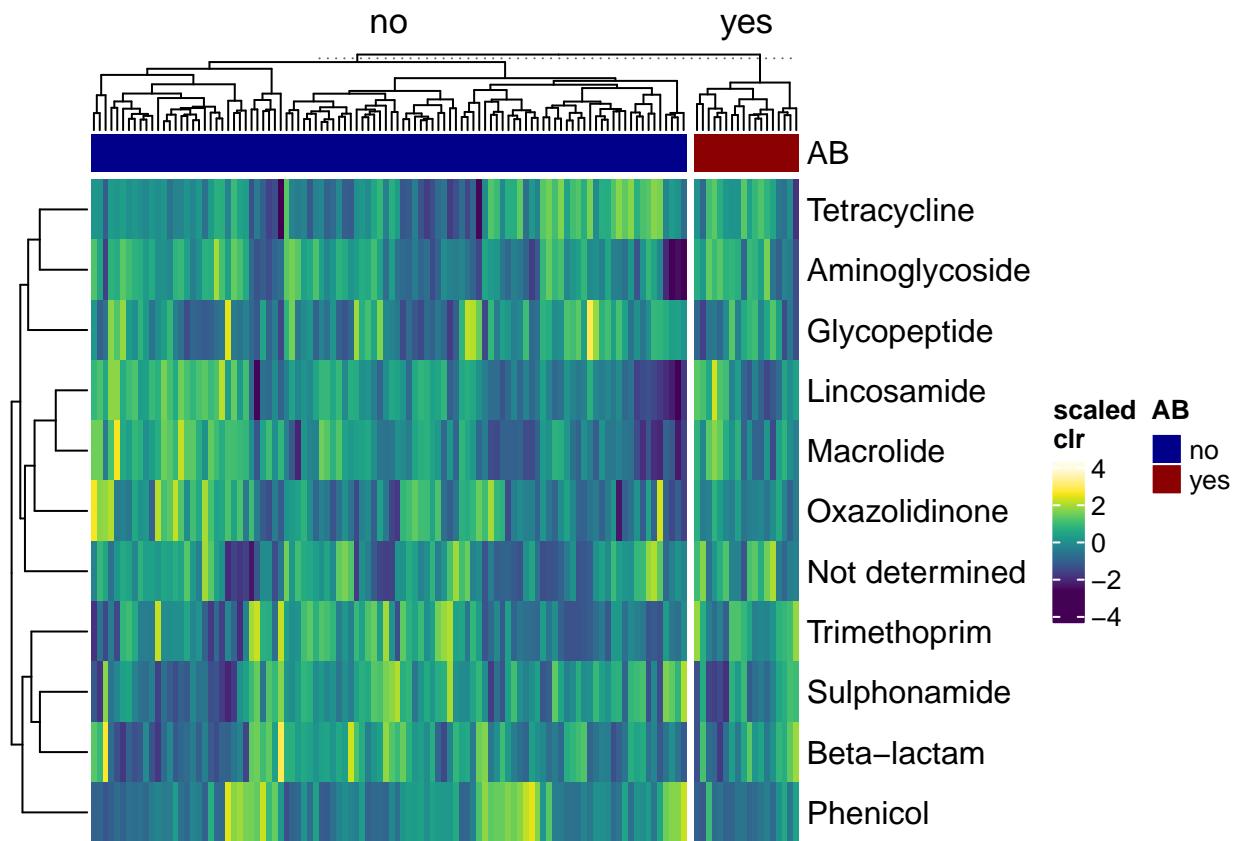
sechm(tse_AMRClass,
       features = rownames(tse_AMRClass),

```

```

assayName = "clr",
do.scale = TRUE,
top_annotation = c("AB"),
gaps_at = "AB",
hmcols = viridis(256),
cluster_cols = TRUE, cluster_rows = TRUE)

```

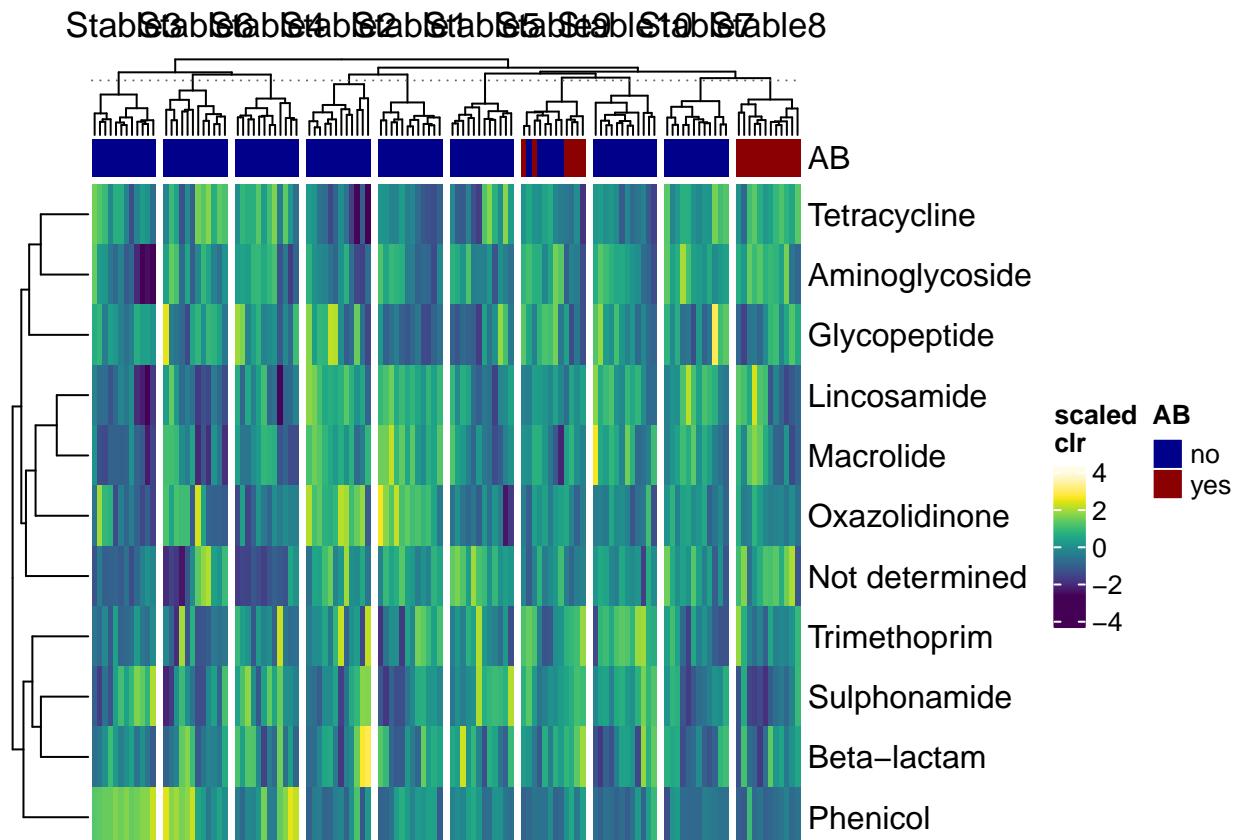


```

# heatmap with AB and stable

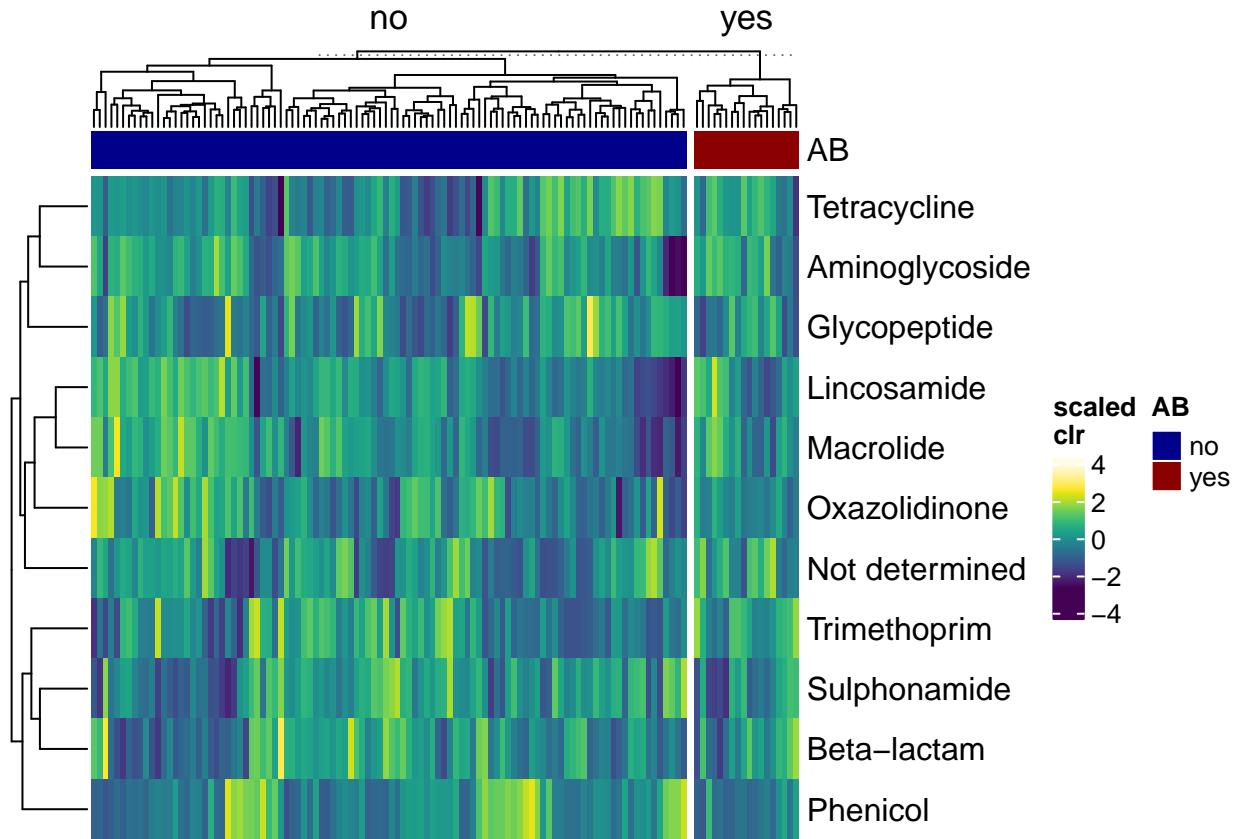
sechm(tse_AMRClass,
      features = rownames(tse_AMRClass),
      assayName = "clr",
      do.scale = TRUE,
      top_annotation = c("AB"),
      gaps_at = "Stables",
      hmcols = viridis(256),
      cluster_cols = TRUE, cluster_rows = TRUE)

```



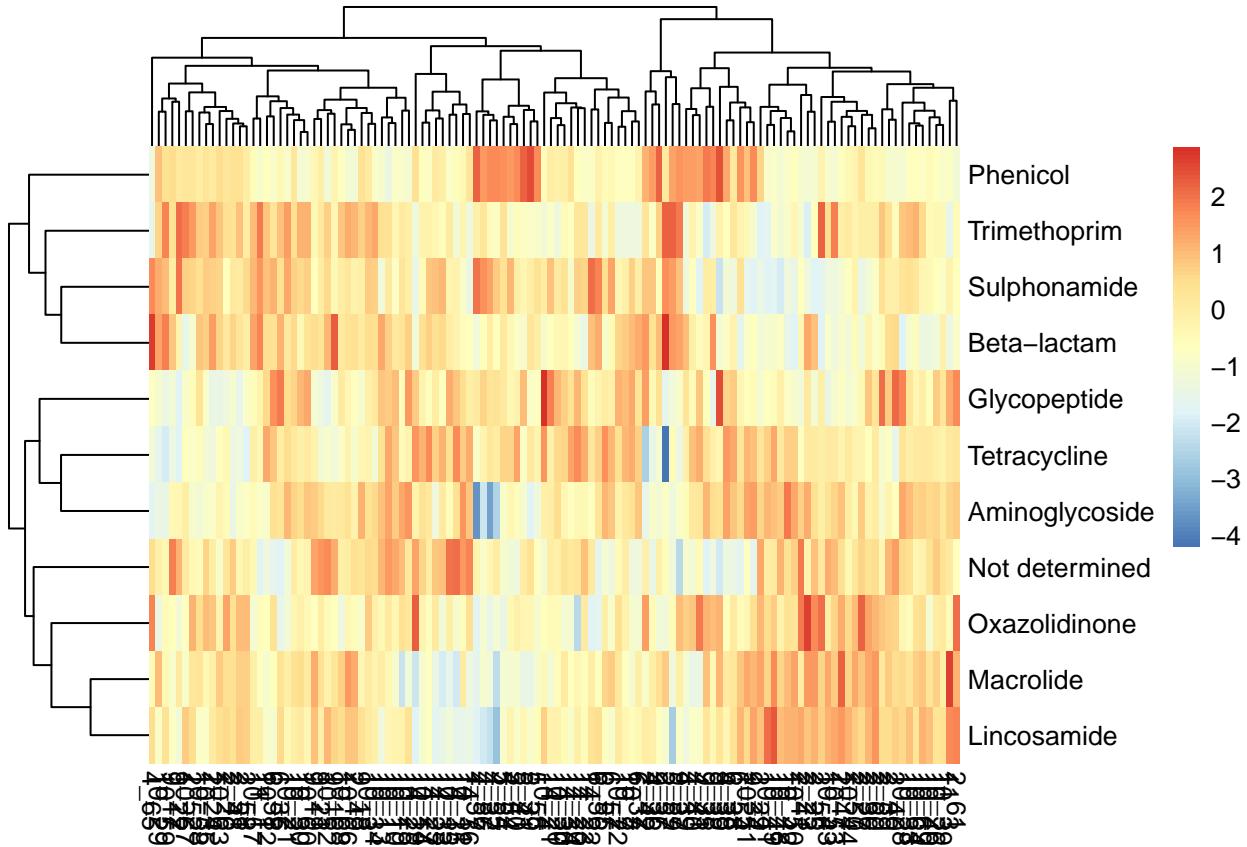
```
tse_AMRClass@metadata$anno_colors$Stable = (brewer.pal(n=10, name = "Set3"))
```

```
sechm(tse_AMRClass,
  features = rownames(tse_AMRClass),
  assayName = "clr",
  do.scale = TRUE,
  top_annotation = c("AB"),
  gaps_at = "AB",
  hmcols = viridis(256),
  cluster_cols = TRUE, cluster_rows = TRUE)
```



```
# Phylum heatmap
mat <- assay(tse_AMRClass, "clr_z")

pheatmap(mat)
```



```

# Phylum heatmap hierachal clustering with AB

# Clustering both samples and features hierarchically

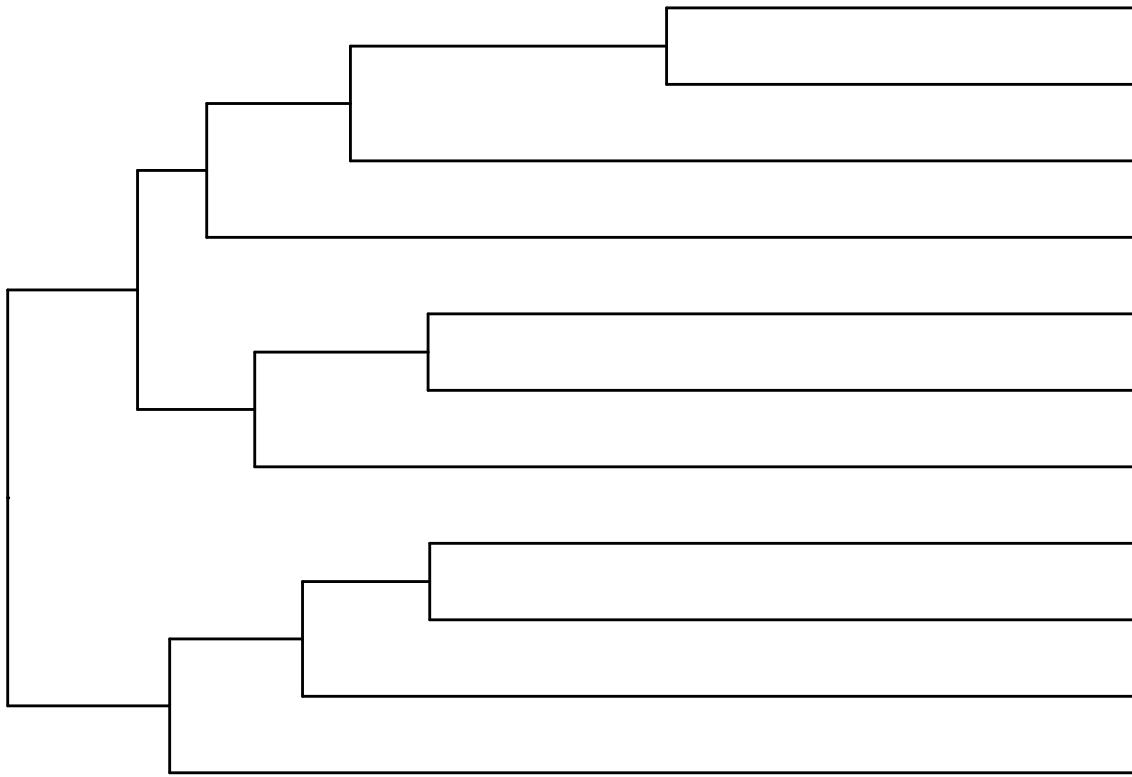
taxa_hclust <- hclust(dist(mat), method = "complete")

# Creates a phylogenetic tree
taxa_tree <- as.phylo(taxa_hclust)

# Plot taxa tree
taxa_tree <- ggtree(taxa_tree) +
  theme(plot.margin=margin(0,0,0,0)) # removes margins

taxa_tree # based on this three, we'll create two clusters

```



```

# Get order of taxa in plot
taxa_ordered <- get_taxa_name(taxa_tree)

taxa_clusters <- cutree(tree = taxa_hclust, k = 2) # 2 clusters based on tree figure

# Converts into data frame
taxa_clusters <- data.frame(clusters = taxa_clusters)
taxa_clusters$clusters <- factor(taxa_clusters$clusters)

# Order data so that it's same as in phylo tree
taxa_clusters <- taxa_clusters[taxa_ordered, , drop = FALSE]

```

```

# Prints taxa and their clusters
taxa_clusters

```

##	clusters
## Lincosamide	1
## Macrolide	1
## Oxazolidinone	1
## Not determined	1
## Aminoglycoside	1
## Tetracycline	1
## Glycopeptide	1
## Beta-lactam	2
## Sulphonamide	2
## Trimethoprim	2

```

## Phenicol          2

#rowData(tse_AMRClass)$clusters <- taxa_clusters[order(match(rownames(taxa_clusters), rownames(tse_AMRC

# Prints taxa and their clusters
rowData(tse_AMRClass)$clusters

## NULL

sample_hclust <- hclust(dist(t(mat)), method = "complete")

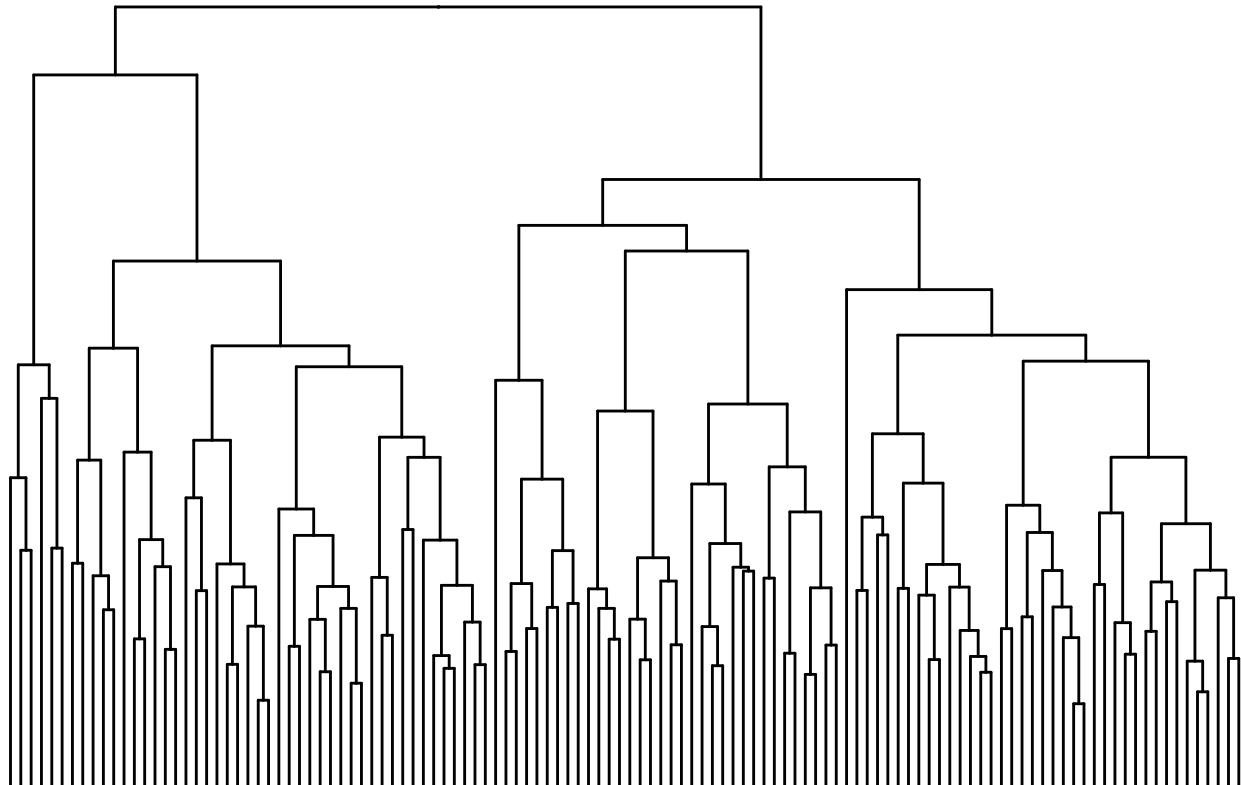
# Creates a phylogenetic tree
sample_tree <- as.phylo(sample_hclust)

# Plot sample tree
sample_tree <- ggtree(sample_tree) + layout_dendrogram() +
  theme(plot.margin=margin(0,0,0,0)) # removes margins

# Get order of samples in plot
samples_ordered <- rev(get_taxa_name(sample_tree))

# to view the tree, run
sample_tree

```



```

# Creates clusters
sample_clusters <- factor(cutree(tree = sample_hclust, k = 2)) # 2 clusters based on methods in Cluster

# Converts into data frame
sample_data <- data.frame(clusters = sample_clusters)

# Order data so that it's same as in phylo tree
sample_data <- sample_data[samples_ordered, , drop = FALSE]

# Order data based on
tse_AMRClass <- tse_AMRClass[ , rownames(sample_data)]

# Add sample type data
sample_data$sample_types <- colData(tse_AMRClass)$AB

sample_data

```

	clusters	sample_types
## 2_42	2	no
## 4_36	2	no
## 9_36	2	no
## 2_57	2	no
## 5_59	2	no
## 9_37	2	no
## 9_35	2	no
## 9_38	2	no
## 4_39	2	no
## 4_40	2	no
## 4_41	2	no
## 9_39	2	no
## 4_38	2	no
## 6_38	2	no
## 6_54	2	no
## 10_51	2	yes
## 9_34	2	no
## 2_41	2	no
## 2_23	2	no
## 2_25	2	no
## 2_29	2	no
## 10_41	2	no
## 10_49	2	yes
## 10_48	2	yes
## 10_42	2	no
## 10_50	2	yes
## 2_56	2	no
## 10_43	2	no
## 10_53	2	yes
## 2_47	2	no
## 10_44	2	no
## 2_50	2	no
## 2_27	2	no
## 2_36	2	no
## 2_60	2	no

## 2_59	2	no
## 2_39	2	no
## 2_40	2	no
## 11_3	2	no
## 2_61	2	no
## 10_28	2	no
## 10_33	2	no
## 10_64	2	yes
## 10_69	2	no
## 10_40	2	no
## 10_26	2	no
## 10_39	2	no
## 14_29	1	no
## 10_14	1	yes
## 14_27	1	no
## 14_25	1	no
## 14_33	1	no
## 10_15	1	yes
## 14_35	1	no
## 10_11	1	yes
## 14_36	1	no
## 4_57	1	no
## 4_56	1	no
## 4_37	1	no
## 4_55	1	no
## 5_54	1	no
## 4_54	1	no
## 5_40	1	no
## 5_41	1	no
## 5_39	1	no
## 5_55	1	no
## 10_4	1	no
## 14_21	1	no
## 10_20	1	no
## 14_30	1	no
## 14_34	1	no
## 14_20	1	no
## 14_23	1	no
## 14_22	1	no
## 6_56	1	no
## 6_55	1	no
## 6_57	1	no
## 10_22	1	no
## 10_7	1	no
## 10_8	1	no
## 6_37	1	no
## 4_65	1	no
## 10_29	1	no
## 10_59	1	yes
## 9_17	1	no
## 9_22	1	no
## 10_67	1	no
## 2_52	1	no
## 10_58	1	yes

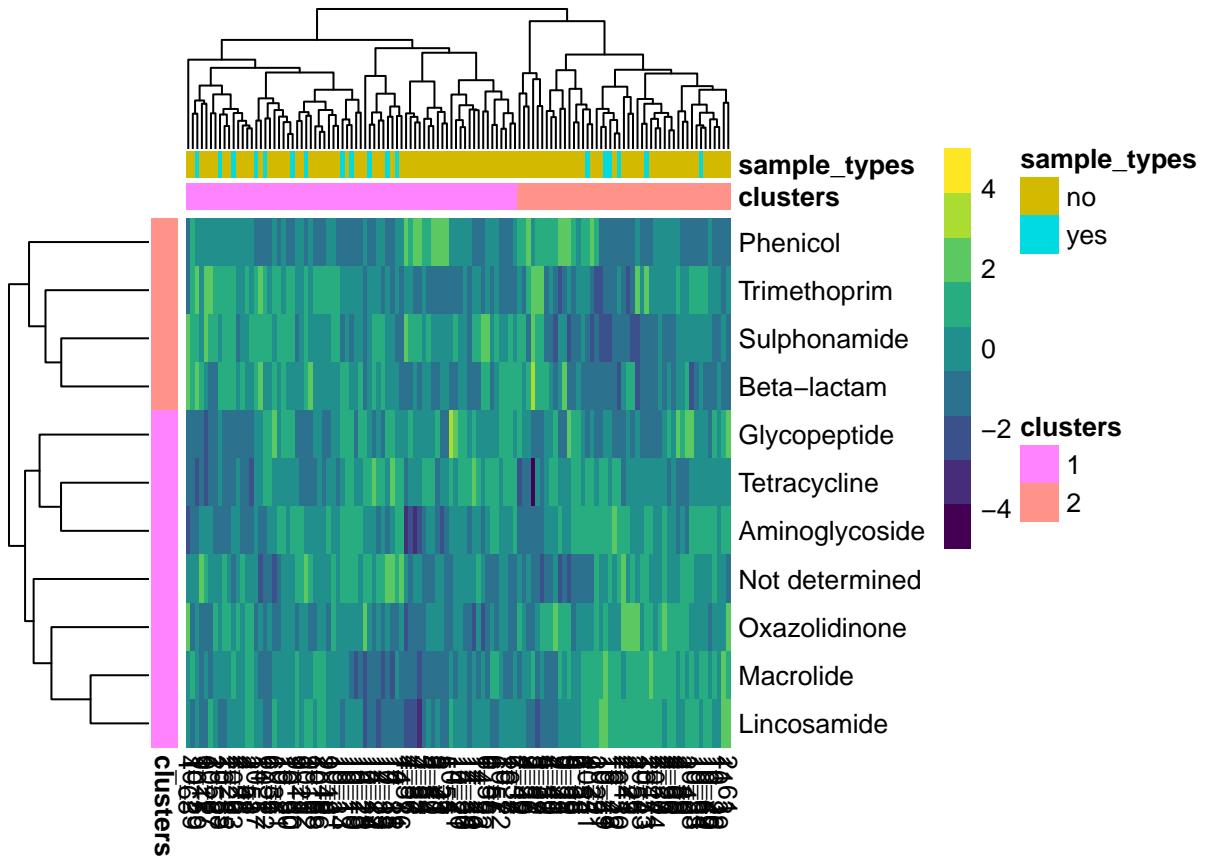
```

## 10_35      1      no
## 2_24      1      no
## 10_63      1     yes
## 2_26      1      no
## 2_48      1      no
## 2_51      1      no
## 2_58      1      no
## 10_57      1     yes
## 11_1       1      no
## 10_12      1     yes
## 6_58       1      no
## 6_36       1      no
## 10_21      1      no
## 10_2       1      no
## 10_30      1      no
## 10_60      1     yes
## 10_19      1      no
## 10_25      1      no
## 10_13      1     yes
## 10_1       1      no
## 10_10      1     yes
## 9_19       1      no
## 9_21       1      no
## 10_52      1     yes
## 9_16       1      no
## 10_68      1      no
## 10_66      1      no
## 2_49       1      no
## 9_18       1      no
## 10_3       1      no
## 10_34      1      no

breaks <- seq(-ceiling(max(abs(mat))), ceiling(max(abs(mat))),
               length.out = ifelse( max(abs(mat))>5, 2*ceiling(max(abs(mat))), 10 ) )
#colors <- colorRampPalette(c("darkblue", "blue", "white", "red", "darkred"))(length(breaks)-1) replace

pheatmap(mat, annotation_row = taxa_clusters,
         annotation_col = sample_data,
         breaks = breaks,
         color = colorRampPalette(viridis(256))(length(breaks)-1))

```

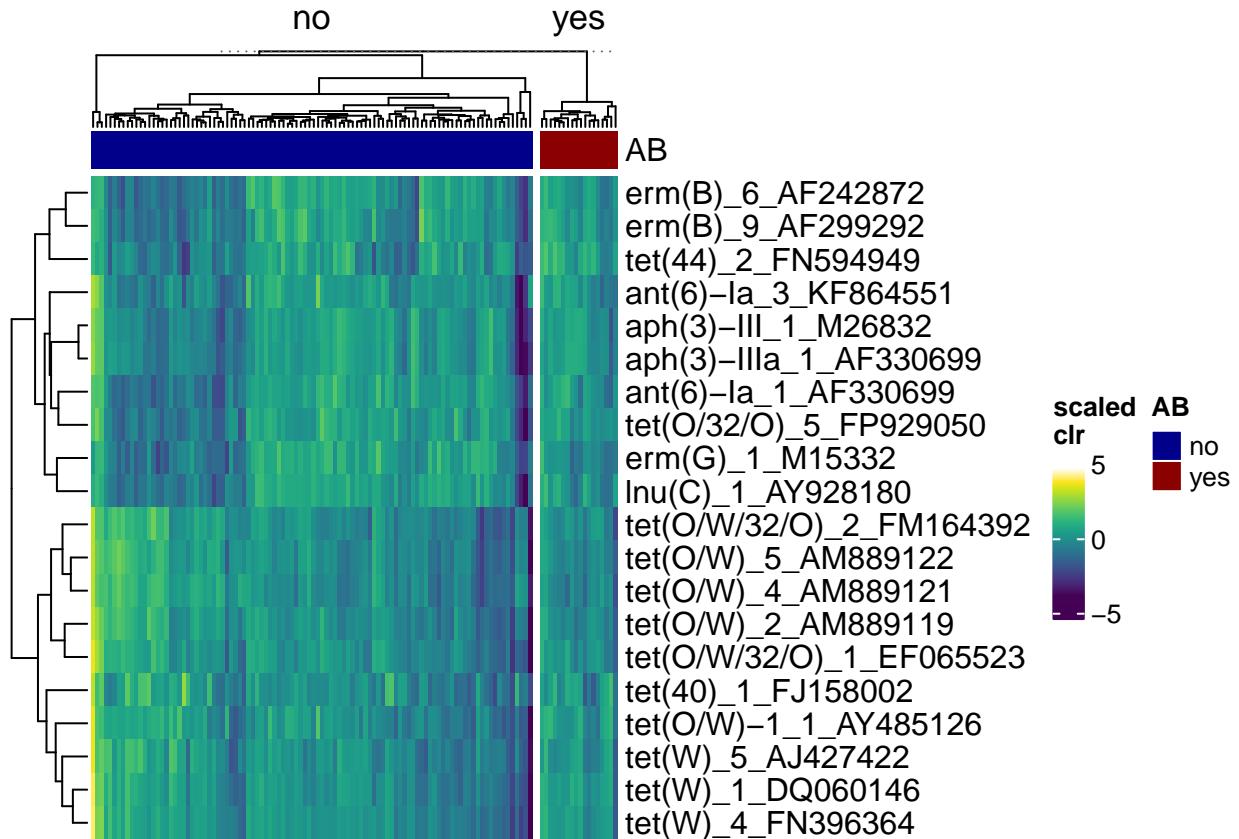


heatmaps on ARG level

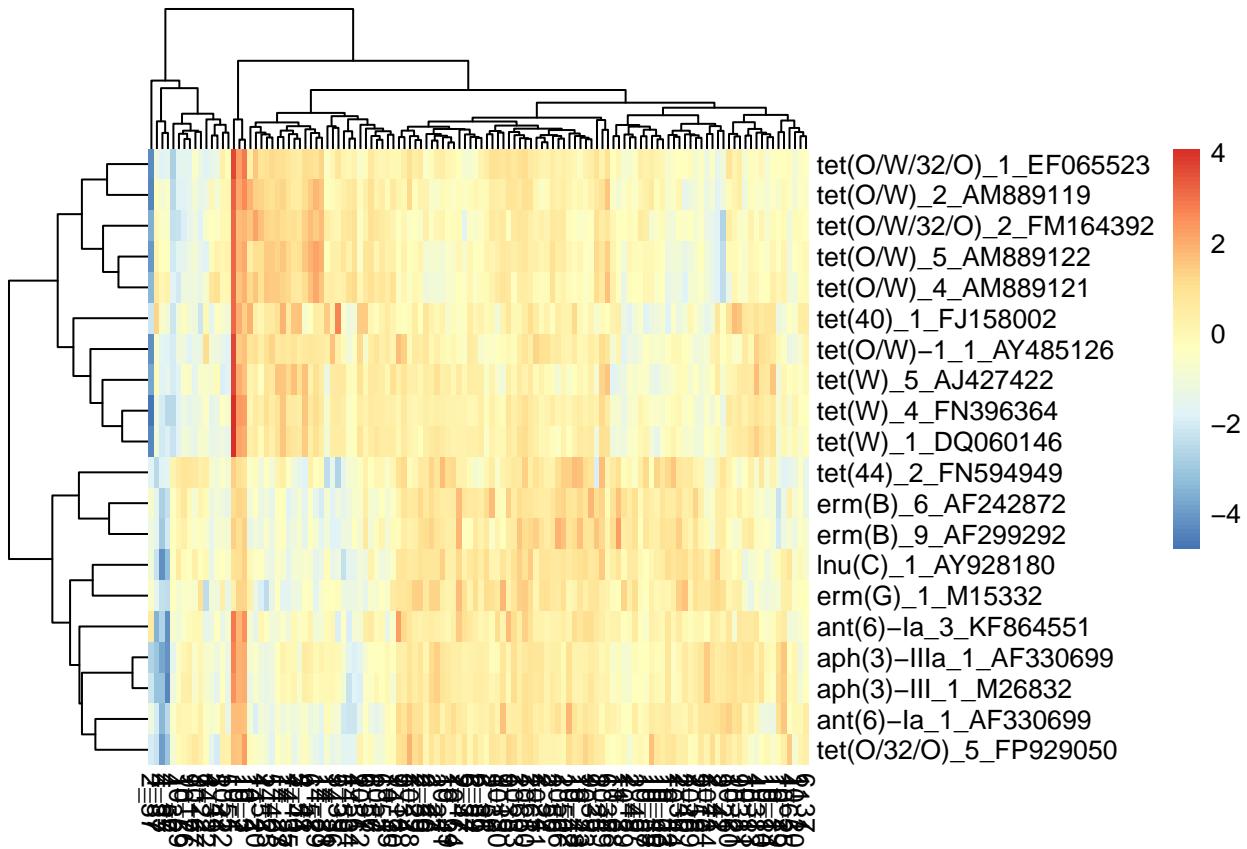
```
tse = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)
tse <- transformCounts(tse, method = "relabundance")
tse <- transformCounts(tse, MARGIN = "samples", method = "clr", assay.type = "counts", pseudocount=1)
tse <- transformCounts(tse, assay.type = "clr",
                      MARGIN = "features",
                      method = "z", name = "clr_z")
top_taxa <- getTopTaxa(tse, top = 20)
tse <- tse[top_taxa, ]

# ARG heatmap AB
tse@metadata$anno_colors$AB = c(yes = "darkred", no = "darkblue")

sechm(tse,
       features = rownames(tse),
       assayName = "clr",
       do.scale = TRUE,
       top_annotation = c("AB"),
       gaps_at = "AB",
       hmcols = viridis(256),
       cluster_cols = TRUE, cluster_rows = TRUE)
```



```
# ARG heatmap
mat <- assay(tse, "clr_z")
pheatmap(mat)
```



```

# ARG heatmap hierachal clustering with AB

# Clustering both samples and features hierarchically

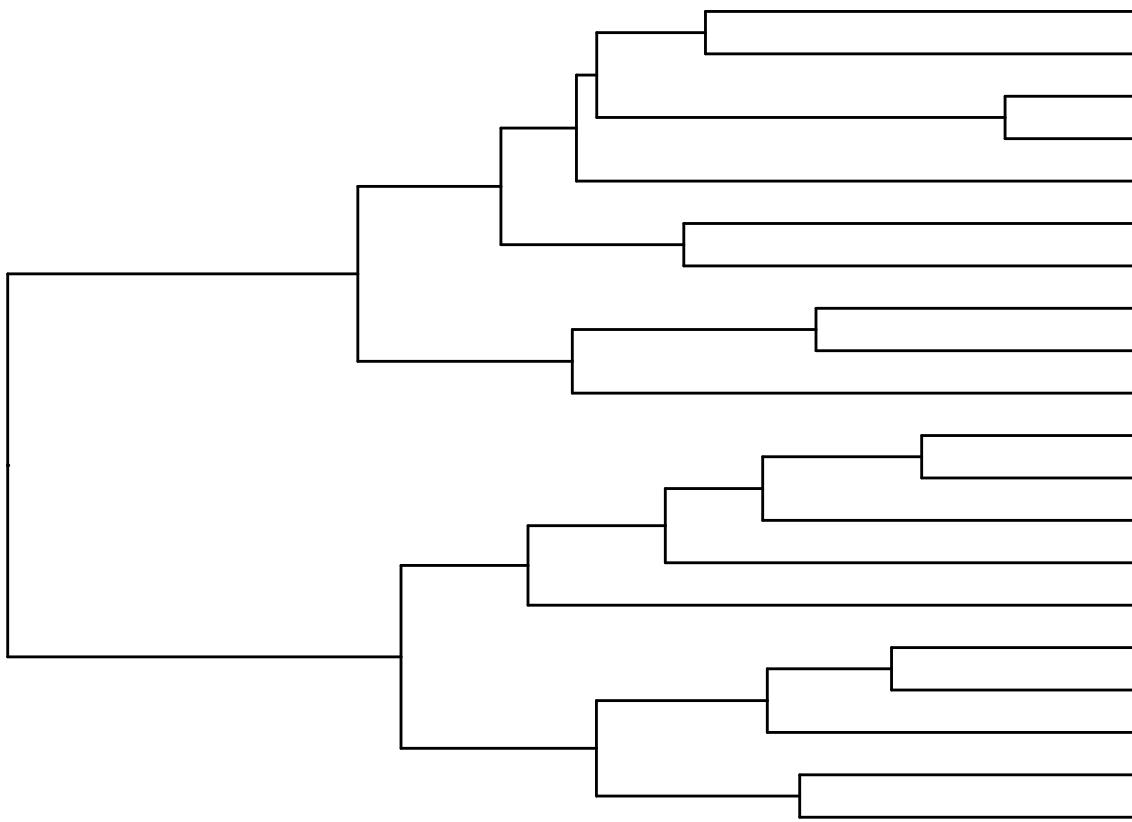
taxa_hclust <- hclust(dist(mat), method = "complete")

# Creates a phylogenetic tree
taxa_tree <- as.phylo(taxa_hclust)

# Plot taxa tree
taxa_tree <- ggtree(taxa_tree) +
  theme(plot.margin=margin(0,0,0,0)) # removes margins

taxa_tree # based on this three, we'll create two clusters

```



```

# Get order of taxa in plot
taxa_ordered <- get_taxa_name(taxa_tree)

taxa_clusters <- cutree(tree = taxa_hclust, k = 2) # 2 clusters based on methods in Clustering.R script

# Converts into data frame
taxa_clusters <- data.frame(clusters = taxa_clusters)
taxa_clusters$clusters <- factor(taxa_clusters$clusters)

# Order data so that it's same as in phylo tree
taxa_clusters <- taxa_clusters[taxa_ordered, , drop = FALSE]

# Prints taxa and their clusters
taxa_clusters

##                                     clusters
## tet(0/32/0)_5_FP929050          2
## ant(6)-Ia_1_AF330699           2
## aph(3)-III_1_M26832            2
## aph(3)-IIIa_1_AF330699          2
## ant(6)-Ia_3_KF864551           2
## erm(G)_1_M15332                2
## lnu(C)_1_AY928180              2
## erm(B)_9_AF299292               2
## erm(B)_6_AF242872               2
## tet(44)_2_FN594949               2

```

```

## tet(W)_1_DQ060146           1
## tet(W)_4_FN396364           1
## tet(W)_5_AJ427422           1
## tet(O/W)-1_1_AY485126       1
## tet(40)_1_FJ158002          1
## tet(O/W)_4_AM889121          1
## tet(O/W)_5_AM889122          1
## tet(O/W/32/0)_2_FM164392      1
## tet(O/W)_2_AM889119          1
## tet(O/W/32/0)_1_EF065523      1

#rowData(tse_AMRClass)$clusters <- taxa_clusters[order(match(rownames(taxa_clusters), rownames(tse_AMRC

# Prints taxa and their clusters
rowData(tse_AMRClass)$clusters

## NULL

sample_hclust <- hclust(dist(t(mat)), method = "complete")

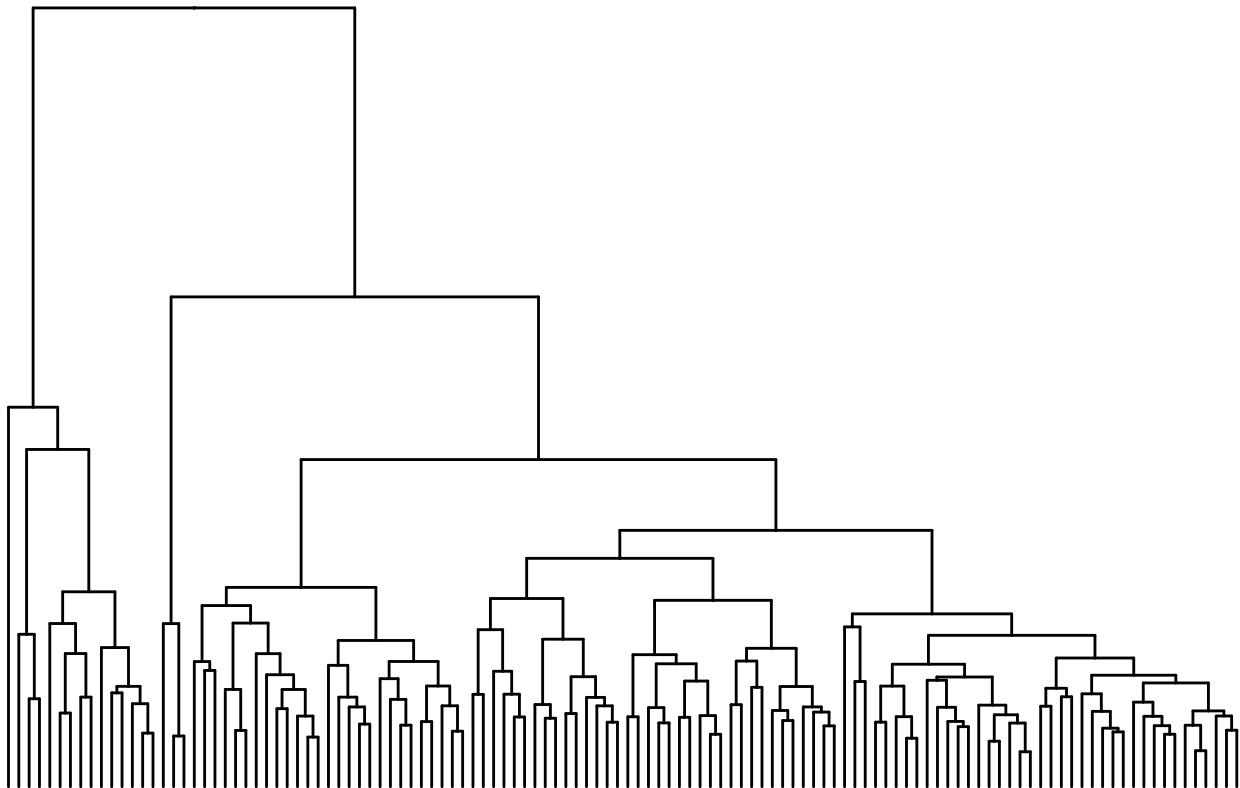
# Creates a phylogenetic tree
sample_tree <- as.phylo(sample_hclust)

# Plot sample tree
sample_tree <- ggtree(sample_tree) + layout_dendrogram() +
  theme(plot.margin=margin(0,0,0,0)) # removes margins

# Get order of samples in plot
samples_ordered <- rev(get_taxa_name(sample_tree))

# to view the tree, run
sample_tree

```



```

# Creates clusters
sample_clusters <- factor(cutree(tree = sample_hclust, k = 2))

# Converts into data frame
sample_data <- data.frame(clusters = sample_clusters)

# Order data so that it's same as in phylo tree
sample_data <- sample_data[samples_ordered, , drop = FALSE]

# Order data based on
tse_AMRClass <- tse_AMRClass[ , rownames(sample_data)]

# Add sample type data
sample_data$sample_types <- colData(tse_AMRClass)$AB

sample_data

##           clusters sample_types
## 2_57          2        no
## 4_37          2        no
## 4_55          2        no
## 4_57          2        no
## 14_22         2        no
## 4_39          2        no
## 5_41          2        no
## 10_22         2        no

```

## 5_55	2	no
## 4_36	2	no
## 10_67	2	no
## 9_22	2	no
## 10_59	2	yes
## 10_26	2	no
## 9_17	2	no
## 10_1	1	no
## 10_2	1	no
## 10_3	1	no
## 14_21	1	no
## 14_36	1	no
## 5_39	1	no
## 5_59	1	no
## 14_34	1	no
## 4_56	1	no
## 10_12	1	yes
## 6_57	1	no
## 10_15	1	yes
## 14_30	1	no
## 6_56	1	no
## 10_7	1	no
## 6_55	1	no
## 10_4	1	no
## 14_20	1	no
## 4_54	1	no
## 14_25	1	no
## 5_40	1	no
## 4_41	1	no
## 6_58	1	no
## 14_29	1	no
## 14_33	1	no
## 14_23	1	no
## 14_27	1	no
## 14_35	1	no
## 4_40	1	no
## 5_54	1	no
## 10_19	1	no
## 10_25	1	no
## 4_65	1	no
## 10_20	1	no
## 11_1	1	no
## 6_37	1	no
## 10_60	1	yes
## 10_21	1	no
## 9_38	1	no
## 10_11	1	yes
## 10_13	1	yes
## 4_38	1	no
## 10_10	1	yes
## 10_14	1	yes
## 10_8	1	no
## 10_68	1	no
## 11_3	1	no

## 10_29	1	no
## 2_48	1	no
## 2_49	1	no
## 10_57	1	yes
## 10_58	1	yes
## 10_35	1	no
## 10_40	1	no
## 10_63	1	yes
## 10_34	1	no
## 2_41	1	no
## 2_42	1	no
## 9_21	1	no
## 10_52	1	yes
## 10_44	1	no
## 2_36	1	no
## 2_40	1	no
## 10_69	1	no
## 2_58	1	no
## 9_16	1	no
## 2_25	1	no
## 10_49	1	yes
## 6_38	1	no
## 9_18	1	no
## 9_34	1	no
## 10_28	1	no
## 2_29	1	no
## 2_39	1	no
## 2_47	1	no
## 9_37	1	no
## 6_54	1	no
## 2_59	1	no
## 6_36	1	no
## 2_26	1	no
## 2_23	1	no
## 2_27	1	no
## 10_41	1	no
## 10_39	1	no
## 10_64	1	yes
## 2_51	1	no
## 9_36	1	no
## 10_30	1	no
## 9_19	1	no
## 9_35	1	no
## 10_33	1	no
## 2_60	1	no
## 10_50	1	yes
## 2_61	1	no
## 10_42	1	no
## 10_48	1	yes
## 10_43	1	no
## 10_53	1	yes
## 9_39	1	no
## 10_51	1	yes
## 2_24	1	no

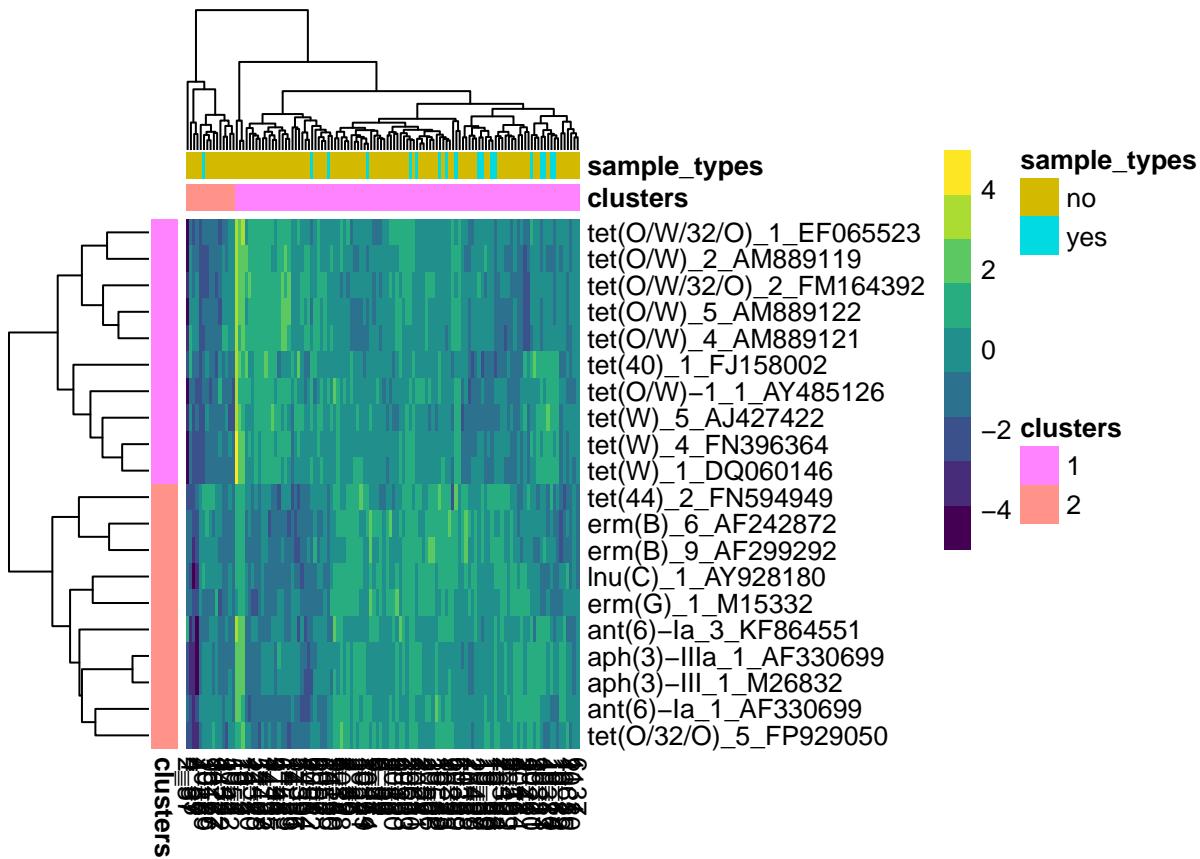
```

## 2_52      1      no
## 2_56      1      no
## 10_66     1      no
## 2_50      1      no

breaks <- seq(-ceiling(max(abs(mat))), ceiling(max(abs(mat))),
               length.out = ifelse( max(abs(mat))>5, 2*ceiling(max(abs(mat))), 10 ) )

pheatmap(mat, annotation_row = taxa_clusters,
          annotation_col = sample_data,
          breaks = breaks,
          color = colorRampPalette(viridis(256))(length(breaks)-1))

```



```
## Alpha diversity
```

```

# For rarefaction to function, and good coverage tests to make sense, we need to have integers for our

Rps_copy = Rps
Rps_mp_copy = Rps_mp
Rps_tpm_copy = Rps_tpm

#rounding the "counts"
otu_table(Rps_copy) = otu_table(round(as((otu_table(Rps_copy)), "matrix")), taxa_are_rows(Rps_copy))
otu_table(Rps_tpm_copy) = otu_table(round(as((otu_table(Rps_tpm_copy)), "matrix")), taxa_are_rows(Rps_tpm_copy))
otu_table(Rps_mp_copy) = otu_table(round(as((otu_table(Rps_mp_copy)), "matrix")), taxa_are_rows(Rps_mp_copy))

```

```

# Earlier renditions of this script also scaled the otu table by a factor 1000 to preserve data, but th
Rps_scaled_copy = Rps
Rps_mp_scaled_copy = Rps_mp

otu_table(Rps_scaled_copy) = otu_table(Rps_scaled_copy) * 1000
otu_table(Rps_mp_scaled_copy) = otu_table(Rps_mp_scaled_copy) * 1000

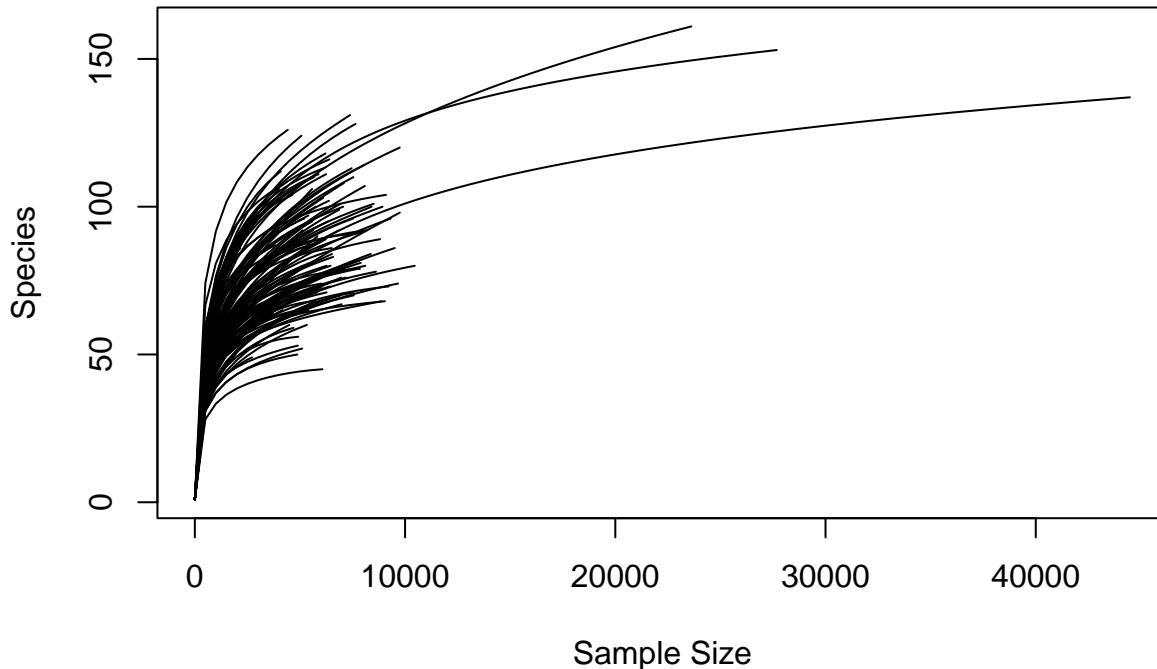
otu_table(Rps_scaled_copy) = otu_table(round(as((otu_table(Rps_scaled_copy)), "matrix")), taxa_are_rows)
otu_table(Rps_mp_scaled_copy) = otu_table(round(as((otu_table(Rps_mp_scaled_copy)), "matrix")), taxa_are

otu_tab <- t(abundances(Rps_copy)) # can use veganotu() function for this
otu_tab2 <- t(abundances(Rps_tpm_copy))
otu_tab3 <- t(abundances(Rps_mp_copy))

otu_tab4 <- t(abundances(Rps_scaled_copy))
otu_tab5 <- t(abundances(Rps_mp_scaled_copy))

# rarefaction curves
vegan::rarecurve(otu_tab, step = 500, label = FALSE) # kraken2

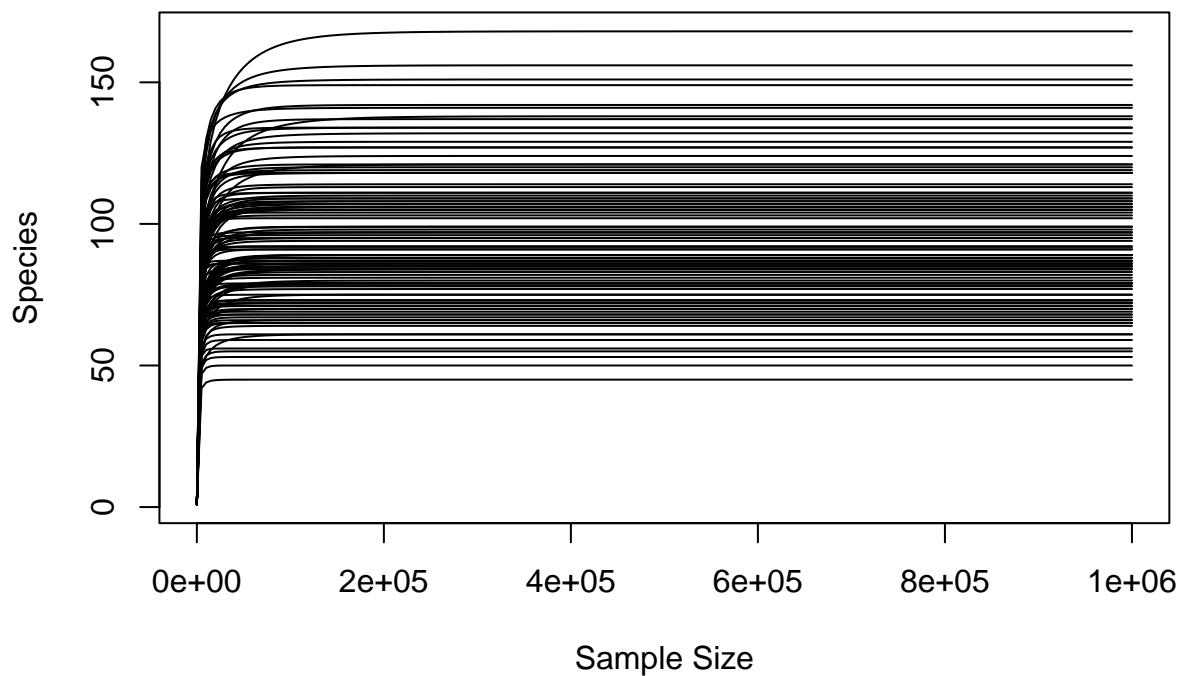
```



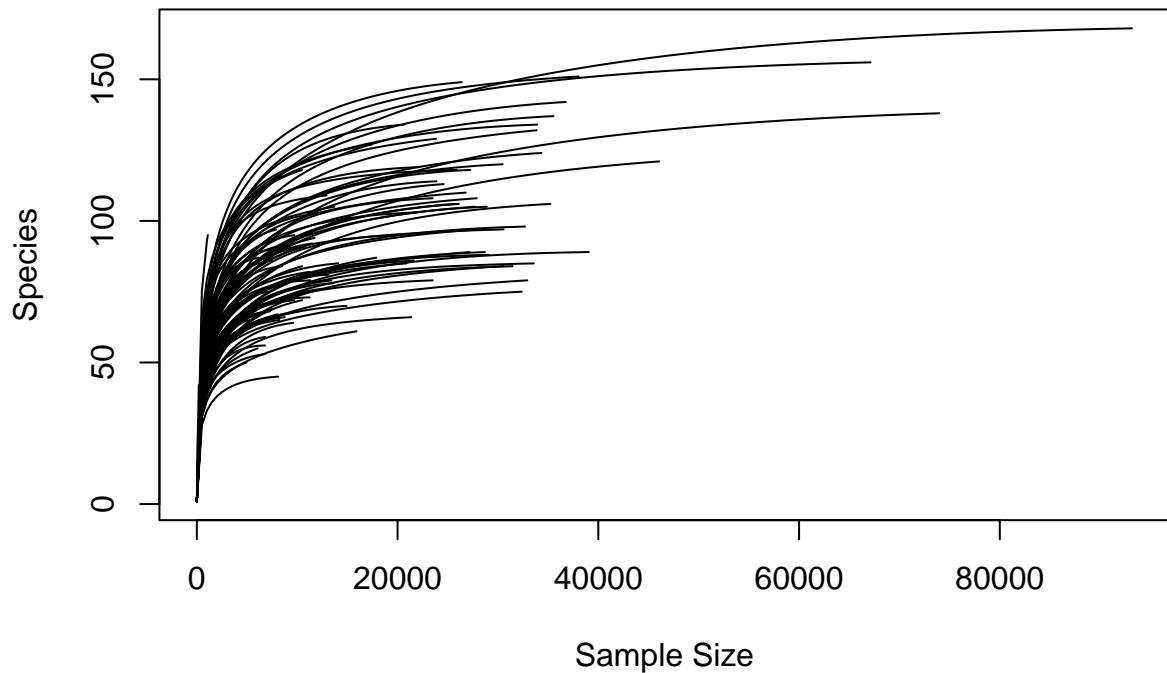
```

vegan::rarecurve(otu_tab2, step = 5000, label = FALSE) # TPM k2 (this data has smallest count 10)

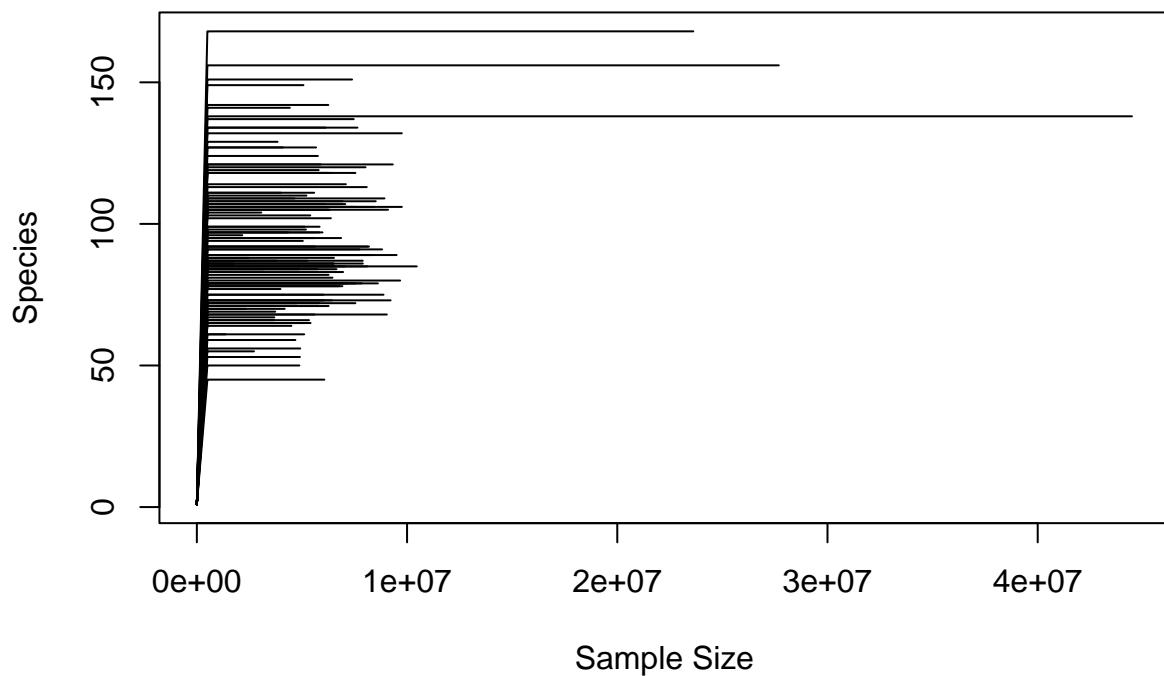
```



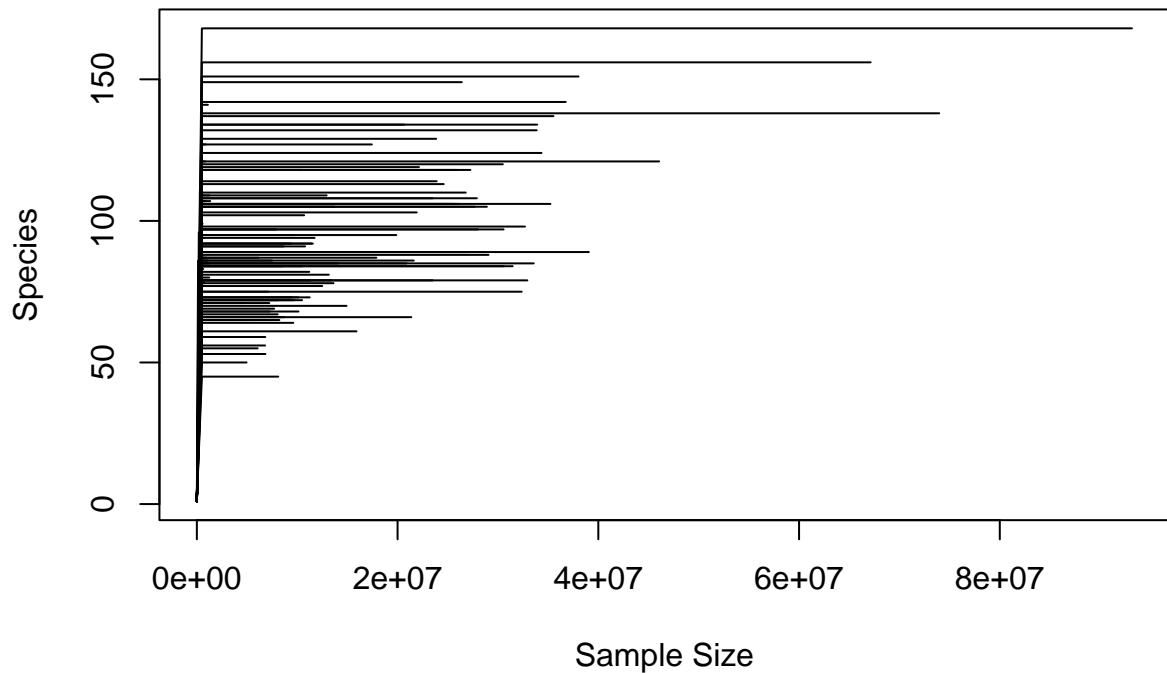
```
# rarefaction curves of TPM data are all converging towards the plateau, no rarefaction required  
vegan::rarecurve(otu_tab3, step = 500, label = FALSE) #MP
```



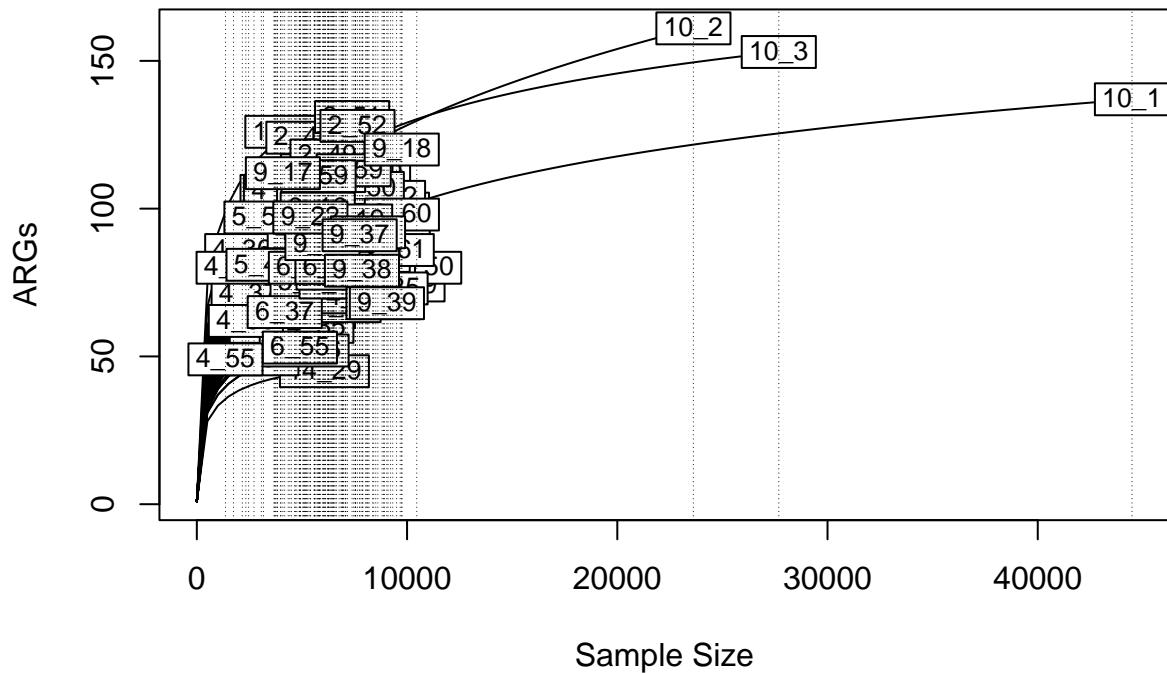
```
vegan::rarecurve(otu_tab4, step = 500000, label = FALSE) # k2 scaled
```



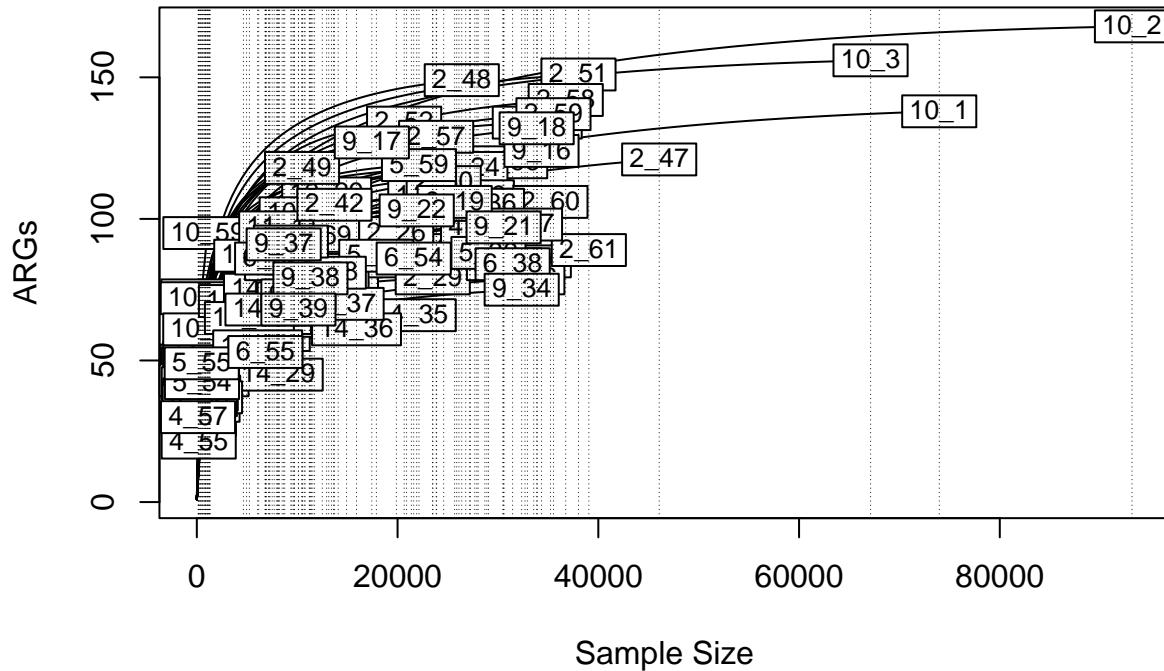
```
vegan::rarecurve(otu_tab5, step = 500000, label = FALSE) #MP scaled
```



```
# These two curves completely artificially flatten out, clearly because of the scaling
# we can add lines to show sampling depths
rarecurve(otu_tab, step=500, ylab = "ARGs")
abline(v=sample_sums(Rps), lty='dotted', lwd=0.5) # k2
```



```
rarecurve(otu_tab3, step=500, ylab = "ARGs")
abline(v=sample_sums(Rps_mp), lty='dotted', lwd=0.5) # MP
```



```
# virtually no samples are reaching a plateau so sequencing depth is not appropriate, undersampling for k2

# we use Good's coverage test to see the amount of singletons in the samples (might have to add round())

summary(goods(round(otu_tab3))) # on average, 0.65% of the reads in the samples are singletons for mp

##      no.sing      no.seqs      goods
##  Min.   : 0.000  Min.   : 118  Min.   : 87.29
##  1st Qu.: 4.000  1st Qu.: 6815  1st Qu.: 99.90
##  Median : 6.000  Median :11280  Median : 99.96
##  Mean   : 7.408  Mean   :16407  Mean   : 99.35
##  3rd Qu.: 9.000  3rd Qu.:26183  3rd Qu.: 99.98
##  Max.   :30.000  Max.   :93167  Max.   :100.00

summary(goods(round(otu_tab))) # on average, 0.31% of the reads in the samples are singletons for k2

##      no.sing      no.seqs      goods
##  Min.   : 3.00  Min.   : 1363  Min.   :98.74
##  1st Qu.:12.00  1st Qu.: 5062  1st Qu.:99.62
##  Median :17.50  Median : 6240  Median :99.71
##  Mean   :17.84  Mean   : 6826  Mean   :99.69
##  3rd Qu.:22.00  3rd Qu.: 7580  3rd Qu.:99.82
##  Max.   :43.00  Max.   :44480  Max.   :99.96
```

```
summary(goods(otu_tab2)) # there are no singletons in tpm
```

```
##      no.sing      no.seqs       goods
##  Min.   :0    Min.   :1e+06   Min.   :100
##  1st Qu.:0    1st Qu.:1e+06   1st Qu.:100
##  Median :0    Median :1e+06   Median :100
##  Mean   :0    Mean   :1e+06   Mean   :100
##  3rd Qu.:0    3rd Qu.:1e+06   3rd Qu.:100
##  Max.   :0    Max.   :1e+06   Max.   :100
```

```
Rps_mp_copy %>% ps_filter(FarmRoundStable == c("Farm2R1S1")) %>% veganotu() %>% goods() %>% summary()
```

```
##      no.sing      no.seqs       goods
##  Min.   : 6.00   Min.   :118.0   Min.   :87.29
##  1st Qu.:10.75  1st Qu.:202.8   1st Qu.:95.04
##  Median :13.50  Median :365.0   Median :95.91
##  Mean   :13.33  Mean   :395.1   Mean   :95.51
##  3rd Qu.:15.00  3rd Qu.:527.0   3rd Qu.:97.30
##  Max.   :21.00  Max.   :816.0   Max.   :98.90
```

```
# for the stable Farm2R1S1 on mp, on average, 4.5% of the reads in the samples are singletons
Rps_copy %>% ps_filter(FarmRoundStable == c("Farm2R1S1")) %>% veganotu() %>% goods() %>% summary()
```

```
##      no.sing      no.seqs       goods
##  Min.   :11.00   Min.   :1363   Min.   :98.74
##  1st Qu.:16.50  1st Qu.:2289   1st Qu.:99.16
##  Median :21.50  Median :3538   Median :99.27
##  Mean   :23.33  Mean   :4017   Mean   :99.31
##  3rd Qu.:30.00  3rd Qu.:5644   3rd Qu.:99.46
##  Max.   :41.00  Max.   :7559   Max.   :99.84
```

```
# for k2, this is only 0.73%
```

```
# rarefy to equal library size or not? (mp)
lib.div <- microbiome::alpha(Rps_mp, index = "all")
lib.div2 <- richness(Rps_mp)
lib.div$ReadsPerSample <- sample_sums(Rps_mp)
lib.div$chao1 <- lib.div2$chao1
colnames(lib.div)
```

```
## [1] "observed"                  "chao1"
## [3] "diversity_inverse_simpson" "diversity_gini_simpson"
## [5] "diversity_shannon"         "diversity_coverage"
## [7] "evenness_camargo"          "evenness_pielou"
## [9] "evenness_simpson"          "evenness_evar"
## [11] "evenness_bulla"            "dominance_dbp"
## [13] "dominance_dmn"             "dominance_absolute"
## [15] "dominance_relative"        "dominance_simpson"
## [17] "dominance_core_abundance" "dominance_gini"
## [19] "rarity_log_modulo_skewness" "rarity_low_abundance"
## [21] "rarity_rare_abundance"     "ReadsPerSample"
```

```

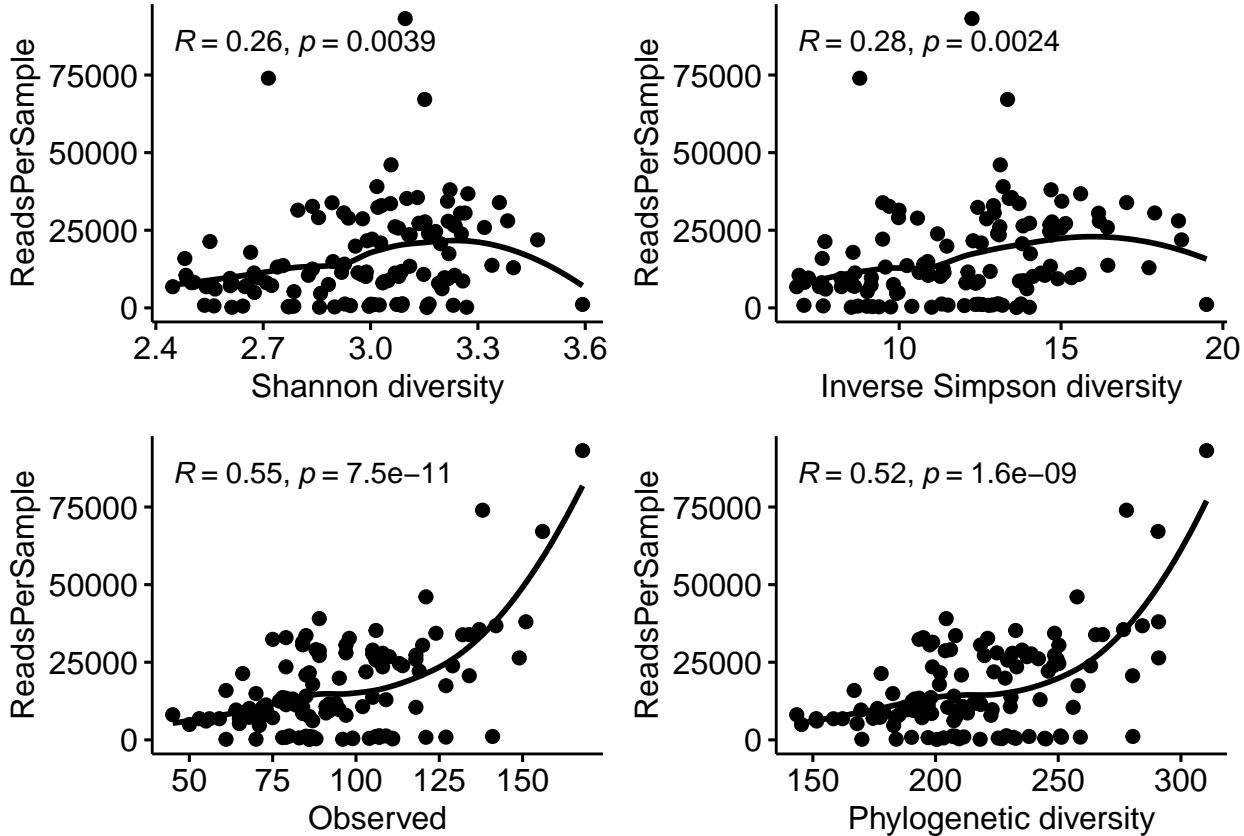
p1 = ggscatter(lib.div, "diversity_shannon", "ReadsPerSample", xlab = "Shannon diversity", add = "loess"
  stat_cor(method = "pearson")
p2 = ggscatter(lib.div, "diversity_inverse_simpson", "ReadsPerSample", xlab = "Inverse Simpson diversity",
  stat_cor(method = "pearson")
p3 = ggscatter(lib.div, "observed", "ReadsPerSample", xlab = "Observed", add = "loess") +
  stat_cor(method = "pearson")

df.pd <- pd(t(as.data.frame(Rps_mp@otu_table)), Rps_mp@phy_tree, include.root=T) # transposing for use in ggplot
lib.div$Phylogenetic_Diversity <- df.pd$PD

p4 = ggscatter(lib.div, "Phylogenetic_Diversity", "ReadsPerSample", xlab = "Phylogenetic diversity", add = "loess"
  stat_cor(method = "pearson")

ggarrange(p1, p2, p3, p4, ncol = 2, nrow = 2)

```



```
# there seems to be increases of reads/sample for PD and observed
```

```
# However, in the rarefaction curves, we can clearly see three outliers, with very large sample sizes at
```

```
sample_data(Rps_mp)$Sample_Unique = sample_names(Rps_mp)
sample_variables(Rps_mp)
```

```
## [1] "SampleIdentifier" "ResCap"           "Conc...ng..pl."    "SampleID"
```

```

## [5] "LibraryNumber"      "Sample_Unique"      "LibraryName"      "Farm"
## [9] "Farm2"               "Stable"            "FarmRoundStable"  "Days"
## [13] "Age"                 "Sname"             "WeightAnimal"     "Gender"
## [17] "AgeParentStock"     "Hatchery"          "Researcher"       "AB"
## [21] "Abday"               "FlockSize"          "FeedF"             "FeedType"
## [25] "FeedProducent"      "Cox"                "OPG"               "Cluster"
## [29] "LitterType"          "Metagenomics"       "ReadPerc"          "ReadTot"
## [33] "Stables"

Rps_trim = Rps_mp %>% subset_samples(Sample_Unique != "10_1" & Sample_Unique != "10_2" & Sample_Unique != "10_3")

lib.div <- microbiome::alpha(Rps_trim, index = "all")
lib.div2 <- richness(Rps_trim)
lib.div$ReadsPerSample <- sample_sums(Rps_trim)
lib.div$chao1 <- lib.div2$chao1
colnames(lib.div)

## [1] "observed"                  "chao1"
## [3] "diversity_inverse_simpson" "diversity_gini_simpson"
## [5] "diversity_shannon"         "diversity_coverage"
## [7] "evenness_camargo"          "evenness_pielou"
## [9] "evenness_simpson"          "evenness_evar"
## [11] "evenness_bulla"            "dominance_dbp"
## [13] "dominance_dmn"             "dominance_absolute"
## [15] "dominance_relative"        "dominance_simpson"
## [17] "dominance_core_abundance"  "dominance_gini"
## [19] "rarity_log_modulo_skewness" "rarity_low_abundance"
## [21] "rarity_rare_abundance"     "ReadsPerSample"

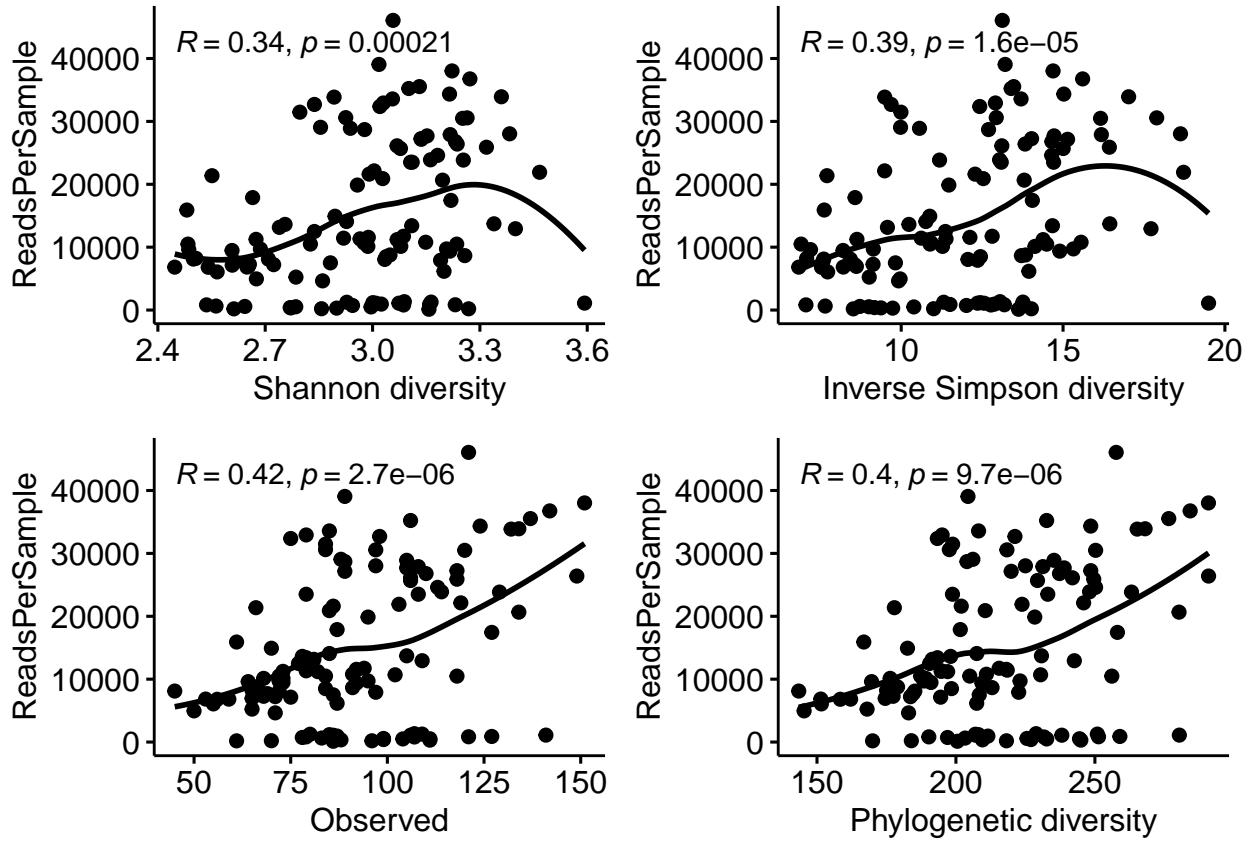
p1 = ggscatter(lib.div, "diversity_shannon", "ReadsPerSample", xlab = "Shannon diversity", add = "loess",
  stat_cor(method = "pearson")
p2 = ggscatter(lib.div, "diversity_inverse_simpson", "ReadsPerSample", xlab = "Inverse Simpson diversity",
  stat_cor(method = "pearson")
p3 = ggscatter(lib.div, "observed", "ReadsPerSample", xlab = "Observed", add = "loess") +
  stat_cor(method = "pearson")

df.pd <- pd(t(as.data.frame(Rps_trim@otu_table)), Rps_trim@phy_tree, include.root=T) # transposing for use
lib.div$Phylogenetic_Diversity <- df.pd$PD

p4 = ggscatter(lib.div, "Phylogenetic_Diversity", "ReadsPerSample", xlab = "Phylogenetic diversity", add = "loess",
  stat_cor(method = "pearson"))

ggarrange(p1, p2, p3, p4, ncol = 2, nrow = 2)

```



```
# even when we remove the outliers, there is a positive correlation
```

```
# repeat this for k2
```

```
lib.div <- microbiome::alpha(Rps, index = "all")
lib.div2 <- richness(Rps)
lib.div$ReadsPerSample <- sample_sums(Rps)
lib.div$chao1 <- lib.div2$chao1
colnames(lib.div)
```

```
## [1] "observed"                      "chao1"
## [3] "diversity_inverse_simpson"     "diversity_gini_simpson"
## [5] "diversity_shannon"              "diversity_coverage"
## [7] "evenness_camargo"               "evenness_pielou"
## [9] "evenness_simpson"                "evenness_evar"
## [11] "evenness_bulla"                  "dominance_dbp"
## [13] "dominance_dmn"                  "dominance_absolute"
## [15] "dominance_relative"              "dominance_simpson"
## [17] "dominance_core_abundance"       "dominance_gini"
## [19] "rarity_log_modulo_skewness"     "rarity_low_abundance"
## [21] "rarity_rare_abundance"          "ReadsPerSample"
```

```
p1 = ggscatter(lib.div, "diversity_shannon", "ReadsPerSample", xlab = "Shannon diversity", add = "loess"
  stat_cor(method = "pearson")
p2 = ggscatter(lib.div, "diversity_inverse_simpson", "ReadsPerSample", xlab = "Inverse Simpson diversi
```

```

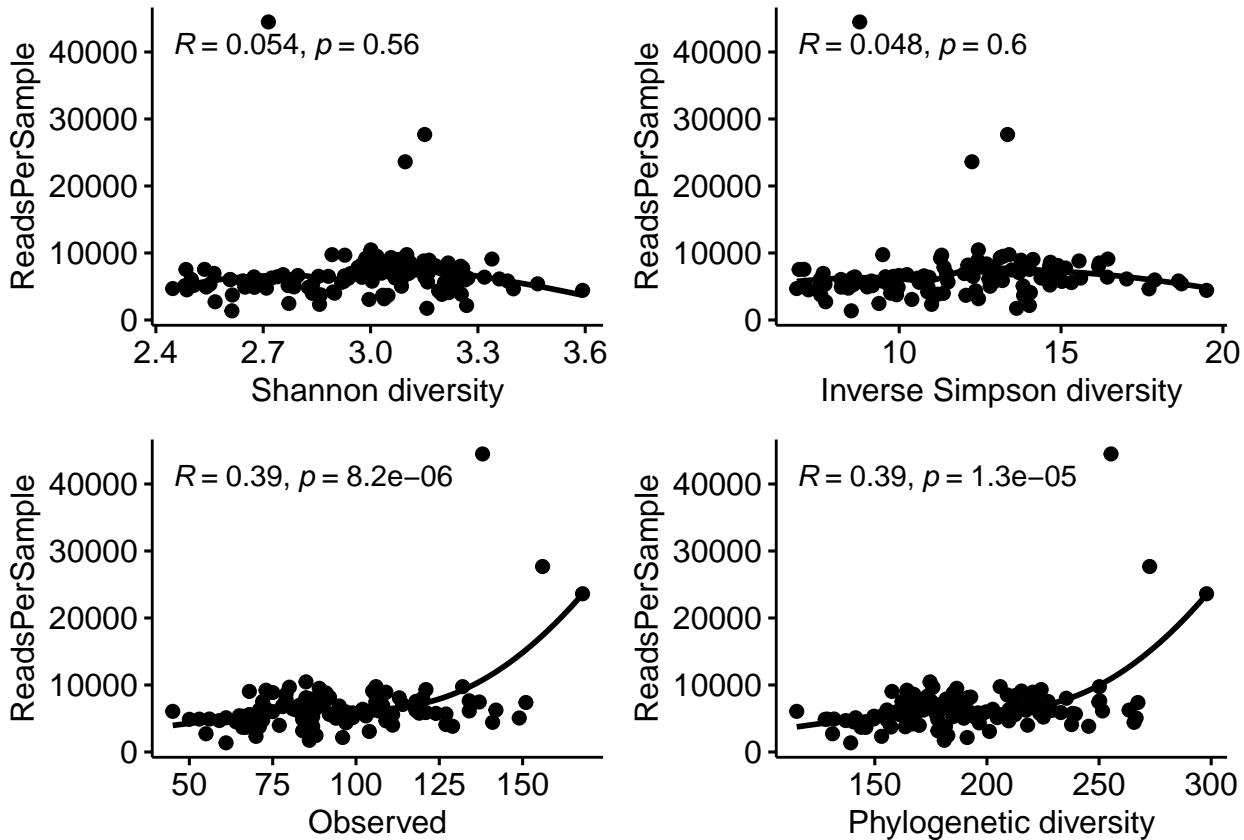
stat_cor(method = "pearson")
p3 = ggscatter(lib.div, "observed", "ReadsPerSample", xlab = "Observed", add = "loess") +
  stat_cor(method = "pearson")

df.pd <- pd(t(as.data.frame(Rps@otu_table)), Rps@phy_tree, include.root=T) # transposing for use in pica
lib.div$Phylogenetic_Diversity <- df.pd$PD

p4 = ggscatter(lib.div, "Phylogenetic_Diversity", "ReadsPerSample", xlab = "Phylogenetic diversity", add =
  stat_cor(method = "pearson"))

ggarrange(p1, p2, p3, p4, ncol = 2, nrow = 2)

```



```

# we can clearly see an increase in reads/sample when increasing abundance (at least for PD and observed)

# rarefy to equal library size or not? (k2)

Rps_trim = Rps %>% subset_samples(Sample_Unique != "10_1" & Sample_Unique != "10_2" & Sample_Unique != "10_3")

lib.div <- microbiome::alpha(Rps_trim, index = "all")
lib.div2 <- richness(Rps_trim)
lib.div$ReadsPerSample <- sample_sums(Rps_trim)
lib.div$chao1 <- lib.div2$chao1
colnames(lib.div)

```

```

## [1] "observed"          "chao1"
## [3] "diversity_inverse_simpson" "diversity_gini_simpson"
## [5] "diversity_shannon"      "diversity_coverage"
## [7] "evenness_camargo"       "evenness_pielou"
## [9] "evenness_simpson"       "evenness_evar"
## [11] "evenness_bulla"         "dominance_dbp"
## [13] "dominance_dmn"         "dominance_absolute"
## [15] "dominance_relative"    "dominance_simpson"
## [17] "dominance_core_abundance" "dominance_gini"
## [19] "rarity_log_modulo_skewness" "rarity_low_abundance"
## [21] "rarity_rare_abundance"   "ReadsPerSample"

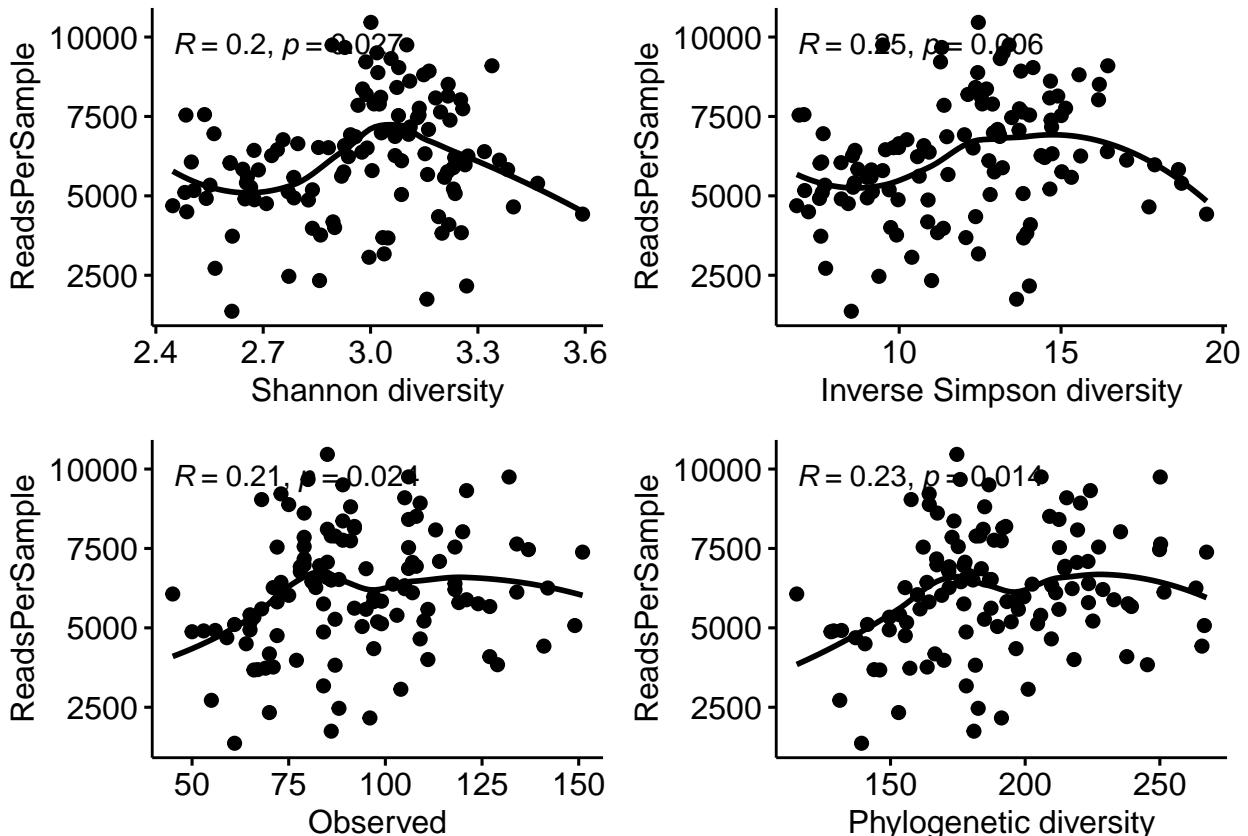
p1 = ggscatter(lib.div, "diversity_shannon", "ReadsPerSample", xlab = "Shannon diversity", add = "loess",
  stat_cor(method = "pearson")
p2 = ggscatter(lib.div, "diversity_inverse_simpson", "ReadsPerSample", xlab = "Inverse Simpson diversity",
  stat_cor(method = "pearson")
p3 = ggscatter(lib.div, "observed", "ReadsPerSample", xlab = "Observed", add = "loess") +
  stat_cor(method = "pearson")

df.pd <- pd(t(as.data.frame(Rps_trim@otu_table)), Rps_trim@phy_tree, include.root=T) # transposing for use
lib.div$Phylogenetic_Diversity <- df.pd$PD

p4 = ggscatter(lib.div, "Phylogenetic_Diversity", "ReadsPerSample", xlab = "Phylogenetic diversity", add = "loess",
  stat_cor(method = "pearson")

ggarrange(p1, p2, p3, p4, ncol = 2, nrow = 2)

```



```

# Now, this positive correlation has completely disappeared

lib.div <- microbiome::alpha(Rps, index = "all")
lib.div2 <- richness(Rps)
lib.div$ReadsPerSample <- sample_sums(Rps)
lib.div$chao1 <- lib.div2$chao1
df.pd <- pd(t(as.data.frame(Rps@otu_table)), Rps@phy_tree, include.root=T) # transposing for use in picante
lib.div$Phylogenetic_Diversity <- df.pd$PD

set.seed(1337)

ps0.rar <- rarefy_even_depth(Rps, sample.size = 118) # we do not want to lose samples so lowest sample size is 118
#ps0.rar <- srs_p(Rps) alternative way of rarefaction

# We will use k2 corrected data

hmp.div <- microbiome::alpha(Rps, index = "all")

datatable(hmp.div)

hmp.meta <- meta(Rps)
hmp.meta$sam_name <- rownames(hmp.meta)
hmp.div$sam_name <- rownames(hmp.div)
div.df <- merge(hmp.div,hmp.meta, by = "sam_name")
colnames(div.df)

## [1] "sam_name"                      "observed"
## [3] "chao1"                          "diversity_inverse_simpson"
## [5] "diversity_gini_simpson"        "diversity_shannon"
## [7] "diversity_coverage"            "evenness_camargo"
## [9] "evenness_pielou"                "evenness_simpson"
## [11] "evenness_evar"                  "evenness_bulla"
## [13] "dominance_dbp"                 "dominance_dmn"
## [15] "dominance_absolute"            "dominance_relative"
## [17] "dominance_simpson"             "dominance_core_abundance"
## [19] "dominance_gini"                "rarity_log_modulo_skewness"
## [21] "rarity_low_abundance"          "rarity_rare_abundance"
## [23] "SampleIdentifier"              "ResCap"
## [25] "Conc...ng..pl."                "SampleID"
## [27] "LibraryNumber"                 "Sample_Unique"
## [29] "LibraryName"                   "Farm"
## [31] "Farm2"                         "Stable"
## [33] "FarmRoundStable"               "Days"
## [35] "Age"                            "Sname"
## [37] "WeightAnimal"                  "Gender"
## [39] "AgeParentStock"                "Hatchery"
## [41] "Researcher"                    "AB"
## [43] "Abday"                         "FlockSize"
## [45] "FeedF"                          "FeedType"
## [47] "FeedProducent"                 "Cox"
## [49] "OPG"                            "Cluster"

```

```

## [51] "LitterType"                      "Metagenomics"
## [53] "ReadPerc"                         "ReadTot"
## [55] "Stables"

#based on microbial agent
div.df2 <- div.df[, c("Cox", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon")]
colnames(div.df2) <- c("Agent", "Inverse Simpson", "Gini-Simpson", "Shannon", "chao1", "Coverage", "Pie")

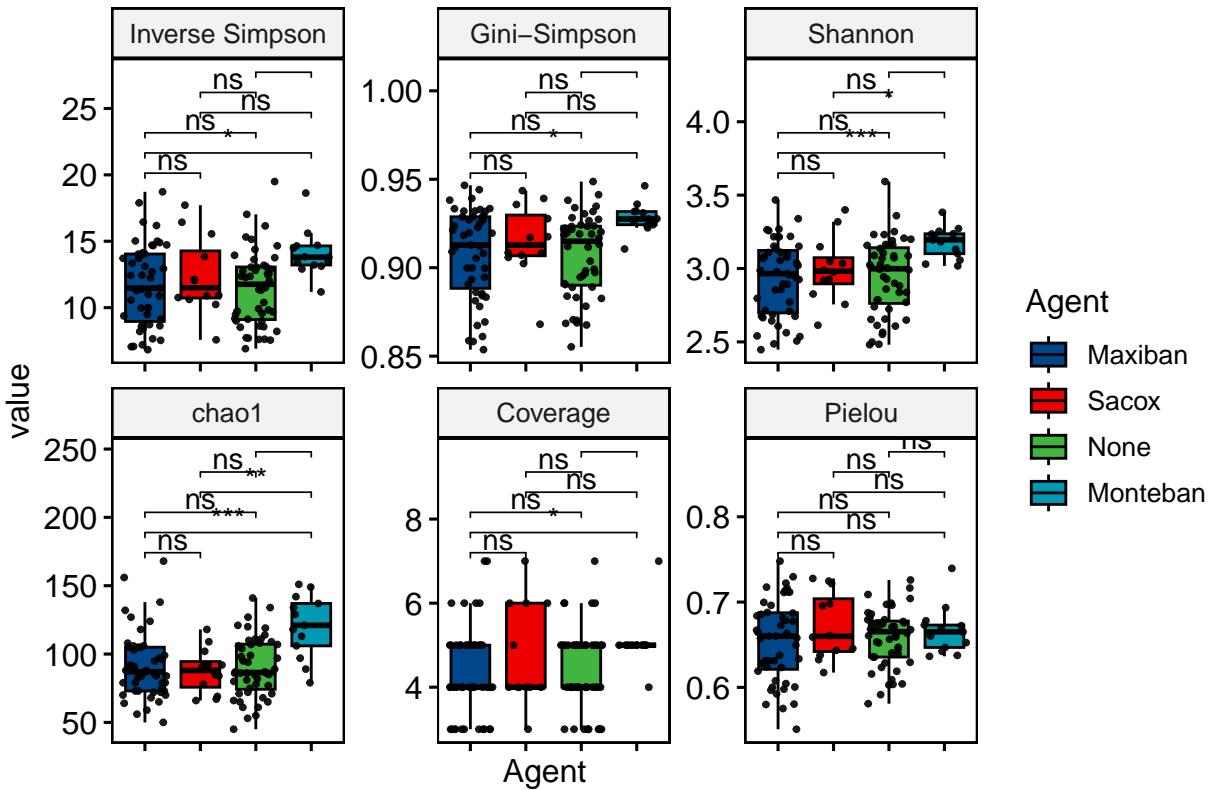
div_df_melt <- reshape2::melt(div.df2)

lev = c("Maxiban", "Scox", "Monteban", "None")
L.pairs <- combn(seq_along(lev), 2, simplify = FALSE, FUN = function(i) lev[i])

ggboxplot(div_df_melt, x = "Agent", y = "value",
          fill = "Agent",
          palette = "lancet",
          legend = "right",
          facet.by = "variable",
          scales = "free",
          title = "FPKM Alpha diversity metrics by microbial agent",
          outlier.shape = NA) +
rremove("x.text") + stat_compare_means(
  comparisons = L.pairs,
  method = "wilcox.test",
  label = "p.signif"
) + geom_jitter(size = 0.7, alpha = 0.9)

```

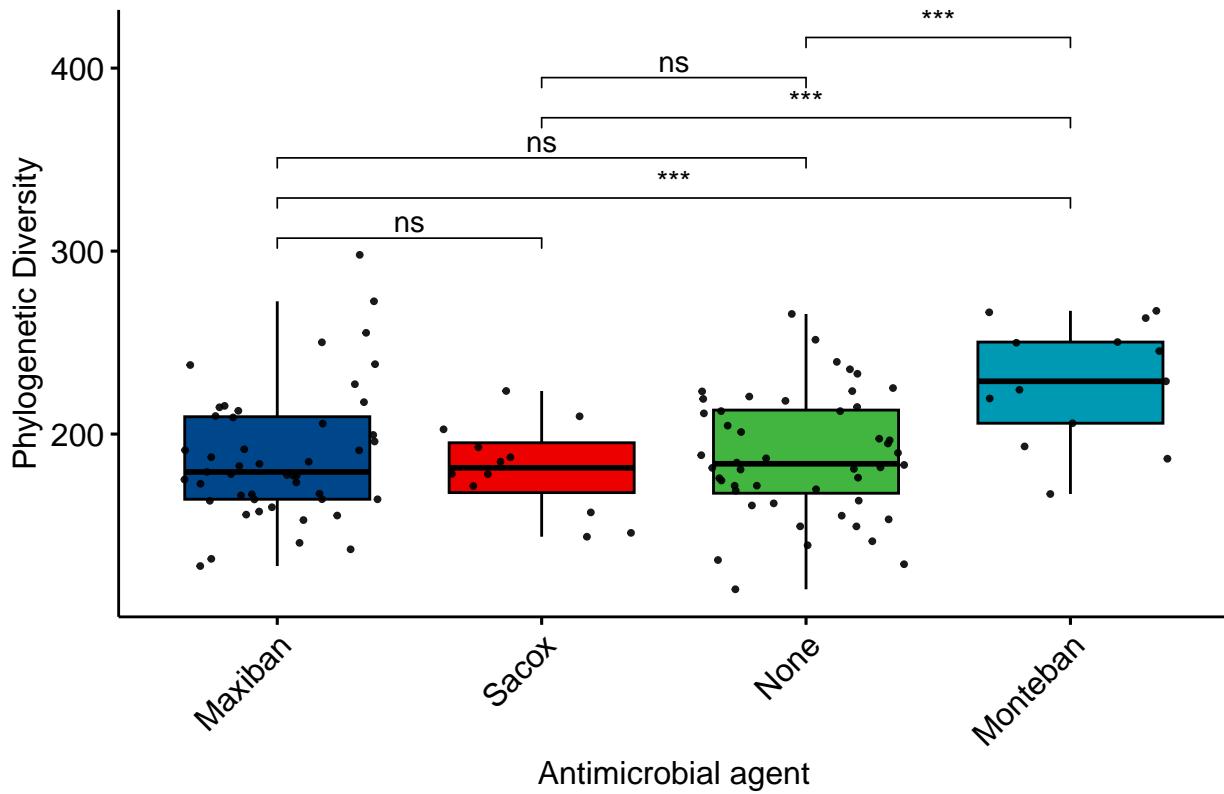
FPKM Alpha diversity metrics by microbial agent



```
df.pd <- pd(t(as.data.frame(Rps@otu_table)), Rps@phy_tree, include.root=T) # transposing for use in pica
hmp.meta$Phylogenetic_Diversity <- df.pd$PD

ggboxplot(hmp.meta,
          x = "Cox",
          y = "Phylogenetic_Diversity",
          fill = "Cox",
          palette = "lancet",
          ylab = "Phylogenetic Diversity",
          xlab = "Antimicrobial agent",
          legend = "right",
          title = "FPKM phylogenetic diversity by microbial agent",
          outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) +
  stat_compare_means(
    comparisons = L.pairs,
    label = "p.signif",) + geom_jitter(size = 0.7, alpha = 0.9)
```

FPKM phylogenetic diversity by microbial agent



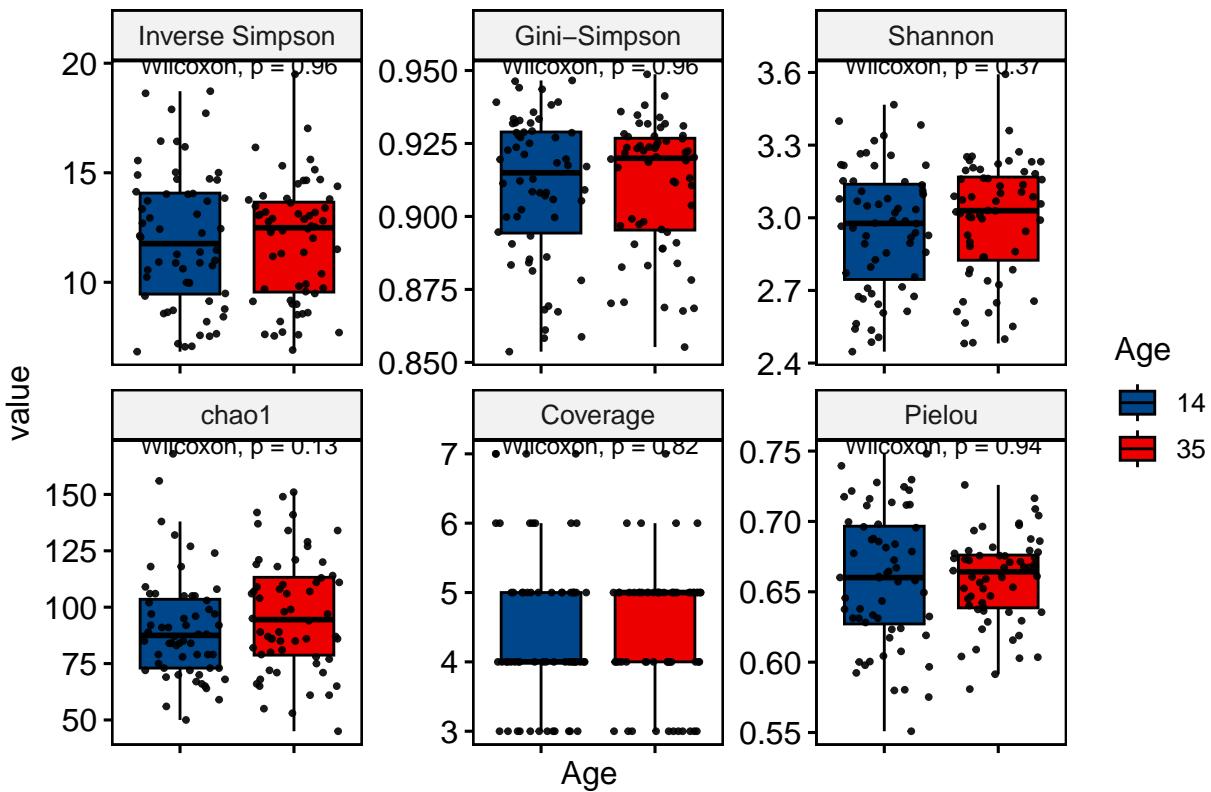
```
# age / days

div.df2 <- div.df[, c("Age", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon")]
colnames(div.df2) <- c("Age", "Inverse Simpson", "Gini-Simpson", "Shannon", "chao1", "Coverage", "Pielou")

div.df2$Age = as.factor(div.df2$Age)
div_df_melt <- reshape2::melt(div.df2)

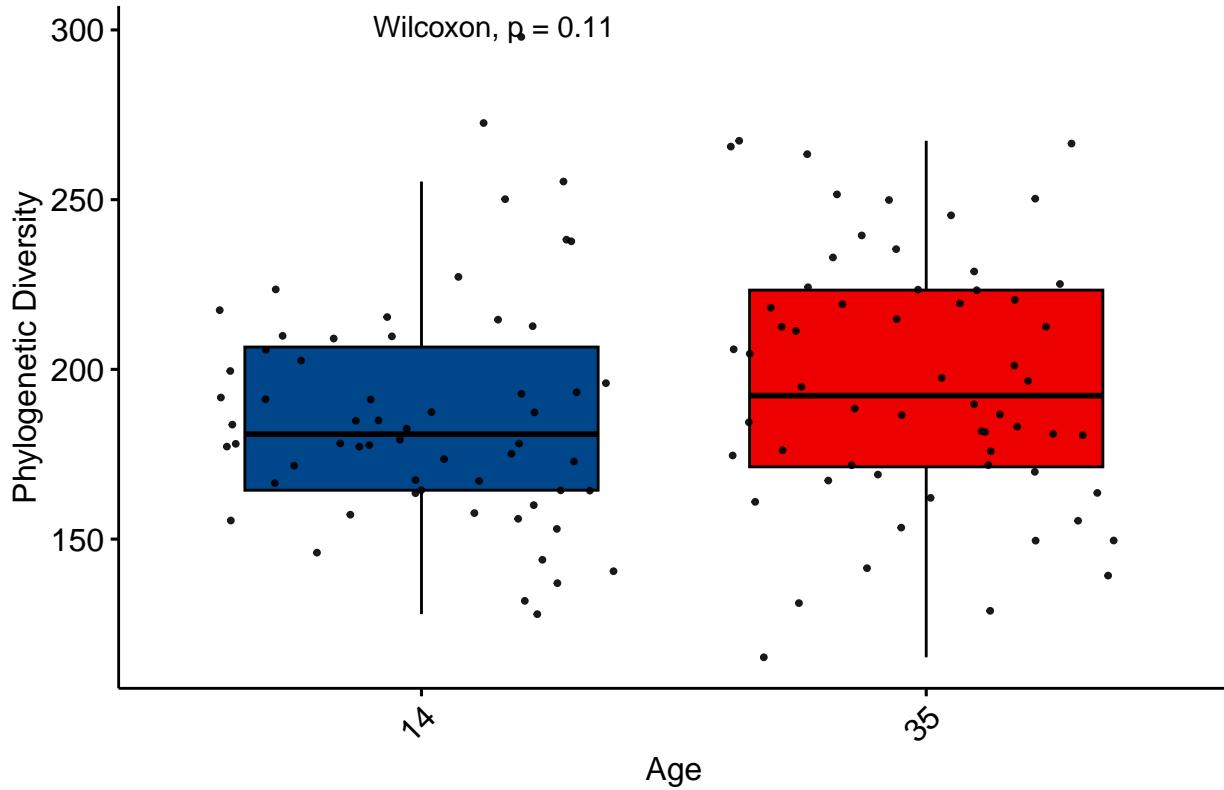
ggboxplot(div_df_melt, x = "Age", y = "value",
          fill = "Age",
          palette = "lancet",
          legend= "right",
          facet.by = "variable",
          scales = "free",
          title = "FPKM Alpha diversity metrics by age",
          outlier.shape = NA) +
  rremove("x.text") + stat_compare_means(method = "wilcox.test", size = 3.1) + geom_jitter(size = 0.7, alpha = 0.5)
```

FPKM Alpha diversity metrics by age



```
ggboxplot(hmp.meta,
  x = "Age",
  y = "Phylogenetic_Diversity",
  fill = "Age",
  palette = "lancet",
  ylab = "Phylogenetic Diversity",
  xlab = "Age",
  legend = "right",
  title = "FPKM phylogenetic diversity by age",
  outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) +
  stat_compare_means(method = "wilcox.test", paired = TRUE) + geom_jitter(size = 0.7, alpha = 0.9)
```

FPKM phylogenetic diversity by age



```
# farms / company

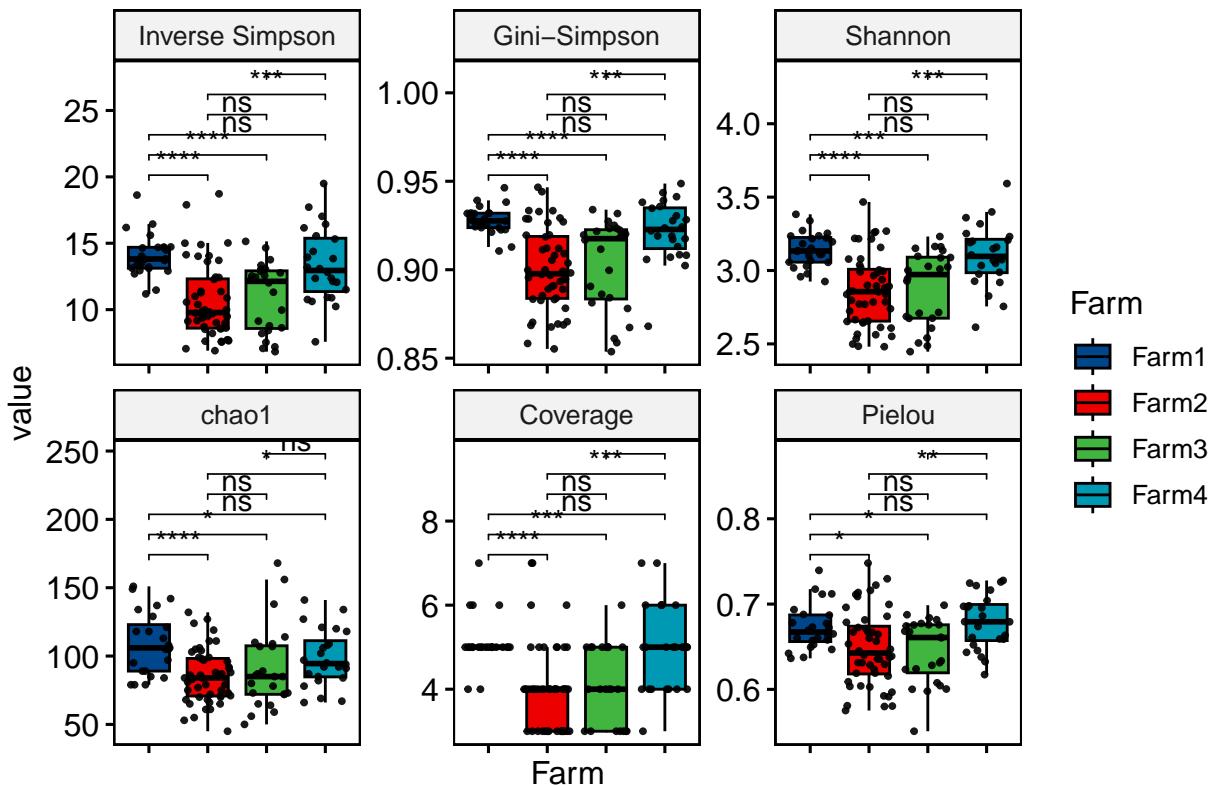
div.df2 <- div.df[, c("Farm2", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon")]
colnames(div.df2) <- c("Farm", "Inverse Simpson", "Gini-Simpson", "Shannon", "chao1", "Coverage", "Pielou")

div_df_melt <- reshape2::melt(div.df2)

lev = c("Farm1", "Farm2", "Farm3", "Farm4")
L.pairs <- combn(seq_along(lev), 2, simplify = FALSE, FUN = function(i) lev[i])

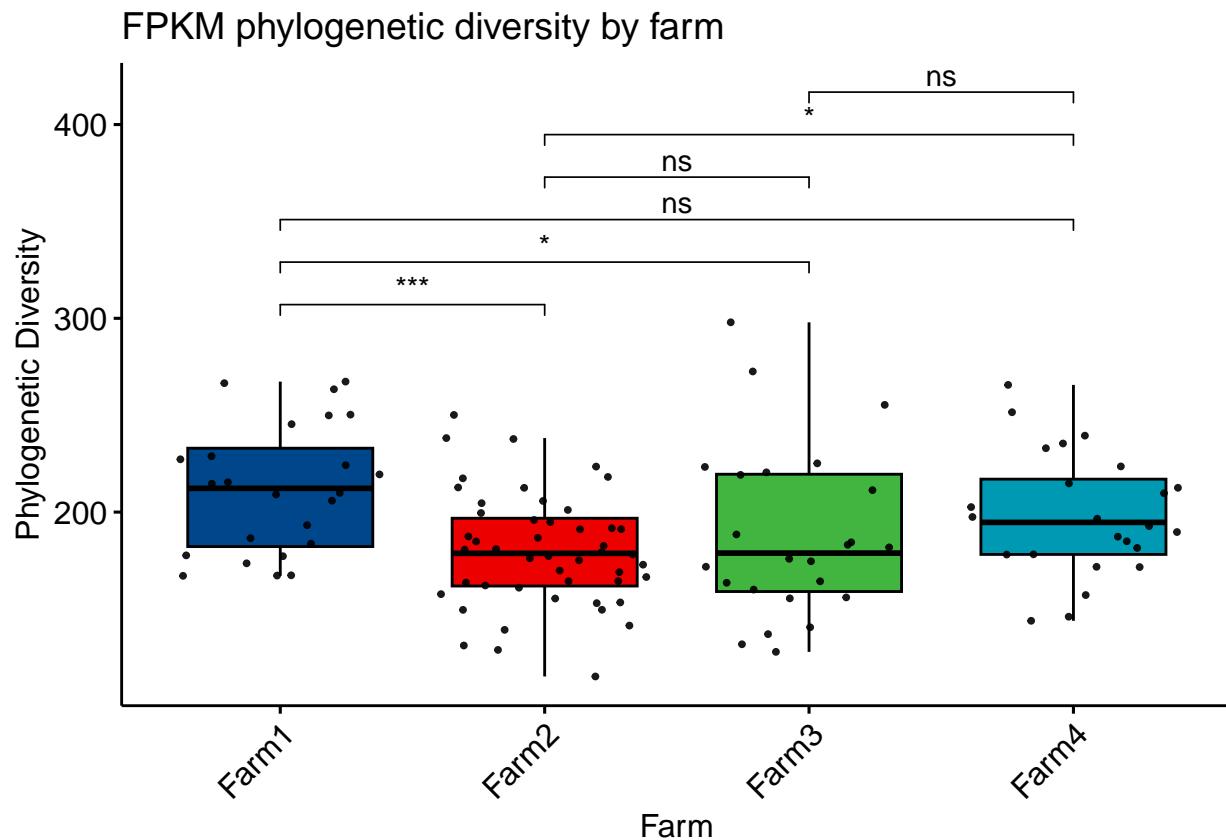
ggboxplot(div_df_melt, x = "Farm", y = "value",
          fill = "Farm",
          palette = "lancet",
          legend= "right",
          facet.by = "variable",
          scales = "free",
          order = lev,
          title = "FPKM Alpha diversity metrics by farm",
          outlier.shape = NA) + rotate_x_text() + rremove("x.text") +
stat_compare_means(method = "wilcox.test",
                    comparisons = L.pairs,
                    label = "p.signif"
) + geom_jitter(size = 0.7, alpha = 0.9)
```

FPKM Alpha diversity metrics by farm



```
df.pd <- pd(t(as.data.frame(Rps@otu_table)), Rps@phy_tree, include.root=T) # transposing for use in picante
hmp.meta$Phylogenetic_Diversity <- df.pd$PD
```

```
ggboxplot(hmp.meta,
          x = "Farm2",
          y = "Phylogenetic_Diversity",
          fill = "Farm2",
          palette = "lancet",
          ylab = "Phylogenetic Diversity",
          xlab = "Farm",
          legend = "right",
          order = lev,
          title = "FPKM phylogenetic diversity by farm",
          outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) +
  stat_compare_means(
    comparisons = L.pairs,
    label = "p.signif"
  ) + geom_jitter(size = 0.7, alpha = 0.9)
```



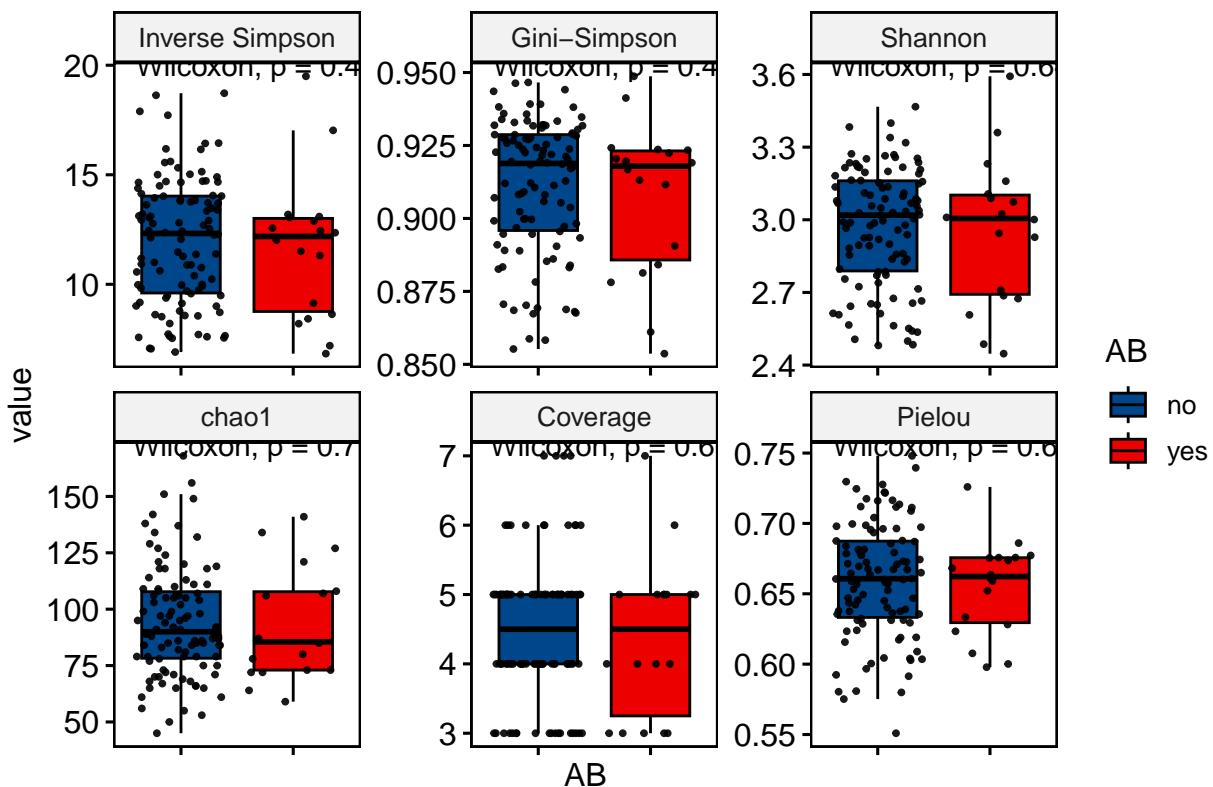
```
# based on AB

div.df2 <- div.df[, c("AB", "diversity_inverse_simpson", "diversity_gini_simpson", "diversity_shannon",
colnames(div.df2) <- c("AB", "Inverse Simpson", "Gini-Simpson", "Shannon", "chao1", "Coverage", "Pielou")

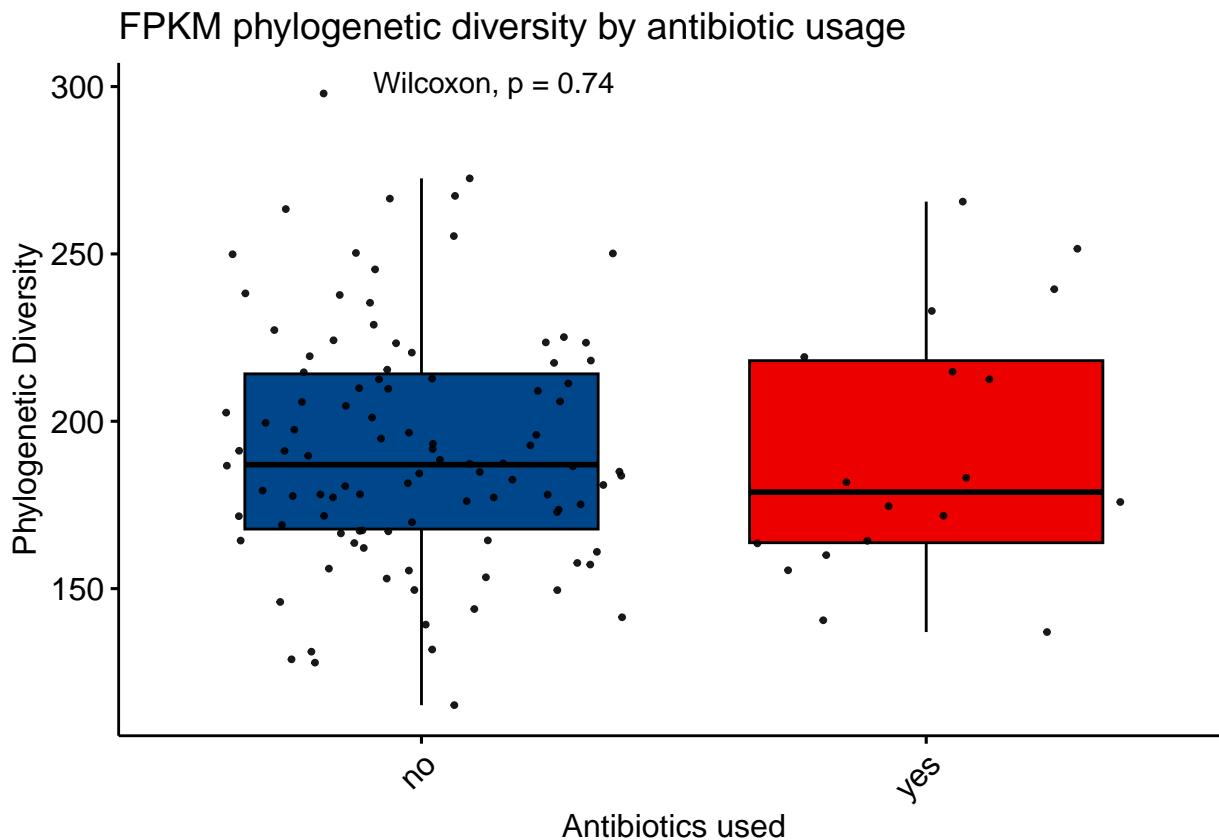
div_df_melt <- reshape2::melt(div.df2)

ggboxplot(div_df_melt, x = "AB", y = "value",
fill = "AB",
palette = "lancet",
legend= "right",
facet.by = "variable",
scales = "free",
title = "FPKM Alpha diversity metrics by antibiotic usage",
outlier.shape = NA) +
rremove("x.text") + stat_compare_means(
method = "wilcox.test") + geom_jitter(size = 0.7, alpha = 0.9)
```

FPKM Alpha diversity metrics by antibiotic usage



```
ggboxplot(hmp.meta,
  x = "AB",
  y = "Phylogenetic_Diversity",
  fill = "AB",
  palette = "lancet",
  ylab = "Phylogenetic Diversity",
  xlab = "Antibiotics used",
  legend = "right",
  title = "FPKM phylogenetic diversity by antibiotic usage",
  outlier.shape = NA) + rotate_x_text() +
  theme(legend.position="none", axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) +
  stat_compare_means() + geom_jitter(size = 0.7, alpha = 0.9)
```



```
# based on stable and age

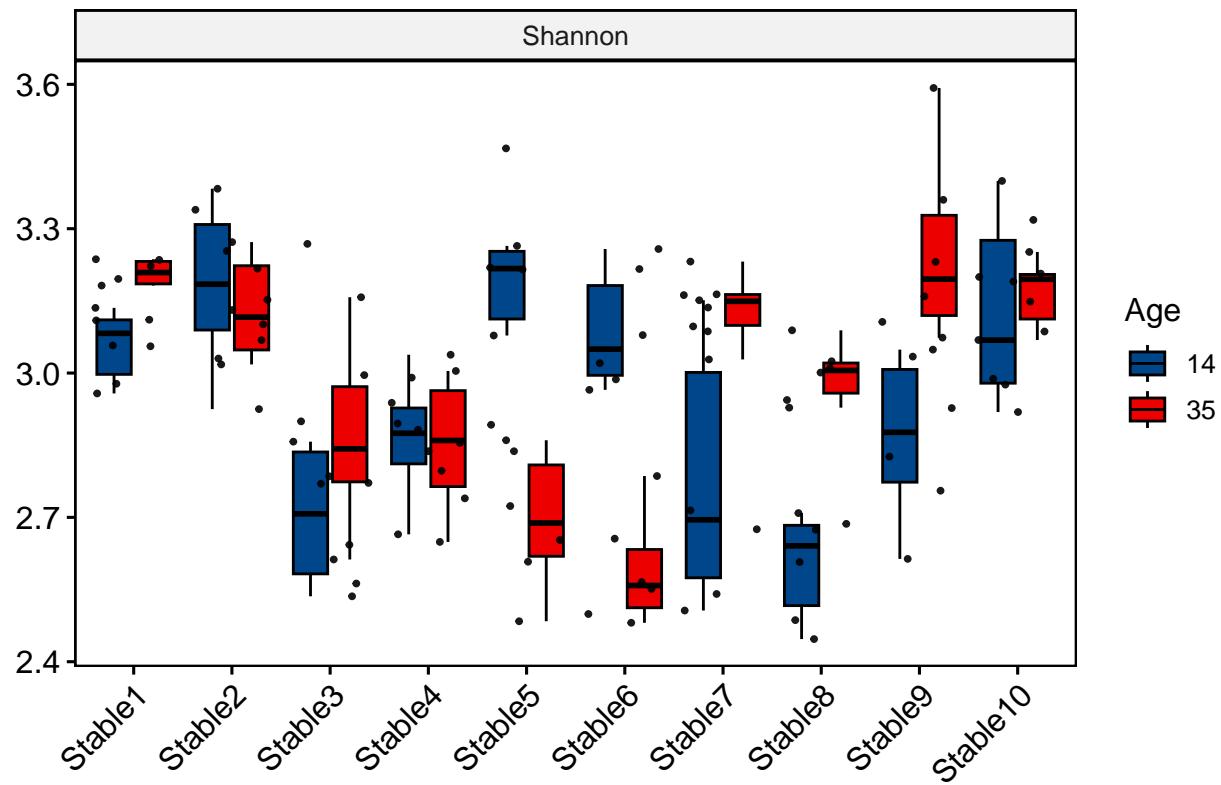
div.df2 <- div.df[, c("Stables", "Age", "diversity_shannon")]
colnames(div.df2) <- c("Stable", "Age", "Shannon")

div_df_melt <- reshape2::melt(div.df2)

lev = c("Stable1", "Stable2", "Stable3", "Stable4", "Stable5", "Stable6", "Stable7", "Stable8", "Stable9", "Stab

ggboxplot(div_df_melt, x = "Stable", y = "value",
  fill = "Age",
  palette = "lancet",
  legend= "right",
  facet.by = "variable",
  scales = "free",
  order = lev,
  title = "Shannon diversity by stable and age",
  xlab = FALSE,
  ylab = FALSE,
  outlier.shape = NA) + rotate_x_text() +
  theme(axis.text.x=element_text(angle=45,hjust=1,vjust=1,size=12)) + geom_jitter(size = 0.7, alpha = 0.5)
```

Shannon diversity by stable and age

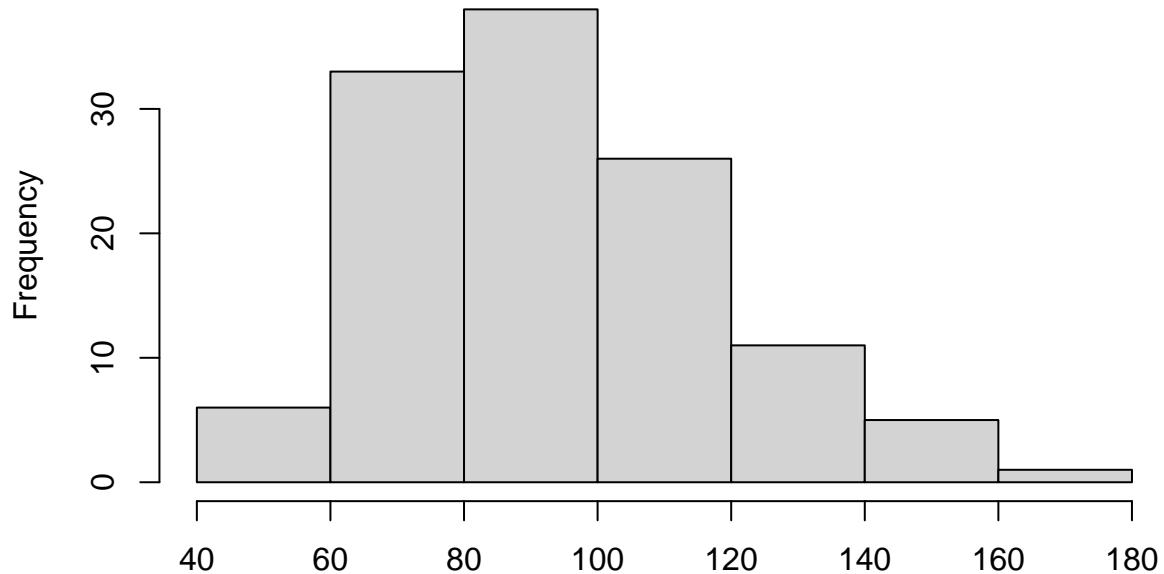


Looking at significance

Checking for normality

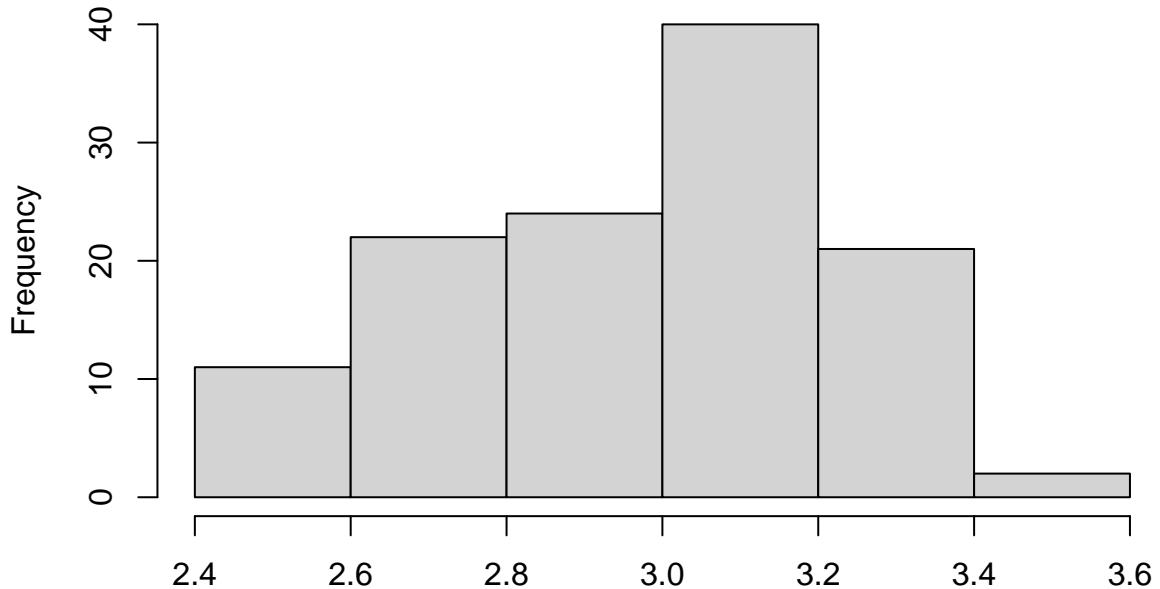
```
hist(lib.div$chao1, main="chao1 richness", xlab="")
```

chao1 richness



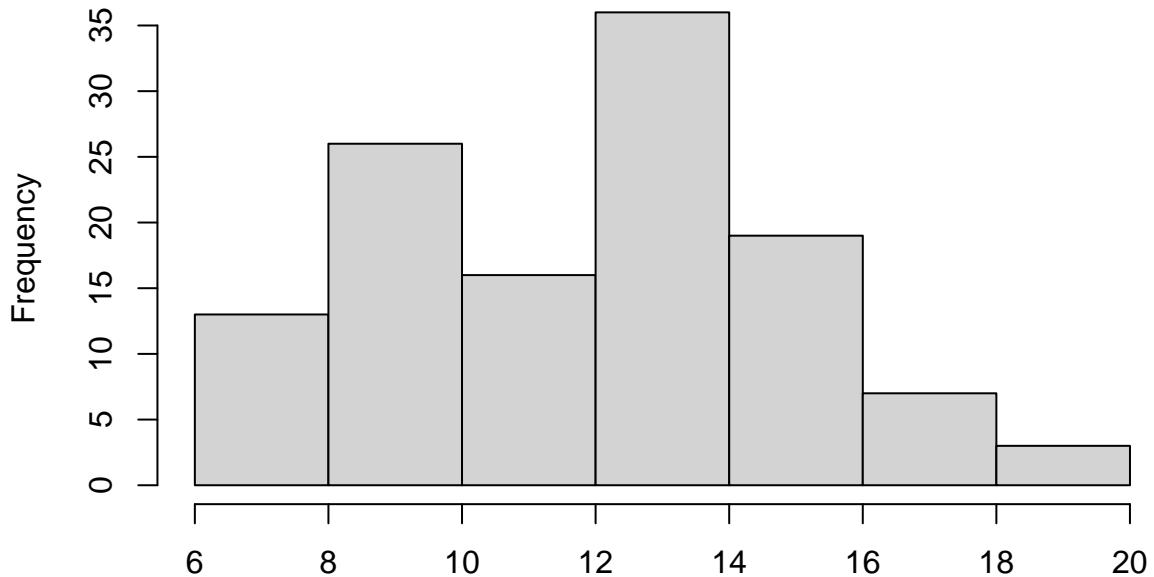
```
hist(lib.div$diversity_shannon, main="Shannon diversity", xlab="")
```

Shannon diversity



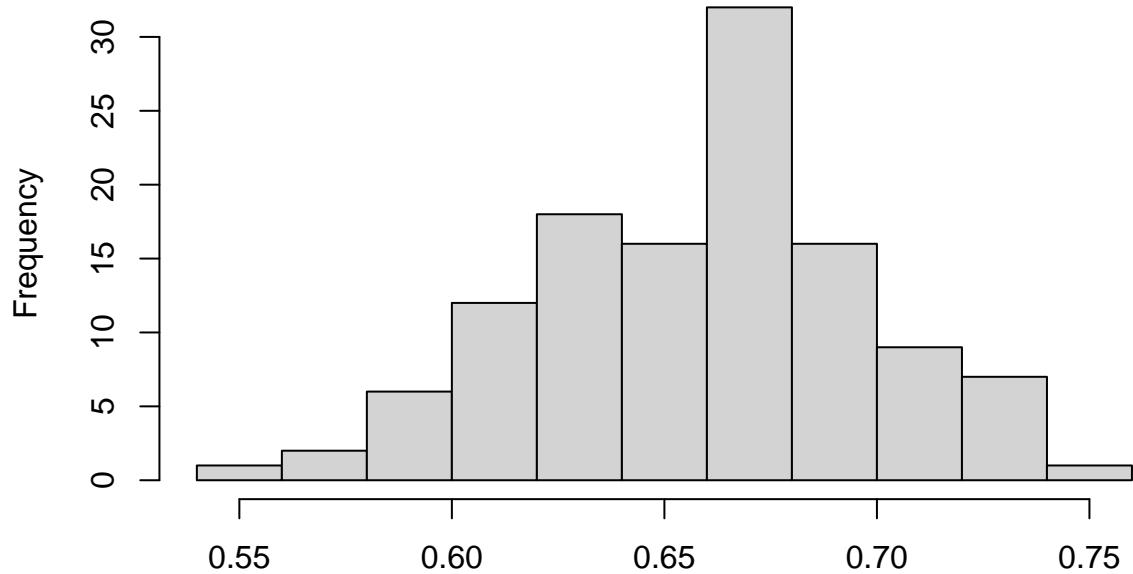
```
# Fisher diversity does not work when integers are not used  
hist(lib.div$diversity_inverse_simpson, main="Inverse Simpson evenness", xlab="")
```

Inverse Simpson evenness



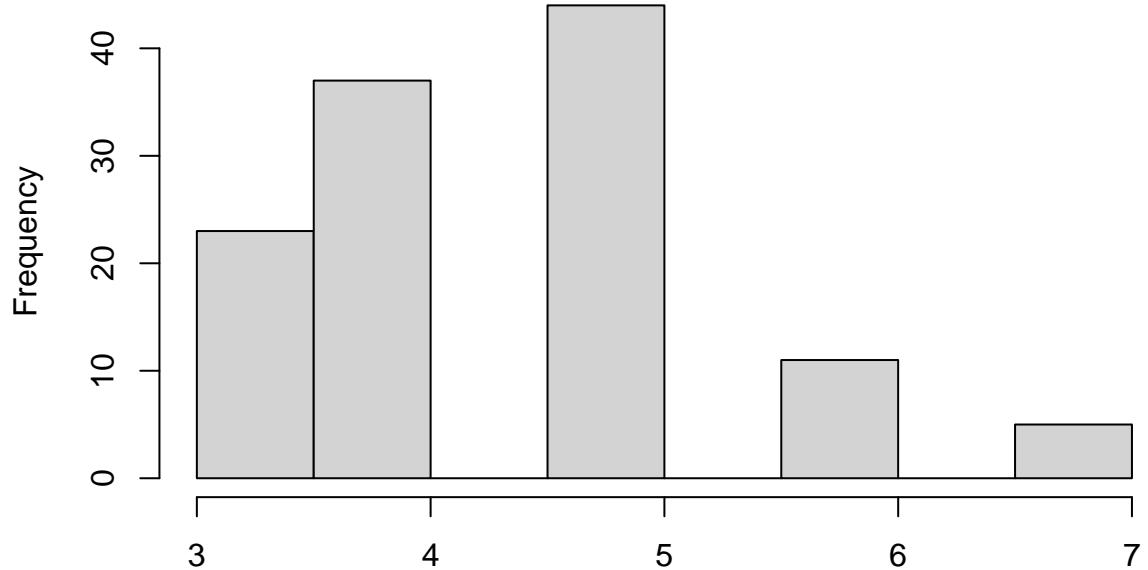
```
hist(lib.div$evenness_pielou, main="Pielou evenness", xlab="")
```

Pielou evenness



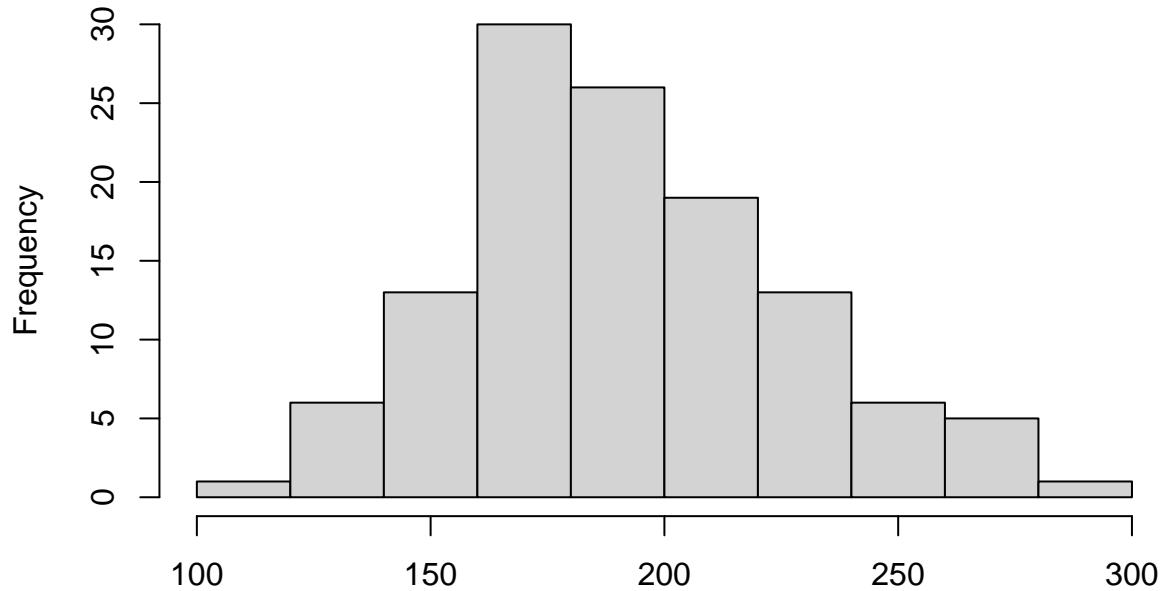
```
hist(lib.div$diversity_coverage, main="Coverage diversity", xlab="")
```

Coverage diversity



```
hist(lib.div$Phylogenetic_Diversity, main="Coverage diversity", xlab="")
```

Coverage diversity



```
# If data is normally distributed we can use ANOVA / t-tests, if not we will use Kruskal-Wallis tests  
# In this case, the data seems roughly normally distributed, we can use Shapiro-Wilk tests to test for  
shapiro.test(lib.div$chao1) # test deems it not normally distributed p<0,05
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: lib.div$chao1  
## W = 0.97275, p-value = 0.01539
```

```
shapiro.test(lib.div$diversity_shannon) # test deems this measure not normally distributed p<0,05
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: lib.div$diversity_shannon  
## W = 0.97091, p-value = 0.01055
```

```
#shapiro.test(lib.div$diversity_fisher) # impossible to calculate  
shapiro.test(lib.div$diversity_gini_simpson) # test deems this measure not normally distributed p<0,05
```

```
##  
## Shapiro-Wilk normality test  
##
```

```

## data: lib.div$diversity_gini_simpson
## W = 0.93447, p-value = 1.847e-05

shapiro.test(lib.div$diversity_inverse_simpson) # test deems this measure not normally distributed p<0,05

## 
## Shapiro-Wilk normality test
##
## data: lib.div$diversity_inverse_simpson
## W = 0.97581, p-value = 0.02916

shapiro.test(lib.div$evenness_pielou) # test deems this measure normally distributed p>0,05

## 
## Shapiro-Wilk normality test
##
## data: lib.div$evenness_pielou
## W = 0.99137, p-value = 0.6612

shapiro.test(lib.div$diversity_coverage) # test deems this measure not normally distributed p<0,05

## 
## Shapiro-Wilk normality test
##
## data: lib.div$diversity_coverage
## W = 0.89358, p-value = 9.34e-08

shapiro.test(lib.div$Phylogenetic_Diversity) # test deems this measure normally distributed p>0,05

## 
## Shapiro-Wilk normality test
##
## data: lib.div$Phylogenetic_Diversity
## W = 0.98144, p-value = 0.09698

# Fairly small sample sizes however, and the shapiro-wilk test is not perfect, we will assume normality
# The variables that we are interested in are the Age, which Farm the samples are from, and whether ant

# We will run ANOVAs for the normally distributed variables

# Age

# Normally distributed with only 2 levels, so we can use t-tests :

t.test(lib.div$evenness_pielou ~ sample_data(Rps)$Age) # not significant

## 
## Welch Two Sample t-test
##
## data: lib.div$evenness_pielou by sample_data(Rps)$Age

```

```

## t = 0.11577, df = 101.71, p-value = 0.9081
## alternative hypothesis: true difference in means between group 14 and group 35 is not equal to 0
## 95 percent confidence interval:
## -0.01364378 0.01533518
## sample estimates:
## mean in group 14 mean in group 35
## 0.6593399 0.6584942

t.test(lib.div$Phylogenetic_Diversity ~ sample_data(Rps)$Age) # not significant

## 
## Welch Two Sample t-test
##
## data: lib.div$Phylogenetic_Diversity by sample_data(Rps)$Age
## t = -1.4457, df = 116.81, p-value = 0.1509
## alternative hypothesis: true difference in means between group 14 and group 35 is not equal to 0
## 95 percent confidence interval:
## -22.24275 3.47167
## sample estimates:
## mean in group 14 mean in group 35
## 187.1559 196.5414

# Non-normally distributed

wilcox.test(lib.div$chao1 ~ sample_data(Rps)$Age) # not sign

## 
## Wilcoxon rank sum test with continuity correction
##
## data: lib.div$chao1 by sample_data(Rps)$Age
## W = 1512.5, p-value = 0.1319
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(lib.div$diversity_shannon ~ sample_data(Rps)$Age) # shannon diversity does not seem to sign

## 
## Wilcoxon rank sum test with continuity correction
##
## data: lib.div$diversity_shannon by sample_data(Rps)$Age
## W = 1628, p-value = 0.368
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(lib.div$diversity_gini_simpson ~ sample_data(Rps)$Age) # not significant

## 
## Wilcoxon rank sum test with continuity correction
##
## data: lib.div$diversity_gini_simpson by sample_data(Rps)$Age
## W = 1810, p-value = 0.9602
## alternative hypothesis: true location shift is not equal to 0

```

```
wilcox.test(lib.div$diversity_inverse_simpson ~ sample_data(Rps)$Age) # not significant
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: lib.div$diversity_inverse_simpson by sample_data(Rps)$Age  
## W = 1810, p-value = 0.9602  
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(lib.div$diversity_coverage ~ sample_data(Rps)$Age) # not significant
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: lib.div$diversity_coverage by sample_data(Rps)$Age  
## W = 1758, p-value = 0.8197  
## alternative hypothesis: true location shift is not equal to 0
```

For age, none of the groups significantly differ

Antibiotics

```
t.test(lib.div$Phylogenetic_Diversity ~ sample_data(Rps)$AB) # not significant
```

```
##  
## Welch Two Sample t-test  
##  
## data: lib.div$Phylogenetic_Diversity by sample_data(Rps)$AB  
## t = 0.060704, df = 22.334, p-value = 0.9521  
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0  
## 95 percent confidence interval:  
## -19.61118 20.79492  
## sample estimates:  
## mean in group no mean in group yes  
## 191.9374 191.3456
```

```
t.test(lib.div$evenness_pielou ~ sample_data(Rps)$AB) # not significant
```

```
##  
## Welch Two Sample t-test  
##  
## data: lib.div$evenness_pielou by sample_data(Rps)$AB  
## t = 0.55959, df = 26.811, p-value = 0.5804  
## alternative hypothesis: true difference in means between group no and group yes is not equal to 0  
## 95 percent confidence interval:  
## -0.01326442 0.02320819  
## sample estimates:  
## mean in group no mean in group yes  
## 0.65966628 0.6546909
```

```
wilcox.test(lib.div$diversity_shannon ~ sample_data(Rps)$AB) # shannon diversity does not seem to signif
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: lib.div$diversity_shannon by sample_data(Rps)$AB  
## W = 982, p-value = 0.6407  
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(lib.div$chao1 ~ sample_data(Rps)$AB)
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: lib.div$chao1 by sample_data(Rps)$AB  
## W = 958.5, p-value = 0.7687  
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(lib.div$diversity_gini_simpson ~ sample_data(Rps)$AB) # not sign
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: lib.div$diversity_gini_simpson by sample_data(Rps)$AB  
## W = 1033, p-value = 0.4001  
## alternative hypothesis: true location shift is not equal to 0
```

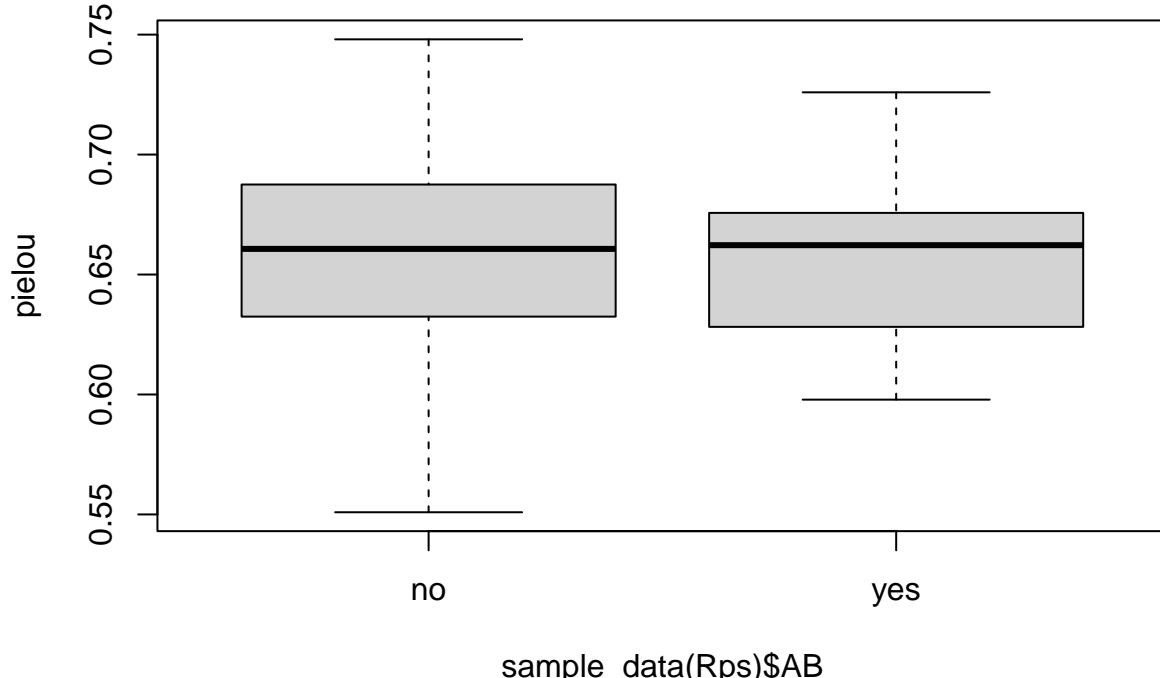
```
wilcox.test(lib.div$diversity_inverse_simpson ~ sample_data(Rps)$AB) # not significant
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: lib.div$diversity_inverse_simpson by sample_data(Rps)$AB  
## W = 1033, p-value = 0.4001  
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(lib.div$diversity_coverage ~ sample_data(Rps)$AB)
```

```
##  
## Wilcoxon rank sum test with continuity correction  
##  
## data: lib.div$diversity_coverage by sample_data(Rps)$AB  
## W = 973.5, p-value = 0.6724  
## alternative hypothesis: true location shift is not equal to 0
```

```
boxplot(lib.div$evenness_pielou ~ sample_data(Rps)$AB, ylab="pielou") # the boxplots are quite similar
```



```
# AB does not seem to significantly differ in their alpha diversities
# normally distributed
# Farm has more than 2 levels, so we will use ANOVAs
aov.pd.farm = aov(lib.div$Phylogenetic_Diversity ~ sample_data(Rps)$Farm2)
summary(aov.pd.farm)
```

```
##                                Df Sum Sq Mean Sq F value    Pr(>F)
## sample_data(Rps)$Farm2     3 17895   5965   5.166 0.00218 **
## Residuals                  116 133943    1155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aov.pd.farm) # only difference between 2 and 1
```

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lib.div$Phylogenetic_Diversity ~ sample_data(Rps)$Farm2)
##
## $`sample_data(Rps)$Farm2`
##          diff      lwr      upr     p adj
## Farm2-Farm1 -32.061495 -54.205526 -9.917464 0.0014365
## Farm3-Farm1 -23.717827 -49.287551  1.851898 0.0793254
```

```

## Farm4-Farm1 -13.578515 -39.148239 11.991210 0.5117422
## Farm3-Farm2  8.343668 -13.800363 30.487699 0.7598745
## Farm4-Farm2  18.482980 -3.661051 40.627011 0.1361507
## Farm4-Farm3  10.139312 -15.430413 35.709036 0.7300928

aov.pielou.farm = aov(lib.div$evenness_pielou ~ sample_data(Rps)$Farm2)
summary(aov.pielou.farm)

```

```

##                               Df  Sum Sq  Mean Sq F value Pr(>F)
## sample_data(Rps)$Farm2     3 0.02383 0.007942   5.58 0.0013 **
## Residuals                  116 0.16509 0.001423
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```
TukeyHSD(aov.pielou.farm) # only difference between 4 and 2 and 4 and 3
```

```

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = lib.div$evenness_pielou ~ sample_data(Rps)$Farm2)
##
## $`sample_data(Rps)$Farm2`
##      diff          lwr          upr      p adj
## Farm2-Farm1 -0.024188099 -0.048772296 0.0003960974 0.0555715
## Farm3-Farm1 -0.027460151 -0.055847537 0.0009272338 0.0618334
## Farm4-Farm1  0.006198412 -0.022188973 0.0345857974 0.9410499
## Farm3-Farm2 -0.003272052 -0.027856249 0.0213121448 0.9855840
## Farm4-Farm2  0.030386512  0.005802315 0.0549707084 0.0088564
## Farm4-Farm3  0.033658564  0.005271178 0.0620459489 0.0131688

```

```
# Non-normally distributed
```

```
kruskal.test(lib.div$chao1 ~ sample_data(Rps)$Farm2)
```

```

##
## Kruskal-Wallis rank sum test
##
## data: lib.div$chao1 by sample_data(Rps)$Farm2
## Kruskal-Wallis chi-squared = 17.587, df = 3, p-value = 0.0005352

```

```
pairwise.wilcox.test(lib.div$chao1, sample_data(Rps)$Farm2, p.adjust.method="fdr") # it seems that only
```

```

##
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data: lib.div$chao1 and sample_data(Rps)$Farm2
##
##      Farm1    Farm2    Farm3
## Farm2 0.00044 -      -
## Farm3 0.02910 0.68895 -
## Farm4 0.17841 0.02910 0.17548
##
## P value adjustment method: fdr

```

```
kruskal.test(lib.div$diversity_shannon ~ sample_data(Rps)$Farm2)

## 
##  Kruskal-Wallis rank sum test
##
##  data: lib.div$diversity_shannon by sample_data(Rps)$Farm2
##  Kruskal-Wallis chi-squared = 29.664, df = 3, p-value = 1.624e-06
```

```
pairwise.wilcox.test(lib.div$diversity_shannon, sample_data(Rps)$Farm2, p.adjust.method="fdr") # farm 1

## 
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
##  data: lib.div$diversity_shannon and sample_data(Rps)$Farm2
##
##      Farm1     Farm2     Farm3
## Farm2 6.9e-06 -      
## Farm3 0.00034 0.62245 -
## Farm4 0.59184 0.00034 0.00830
##
##  P value adjustment method: fdr
```

```
kruskal.test(lib.div$diversity_gini_simpson ~ sample_data(Rps)$Farm2)
```

```
## 
##  Kruskal-Wallis rank sum test
##
##  data: lib.div$diversity_gini_simpson by sample_data(Rps)$Farm2
##  Kruskal-Wallis chi-squared = 32.804, df = 3, p-value = 3.542e-07
```

```
pairwise.wilcox.test(lib.div$diversity_gini_simpson, sample_data(Rps)$Farm2, p.adjust.method="fdr") # s

## 
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
##  data: lib.div$diversity_gini_simpson and sample_data(Rps)$Farm2
##
##      Farm1     Farm2     Farm3
## Farm2 2.8e-06 -      
## Farm3 2.1e-05 0.54105 -
## Farm4 0.18371 0.00021 0.02164
##
##  P value adjustment method: fdr
```

```
kruskal.test(lib.div$diversity_inverse_simpson ~ sample_data(Rps)$Farm2) # not significant
```

```
## 
##  Kruskal-Wallis rank sum test
##
##  data: lib.div$diversity_inverse_simpson by sample_data(Rps)$Farm2
##  Kruskal-Wallis chi-squared = 32.804, df = 3, p-value = 3.542e-07
```

```

pairwise.wilcox.test(lib.div$diversity_inverse_simpson, sample_data(Rps)$Farm2, p.adjust.method="fdr")

##
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$diversity_inverse_simpson and sample_data(Rps)$Farm2
##
##      Farm1   Farm2   Farm3
## Farm2 2.8e-06 -      -
## Farm3 2.1e-05 0.54105 -
## Farm4 0.18371 0.00021 0.02164
##
## P value adjustment method: fdr

kruskal.test(lib.div$evenness_pielou ~ sample_data(Rps)$Farm2)

##
##  Kruskal-Wallis rank sum test
##
## data: lib.div$evenness_pielou by sample_data(Rps)$Farm2
## Kruskal-Wallis chi-squared = 13.779, df = 3, p-value = 0.003221

pairwise.wilcox.test(lib.div$evenness_pielou, sample_data(Rps)$Farm2, p.adjust.method="fdr")

##
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
## data: lib.div$evenness_pielou and sample_data(Rps)$Farm2
##
##      Farm1 Farm2 Farm3
## Farm2 0.022 -      -
## Farm3 0.069 0.967 -
## Farm4 0.475 0.018 0.018
##
## P value adjustment method: fdr

# In addition, it could be interesting to look at the concentration of DNA as a continuous variable

# Normally distributed

glm.chao1.age = glm(lib.div$chao1 ~ sample_data(Rps)$Conc...ng..μl.)
summary(glm.chao1.age)

##
## Call:
## glm(formula = lib.div$chao1 ~ sample_data(Rps)$Conc...ng..μl.)
##
## Coefficients:
## (Intercept)             Estimate Std. Error t value Pr(>|t|)
## 88.94210      5.60816   15.859   <2e-16 ***

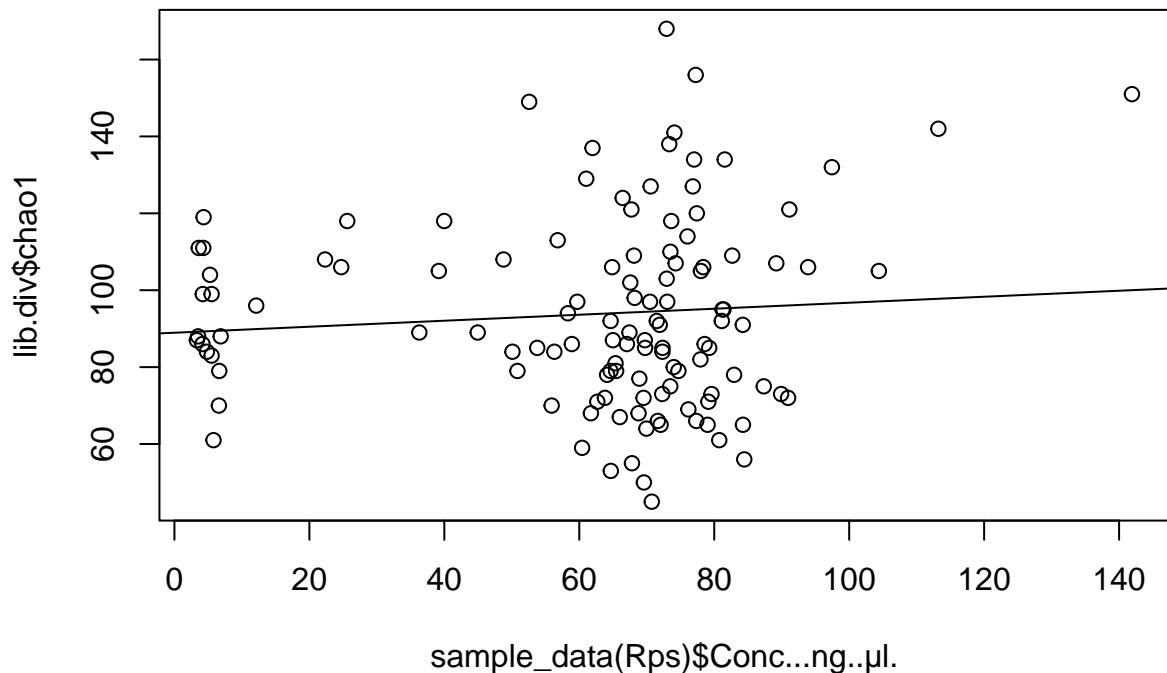
```

```

## sample_data(Rps)$Conc...ng..μl. 0.07800   0.08318   0.938     0.35
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 597.4756)
##
## Null deviance: 71027 on 119 degrees of freedom
## Residual deviance: 70502 on 118 degrees of freedom
## AIC: 1111.7
##
## Number of Fisher Scoring iterations: 2

plot(lib.div$chao1 ~ sample_data(Rps)$Conc...ng..μl.)
abline(glm.chao1.age)

```



```

glm.shannon.age = glm(lib.div$diversity_shannon ~ sample_data(Rps)$Conc...ng..μl.)
summary(glm.shannon.age)

```

```

##
## Call:
## glm(formula = lib.div$diversity_shannon ~ sample_data(Rps)$Conc...ng..μl.)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.8761921  0.0569751 50.482 <2e-16 ***

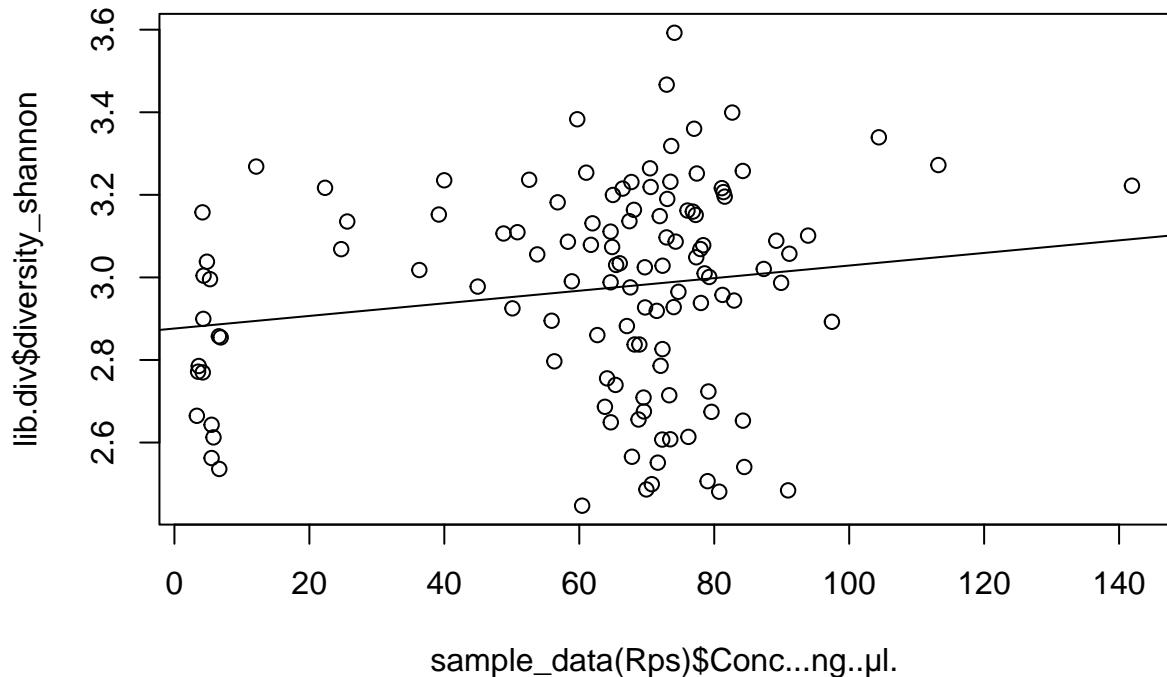
```

```

## sample_data(Rps)$Conc...ng..μl. 0.0015253 0.0008451 1.805 0.0736 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.06166658)
##
## Null deviance: 7.4775 on 119 degrees of freedom
## Residual deviance: 7.2767 on 118 degrees of freedom
## AIC: 10.207
##
## Number of Fisher Scoring iterations: 2

plot(lib.div$diversity_shannon ~ sample_data(Rps)$Conc...ng..μl.)
abline(glm.shannon.age)

```



```

glm.gini_simpson.age = glm(lib.div$diversity_gini_simpson ~ sample_data(Rps)$Conc...ng..μl.)
summary(glm.gini_simpson.age)

```

```

##
## Call:
## glm(formula = lib.div$diversity_gini_simpson ~ sample_data(Rps)$Conc...ng..μl.)
##
## Coefficients:
## (Intercept) Estimate Std. Error t value Pr(>|t|)
## 9.018e-01 5.324e-03 169.380 <2e-16 ***

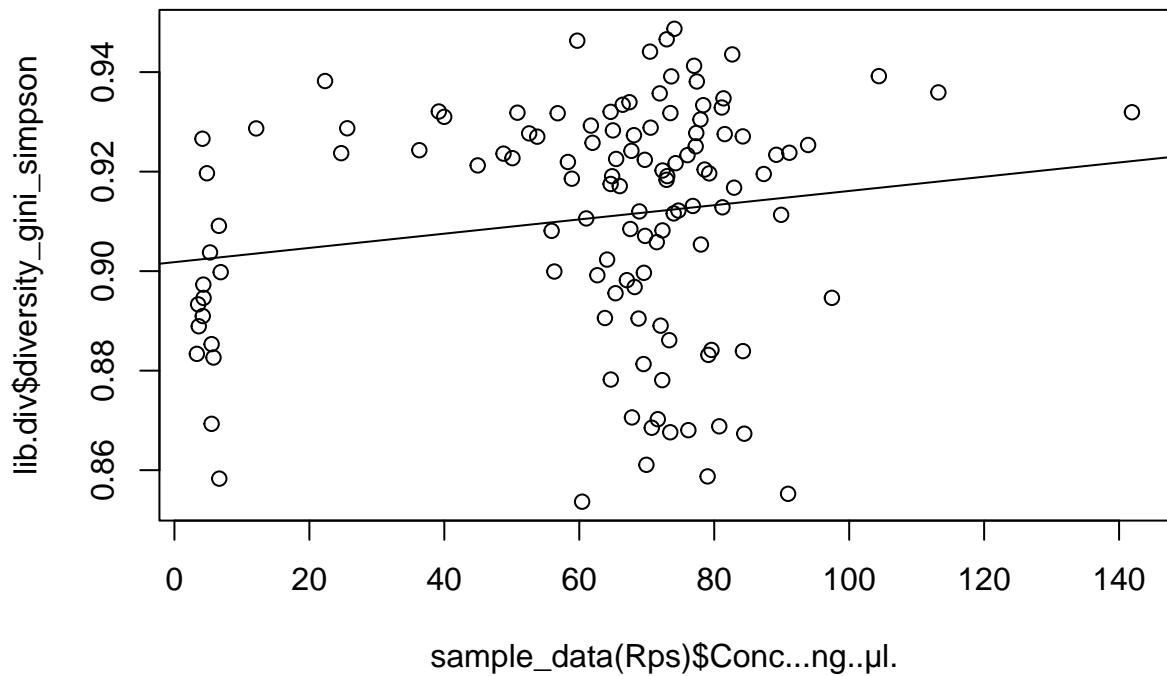
```

```

## sample_data(Rps)$Conc...ng..pl. 1.431e-04 7.897e-05 1.812 0.0726 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.0005384943)
##
## Null deviance: 0.065310 on 119 degrees of freedom
## Residual deviance: 0.063542 on 118 degrees of freedom
## AIC: -558.68
##
## Number of Fisher Scoring iterations: 2

plot(lib.div$diversity_gini_simpson ~ sample_data(Rps)$Conc...ng..pl.)
abline(glm.gini_simpson.age)

```



```

glm.inv_simpson.age = glm(lib.div$diversity_inverse_simpson ~ sample_data(Rps)$Conc...ng..pl.)
summary(glm.inv_simpson.age)

```

```

##
## Call:
## glm(formula = lib.div$diversity_inverse_simpson ~ sample_data(Rps)$Conc...ng..pl.)
##
## Coefficients:
## (Intercept) Estimate Std. Error t value Pr(>|t|)
## 10.606163   0.661297  16.038 <2e-16 ***

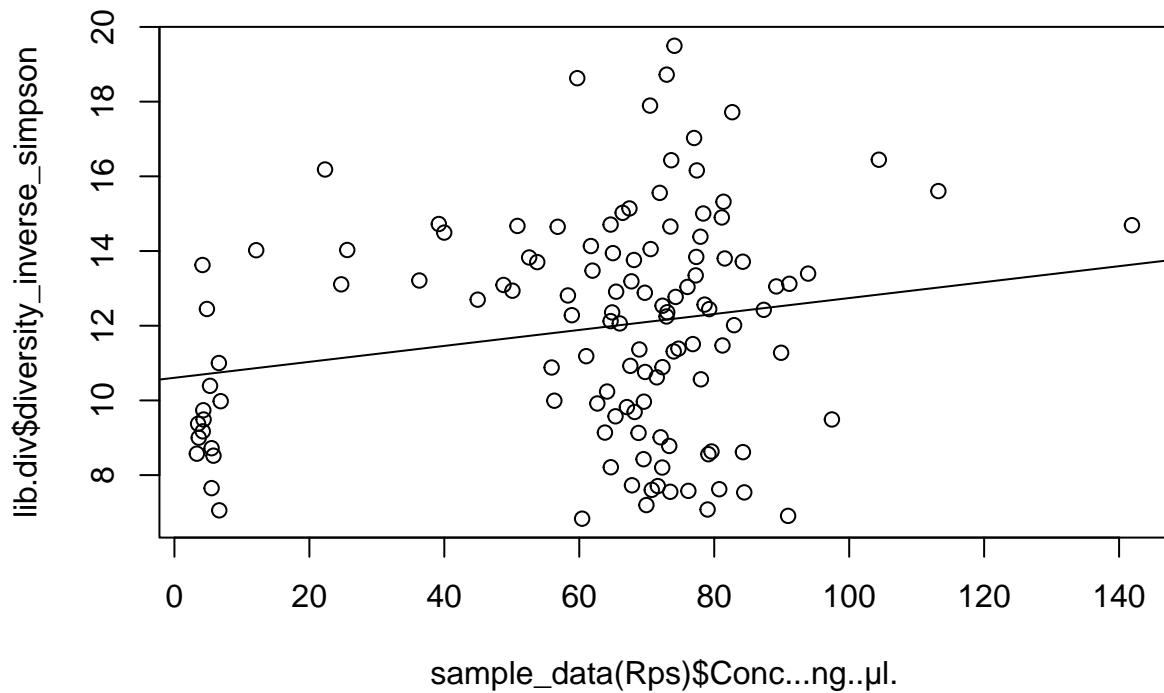
```

```

## sample_data(Rps)$Conc...ng..μl. 0.021346  0.009809  2.176  0.0315 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 8.307548)
##
## Null deviance: 1019.63 on 119 degrees of freedom
## Residual deviance: 980.29 on 118 degrees of freedom
## AIC: 598.59
##
## Number of Fisher Scoring iterations: 2

plot(lib.div$diversity_inverse_simpson ~ sample_data(Rps)$Conc...ng..μl.)
abline(glm.inv_simpson.age)

```



```

glm.pielou.age = glm(lib.div$evenness_pielou ~ sample_data(Rps)$Conc...ng..μl.)
summary(glm.pielou.age)

```

```

##
## Call:
## glm(formula = lib.div$evenness_pielou ~ sample_data(Rps)$Conc...ng..μl.)
##
## Coefficients:
## (Intercept) Estimate Std. Error t value Pr(>|t|)
## 0.6419624  0.0090212  71.162 <2e-16 ***

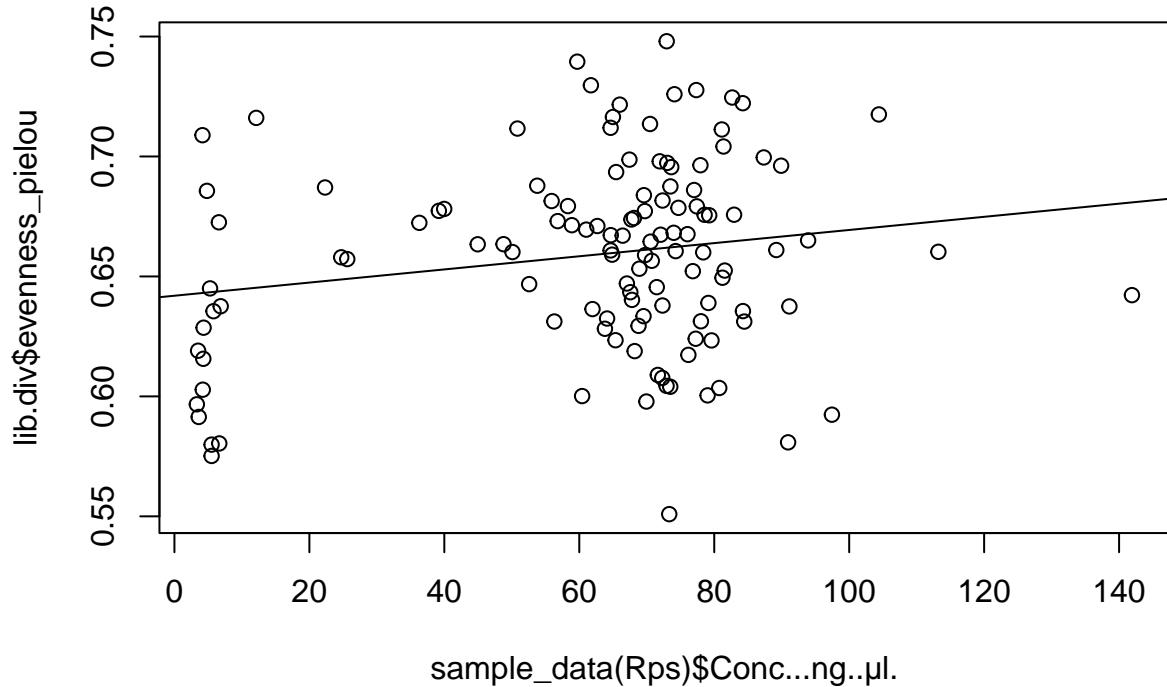
```

```

## sample_data(Rps)$Conc...ng..pl. 0.0002741 0.0001338 2.049 0.0427 *
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.001545991)
##
## Null deviance: 0.18891 on 119 degrees of freedom
## Residual deviance: 0.18243 on 118 degrees of freedom
## AIC: -432.12
##
## Number of Fisher Scoring iterations: 2

plot(lib.div$evenness_pielou ~ sample_data(Rps)$Conc...ng..pl.)
abline(glm.pielou.age)

```

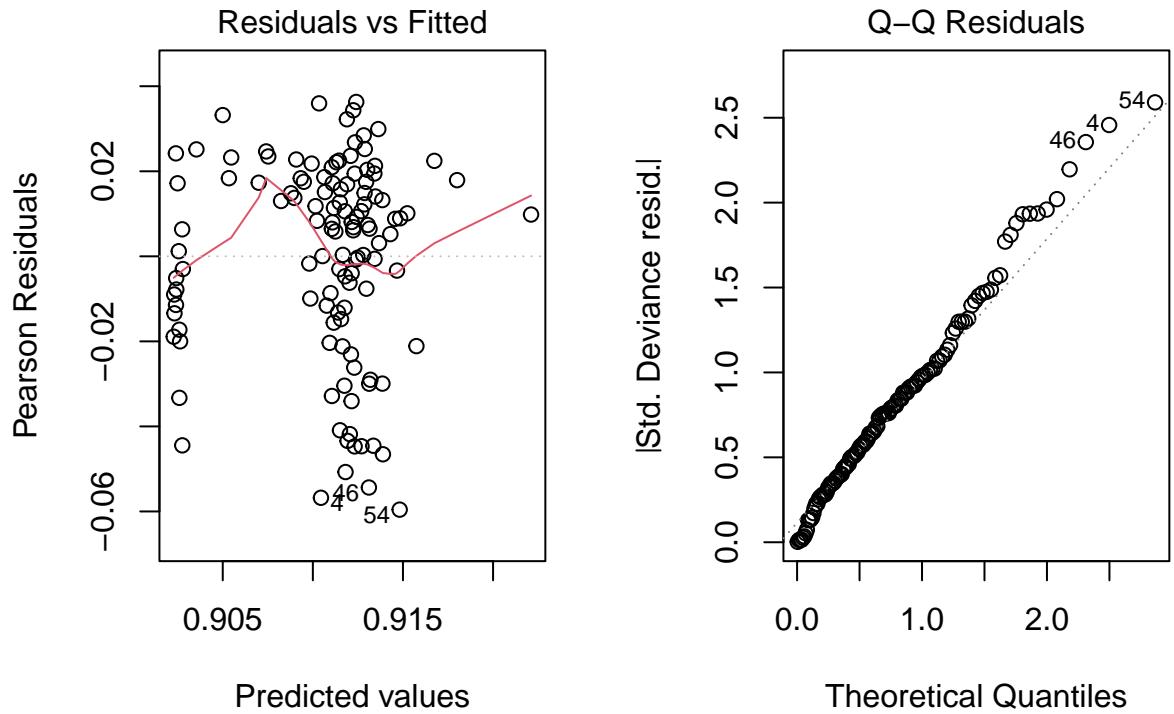


```

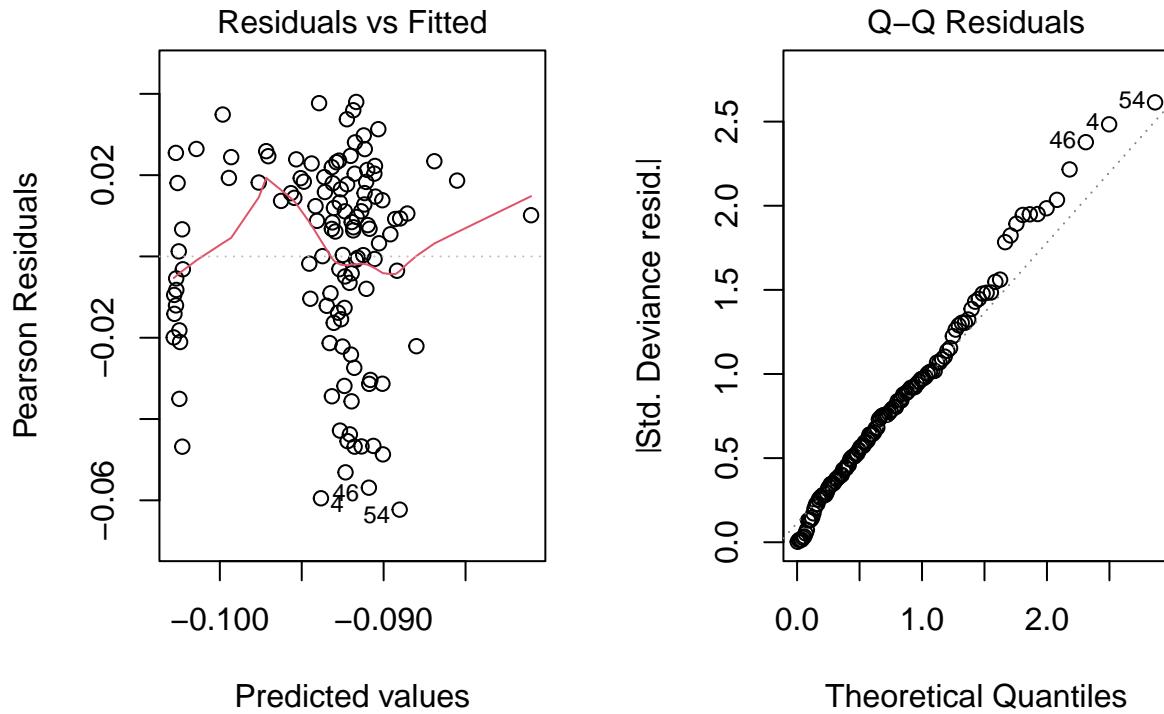
# Non-normally distributed

gaussian.gini_simpson.conc = glm(lib.div$diversity_gini_simpson ~ sample_data(Rps)$Conc...ng..pl., family=gaussian)
par(mfrow = c(1,2))
plot(gaussian.gini_simpson.conc, which=c(1,2))

```



```
qp.gini_simpson.conc = glm(lib.div$diversity_gini_simpson ~ sample_data(Rps)$Conc...ng..pl., family="qua
par(mfrow = c(1,2))
plot(qp.gini_simpson.conc, which=c(1,2)) # there are no huge differences between normal model (gaussian)
```

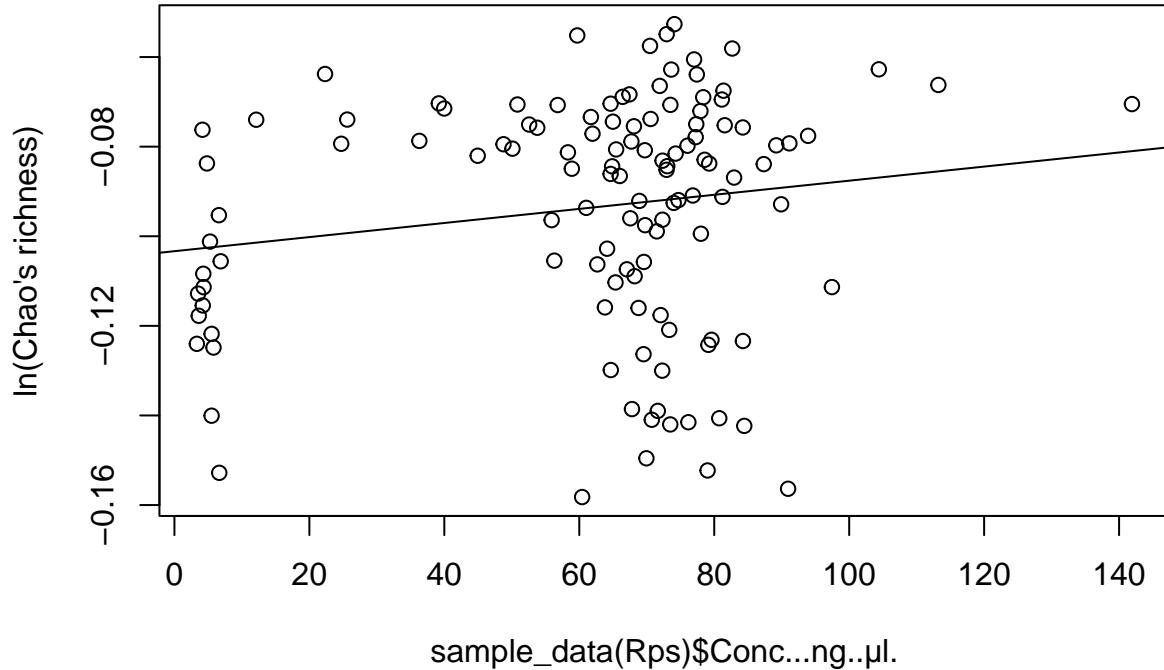


```
summary(qp.gini_simpson.conc)
```

```
##
## Call:
## glm(formula = lib.div$diversity_gini_simpson ~ sample_data(Rps)$Conc...ng..μl.,
##      family = "quasipoisson")
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -1.033e-01  5.863e-03 -17.624 <2e-16 ***
## sample_data(Rps)$Conc...ng..μl.  1.574e-04  8.686e-05   1.812  0.0725 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 0.000591075)
##
## Null deviance: 0.072157 on 119 degrees of freedom
## Residual deviance: 0.070213 on 118 degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

```
par(mfrow = c(1, 1))
#Plot
```

```
plot(log(lib.div$diversity_gini_simpson) ~ sample_data(Rps)$Conc...ng..μl., ylab="ln(Chao's richness)")
abline(qp.gini_simpson.conc)
```



Beta diversity

```
# Visualizing different kinds of ordination methods with BC distance matrix
# DPCoA does not work for resistome data
#estimate_richness(Rps) unrounded data will not find any singletons, because no integers

dist = "bray"
ord_meths = c("DCA", "CCA", "RDA", "NMDS", "MDS", "PCoA")
plist = lapply(as.list(ord_meths), function(i, physeq, dist){
  ordi = ordinate(physeq, method=i, distance=dist)
  plot_ordination(physeq, ordi, "samples", color="Age", shape = "AB")
}, Rps, dist)

## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.216128
## Run 1 stress 0.2177816
## Run 2 stress 0.2168438
## Run 3 stress 0.2171573
## Run 4 stress 0.2215411
```

```

## Run 5 stress 0.2169662
## Run 6 stress 0.2203929
## Run 7 stress 0.2171433
## Run 8 stress 0.2209986
## Run 9 stress 0.2175059
## Run 10 stress 0.2186797
## Run 11 stress 0.2167554
## Run 12 stress 0.251327
## Run 13 stress 0.2175258
## Run 14 stress 0.2161289
## ... Procrustes: rmse 0.001516887 max resid 0.01528869
## Run 15 stress 0.2175058
## Run 16 stress 0.2175059
## Run 17 stress 0.2167521
## Run 18 stress 0.2175236
## Run 19 stress 0.2175191
## Run 20 stress 0.2187365
## *** Best solution was not repeated -- monoMDS stopping criteria:
##      20: stress ratio > sratmax

```

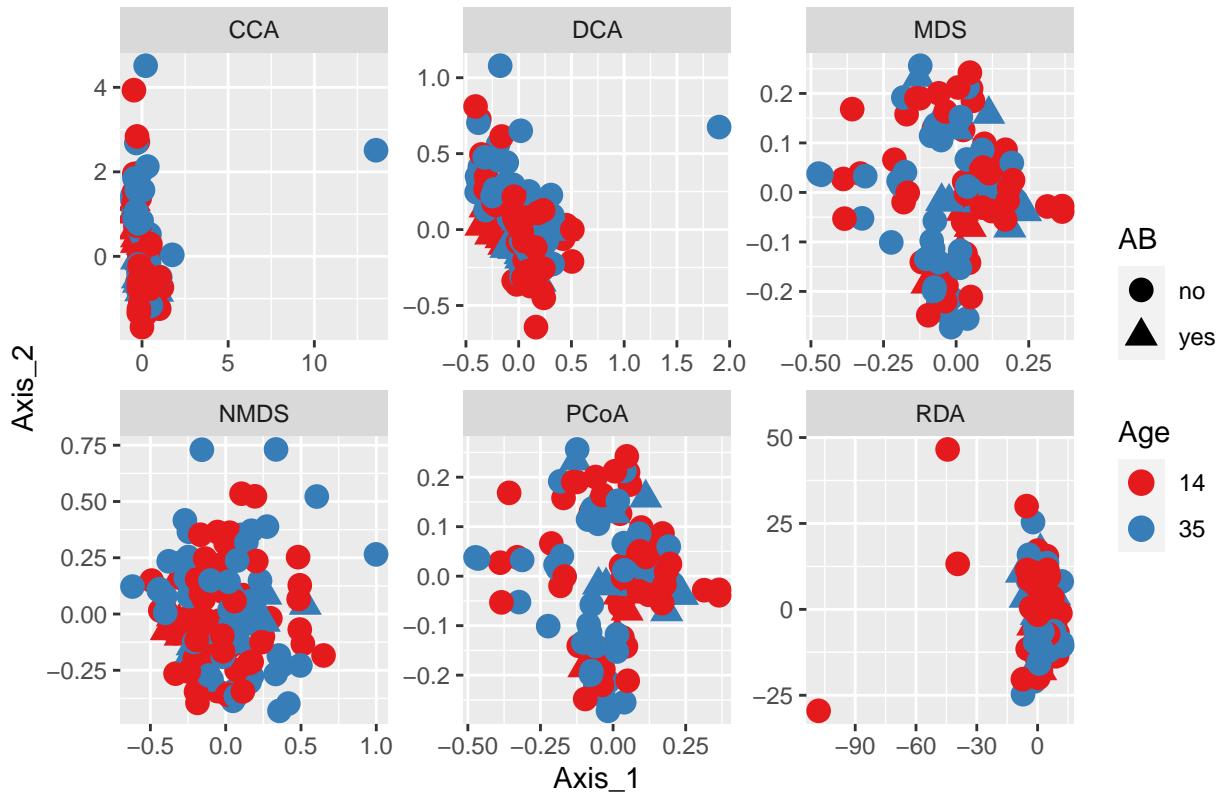
```

names(plist) <- ord_meths

pdataframe = ldply(plist, function(x){
  df = x$data[, 1:2]
  colnames(df) = c("Axis_1", "Axis_2")
  return(cbind(df, x$data))
})
names(pdataframe)[1] = "method"
ggplot(pdataframe, aes(Axis_1, Axis_2, color=Age, shape=AB)) +
  geom_point(size=4) +
  facet_wrap(~method, scales="free") +
  scale_fill_brewer(type="qual", palette="Set1") +
  scale_colour_brewer(type="qual", palette="Set1") +
  ggtitle("Different ordination methods for resistomic data (Bray-Curtis)")

```

Different ordination methods for resistomic data (Bray–Curtis)



```
# Repeat for mp
```

```
plist = llply(as.list(ord_meths), function(i, physeq, dist){
  ordi = ordinate(physeq, method=i, distance=dist)
  plot_ordination(physeq, ordi, "samples", color="Age", shape = "AB")
}, Rps_mp, dist)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.209833
## Run 1 stress 0.2102505
## ... Procrustes: rmse 0.01312371 max resid 0.1277506
## Run 2 stress 0.2100672
## ... Procrustes: rmse 0.0202697 max resid 0.1855523
## Run 3 stress 0.2092721
## ... New best solution
## ... Procrustes: rmse 0.0136935 max resid 0.1290012
## Run 4 stress 0.2098112
## Run 5 stress 0.2090507
## ... New best solution
## ... Procrustes: rmse 0.01893574 max resid 0.1494045
## Run 6 stress 0.2089723
## ... New best solution
## ... Procrustes: rmse 0.01973871 max resid 0.1693078
## Run 7 stress 0.2090269
## ... Procrustes: rmse 0.02150426 max resid 0.1919598
```

```

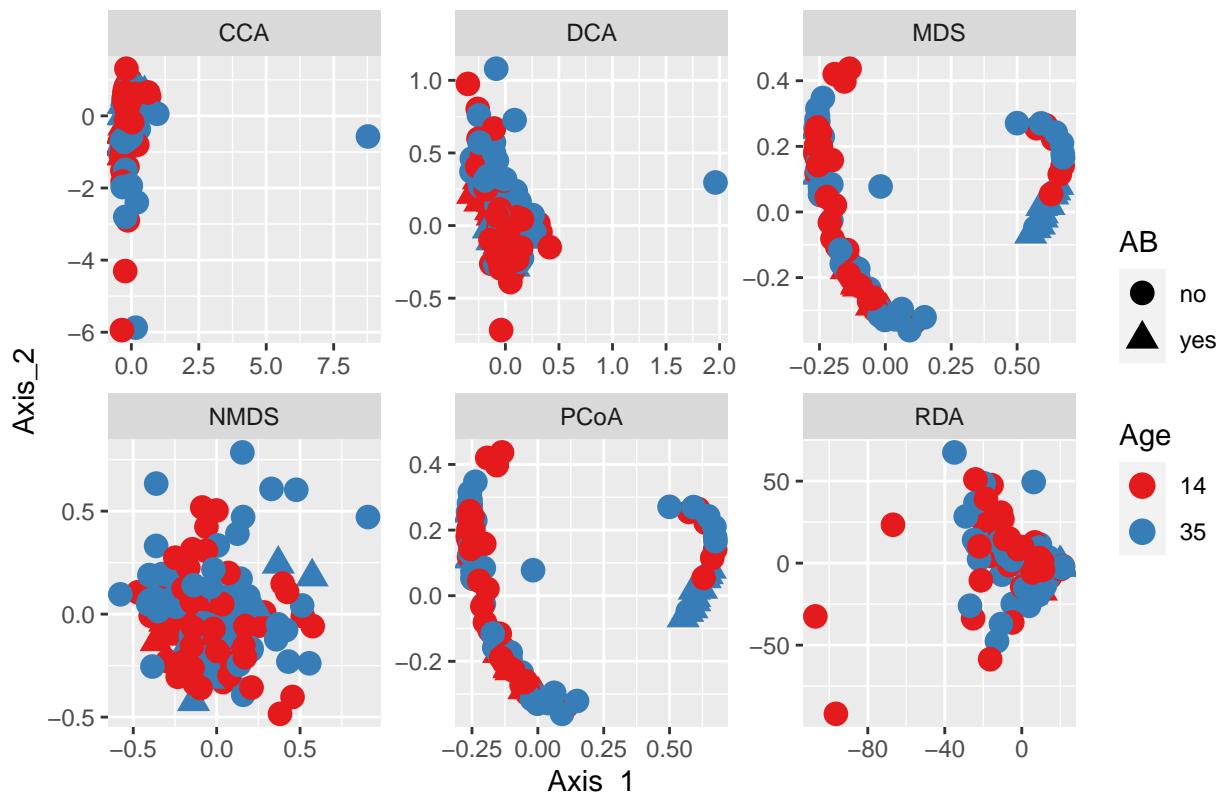
## Run 8 stress 0.2107212
## Run 9 stress 0.2612267
## Run 10 stress 0.2087925
## ... New best solution
## ... Procrustes: rmse 0.01892923 max resid 0.1904573
## Run 11 stress 0.2105235
## Run 12 stress 0.2089511
## ... Procrustes: rmse 0.01887265 max resid 0.1911308
## Run 13 stress 0.2097667
## Run 14 stress 0.2094249
## Run 15 stress 0.208547
## ... New best solution
## ... Procrustes: rmse 0.005540613 max resid 0.04666284
## Run 16 stress 0.2098364
## Run 17 stress 0.2102093
## Run 18 stress 0.209509
## Run 19 stress 0.2102788
## Run 20 stress 0.2098112
## *** Best solution was not repeated -- monoMDS stopping criteria:
##     1: no. of iterations >= maxit
##     17: stress ratio > sratmax
##     2: scale factor of the gradient < sfgrmin

names(plist) <- ord_meths

pdatatable = ldply(plist, function(x){
  df = x$data[, 1:2]
  colnames(df) = c("Axis_1", "Axis_2")
  return(cbind(df, x$data))
})
names(pdatatable)[1] = "method"
ggplot(pdatatable, aes(Axis_1, Axis_2, color=Age, shape=AB)) +
  geom_point(size=4) +
  facet_wrap(~method, scales="free") +
  scale_fill_brewer(type="qual", palette="Set1") +
  scale_colour_brewer(type="qual", palette="Set1") +
  ggtitle("Different ordination methods for resistomic data (Bray-Curtis)")

```

Different ordination methods for resistomic data (Bray–Curtis)



```
# PCoAs for different methods, with Age and Farm as colors, and AB as shape
```

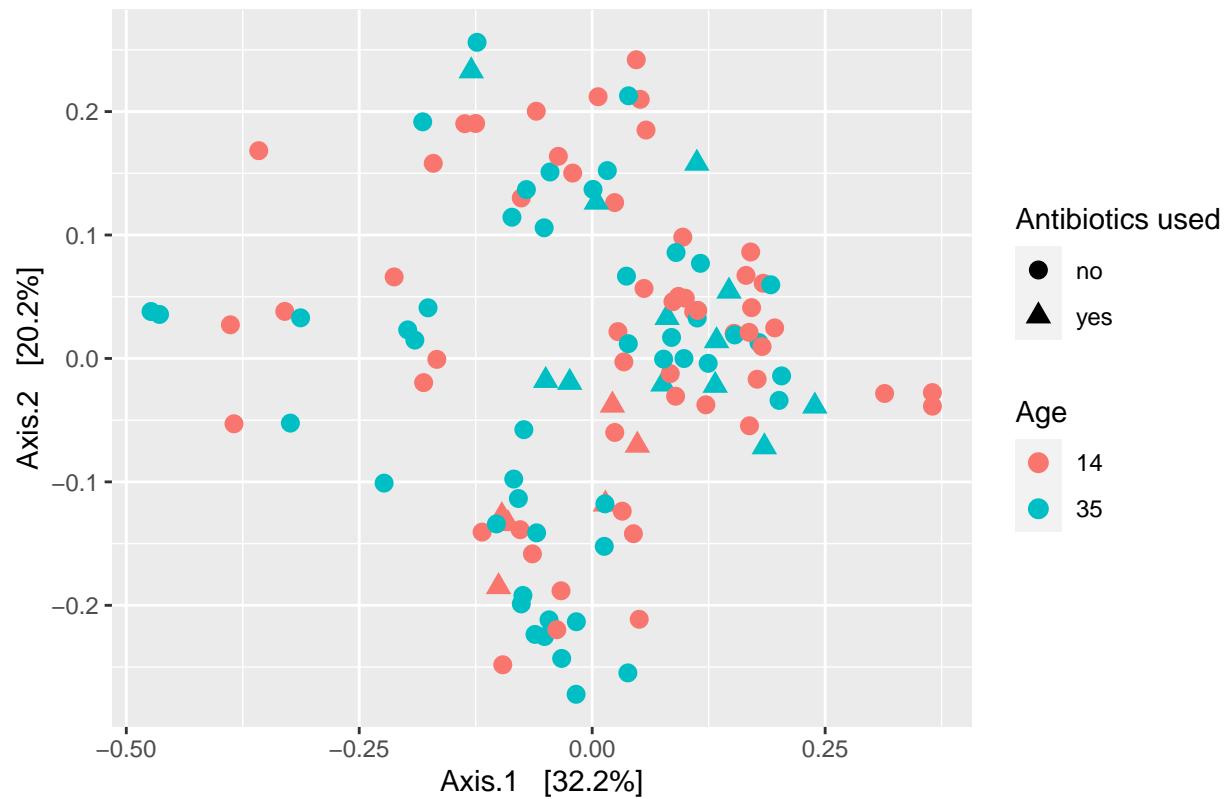
```
plot_pcoa_ordination <- function(data, pcoa, var, title) {
  p <- plot_ordination(data, pcoa, color = var, shape = "AB") +
    geom_point(size = 3) +
    labs(title = title, color = var, shape = "Antibiotics used")

  return(p)
}

pcoa_bc = ordinate(Rps, "PCoA", "bray")
pcoa_unifrac = ordinate(Rps, "PCoA", "unifrac")
pcoa_wunifrac = ordinate(Rps, "PCoA", "wunifrac")
pcoa_jsd = ordinate(Rps, "PCoA", "jsd")
pcoa_jaccard = ordinate(Rps, "PCoA", "jaccard", binary=TRUE)

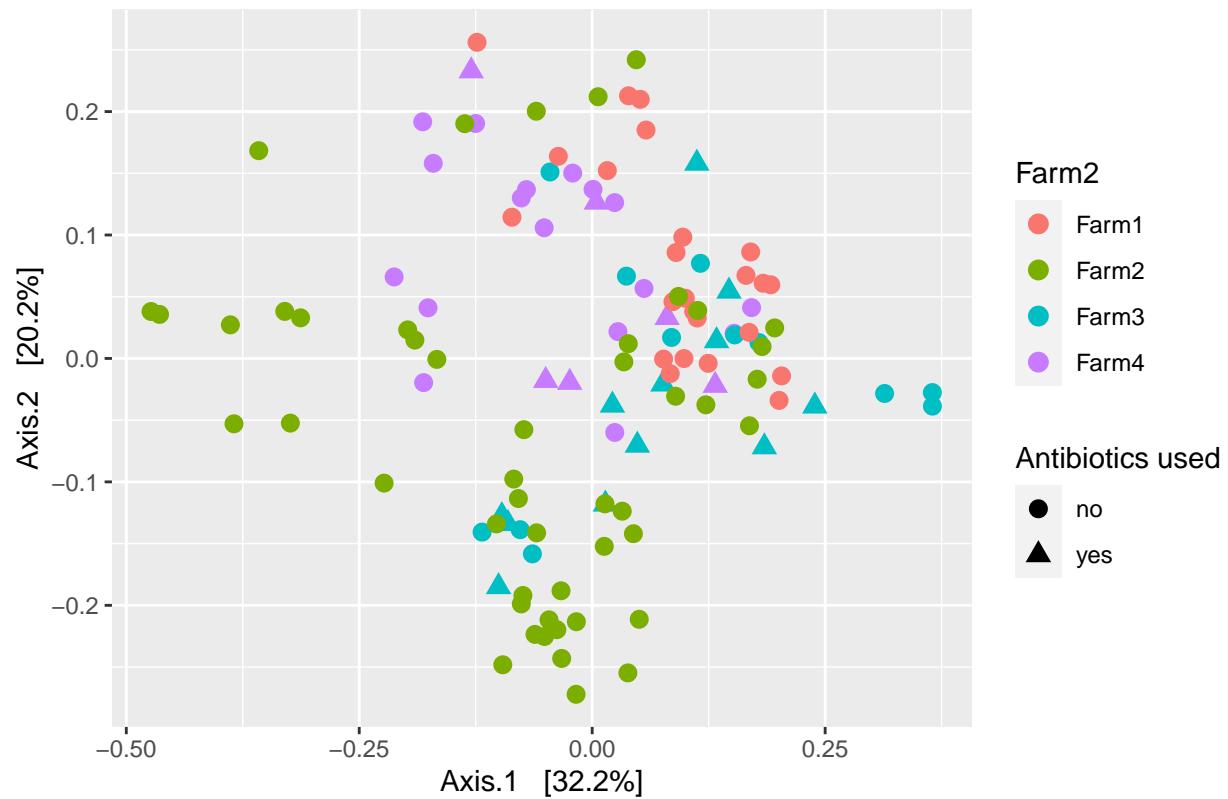
plot_pcoa_ordination(Rps, pcoa_bc, "Age", "PCoA Bray Curtis")
```

PCoA Bray Curtis



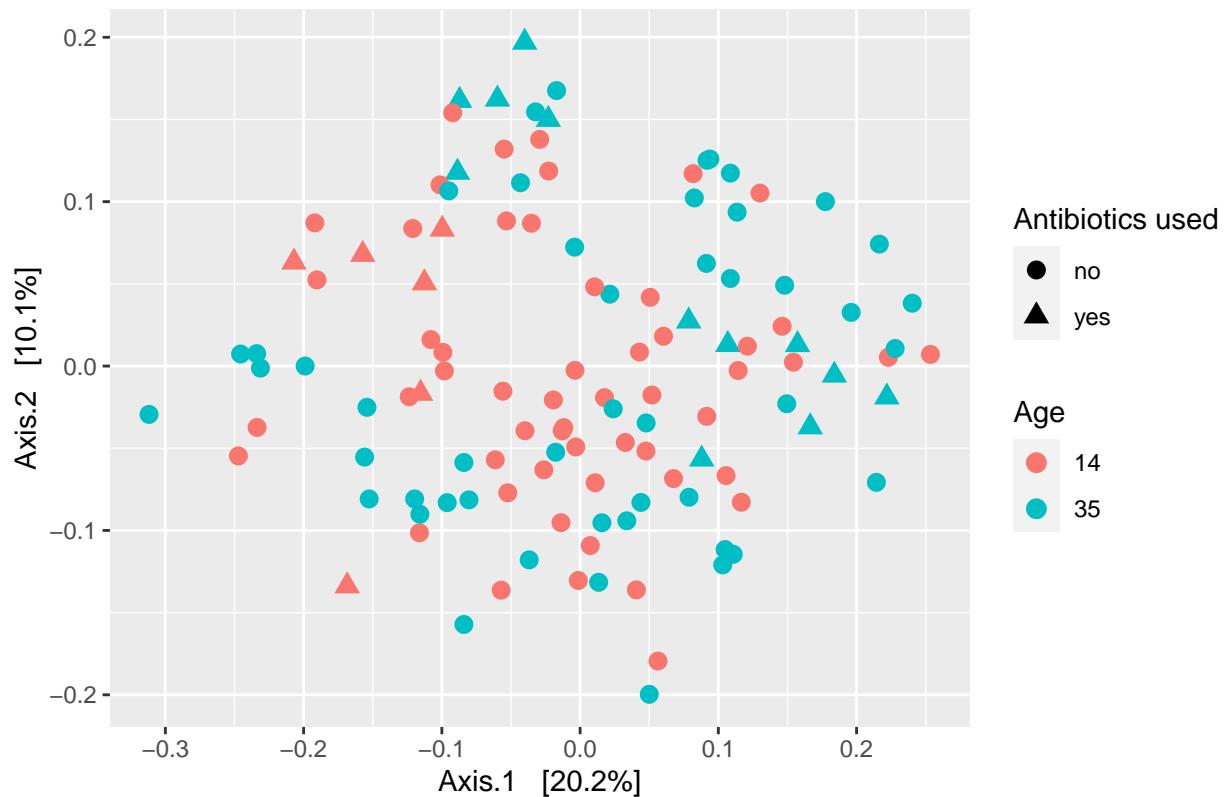
```
plot_pcoa_ordination(Rps, pcoa_bc, "Farm2", "PCoA Bray Curtis")
```

PCoA Bray Curtis



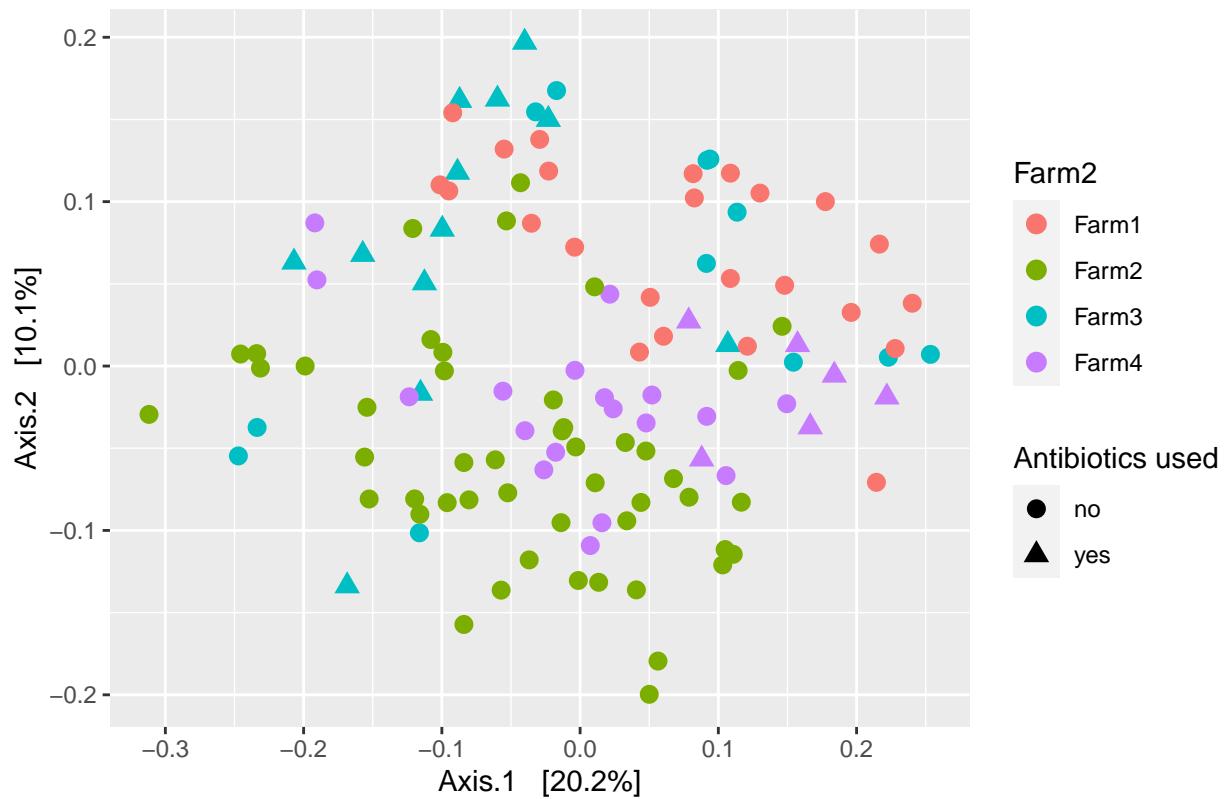
```
plot_pcoa_ordination(Rps, pcoa_unifrac, "Age", "PCoA Unifrac")
```

PCoA Unifrac



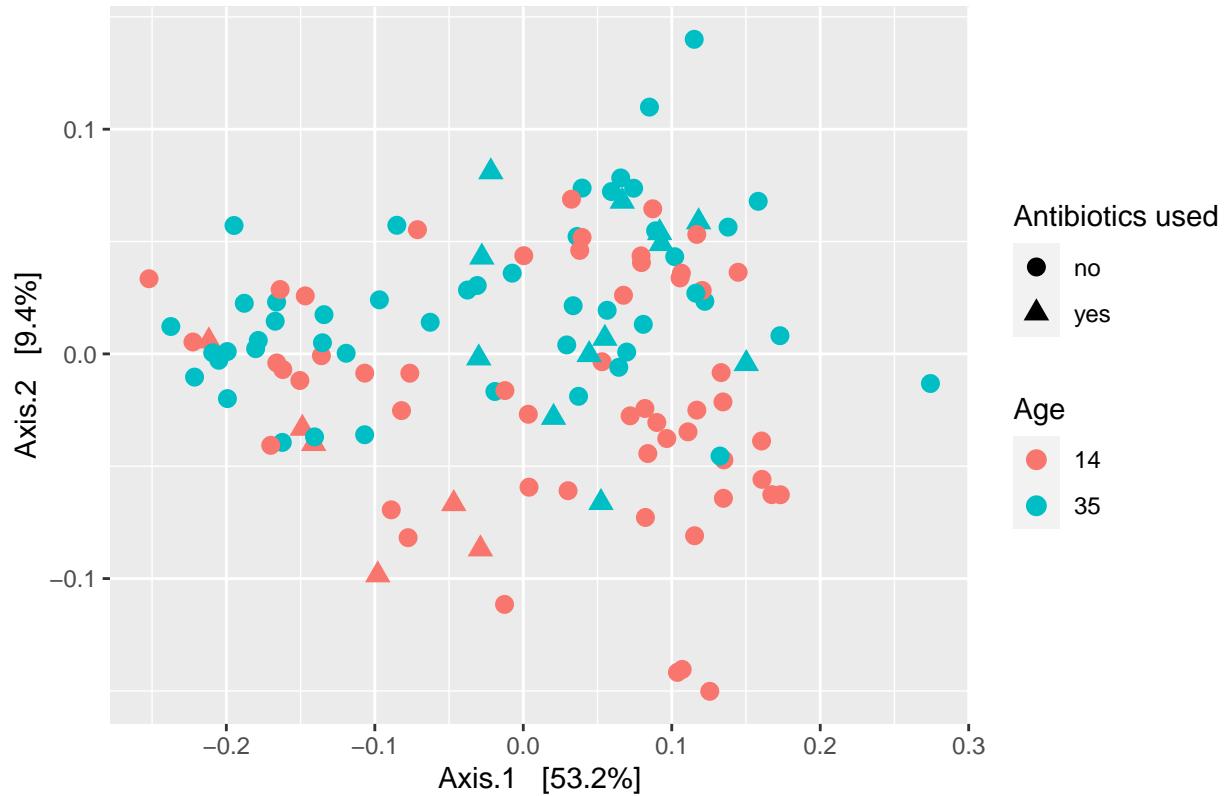
```
plot_pcoa_ordination(Rps, pcoa_unifrac, "Farm2", "PCoA Unifrac")
```

PCoA Unifrac



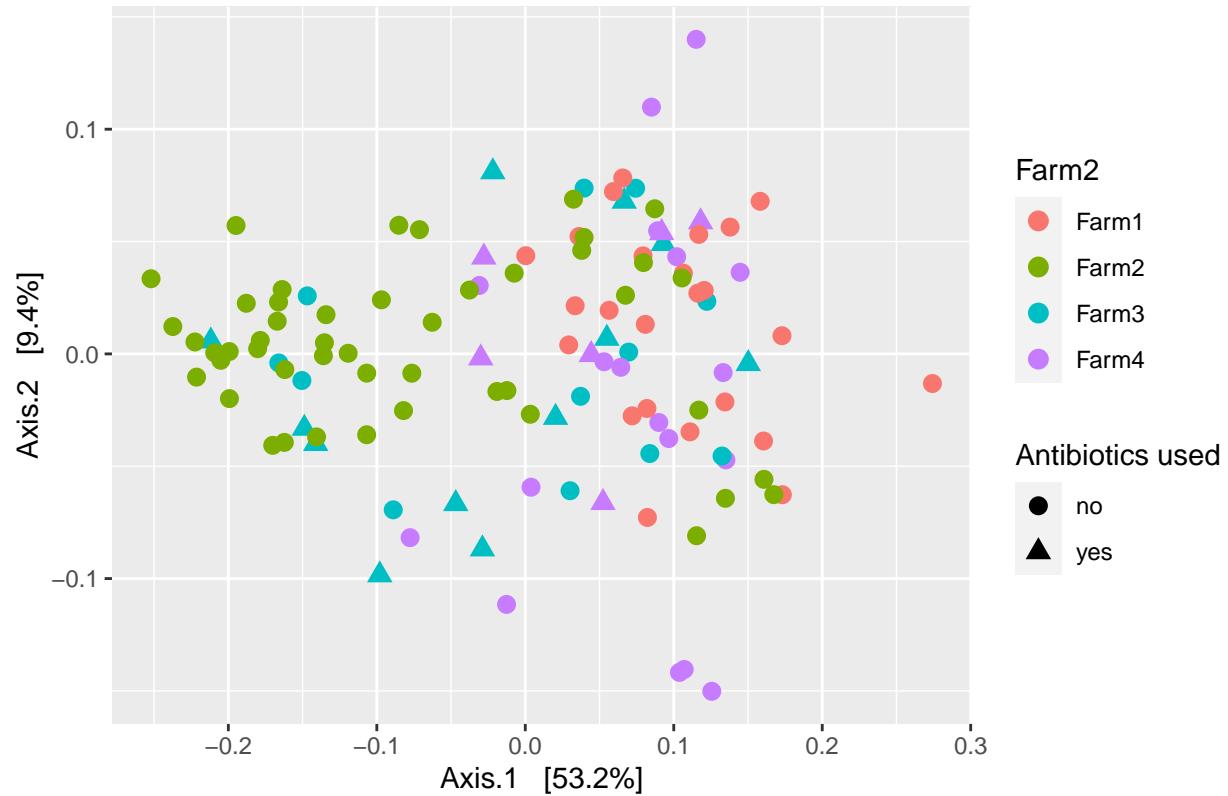
```
plot_pcoa_ordination(Rps, pcoa_wunifrac, "Age", "PCoA Weighted Unifrac")
```

PCoA Weighted Unifrac



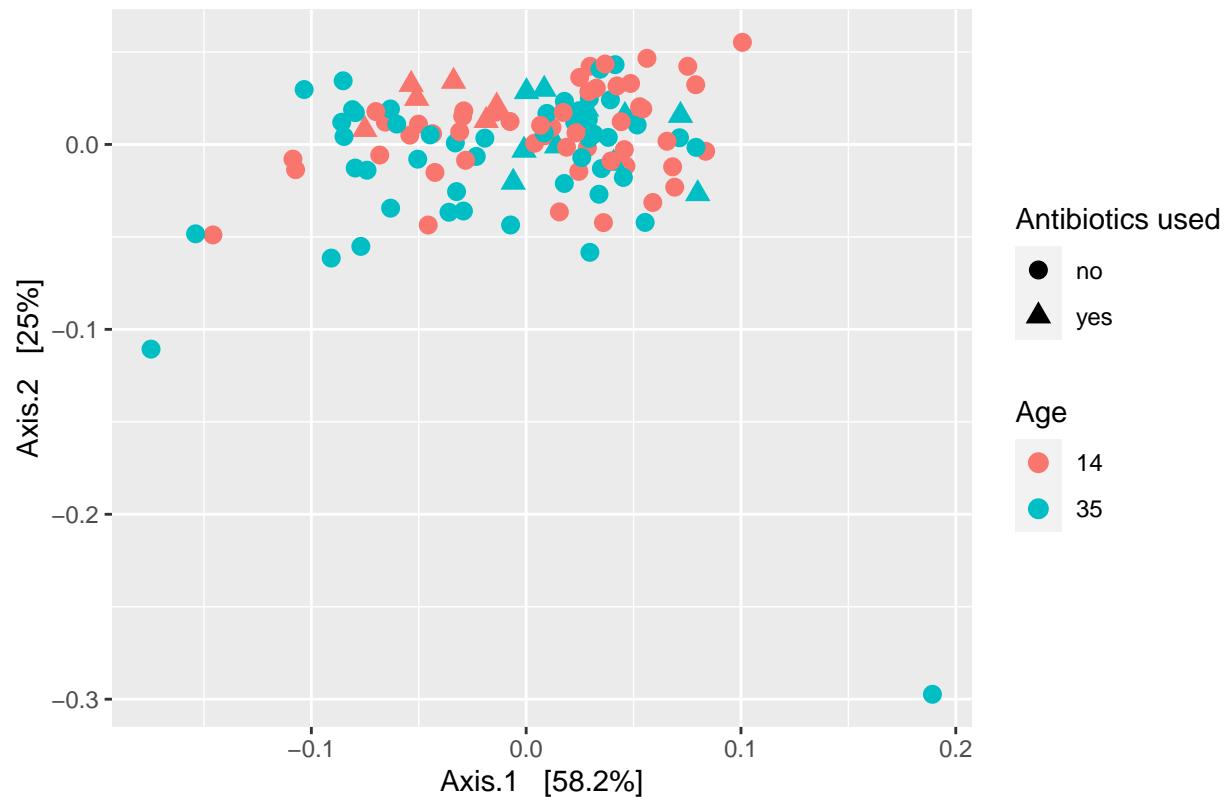
```
plot_pcoa_ordination(Rps, pcoa_wunifrac, "Farm2", "PCoA Weighted Unifrac")
```

PCoA Weighted Unifrac



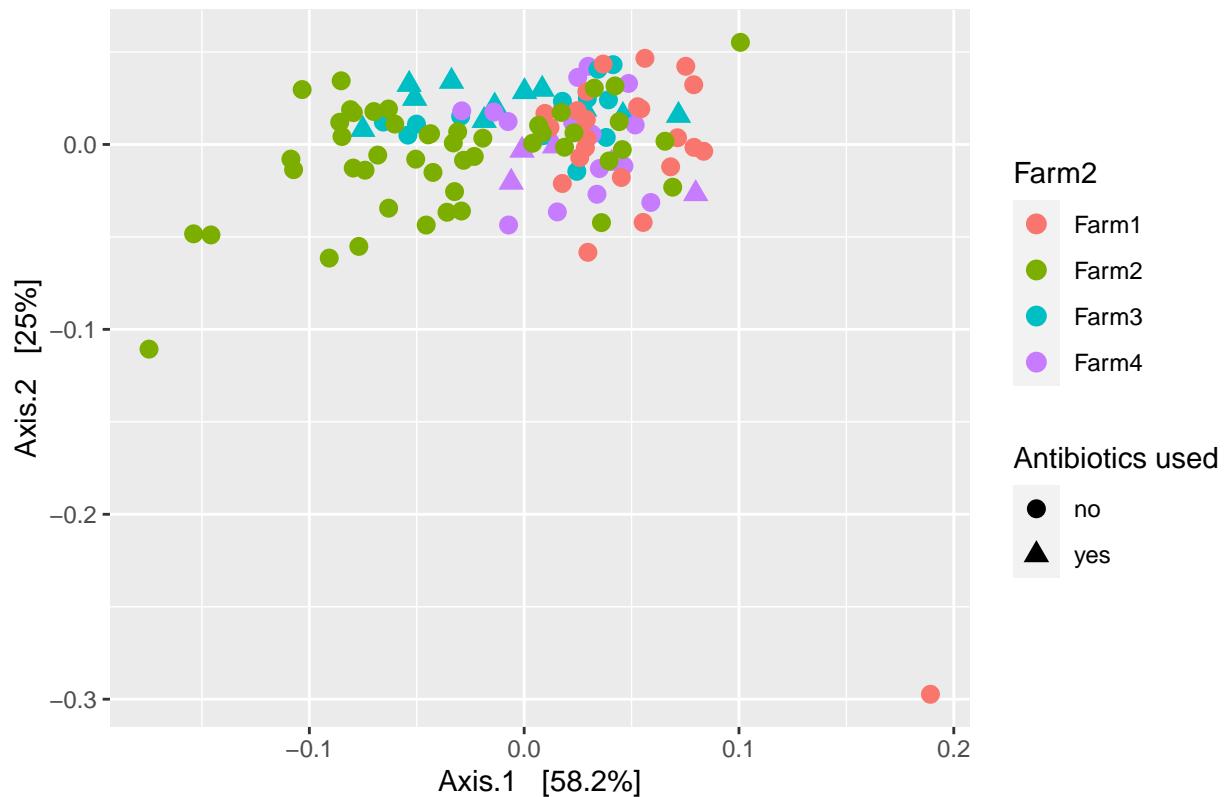
```
plot_pcoa_ordination(Rps, pcoa_jsd, "Age", "PCoA Jensen-Shannon Divergence")
```

PCoA Jensen–Shannon Divergence



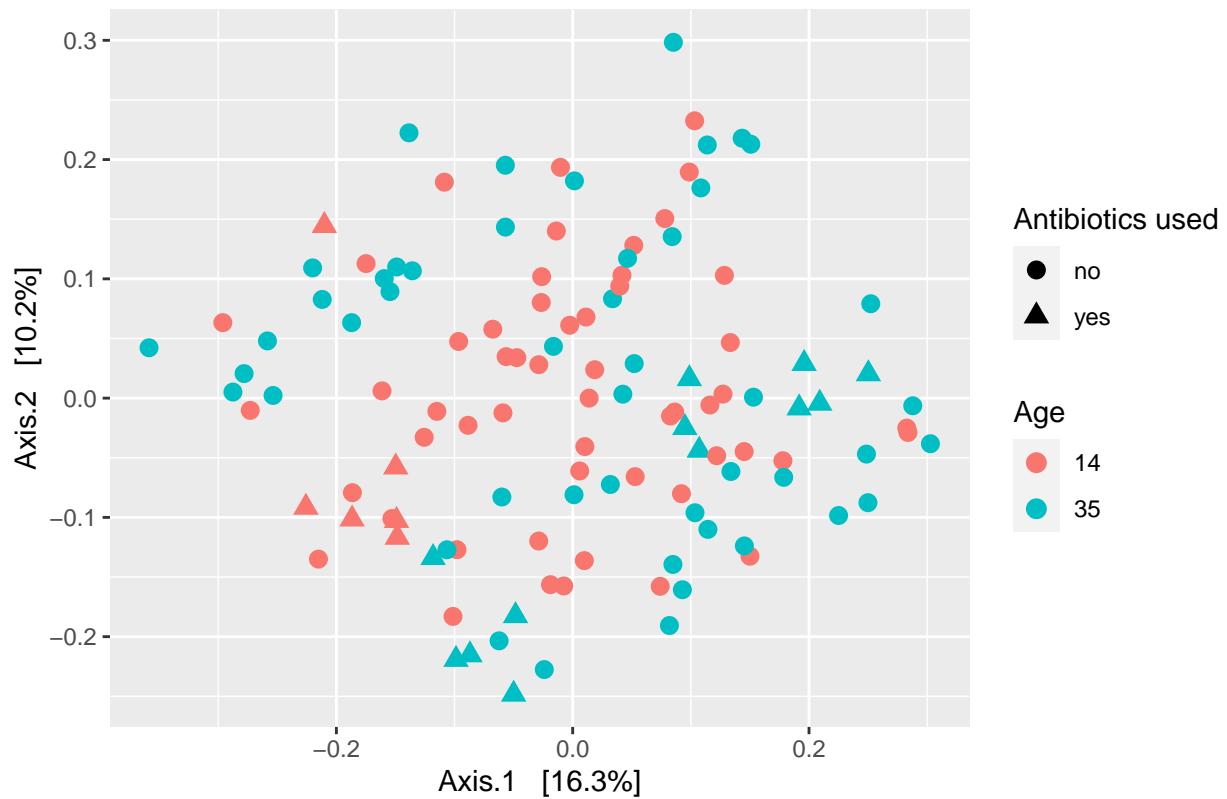
```
plot_pcoa_ordination(Rps, pcoa_jsd, "Farm2", "PCoA Jensen–Shannon Divergence")
```

PCoA Jensen–Shannon Divergence



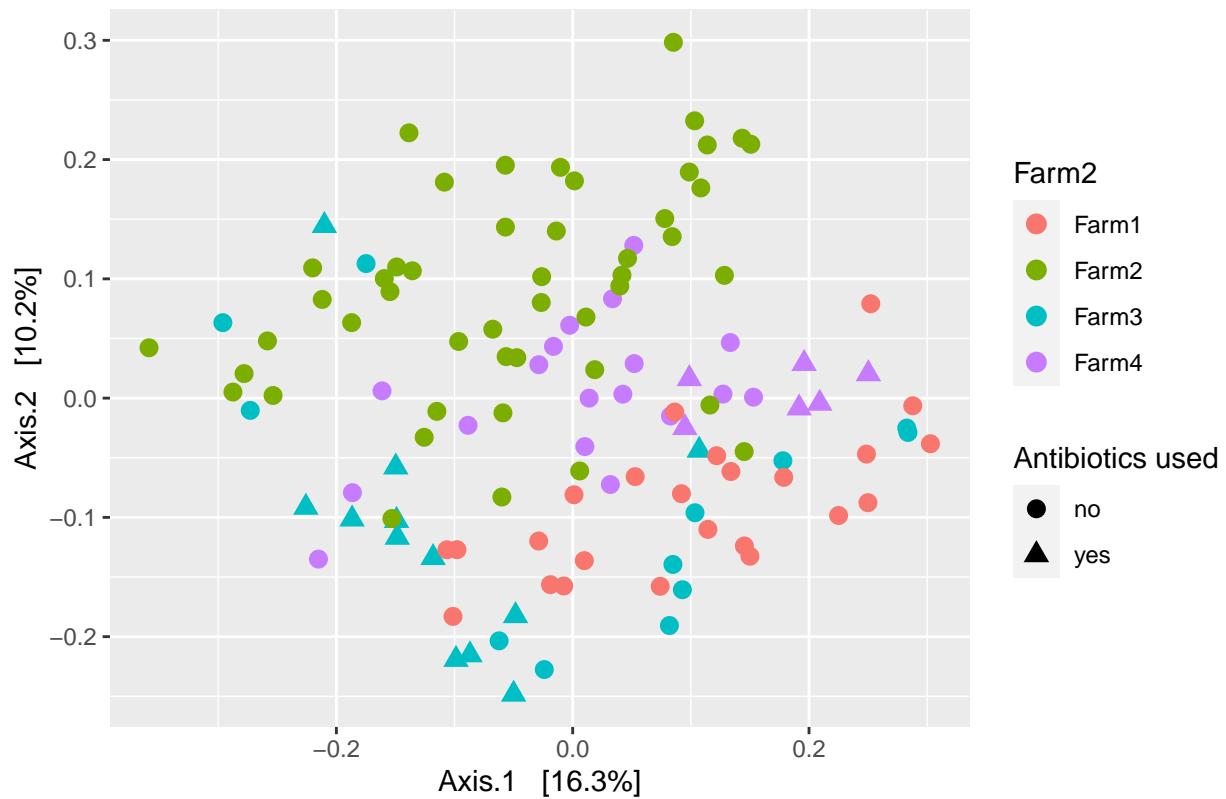
```
plot_pcoa_ordination(Rps, pcoa_jaccard, "Age", "PCoA Jaccard")
```

PCoA Jaccard



```
plot_pcoa_ordination(Rps, pcoa_jaccard, "Farm2", "PCoA Jaccard")
```

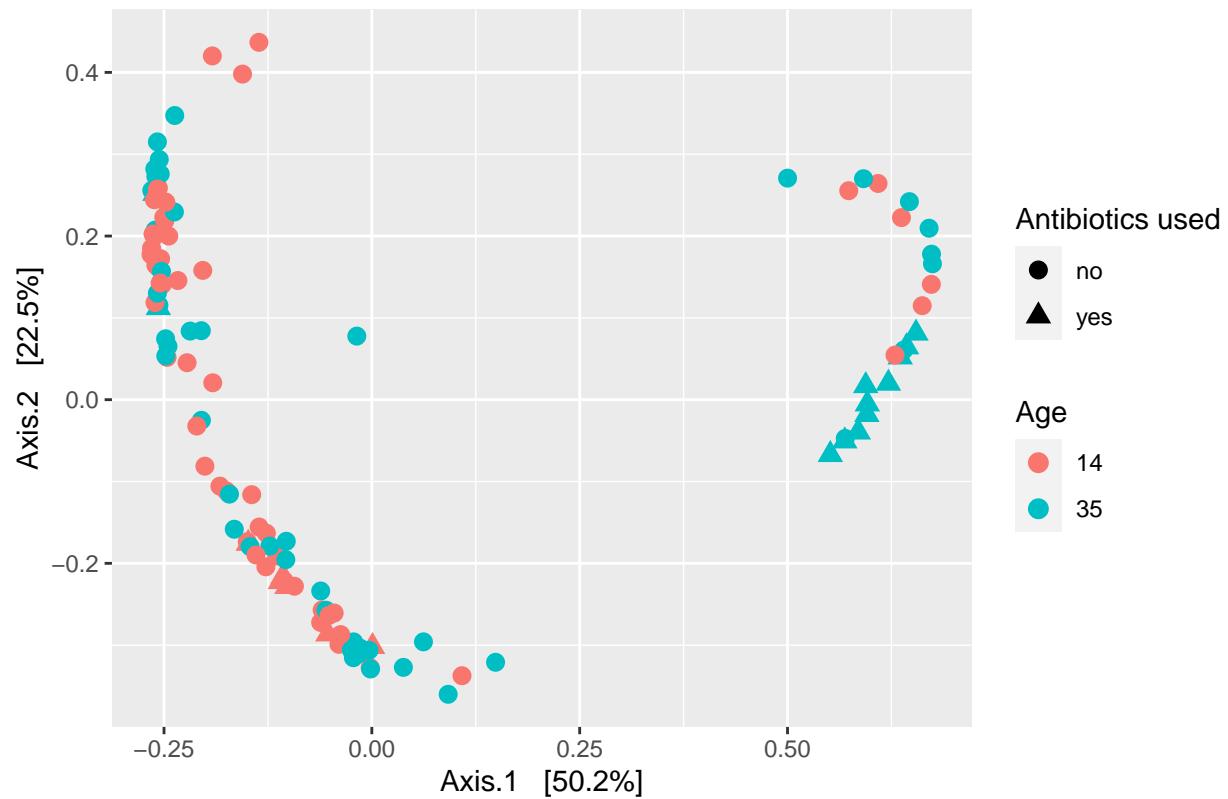
PCoA Jaccard



```
# repeat for metaphlan
pcoa_bc = ordinate(Rps_mp, "PCoA", "bray")
pcoa_unifrac = ordinate(Rps_mp, "PCoA", "unifrac")
pcoa_wunifrac = ordinate(Rps_mp, "PCoA", "wunifrac")
pcoa_jsd = ordinate(Rps_mp, "PCoA", "jsd")
pcoa_jaccard = ordinate(Rps_mp, "PCoA", "jaccard", binary=TRUE)

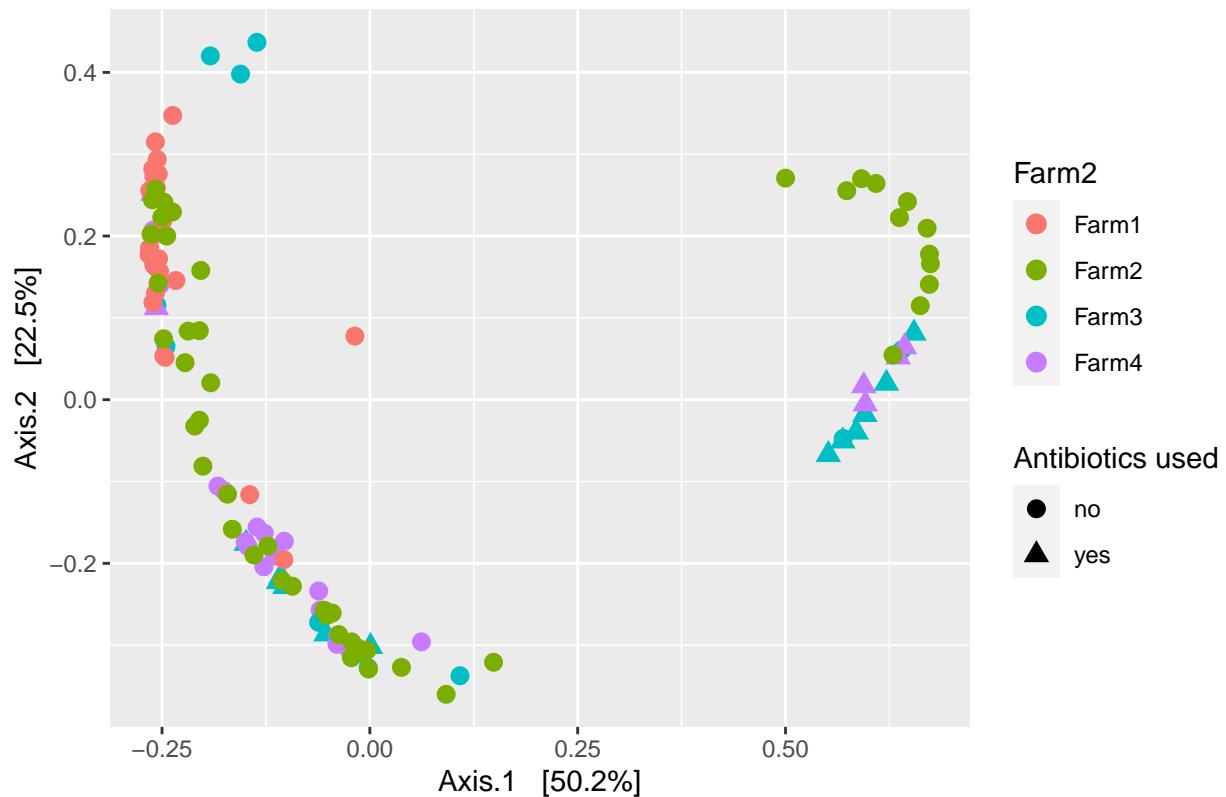
plot_pcoa_ordination(Rps_mp, pcoa_bc, "Age", "PCoA Bray Curtis")
```

PCoA Bray Curtis



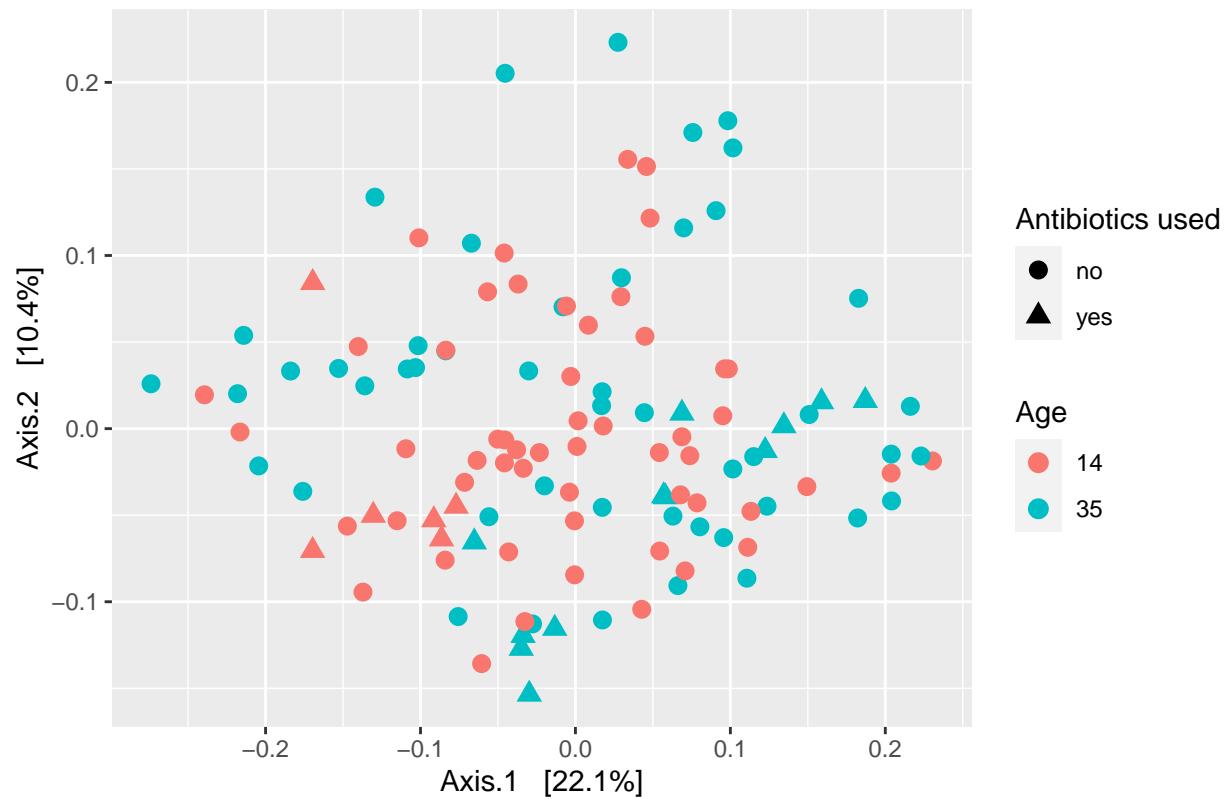
```
plot_pcoa_ordination(Rps_mp, pcoa_bc, "Farm2", "PCoA Bray Curtis")
```

PCoA Bray Curtis



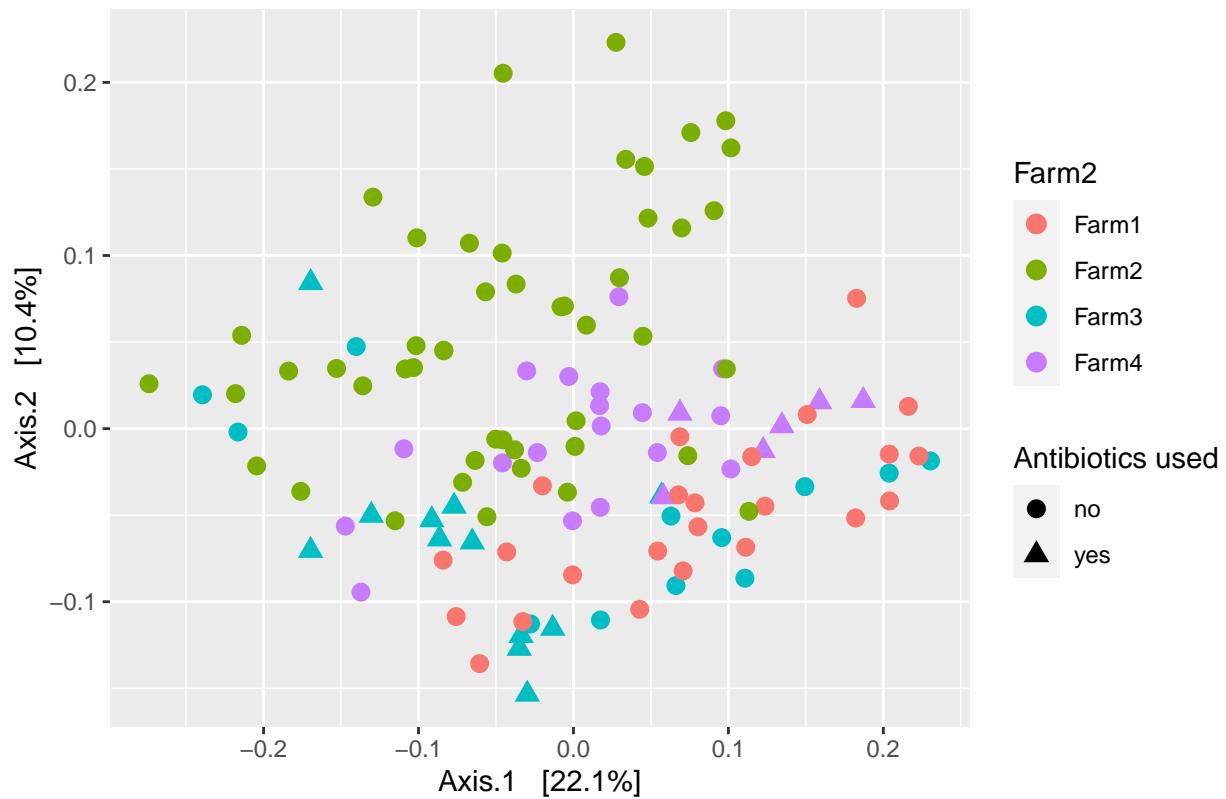
```
plot_pcoa_ordination(Rps_mp, pcoa_unifrac, "Age", "PCoA Unifrac")
```

PCoA Unifrac



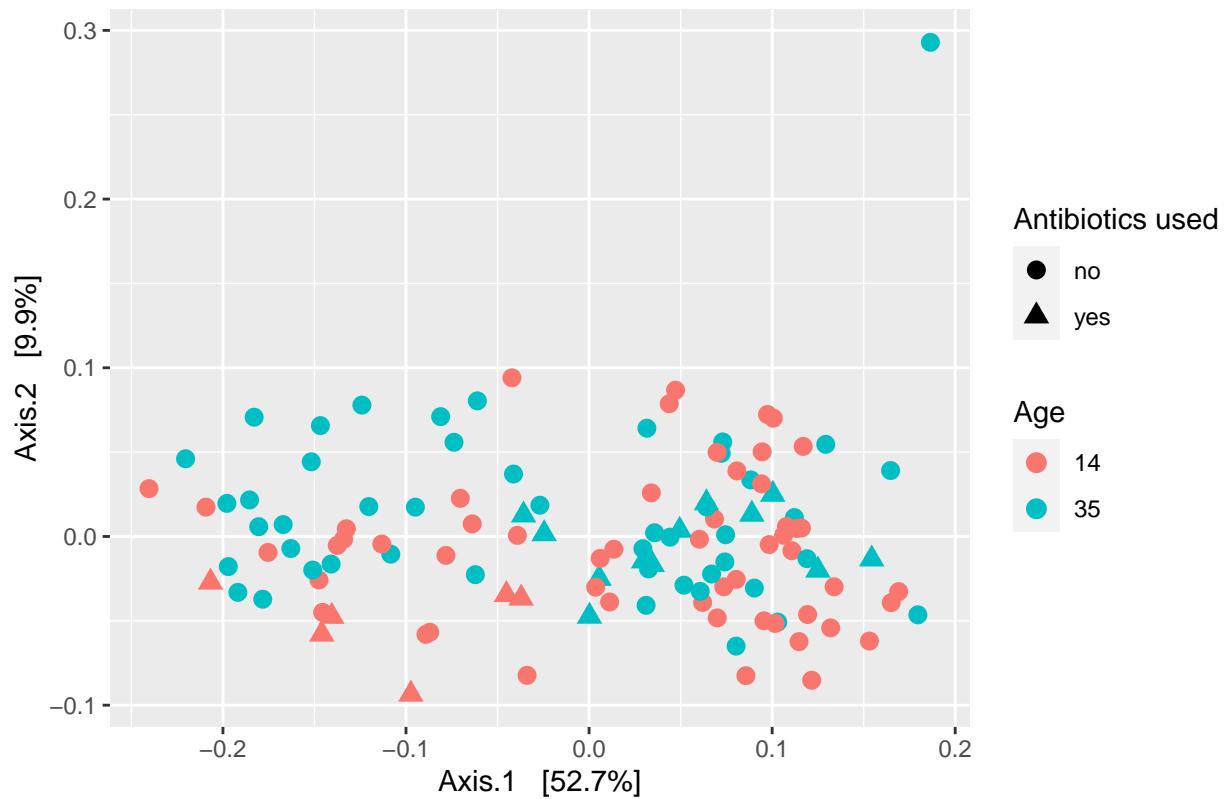
```
plot_pcoa_ordination(Rps_mp, pcoa_unifrac, "Farm2", "PCoA Unifrac")
```

PCoA Unifrac



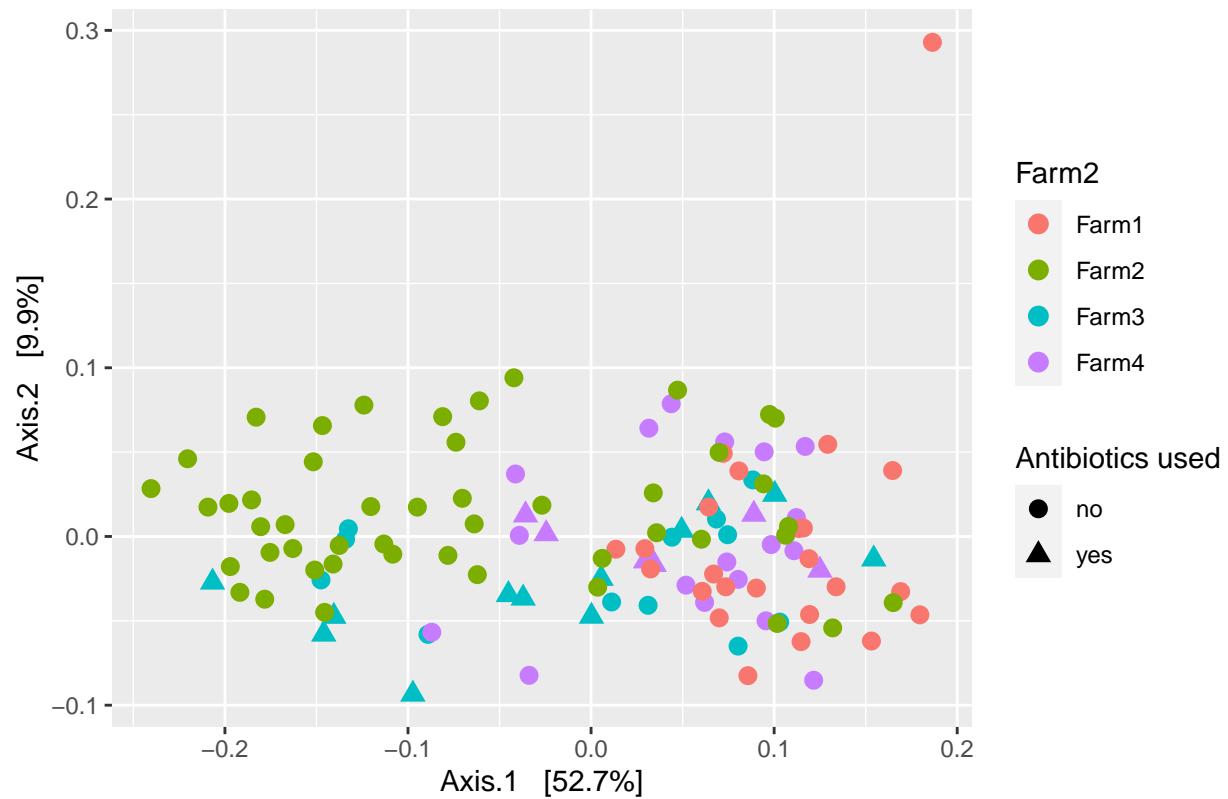
```
plot_pcoa_ordination(Rps_mp, pcoa_wunifrac, "Age", "PCoA Weighted Unifrac")
```

PCoA Weighted Unifrac



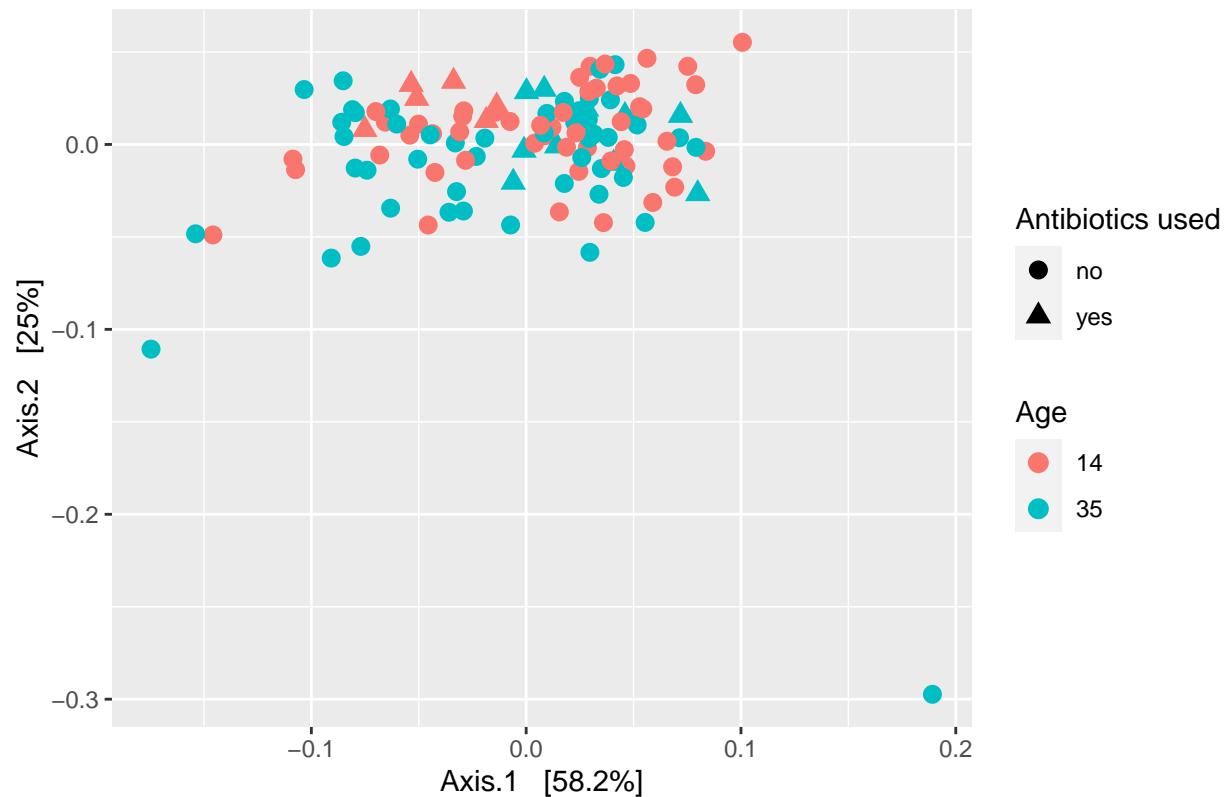
```
plot_pcoa_ordination(Rps_mp, pcoa_wunifrac, "Farm2", "PCoA Weighted Unifrac")
```

PCoA Weighted Unifrac



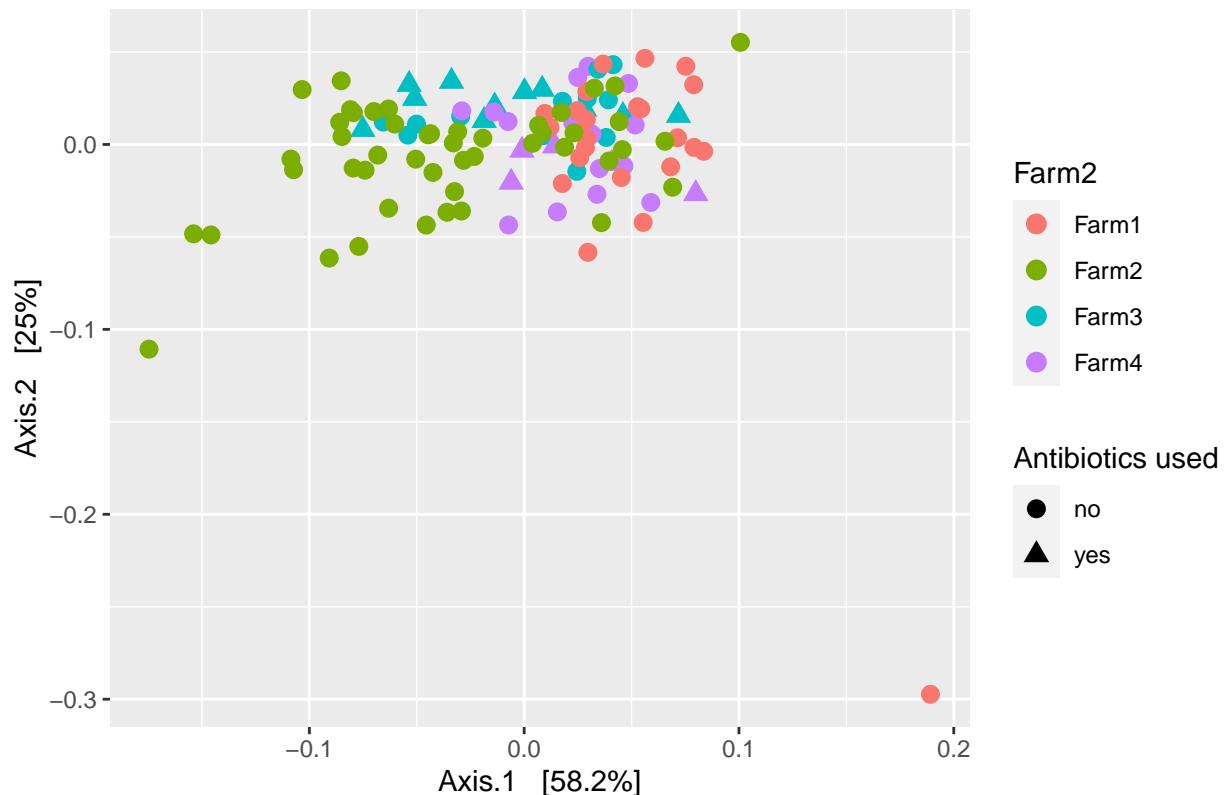
```
plot_pcoa_ordination(Rps_mp, pcoa_jsd, "Age", "PCoA Jensen-Shannon Divergence")
```

PCoA Jensen–Shannon Divergence



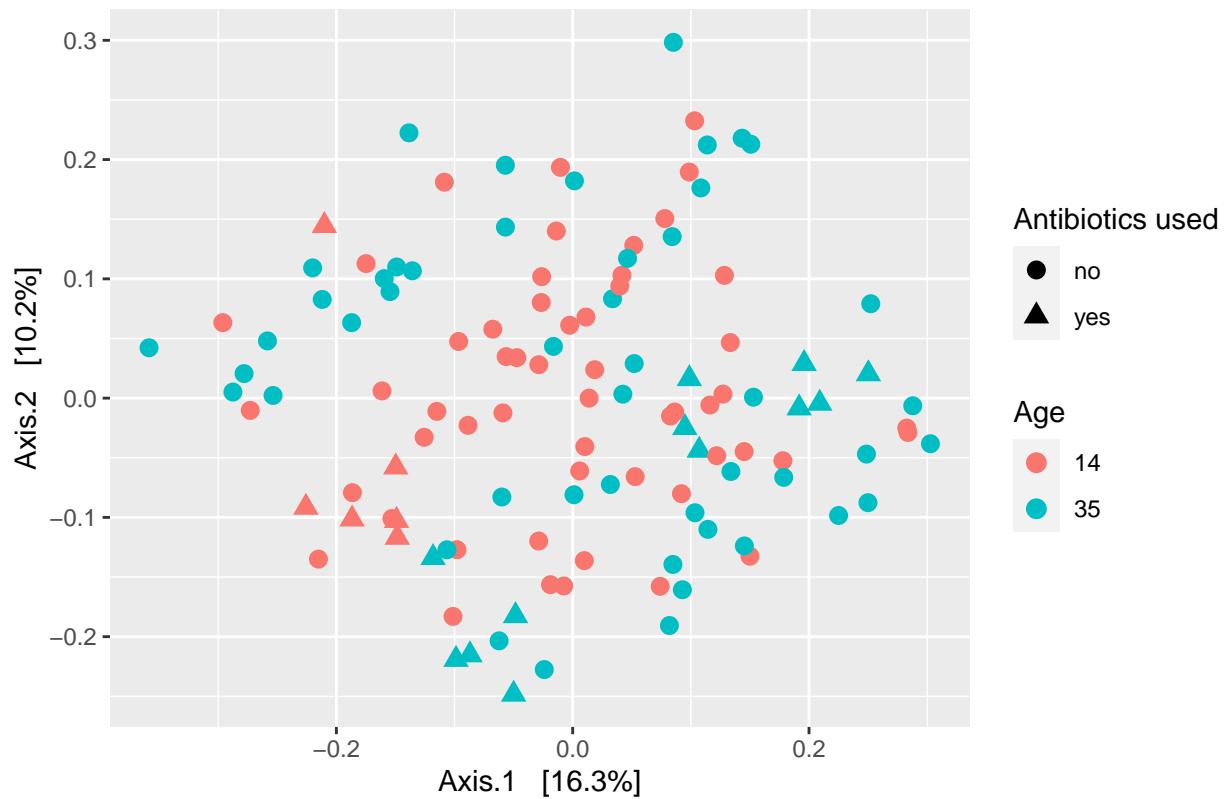
```
plot_pcoa_ordination(Rps_mp, pcoa_jsd, "Farm2", "PCoA Jensen–Shannon Divergence")
```

PCoA Jensen–Shannon Divergence



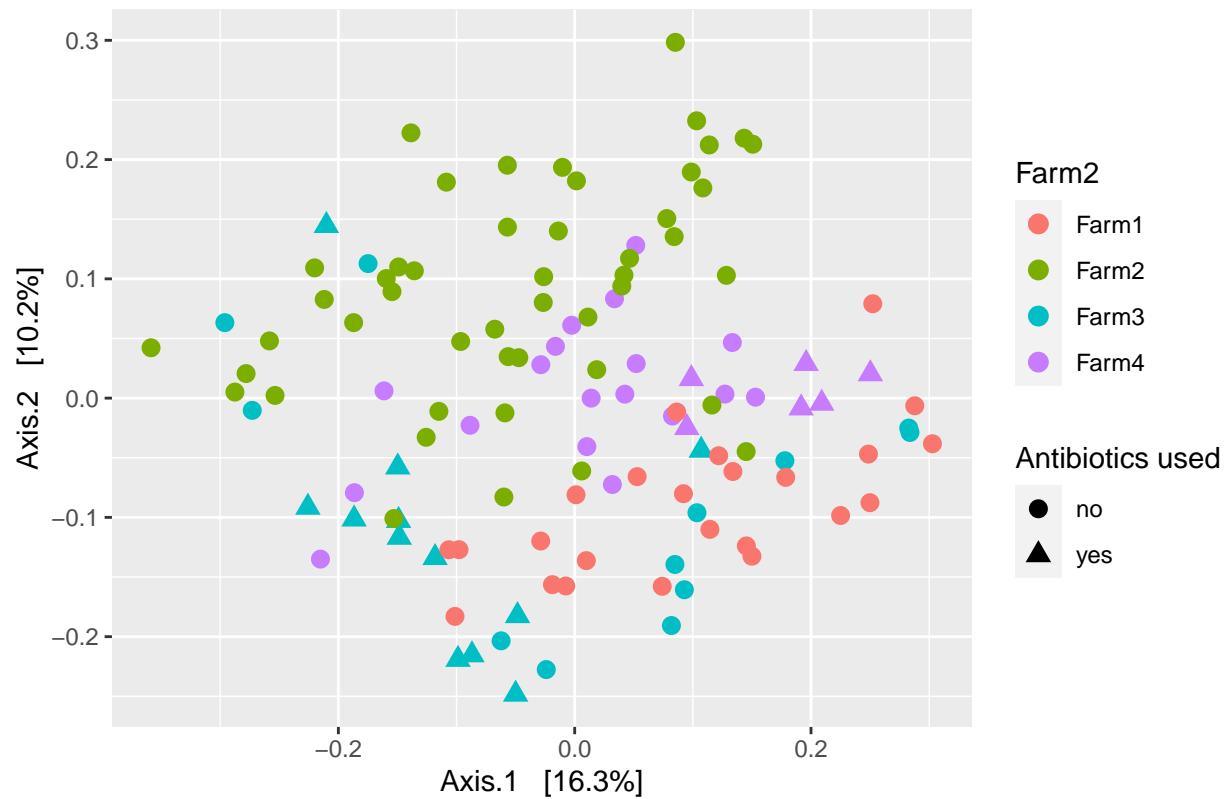
```
plot_pcoa_ordination(Rps_mp, pcoa_jaccard, "Age", "PCoA Jaccard")
```

PCoA Jaccard

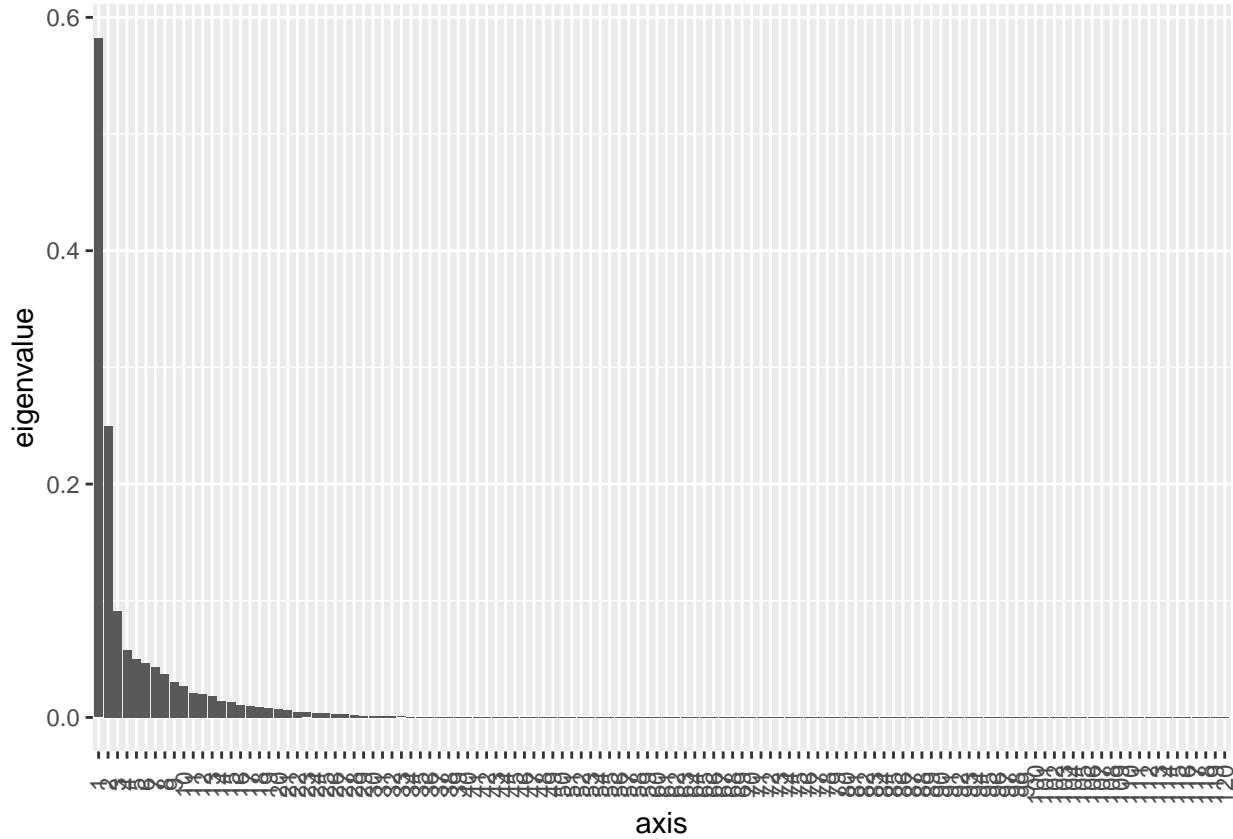


```
plot_pcoa_ordination(Rps_mp, pcoa_jaccard, "Farm2", "PCoA Jaccard")
```

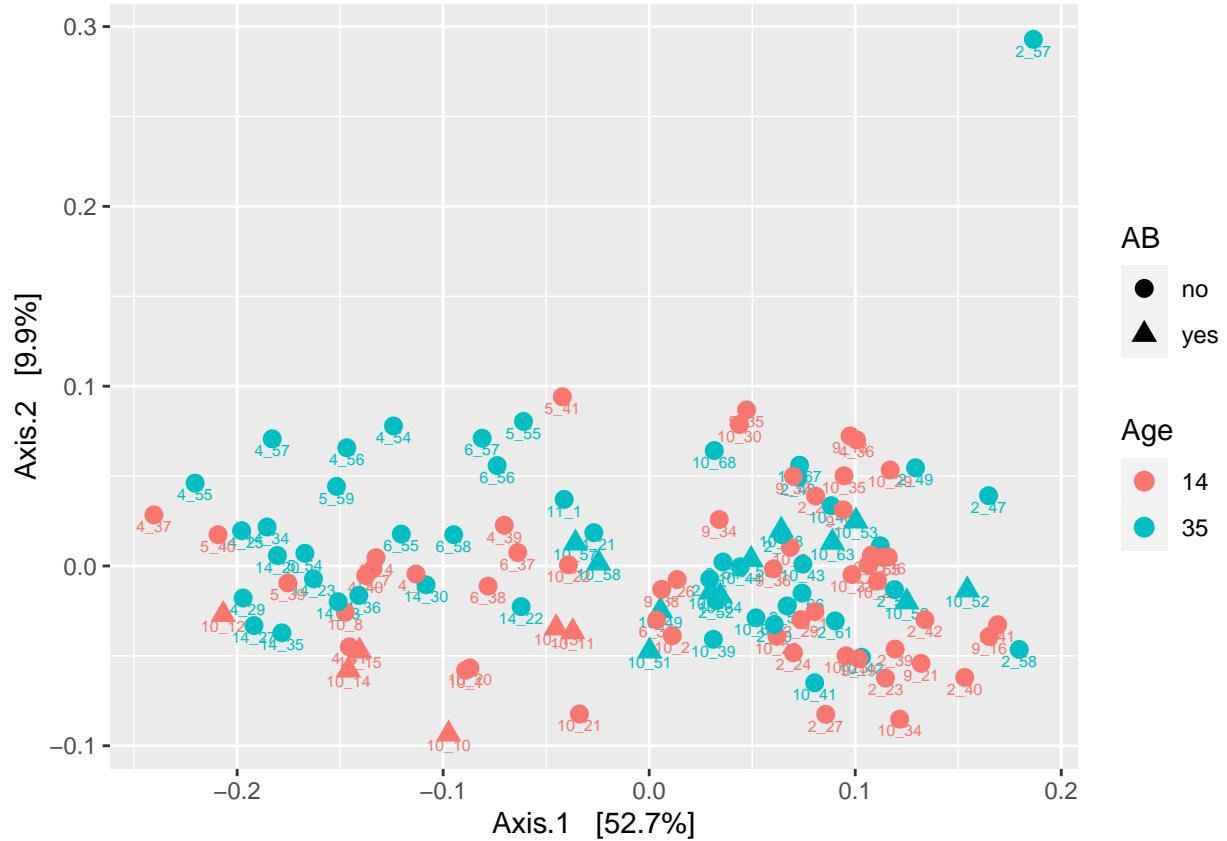
PCoA Jaccard



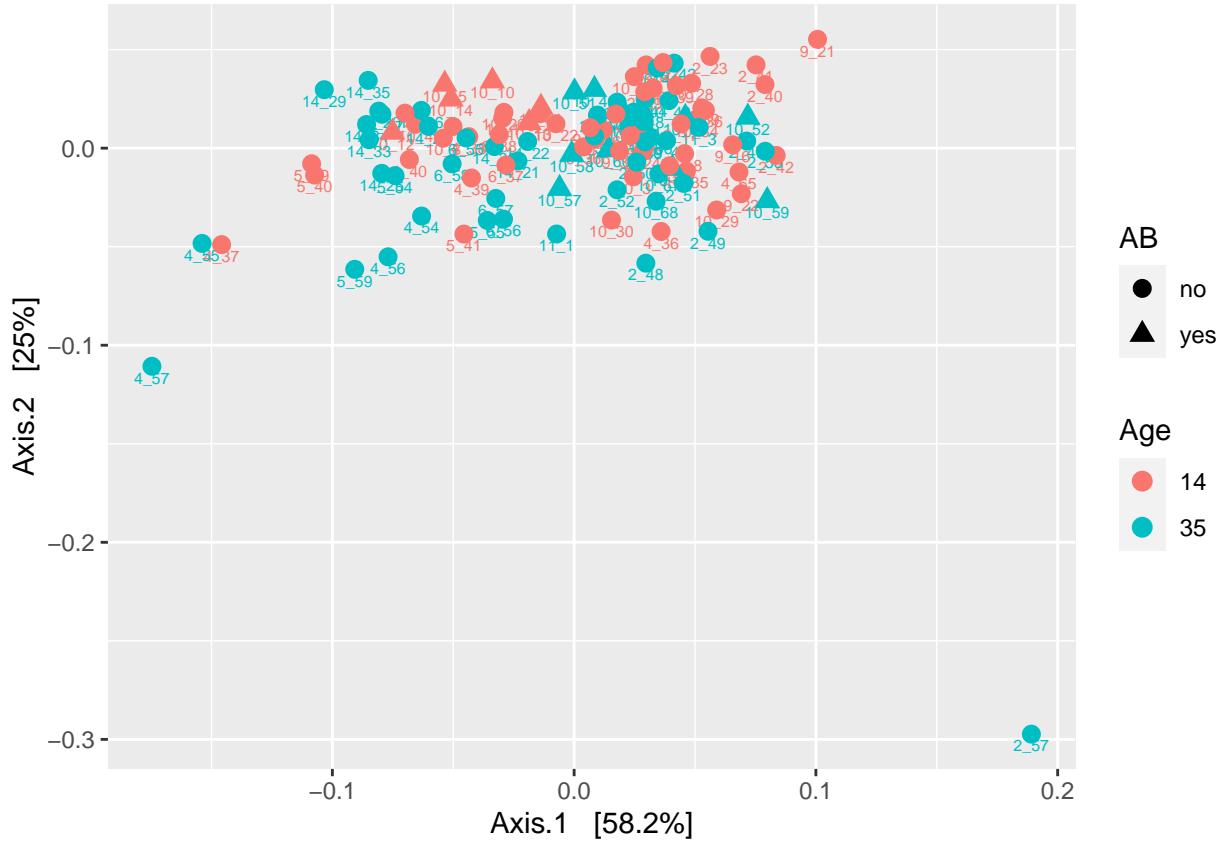
```
#plot_ordination(Rps, pcoa_bc, type = "taxa", color = "AMR_class_primary") +
# geom_point(size = 3) + labs(title = "PCoA primary AMR classes", color = "AMR_class_primary")
plot_scree(pcoa_jsd) #scree plots can be made for any of the PCoAs, and are made for those with less than
```



```
plot_ordination(Rps_mp, pcoa_wunifrac, color = "Age", shape = "AB", label = "Sample_Unique") +  
  geom_point(size = 3) # 2_57 outlier again
```



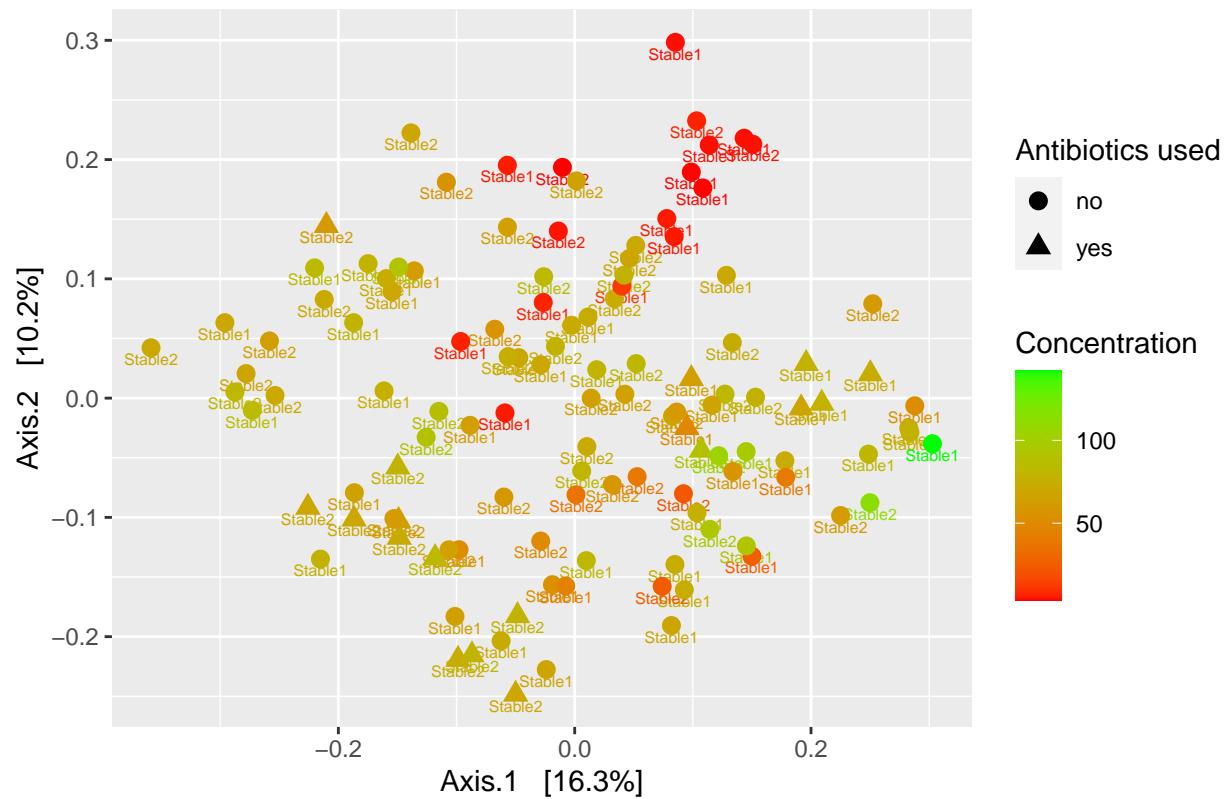
```
plot_ordination(Rps_mp, pcova_jsd, color = "Age", shape = "AB", label = "Sample_Unique") +
  geom_point(size = 3) # 2_57 outlier again
```



```
# plot to look at concentration with a red/green gradient for metaphlan data
# because jaccard is hugely influenced by presence/absence, it will be impacted strongly by singletons
# whenever the data is not rounded, jaccard will not see any of our genes as being present as singleton
# case in point, the following is a plot of metaphlan data which is not rounded
pcoa_jaccard_mp = ordinate(Rps_mp, "PCoA", "jaccard", binary=TRUE)

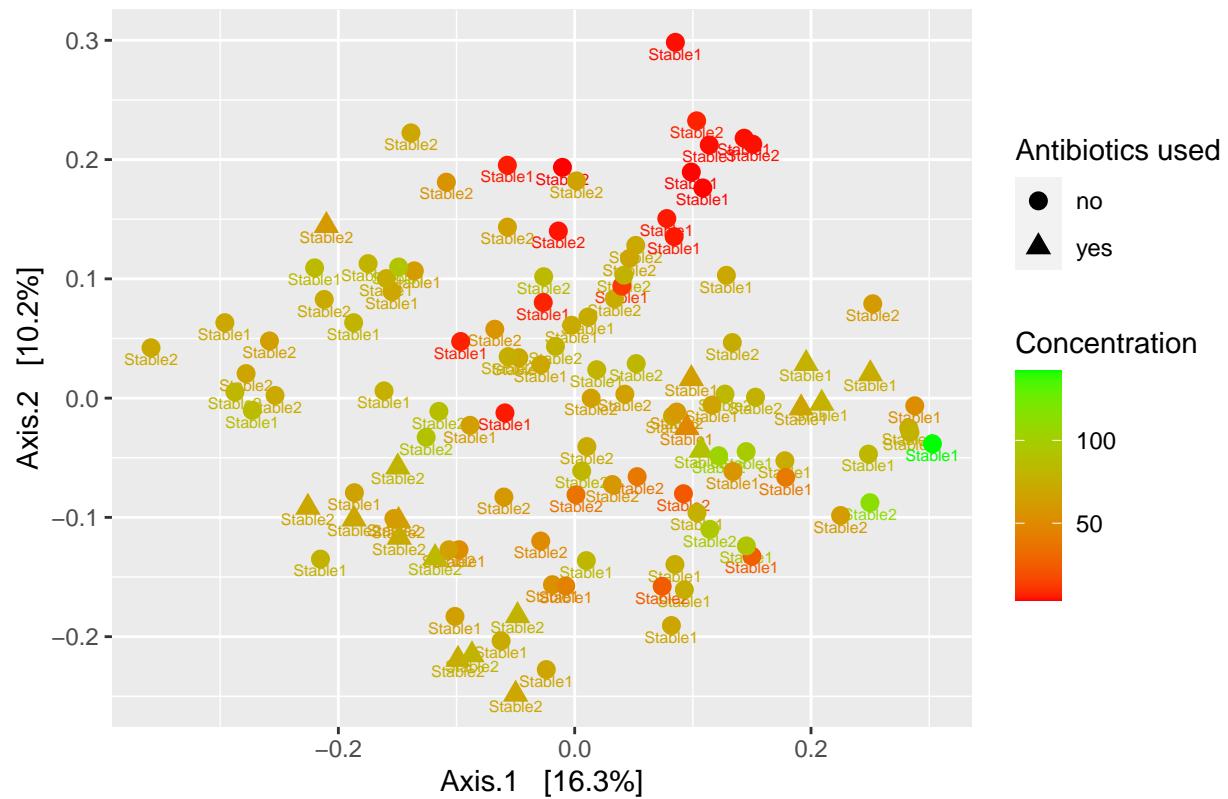
plot_ordination(Rps_mp, pcoa_jaccard_mp, color = "Conc...ng..pl.", shape = "AB", label = "Stable") +
  geom_point(size = 3) + labs(title = "PCoA Jaccard concentration", color = "Concentration", shape = "AB",
  scale_colour_gradient(low = "red", high = "green")
```

PCoA Jaccard concentration



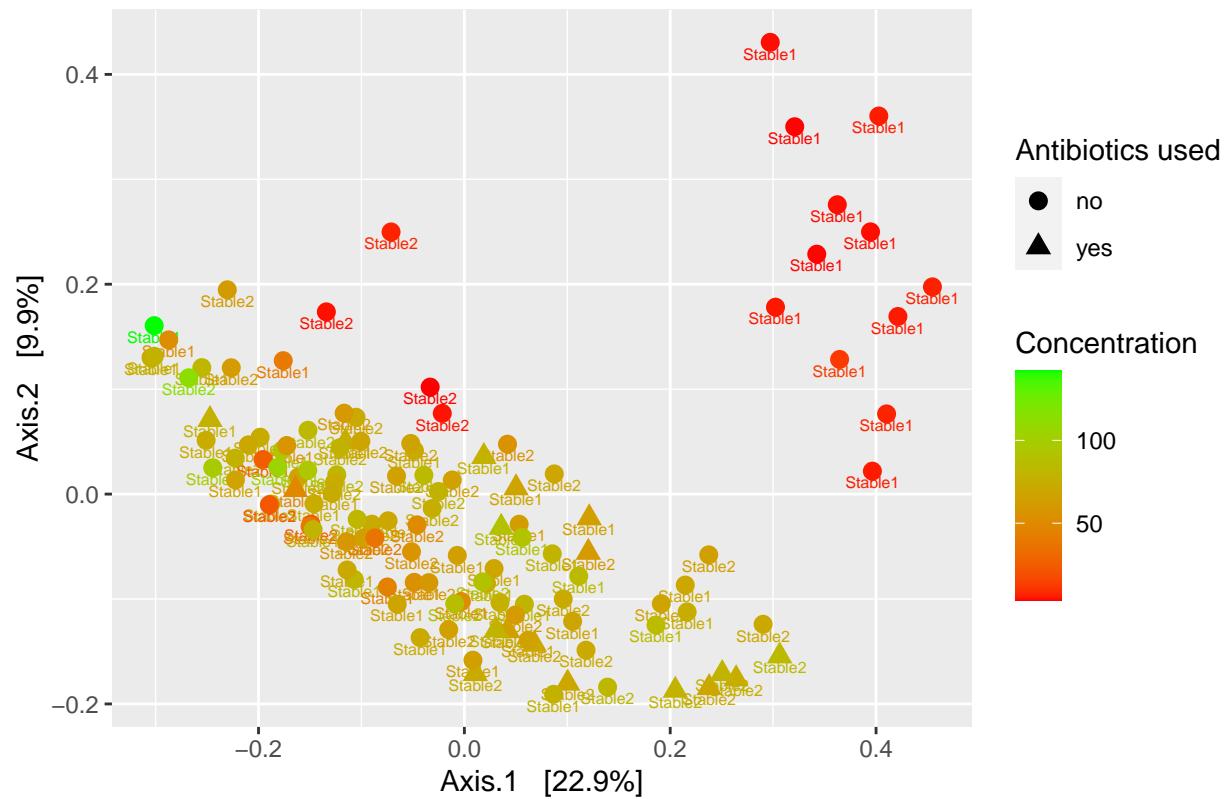
```
# there are also no differences to be found when comparing FPKM/ TPM or metaphlan and kraken2, this is
pcoa_jaccard_tpm = ordinate(Rps_tpm, "PCoA", "jaccard", binary=TRUE)
plot_ordination(Rps_tpm, pcoa_jaccard_tpm, color = "Conc...ng..µl.", shape = "AB", label = "Stable") +
  geom_point(size = 3) + labs(title = "PCoA Jaccard concentration",color = "Concentration", shape = "Antibiotics used", size = 3)
  scale_colour_gradient(low = "red", high = "green")
```

PCoA Jaccard concentration



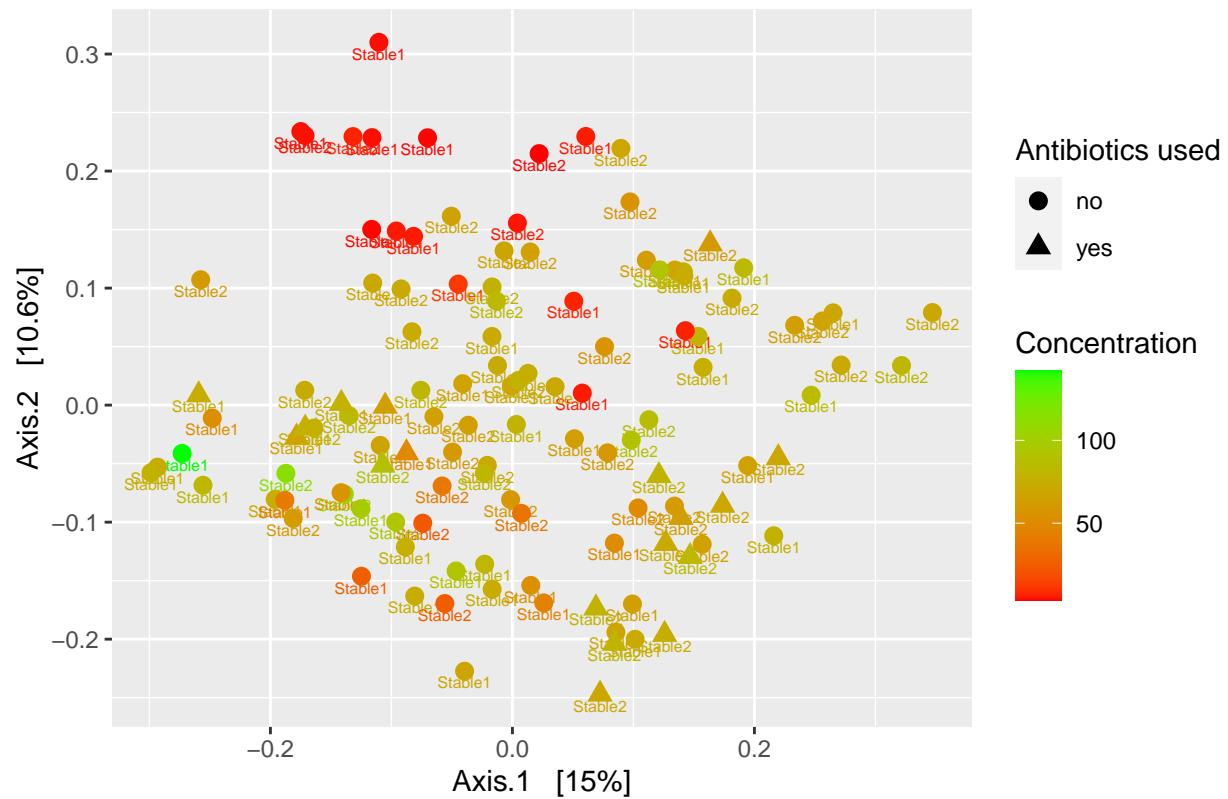
```
# now, when rounding our data a completely different picture emerges
Rps_mp_rounded = Rps_mp
otu_table(Rps_mp_rounded) = otu_table(round(as((otu_table(Rps_mp)), "matrix")), taxa_are_rows(Rps_mp))
pcoa_jaccard_mp = ordinate(Rps_mp_rounded, "PCoA", "jaccard", binary=TRUE)
plot_ordination(Rps_mp_rounded, pcoa_jaccard_mp, color = "Conc...ng..pl.", shape = "AB", label = "Stable1"
geom_point(size = 3) + labs(title = "PCoA Jaccard concentration",color = "Concentration", shape = "Antibiotics used", size = 3)
scale_colour_gradient(low = "red", high = "green")
```

PCoA Jaccard concentration



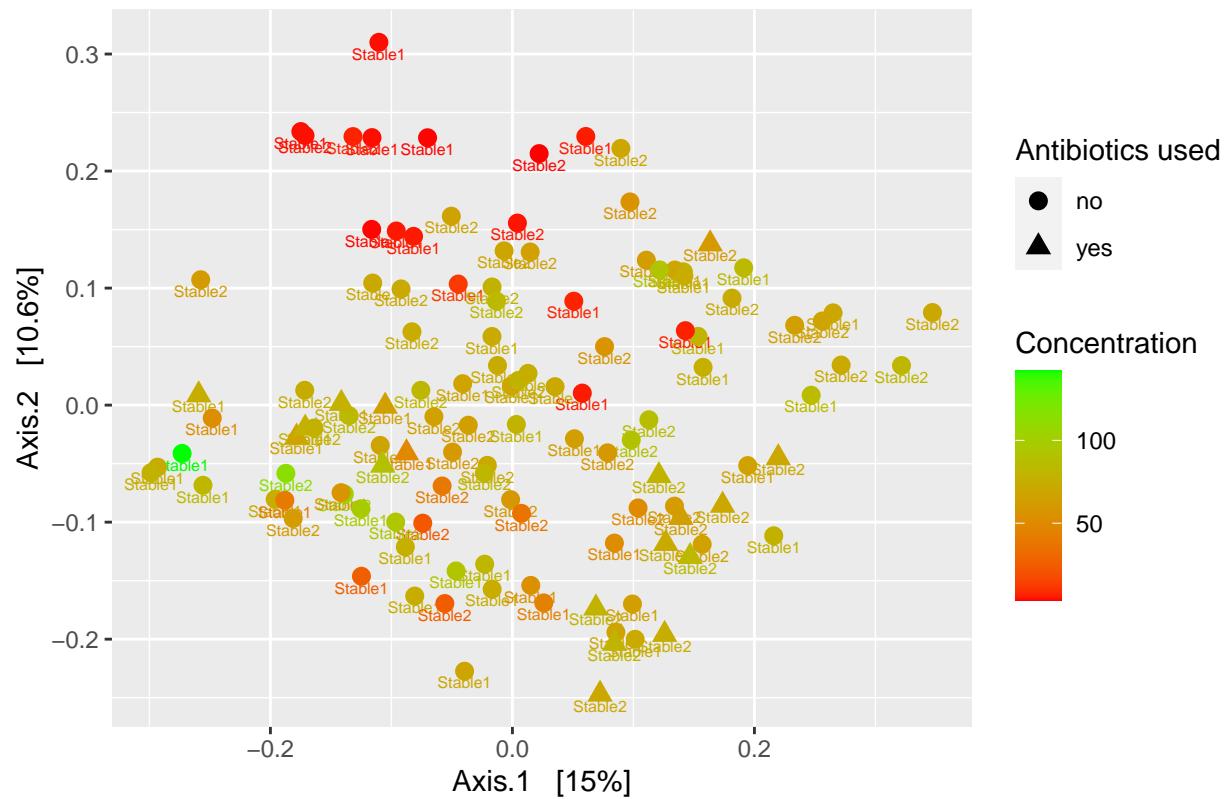
```
Rps_rounded = Rps
otu_table(Rps_rounded) = otu_table(round(as((otu_table(Rps)), "matrix")), taxa_are_rows(Rps))
pcoa_jaccard_k2 = ordinate(Rps_rounded, "PCoA", "jaccard", binary=TRUE)
plot_ordination(Rps_rounded, pcoa_jaccard_k2, color = "Conc...ng..pl.", shape = "AB", label = "Stable")
  geom_point(size = 3) + labs(title = "PCoA Jaccard concentration",color = "Concentration", shape = "Antibiotics used", scale_colour_gradient(low = "red", high = "green"))
```

PCoA Jaccard concentration



```
Rps_k2_scaled = Rps
otu_table(Rps_k2_scaled) = otu_table(round(as((otu_table(Rps_k2_scaled)), "matrix")), taxa_are_rows(Rps))
pcoa_jaccard_k2 = ordinate(Rps_k2_scaled, "PCoA", "jaccard", binary=TRUE)
plot_ordination(Rps_k2_scaled, pcoa_jaccard_k2, color = "Conc...ng..pl.", shape = "AB", label = "Stable"
geom_point(size = 3) + labs(title = "PCoA Jaccard concentration",color = "Concentration", shape = "Antibiotics used", size = 3)
scale_colour_gradient(low = "red", high = "green")
```

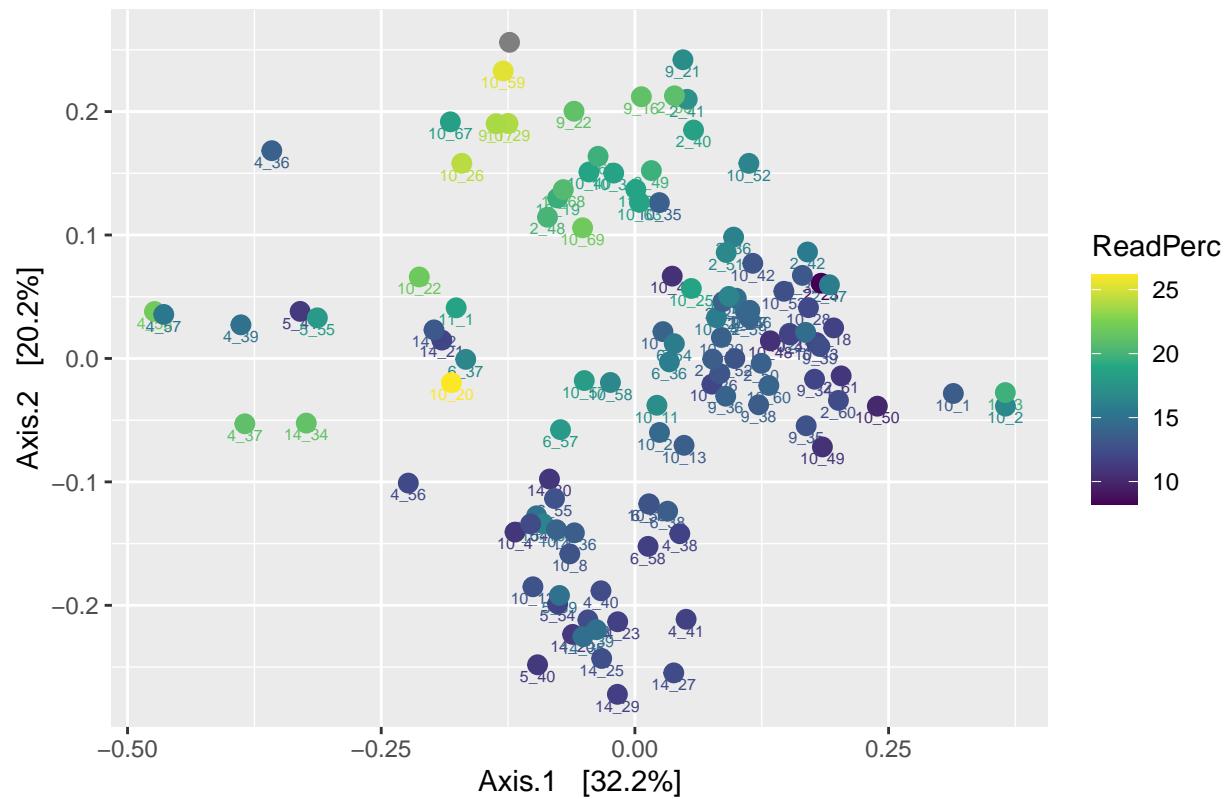
PCoA Jaccard concentration



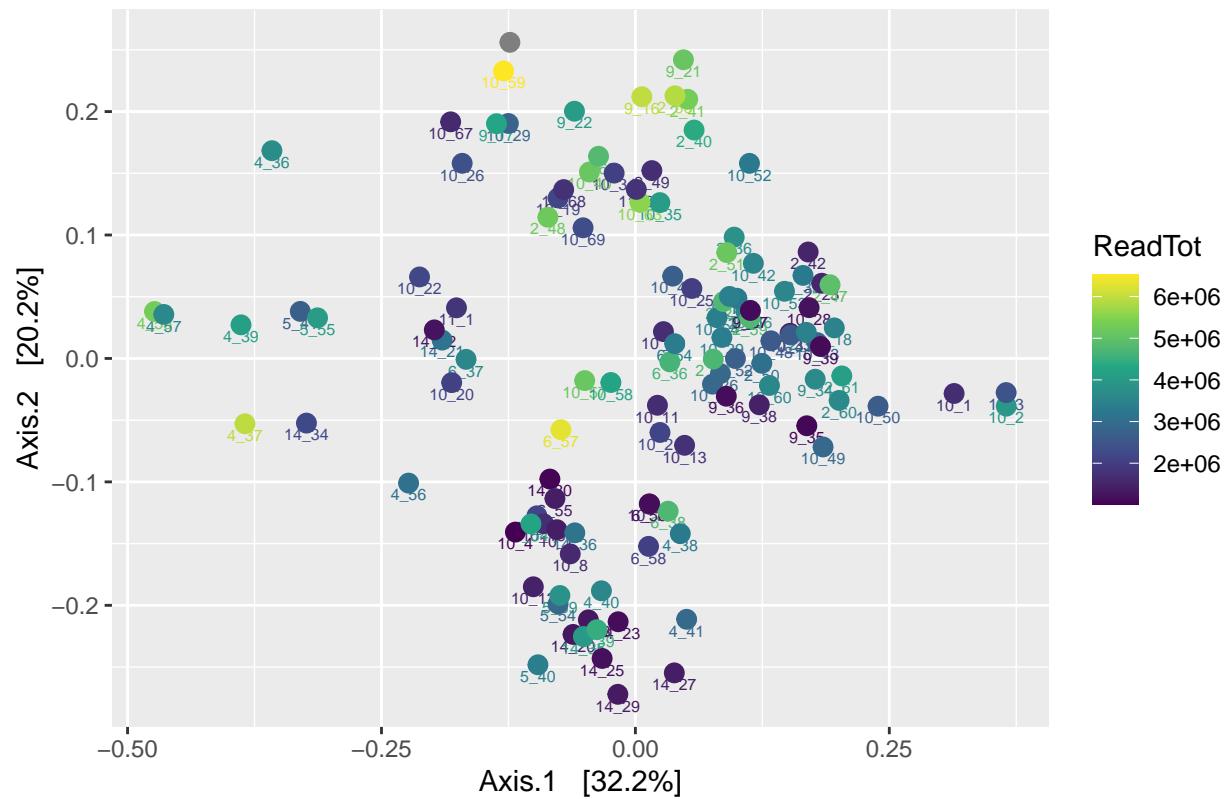
```
pcoa_jaccard_mp_scaled = ordinate(Rps_mp_scaled_copy, "PCoA", "jaccard")
```

```
# BC plots for looking at percentage and total amount of bacterial reads mapped
Rps %>% subset_samples(Sample_Unique != "2_57") %>% plot_ordination(ordinate(Rps, "PCoA", "bray") , col
  geom_point(size = 3) + labs(title = "PCoA BC percentage",color = "ReadPerc") +
  scale_colour_viridis_c()
```

PCoA BC percentage

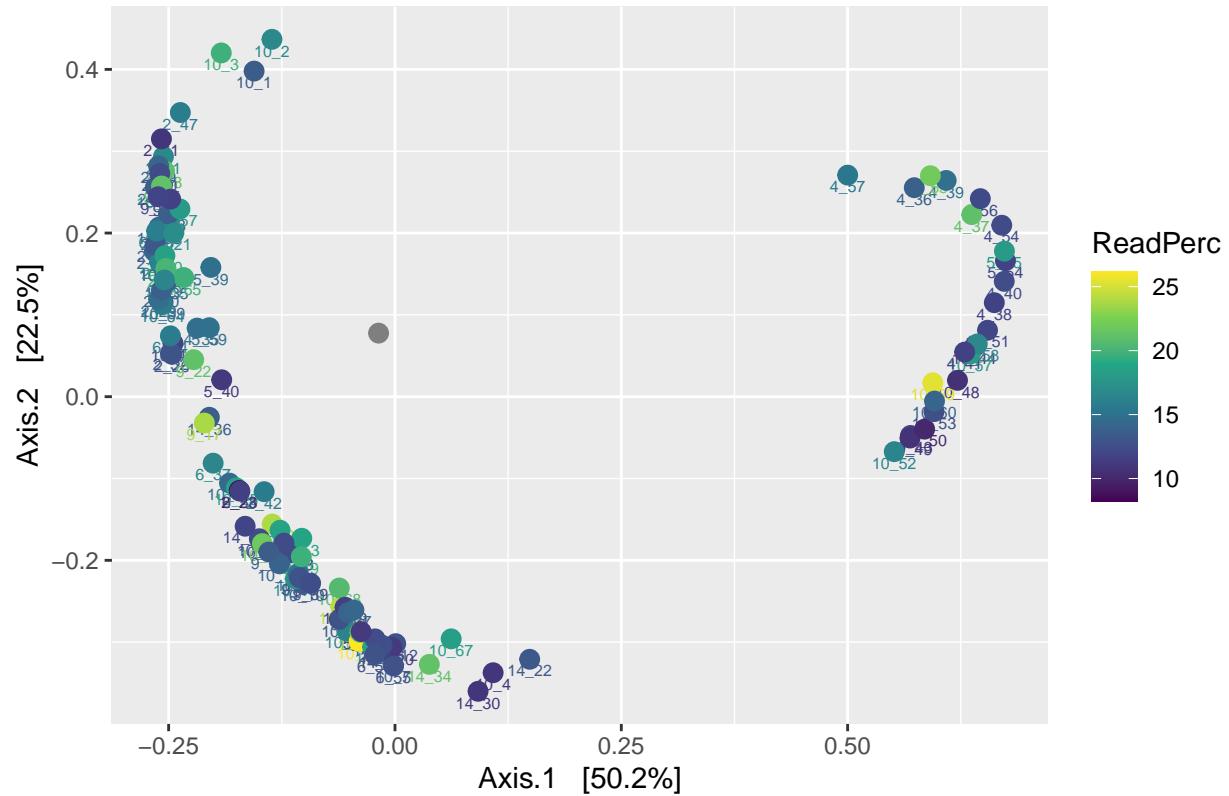


PCoA BC total



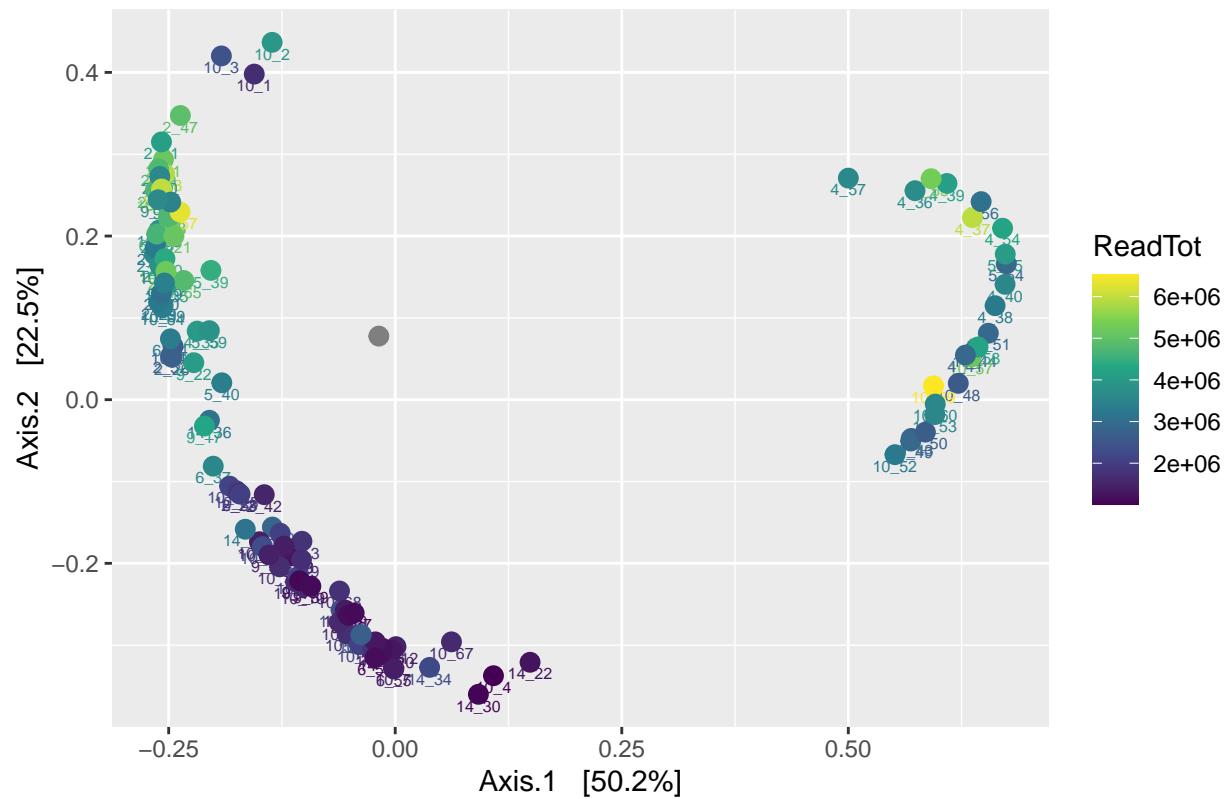
```
# metaphlan
Rps_mp %>% subset_samples(Sample_Unique != "2_57") %>% plot_ordination(ordinate(Rps_mp, "PCoA", "bray")
  geom_point(size = 3) + labs(title = "PCoA BC percentage", color = "ReadPerc") +
  scale_colour_viridis_c()
```

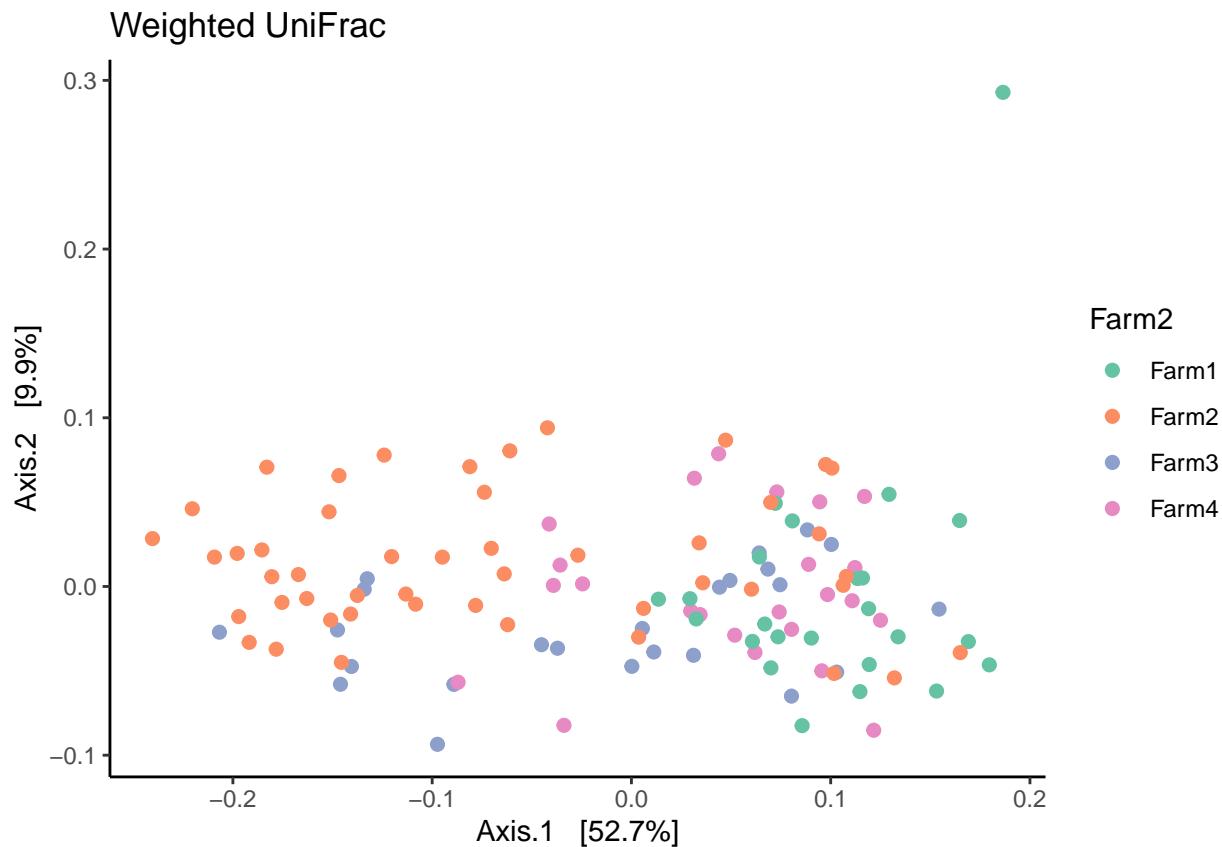
PCoA BC percentage



```
Rps_mp %>% subset_samples(Sample_Unique != "2_57") %>% plot_ordination(ordinate(Rps_mp, "PCoA", "bray") +
  geom_point(size = 3) + labs(title = "PCoA BC total", color = "ReadTot") +
  scale_colour_viridis_c()
```

PCoA BC total

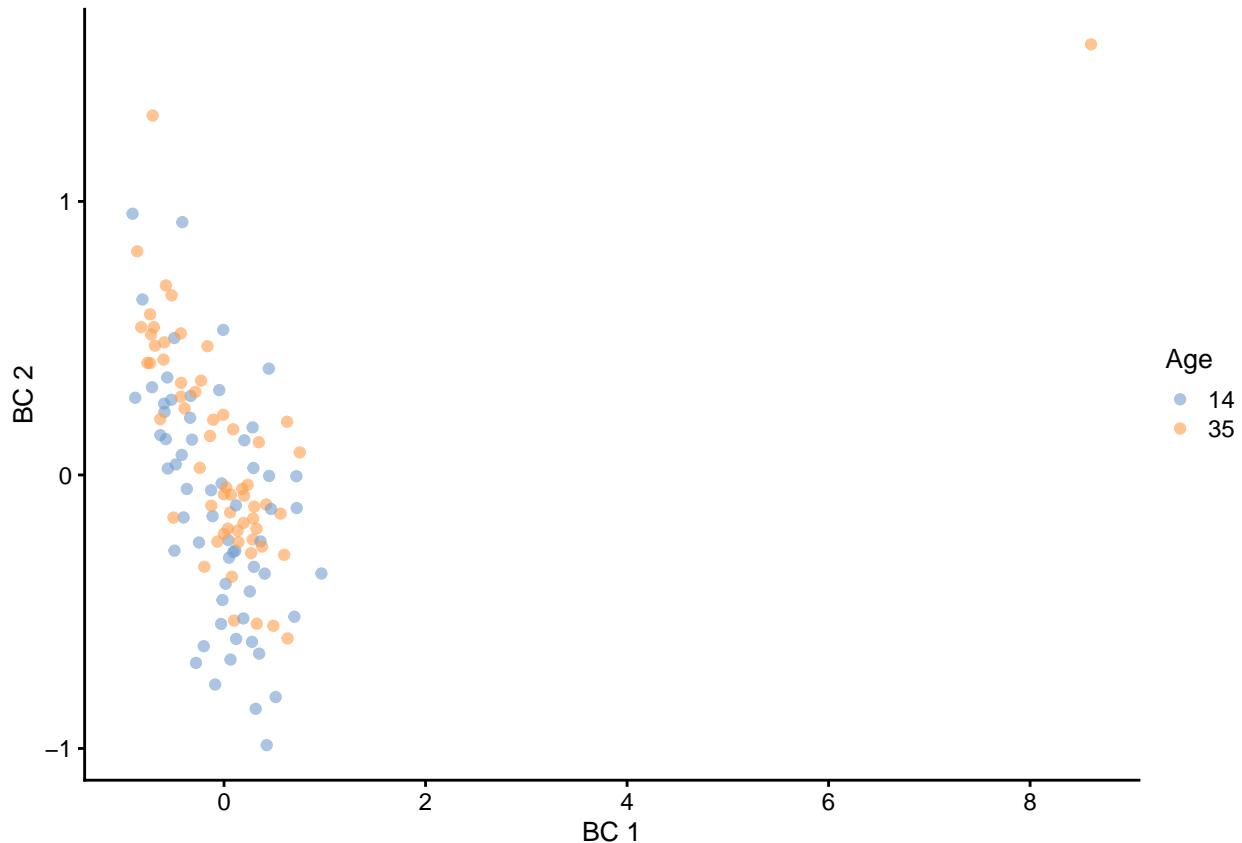




```
# NMDS
```

```
tse2 = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)
tse2 %>% transformCounts( method = "relabundance")
tse2 %>% runNMDS(FUN = vegan::vegdist, name = "BC", nmdsFUN = "monoMDS",
                     exprs_values = "relabundance",
                     keep_dist = TRUE)

tse2 %>% plotReducedDim("BC", colour_by = "Age")
```



PERMANOVAS

```
Rps_copy = Rps
colnames(Rps_copy@tax_table) = c("Phylum", "Class", "Order", "Family")

tse = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)
tse <- transformCounts(tse, method = "relabundance")

adonis2(t(assay(tse, "relabundance")) ~ AB, data = colData(tse), permutations = 9999) # not significant

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ AB, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## AB        1   0.0856 0.01435 1.7182 0.1162
## Residual 118   5.8784 0.98565
## Total    119   5.9640 1.00000

adonis2(t(assay(tse, "relabundance")) ~ Cox, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
```

```

## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Cox, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox       3    0.6906 0.1158 5.0641  1e-04 ***
## Residual 116   5.2733 0.8842
## Total     119   5.9640 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Researcher, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Researcher, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher 4    0.5387 0.09032 2.8546  3e-04 ***
## Residual   115   5.4253 0.90968
## Total      119   5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ FeedProducent, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ FeedProducent, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedProducent 2    0.7301 0.12241 8.1601  1e-04 ***
## Residual     117   5.2339 0.87759
## Total        119   5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ LitterType, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ LitterType, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType   2    1.0650 0.17858 12.718  1e-04 ***
## Residual    117   4.8989 0.82142

```

```

## Total      119  5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ FeedType, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ FeedType, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedType     1    0.1806 0.03028 3.6849 0.0087 **
## Residual 118    5.7834 0.96972
## Total     119    5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Gender, data = colData(tse), permutations = 9999) # not significant

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Gender, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender      1    0.0401 0.00672 0.7988 0.5241
## Residual 118    5.9239 0.99328
## Total     119    5.9640 1.00000

adonis2(t(assay(tse, "relabundance")) ~ Stables, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Stables, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables     9    1.7465 0.29283 5.0612  1e-04 ***
## Residual 110    4.2175 0.70717
## Total     119    5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ FlockSize, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)

```

```

## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ FlockSize, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FlockSize     5    1.3373 0.22423 6.5903  1e-04 ***
## Residual   114    4.6266 0.77577
## Total      119    5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Farm2, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Farm2, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2       3    1.235 0.20707 10.098  1e-04 ***
## Residual  116    4.729 0.79293
## Total      119    5.964 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ AgeParentStock, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ AgeParentStock, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## AgeParentStock  4    1.469 0.24631 9.3955  1e-04 ***
## Residual     115    4.495 0.75369
## Total        119    5.964 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ Age, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ Age, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Age         1    0.1806 0.03028 3.6849 0.0067 **
## Residual  118    5.7834 0.96972

```

```

## Total     119   5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ ReadPerc, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ ReadPerc, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## ReadPerc    1   0.4574 0.07669 9.8009  1e-04 ***
## Residual 118   5.5066 0.92331
## Total     119   5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse, "relabundance")) ~ ReadTot, data = colData(tse), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse, "relabundance")) ~ ReadTot, data = colData(tse), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## ReadTot    1   0.2641 0.04429 5.4678  7e-04 ***
## Residual 118   5.6998 0.95571
## Total     119   5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# variances: AB: 0.0143, Cox: 0.1158, Researcher: 0.090, FP : 0.1224 , LitterType: 0.179, FT : 0.030, Genotype: 0.0001, Gender: 0.0001, Order: 0.0001, Stable: 0.293, FS: 0.224 , Farm 0.207, APS : 0.246, Age: 0.0302, readperc : 0.077, readtot : 0.044
# Order: Stable>APS>FS>Farm>LT>FP>Cox>Researcher>Age>FT>AB>Gender

# basically, composition seems to be different over every single variable, except for gender

# on clust90 level
tse_clust90 <- agglomerateByRank(tse, "Order")
tse_clust90 <- transformCounts(tse_clust90, method = "relabundance")

adonis2(t(assay(tse_clust90, "relabundance")) ~ AB, data = colData(tse_clust90), permutations = 9999) #
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ AB, data = colData(tse_clust90), permutations = 9999)
```

```

##          Df SumOfSqs      R2      F Pr(>F)
## AB          1  0.0494 0.00926 1.1027 0.3044
## Residual  118  5.2824 0.99074
## Total      119  5.3318 1.00000

adonis2(t(assay(tse_clust90, "relabundance")) ~ Cox, data = colData(tse_clust90), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ Cox, data = colData(tse_clust90), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox          3  0.6332 0.11875 5.2106 1e-04 ***
## Residual   116  4.6986 0.88125
## Total       119  5.3318 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_clust90, "relabundance")) ~ Researcher, data = colData(tse_clust90), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ Researcher, data = colData(tse_clust90), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher    4  0.5293 0.09927 3.1686 3e-04 ***
## Residual     115  4.8025 0.90073
## Total        119  5.3318 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_clust90, "relabundance")) ~ FeedProducent, data = colData(tse_clust90), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ FeedProducent, data = colData(tse_clust90), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FeedProducent  2  0.6761 0.1268 8.4952 1e-04 ***
## Residual      117  4.6557 0.8732
## Total         119  5.3318 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

adonis2(t(assay(tse_clust90, "relabundance")) ~ LitterType, data = colData(tse_clust90), permutations = 9999

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ LitterType, data = colData(tse_clust90), p
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType    2   1.0701 0.20069 14.688  1e-04 ***
## Residual    117   4.2617 0.79931
## Total       119   5.3318 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_clust90, "relabundance")) ~ FeedType, data = colData(tse_clust90), permutations = 9999

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ FeedType, data = colData(tse_clust90), per
##          Df SumOfSqs      R2      F Pr(>F)
## FeedType     1   0.1517 0.02845  3.456  0.0169 *
## Residual   118   5.1801 0.97155
## Total       119   5.3318 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_clust90, "relabundance")) ~ Gender, data = colData(tse_clust90), permutations = 9999

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ Gender, data = colData(tse_clust90), perm
##          Df SumOfSqs      R2      F Pr(>F)
## Gender       1   0.0367 0.00688  0.8177  0.455
## Residual   118   5.2951 0.99312
## Total       119   5.3318 1.00000

adonis2(t(assay(tse_clust90, "relabundance")) ~ Stables, data = colData(tse_clust90), permutations = 9999

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ Stables, data = colData(tse_clust90), perm
```

```

##          Df SumOfSqs      R2      F Pr(>F)
## Stables     9   1.7063 0.32002 5.7522  1e-04 ***
## Residual  110   3.6255 0.67998
## Total      119   5.3318 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_clust90, "relabundance")) ~ FlockSize, data = colData(tse_clust90), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ FlockSize, data = colData(tse_clust90), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## FlockSize    5   1.3172 0.24704 7.4806  1e-04 ***
## Residual  114   4.0146 0.75296
## Total      119   5.3318 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_clust90, "relabundance")) ~ Farm2, data = colData(tse_clust90), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ Farm2, data = colData(tse_clust90), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2       3   1.2207 0.22894 11.481  1e-04 ***
## Residual  116   4.1111 0.77106
## Total      119   5.3318 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(assay(tse_clust90, "relabundance")) ~ AgeParentStock, data = colData(tse_clust90), permutations = 9999)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(assay(tse_clust90, "relabundance")) ~ AgeParentStock, data = colData(tse_clust90), permutations = 9999)
##          Df SumOfSqs      R2      F Pr(>F)
## AgeParentStock  4   1.4546 0.27282 10.786  1e-04 ***
## Residual     115   3.8772 0.72718
## Total        119   5.3318 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

# same trends on genus level

# for different ordination methods
ps1.rel <- microbiome::transform(Rps, "compositional")
metadf <- data.frame(sample_data(ps1.rel))

# alternative calculations
#permanova = adonis(t(otu) ~ Age, data = meta, permutations=9999, method = "bray")
#permanova$ao.v.tab

unifrac.dist <- UniFrac(ps1.rel)

adonis2(unifrac.dist ~ Age, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Age, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Age       1   0.2728 0.03107 3.7832  0.001 ***
## Residual 118   8.5093 0.96893
## Total    119   8.7821 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(unifrac.dist ~ AB, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB        1   0.1554 0.0177 2.126  0.017 *
## Residual 118   8.6267 0.9823
## Total    119   8.7821 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(unifrac.dist ~ Farm2, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)

```

```

## Farm2      3   1.1478 0.1307 5.8134  0.001 ***
## Residual  116  7.6343 0.8693
## Total     119  8.7821 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

adonis2(unifrac.dist ~ Cox, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox      3   0.8015 0.09127 3.8834  0.001 ***
## Residual 116  7.9806 0.90873
## Total    119  8.7821 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

adonis2(unifrac.dist ~ Researcher, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher 4   0.5524 0.06291 1.9299  0.001 ***
## Residual   115  8.2297 0.93709
## Total     119  8.7821 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

adonis2(unifrac.dist ~ LitterType, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ LitterType, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType 2   0.8790 0.10009 6.5067  0.001 ***
## Residual   117  7.9031 0.89991
## Total     119  8.7821 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```

adonis2(unifrac.dist ~ Gender, data = metadf) # not sign

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Gender, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Gender     1   0.0417 0.00475 0.5632  0.936
## Residual 118   8.7404 0.99525
## Total     119   8.7821 1.00000

adonis2(unifrac.dist ~ Stables, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = unifrac.dist ~ Stables, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Stables    9   2.2913 0.26091 4.3146  0.001 ***
## Residual 110   6.4908 0.73909
## Total     119   8.7821 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

# same patterns arise, but AB is sign

wunifrac.dist <- UniFrac(ps1.rel,
                           weighted = TRUE)

adonis2(wunifrac.dist ~ Age, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Age, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Age        1   0.0915 0.02747 3.3334  0.022 *
## Residual 118   3.2384 0.97253
## Total     119   3.3298 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

adonis2(wunifrac.dist ~ AB, data = metadf) # not sign

## Permutation test for adonis under reduced model

```

```

## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ AB, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## AB         1   0.0289 0.00869 1.0348  0.332
## Residual  118   3.3009 0.99131
## Total     119   3.3298 1.00000

adonis2(wunifrac.dist ~ Farm2, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Farm2, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2      3   0.7721 0.23189 11.673  0.001 ***
## Residual  116   2.5577 0.76811
## Total     119   3.3298 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(wunifrac.dist ~ Cox, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Cox, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Cox        3   0.4514 0.13556 6.0636  0.001 ***
## Residual  116   2.8785 0.86444
## Total     119   3.3298 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(wunifrac.dist ~ Researcher, data = metadf)

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = wunifrac.dist ~ Researcher, data = metadf)
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher  4   0.2819 0.08465 2.6588  0.002 **
## Residual   115   3.0480 0.91535
## Total      119   3.3298 1.00000

```

```
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(wunifrac.dist ~ LitterType, data = metadf)
```

```
## Permutation test for adonis under reduced model  
## Terms added sequentially (first to last)  
## Permutation: free  
## Number of permutations: 999  
##  
## adonis2(formula = wunifrac.dist ~ LitterType, data = metadf)  
##          Df SumOfSqs      R2      F Pr(>F)  
## LitterType    2   0.6606 0.19839 14.478  0.001 ***  
## Residual  117   2.6692 0.80161  
## Total     119   3.3298 1.00000  
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis2(wunifrac.dist ~ Gender, data = metadf) # not sign
```

```
## Permutation test for adonis under reduced model  
## Terms added sequentially (first to last)  
## Permutation: free  
## Number of permutations: 999  
##  
## adonis2(formula = wunifrac.dist ~ Gender, data = metadf)  
##          Df SumOfSqs      R2      F Pr(>F)  
## Gender      1   0.0216 0.00648 0.7698  0.487  
## Residual 118   3.3083 0.99352  
## Total     119   3.3298 1.00000
```

```
adonis2(wunifrac.dist ~ Stables, data = metadf)
```

```
## Permutation test for adonis under reduced model  
## Terms added sequentially (first to last)  
## Permutation: free  
## Number of permutations: 999  
##  
## adonis2(formula = wunifrac.dist ~ Stables, data = metadf)  
##          Df SumOfSqs      R2      F Pr(>F)  
## Stables     9   1.0491 0.31507 5.6222  0.001 ***  
## Residual 110   2.2807 0.68493  
## Total     119   3.3298 1.00000  
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# sameish patterns
```

```
meta <- meta(ps1.rel)  
adonis2(t(abundances(ps1.rel)) ~ Age, data = meta, permutations=9999, method = "bray")
```

```

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Age, data = meta, permutations = 9999, method = "bray")
##          Df SumOfSqs      R2      F Pr(>F)
## Age       1   0.1806 0.03028 3.6849 0.0081 **
## Residual 118   5.7834 0.96972
## Total    119   5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(abundances(ps1.rel)) ~ AB, data = meta, permutations=9999, method = "bray") # not sign

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ AB, data = meta, permutations = 9999, method = "bray")
##          Df SumOfSqs      R2      F Pr(>F)
## AB       1   0.0856 0.01435 1.7182 0.1122
## Residual 118   5.8784 0.98565
## Total    119   5.9640 1.00000

adonis2(t(abundances(ps1.rel)) ~ Farm2, data = meta, permutations=9999, method = "bray")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Farm2, data = meta, permutations = 9999, method = "bray")
##          Df SumOfSqs      R2      F Pr(>F)
## Farm2     3   1.235 0.20707 10.098  1e-04 ***
## Residual 116   4.729 0.79293
## Total    119   5.964 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(abundances(ps1.rel)) ~ Cox, data = meta, permutations=9999, method = "bray")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Cox, data = meta, permutations = 9999, method = "bray")
##          Df SumOfSqs      R2      F Pr(>F)
## Cox       3   0.6906 0.1158 5.0641  1e-04 ***
## Residual 116   5.2733 0.8842

```

```

## Total     119   5.9640 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(abundances(ps1.rel)) ~ Researcher, data = meta, permutations=9999, method = "bray")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Researcher, data = meta, permutations = 9999, method = "bray")
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher    4   0.5387 0.09032 2.8546  5e-04 ***
## Residual    115   5.4253 0.90968
## Total       119   5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(abundances(ps1.rel)) ~ LitterType, data = meta, permutations=9999, method = "bray")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ LitterType, data = meta, permutations = 9999, method = "bray")
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType    2   1.0650 0.17858 12.718  1e-04 ***
## Residual    117   4.8989 0.82142
## Total       119   5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(abundances(ps1.rel)) ~ Gender, data = meta, permutations=9999, method = "bray") # not sign

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Gender, data = meta, permutations = 9999, method = "bray")
##          Df SumOfSqs      R2      F Pr(>F)
## Gender       1   0.0401 0.00672 0.7988   0.51
## Residual   118   5.9239 0.99328
## Total       119   5.9640 1.00000

adonis2(t(abundances(ps1.rel)) ~ Stables, data = meta, permutations=9999, method = "bray")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)

```

```

## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Stables, data = meta, permutations = 9999, method = "bray"
##          Df SumOfSqs      R2      F Pr(>F)
## Stables     9    1.7465 0.29283 5.0612  1e-04 ***
## Residual 110    4.2175 0.70717
## Total     119    5.9640 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# and BC

adonis2(t(abundances(ps1.rel)) ~ Age, data = meta, permutations=9999, method = "jaccard")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Age, data = meta, permutations = 9999, method = "jaccard")
##          Df SumOfSqs      R2      F Pr(>F)
## Age        1    0.356 0.0276 3.3491 0.0062 **
## Residual 118   12.542 0.9724
## Total     119   12.898 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(abundances(ps1.rel)) ~ AB, data = meta, permutations=9999, method = "jaccard") # not sign

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ AB, data = meta, permutations = 9999, method = "jaccard")
##          Df SumOfSqs      R2      F Pr(>F)
## AB        1    0.1971 0.01528 1.8309 0.0759 .
## Residual 118   12.7005 0.98472
## Total     119   12.8976 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(abundances(ps1.rel)) ~ Farm2, data = meta, permutations=9999, method = "jaccard")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Farm2, data = meta, permutations = 9999, method = "jaccard")
##          Df SumOfSqs      R2      F Pr(>F)

```

```

## Farm2      3   2.3128 0.17932 8.4487  1e-04 ***
## Residual  116  10.5848 0.82068
## Total     119  12.8976 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(abundances(ps1.rel)) ~ Cox, data = meta, permutations=9999, method = "jaccard")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Cox, data = meta, permutations = 9999, method = "jaccard")
##          Df SumOfSqs      R2      F Pr(>F)
## Cox      3    1.284 0.09956 4.2752  1e-04 ***
## Residual 116  11.614 0.90044
## Total    119  12.898 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(abundances(ps1.rel)) ~ Researcher, data = meta, permutations=9999, method = "jaccard")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Researcher, data = meta, permutations = 9999, method = "jaccard")
##          Df SumOfSqs      R2      F Pr(>F)
## Researcher 4    1.0404 0.08067 2.5226  2e-04 ***
## Residual   115  11.8572 0.91933
## Total     119  12.8976 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

adonis2(t(abundances(ps1.rel)) ~ LitterType, data = meta, permutations=9999, method = "jaccard")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ LitterType, data = meta, permutations = 9999, method = "jaccard")
##          Df SumOfSqs      R2      F Pr(>F)
## LitterType 2    1.9707 0.1528 10.551  1e-04 ***
## Residual   117  10.9269 0.8472
## Total     119  12.8976 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

adonis2(t(abundances(ps1.rel)) ~ Gender, data = meta, permutations=9999, method = "jaccard") # not signif

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Gender, data = meta, permutations = 9999, method = "jaccard")
##          Df SumOfSqs      R2      F Pr(>F)
## Gender     1    0.085 0.00659 0.783 0.5882
## Residual 118   12.812 0.99341
## Total     119   12.898 1.00000

adonis2(t(abundances(ps1.rel)) ~ Stables, data = meta, permutations=9999, method = "jaccard")

## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = t(abundances(ps1.rel)) ~ Stables, data = meta, permutations = 9999, method = "jaccard")
##          Df SumOfSqs      R2      F Pr(>F)
## Stables    9    3.3364 0.25869 4.2651  1e-04 ***
## Residual 110   9.5611 0.74131
## Total     119   12.8976 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# as well as jaccard

```

PERMANOVA plots - Age

```

permanova_age <- adonis(t(assay(tse, "relabundance")) ~ Age, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_age)[["Age1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

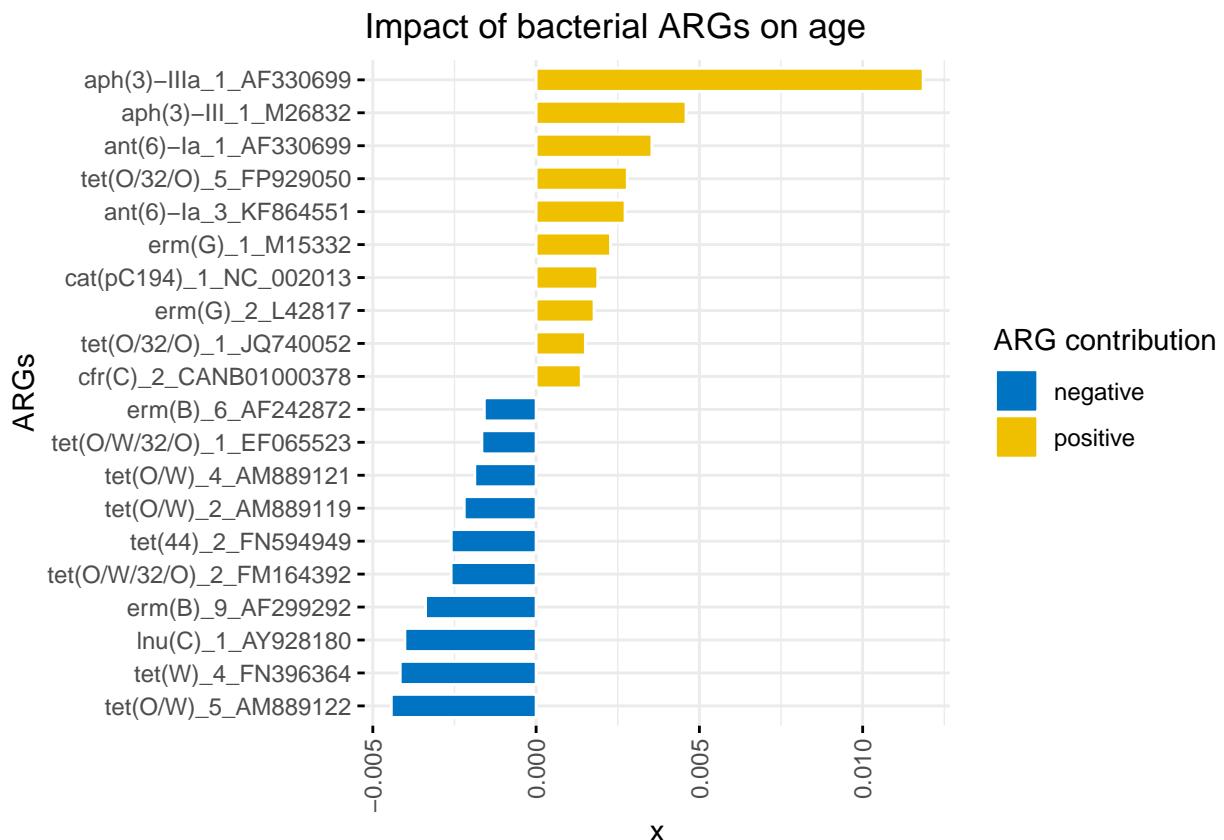
ggbarplot(df, x = "y", y = "x",
           fill = "contr",                      # change fill color by mpg_level
           color = "white",                      # Set bar border colors to white
           palette = "jco",                      # jco journal color palett. see ?ggpar
           sort.val = "asc",                      # Sort the value in ascending order
           sort.by.groups = FALSE,                # Don't sort inside each group

```

```

x.text.angle = 90,           # Rotate vertically x axis texts
xlab = "ARGs",
legend.title = "ARG contribution",
title = "Impact of bacterial ARGs on age",
rotate = TRUE,
ggtheme = theme_minimal())

```



```

# AB

permanova_AB <- adonis(t(assay(tse, "relabundance")) ~ AB, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_AB)[["AB1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                             unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

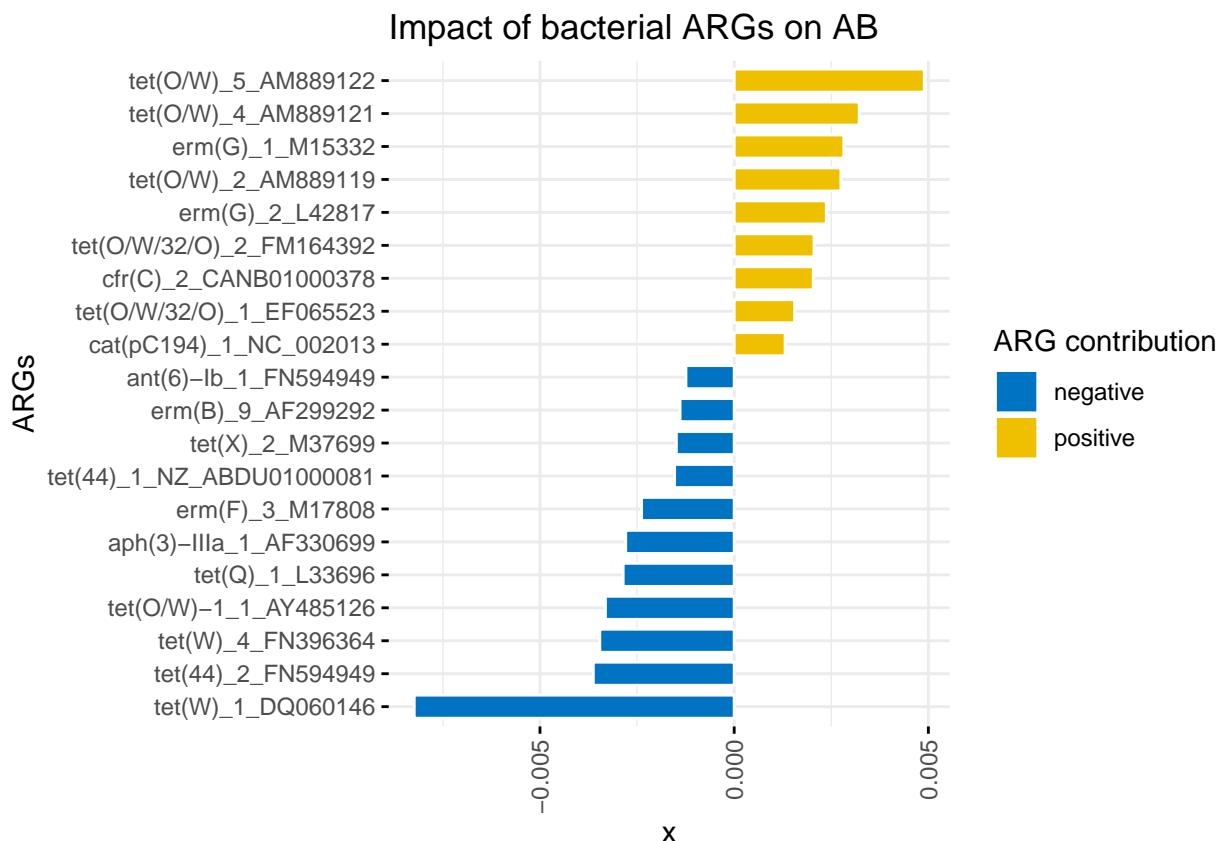
ggbarplot(df, x = "y", y = "x",
          fill = "contr",           # change fill color by mpg_level
          color = "white",          # Set bar border colors to white
          palette = "jco",           # jco journal color palett. see ?ggpar

```

```

sort.val = "asc",           # Sort the value in ascending order
sort.by.groups = FALSE,    # Don't sort inside each group
x.text.angle = 90,         # Rotate vertically x axis texts
xlab = "ARGs",
legend.title = "ARG contribution",
title = "Impact of bacterial ARGs on AB",
rotate = TRUE,
ggtheme = theme_minimal())

```



```

# Stable

permanova_stable <- adonis(t(assay(tse, "relabundance")) ~ Stables, data = colData(tse), permutations = 1000)

coef <- coefficients(permanova_stable)[["Stables1"],]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                           unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

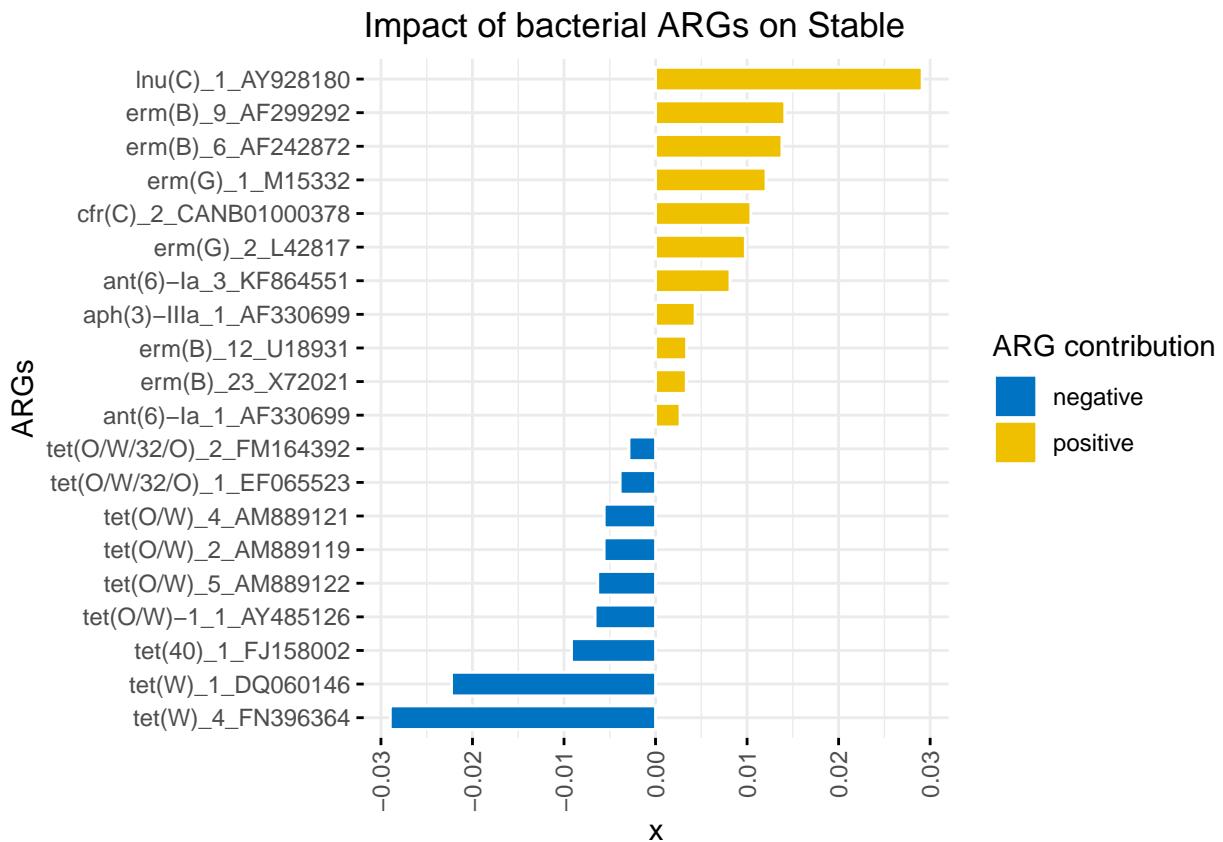
ggbarplot(df, x = "y", y = "x",
          fill = "contr",           # change fill color by mpg_level

```

```

color = "white",           # Set bar border colors to white
palette = "jco",          # jco journal color palett. see ?ggpar
sort.val = "asc",          # Sort the value in ascending order
sort.by.groups = FALSE,    # Don't sort inside each group
x.text.angle = 90,         # Rotate vertically x axis texts
xlab = "ARGs",
legend.title = "ARG contribution",
title = "Impact of bacterial ARGs on Stable",
rotate = TRUE,
ggtheme = theme_minimal())

```



Farm

```

permanova_farm <- adonis(t(assay(tse, "relabundance")) ~ Farm2, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_farm)[["Farm21",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))

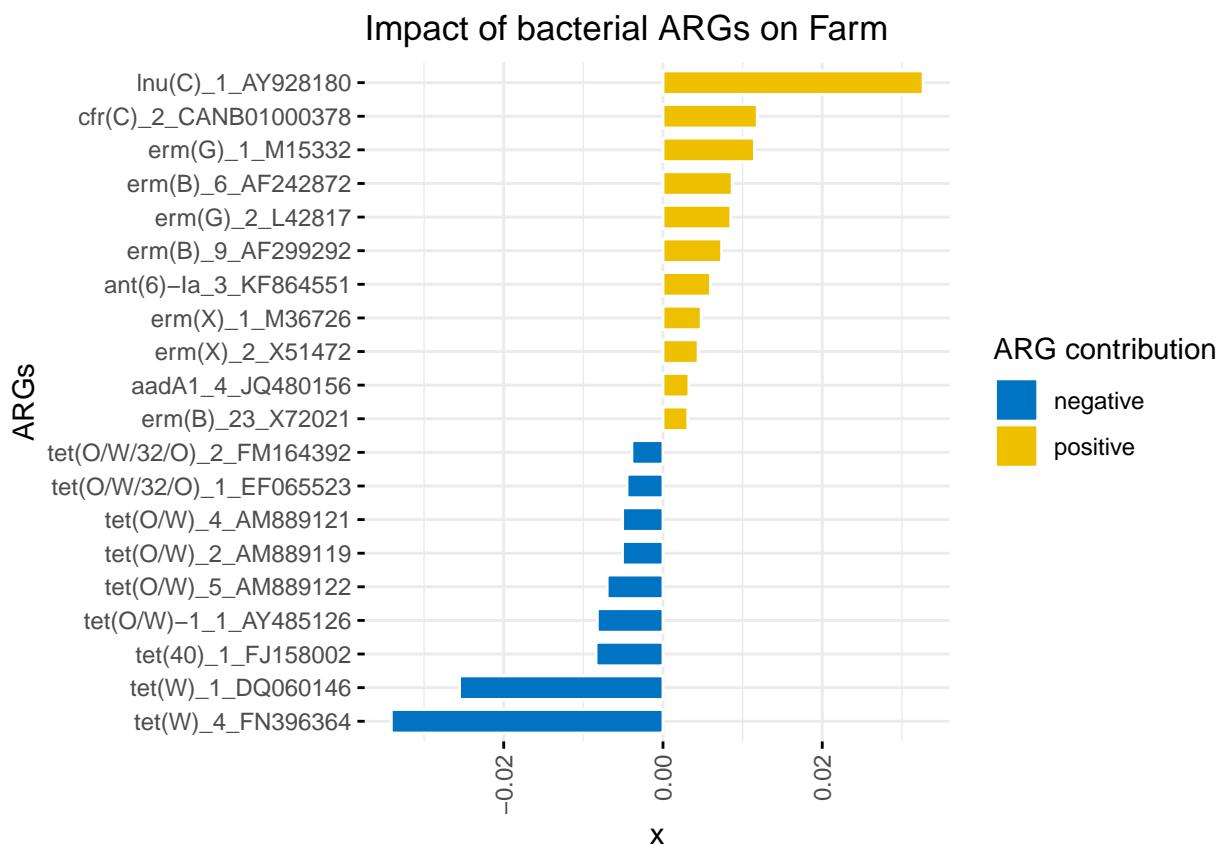
df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

```

```

ggbarplot(df, x = "y", y = "x",
          fill = "contr",           # change fill color by mpg_level
          color = "white",          # Set bar border colors to white
          palette = "jco",          # jco journal color palett. see ?ggpar
          sort.val = "asc",          # Sort the value in ascending order
          sort.by.groups = FALSE,    # Don't sort inside each group
          x.text.angle = 90,         # Rotate vertically x axis texts
          xlab = "ARGs",             # ARG contribution
          legend.title = "ARG contribution",
          title = "Impact of bacterial ARGs on Farm",
          rotate = TRUE,
          ggtheme = theme_minimal())

```



```

# Agent

permanova_agent <- adonis(t(assay(tse, "relabundance")) ~ Cox, data = colData(tse), permutations = 9999)

coef <- coefficients(permanova_agent)[["Cox1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                            unique(names(top.coef)))))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),

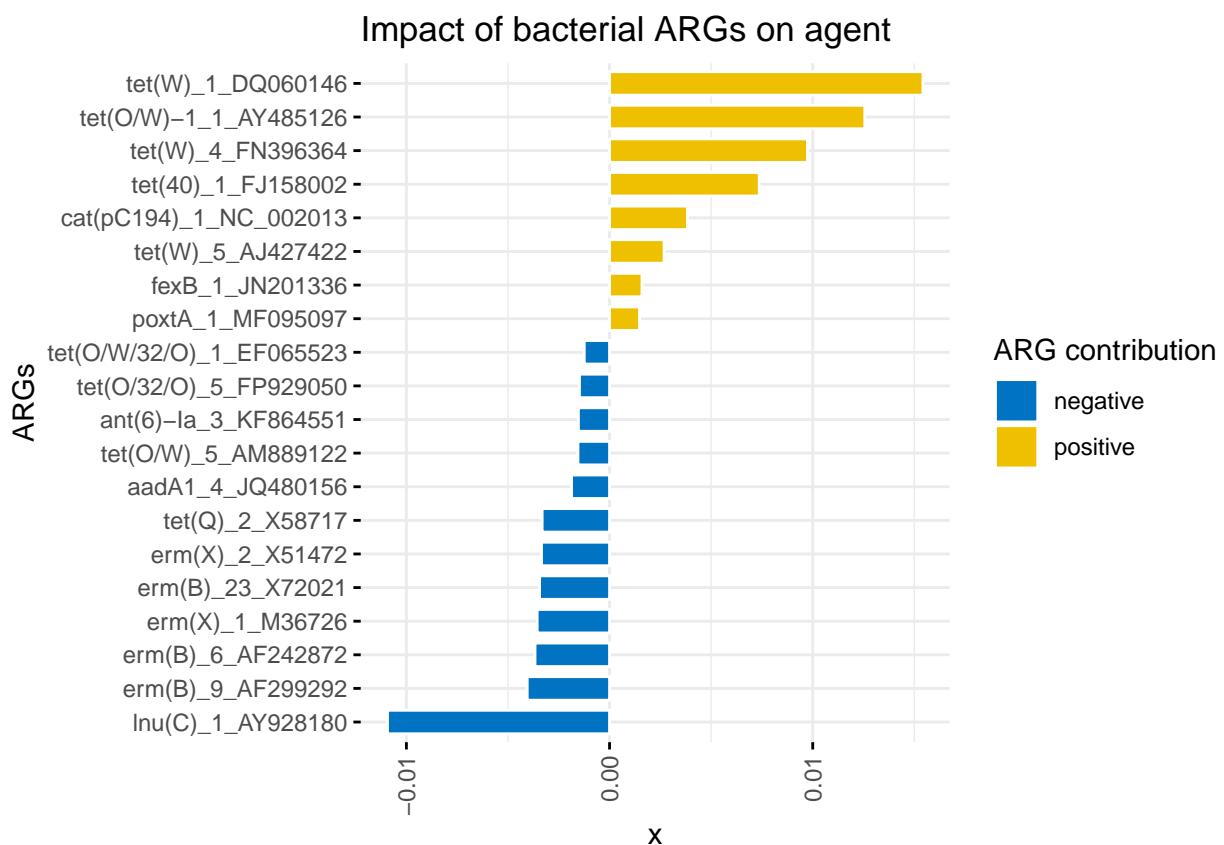
```

```

levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",          # jco journal color palett. see ?ggpar
           sort.val = "asc",          # Sort the value in ascending order
           sort.by.groups = FALSE,    # Don't sort inside each group
           x.text.angle = 90,          # Rotate vertically x axis texts
           xlab = "ARGs",
           legend.title = "ARG contribution",
           title = "Impact of bacterial ARGs on agent",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# same plots but for argclust90

# Age

permanova_age <- adonis(t(assay(tse_clust90, "relabundance")) ~ Age, data = colData(tse_clust90), permutations = 999)

coef <- coefficients(permanova_age)[["Age1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,

```

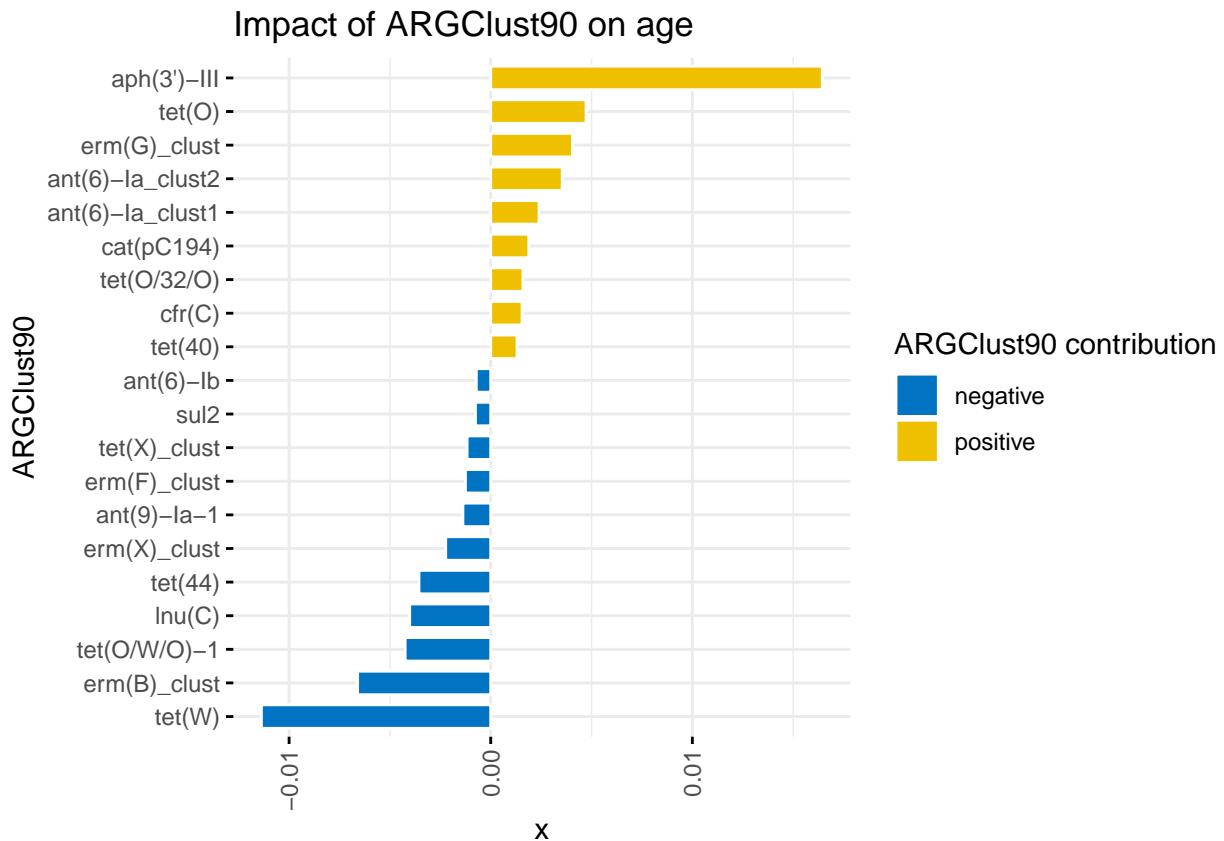
```

y = factor(names(top.coef),
           unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                    levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",                      # change fill color by mpg_level
           color = "white",                     # Set bar border colors to white
           palette = "jco",                   # jco journal color palett. see ?ggpar
           sort.val = "asc",                  # Sort the value in ascending order
           sort.by.groups = FALSE,            # Don't sort inside each group
           x.text.angle = 90,                 # Rotate vertically x axis texts
           xlab = "ARGClust90",
           legend.title = "ARGClust90 contribution",
           title = "Impact of ARGClust90 on age",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# AB

permanova_AB <- adonis(t(assay(tse_clust90, "relabundance")) ~ AB, data = colData(tse_clust90), permutations = 999)

coef <- coefficients(permanova_AB)[["AB1", ]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

```

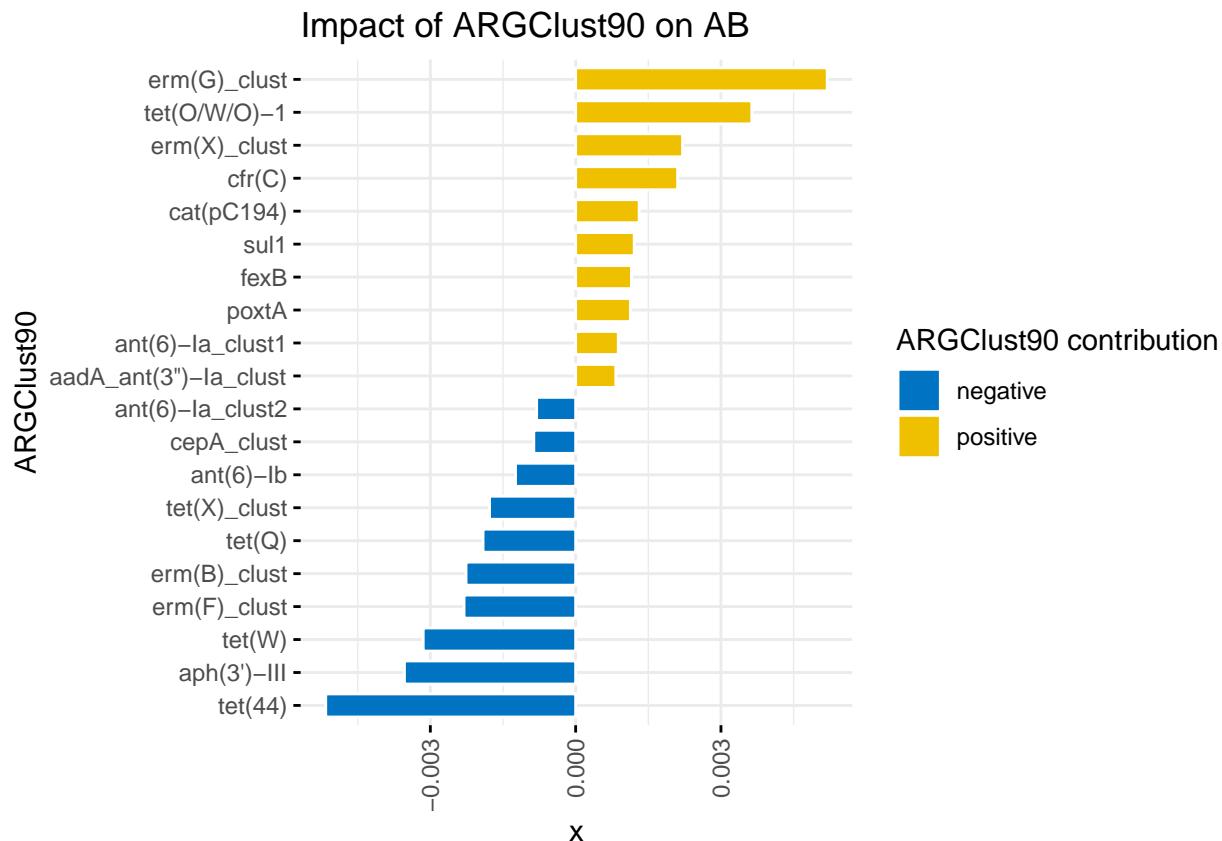
```

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                             unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",         # jco journal color palett. see ?ggpar
           sort.val = "asc",         # Sort the value in ascending order
           sort.by.groups = FALSE,   # Don't sort inside each group
           x.text.angle = 90,        # Rotate vertically x axis texts
           xlab = "ARGClust90",
           legend.title = "ARGClust90 contribution",
           title = "Impact of ARGClust90 on AB",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# Stable

permanova_stable <- adonis(t(assay(tse_clust90, "relabundance")) ~ Stables, data = colData(tse_clust90))

coef <- coefficients(permanova_stable)[["Stables1", ]]

```

```

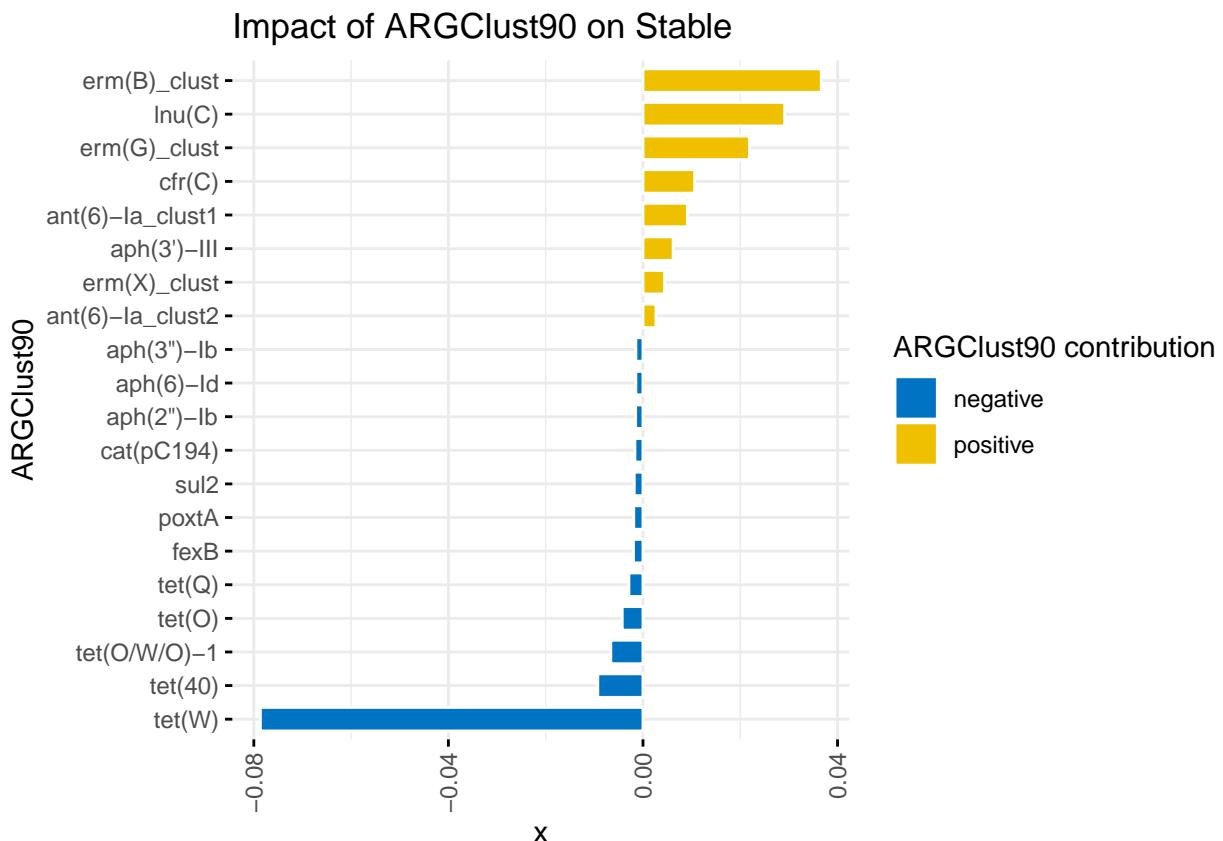
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                             unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",          # change fill color by mpg_level
           color = "white",         # Set bar border colors to white
           palette = "jco",          # jco journal color palett. see ?ggpar
           sort.val = "asc",         # Sort the value in ascending order
           sort.by.groups = FALSE,   # Don't sort inside each group
           x.text.angle = 90,        # Rotate vertically x axis texts
           xlab = "ARGClust90",
           legend.title = "ARGClust90 contribution",
           title = "Impact of ARGClust90 on Stable",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```
# Farm
```

```
permanova_farm <- adonis(t(assay(tse_clust90, "relabundance")) ~ Farm2, data = colData(tse_clust90), pe
```

```

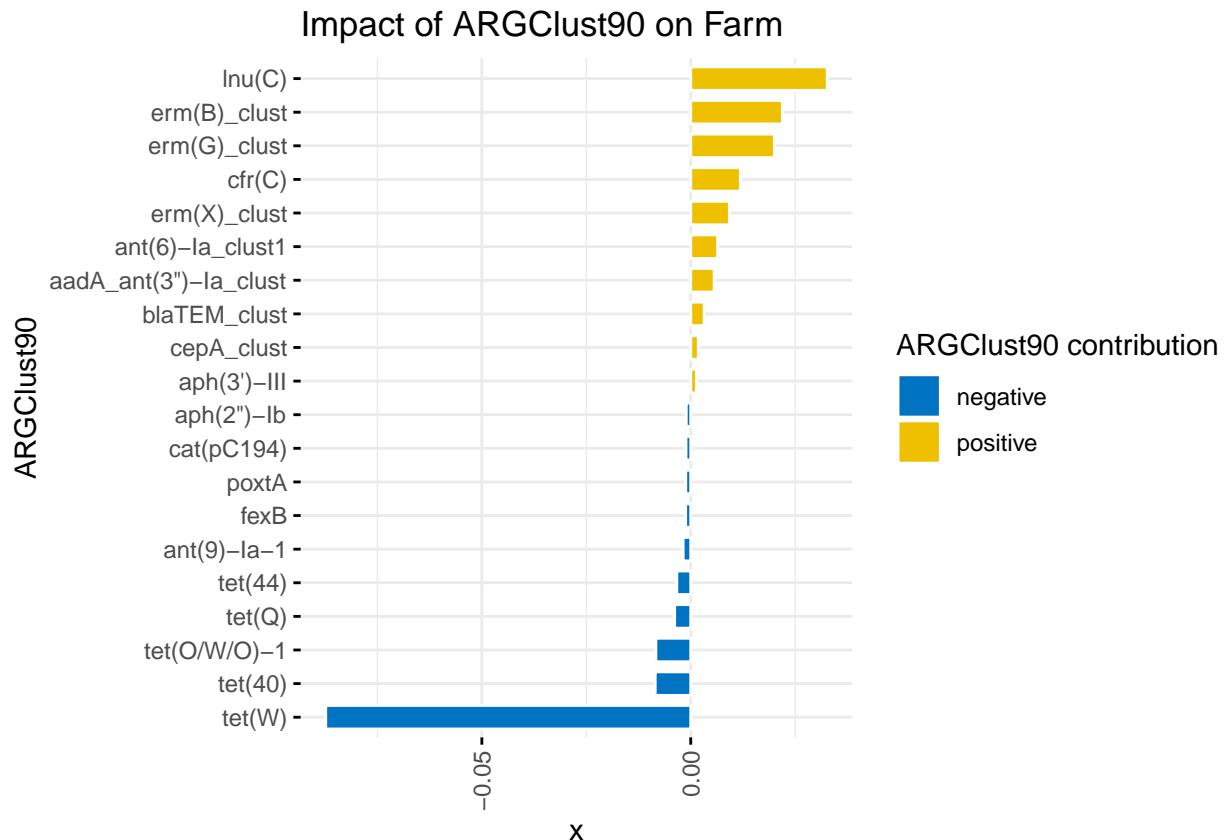
coef <- coefficients(permanova_farm)[["Farm21",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                             unique(names(top.coef)))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",                      # change fill color by mpg_level
           color = "white",                     # Set bar border colors to white
           palette = "jco",                   # jco journal color palett. see ?ggpar
           sort.val = "asc",                  # Sort the value in ascending order
           sort.by.groups = FALSE,            # Don't sort inside each group
           x.text.angle = 90,                # Rotate vertically x axis texts
           xlab = "ARGClust90",              # Label for the x-axis
           legend.title = "ARGClust90 contribution",
           title = "Impact of ARGClust90 on Farm",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



```

# Agent

permanova_agent <- adonis(t(assay(tse_clust90, "relabundance")) ~ Cox, data = colData(tse_clust90), perm = 999)

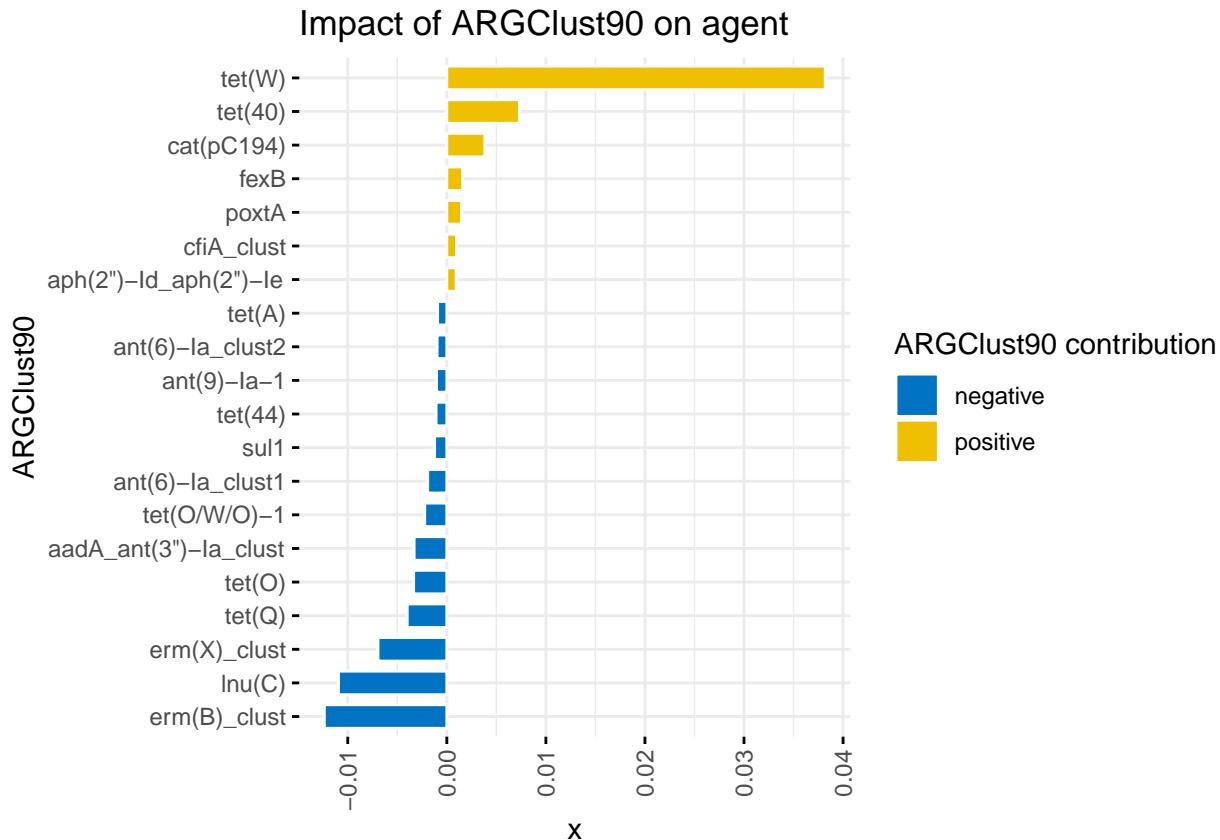
coef <- coefficients(permanova_agent)[["Cox1",]]
top.coef <- sort(head(coef[rev(order(abs(coef)))], 20))

df = data.frame(x = top.coef,
                 y = factor(names(top.coef),
                             unique(names(top.coef)))))

df$contr <- factor(ifelse(df$x < 0, "negative", "positive"),
                     levels = c("negative", "positive"))

ggbarplot(df, x = "y", y = "x",
           fill = "contr",                      # change fill color by mpg_level
           color = "white",                      # Set bar border colors to white
           palette = "jco",                      # jco journal color palett. see ?ggpar
           sort.val = "asc",                      # Sort the value in ascending order
           sort.by.groups = FALSE,                # Don't sort inside each group
           x.text.angle = 90,                     # Rotate vertically x axis texts
           xlab = "ARGClust90",
           legend.title = "ARGClust90 contribution",
           title = "Impact of ARGClust90 on agent",
           rotate = TRUE,
           ggtheme = theme_minimal())

```



checking homogeneity condition - bray ### ANOVAs are performed on betadispers of our rel abund data to test whether groups are more variable than others

```
# Bray
ps.rel = microbiome::transform(Rps, "compositional")
meta = meta(ps.rel)
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Age)) # homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      1 0.00012 0.0001157  0.0223 0.8815
## Residuals 118 0.61151 0.0051823
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$AB))
```

```
## Analysis of Variance Table
##
## Response: Distances
##           Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      1 0.02809 0.0280920  5.6363 0.01921 *
## Residuals 118 0.58813 0.0049841
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Farm2)) # homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      3 0.02740 0.0091332 1.4797 0.2237
## Residuals 116 0.71598 0.0061723
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Stables))
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      9 0.08492 0.0094354 1.9781 0.04848 *
## Residuals 110 0.52469 0.0047699
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Cox)) # homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      3 0.01886 0.0062883 1.2504 0.2948
## Residuals 116 0.58336 0.0050289
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Researcher))
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      4 0.09451 0.0236266 4.7862 0.001314 **
## Residuals 115 0.56768 0.0049364
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$LitterType)) # homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      2 0.02145 0.010727 1.6709 0.1925
## Residuals 117 0.75114 0.006420
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$Gender)) # homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      1 0.00714 0.0071380 1.3832 0.2419
## Residuals 118 0.60892 0.0051603
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$FlockSize)) # homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      5 0.03269 0.0065377 1.0304 0.4032
## Residuals 114 0.72332 0.0063449
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$AgeParentStock))
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value   Pr(>F)
## Groups      4 0.05866 0.0146644 3.0243 0.02058 *
## Residuals 115 0.55761 0.0048488
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$FeedProducent))
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value   Pr(>F)
## Groups      2 0.06454 0.032271 6.3342 0.002444 **
## Residuals 117 0.59608 0.005095
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(betadisper(vegdist(t(abundances(ps.rel))), meta$FeedType)) # homogeneous
```

```
## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      1 0.00012 0.0001157 0.0223 0.8815
## Residuals 118 0.61151 0.0051823
```

```

# Jaccard
anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Age) # homogeneous

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      1 0.00003 0.0000286  0.0048 0.9447
## Residuals 118 0.69816 0.0059166

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$AB))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value   Pr(>F)
## Groups      1 0.04040 0.040404   7.197 0.008351 **
## Residuals 118 0.66246 0.005614
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Farm2) # homogeneous

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      3 0.04535 0.0151151  1.8801 0.1368
## Residuals 116 0.93258 0.0080395

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Stables))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value   Pr(>F)
## Groups      9 0.11305 0.0125607  2.0602 0.03922 *
## Residuals 110 0.67066 0.0060969
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Researcher))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value   Pr(>F)
## Groups      4 0.15688 0.039219  6.7318 6.641e-05 ***
## Residuals 115 0.66999 0.005826
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

```

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$LitterType)) # homogeneous

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      2 0.03683 0.018416  2.1859 0.1169
## Residuals 117 0.98573 0.008425

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$Gender)) # homogeneous

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      1 0.00746 0.0074572  1.2715 0.2618
## Residuals 118 0.69206 0.0058649

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$FlockSize)) # homogeneous

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      5 0.05282 0.0105637  1.2705 0.2815
## Residuals 114 0.94785 0.0083145

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$AgeParentStock))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value  Pr(>F)
## Groups      4 0.08179 0.0204484  3.2496 0.01448 *
## Residuals 115 0.72365 0.0062926
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$FeedProducent))

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value    Pr(>F)
## Groups      2 0.10630 0.053151  8.5942 0.0003293 ***
## Residuals 117 0.72358 0.006184
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

anova(betadisper(vegdist(t(abundances(ps.rel))), method = "jaccard"), meta$FeedType) # homogeneous

## Analysis of Variance Table
##
## Response: Distances
##          Df  Sum Sq  Mean Sq F value Pr(>F)
## Groups      1 0.00003 0.0000286  0.0048 0.9447
## Residuals 118 0.69816 0.0059166

# group variances are not homogenous in most cases, so there are differences in variances between groups

# Tukey tests can be performed to see if and which groups differ in relation to variance

TukeyHSD(betadisper(vegdist(t(abundances(ps.rel)))), meta$Farm2))

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = distances ~ group, data = df)
## 
## $group
##            diff      lwr      upr     p adj
## Farm2-Farm1 0.035908649 -0.01528874 0.08710604 0.2654149
## Farm3-Farm1 0.004754782 -0.05436287 0.06387244 0.9967246
## Farm4-Farm1 0.016405164 -0.04271249 0.07552282 0.8875961
## Farm3-Farm2 -0.031153867 -0.08235126 0.02004353 0.3904962
## Farm4-Farm2 -0.019503485 -0.07070088 0.03169391 0.7536995
## Farm4-Farm3 0.011650382 -0.04746728 0.07076804 0.9556740

# different way of calculating homogeneity, permutation tests, null = no difference in dispersion between groups

permute(betadisper(vegdist(t(abundances(ps.rel)))), meta$Age), pairwise = TRUE)

## 
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##          Df  Sum Sq  Mean Sq      F N.Perm Pr(>F)
## Groups      1 0.00012 0.0001157 0.0223    999  0.888
## Residuals 118 0.61151 0.0051823
## 
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##      14     35
## 14 0.891
## 35 0.8815

permute(betadisper(unifrac.dist, metadf$Age), pairwise = TRUE) # looks like unifrac distances are homogeneous
```

##

```

## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##          Df   Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups      1 0.000595 0.00059526 0.5021    999  0.468
## Residuals 118 0.139893 0.00118553
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##       14     35
## 14     0.464
## 35     0.47997

```

```
permute(betadisper(unifrac.dist, metadf$AB), pairwise = TRUE) # not for AB though
```

```

##
## Permutation test for homogeneity of multivariate dispersions
## Permutation: free
## Number of permutations: 999
##
## Response: Distances
##          Df   Sum Sq   Mean Sq      F N.Perm Pr(>F)
## Groups      1 0.008665 0.0086653 7.9719    999  0.005 **
## Residuals 118 0.128263 0.0010870
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pairwise comparisons:
## (Observed p-value below diagonal, permuted p-value above diagonal)
##       no     yes
## no     0.005
## yes  0.0055783

```

Simper analyses to see which species are most impactful to BC dissimilarity between groups

Test for significance (OTU abundance will not be normally distributed so we will use kruskal wallis tests)

```

# Load data for Rmarkdown
#Age
#simper.pretty(otu_table(Rps), metrics = sample_data(Rps), interesting = c("Age"), perc_cutoff=1, low_c
#Rps_age = data.frame(read.csv("Rps_age_clean_simper.csv"))

#kruskal.pretty(otu_table(Rps), metrics = sample_data(Rps), csv = Rps_age, interesting = c('Age'), outp
KW_Rps_age = data.frame(read.csv("Rps_Age_krusk_simper.csv"))
KW_Rps_age = KW_Rps_age[KW_Rps_age$fdr_krusk_p.val < 0.05,] # filter out non-significant results, based
KW_Rps_age = KW_Rps_age[with(KW_Rps_age, order(SIMPER, decreasing = TRUE)),]

```

```

KW_Rps_age$OTU = as.factor(KW_Rps_age$OTU)

KW_Rps_age %>% mutate_if(is.numeric, format.pval, 2) %>% dplyr::select("SIMPER", "OTU", "fdr_krusk_p.val")
  rowwise() %>% mutate(Combined = paste("ARG =", OTU, ", SIMPER =", SIMPER, ", p-value =", fdr_krusk_p.val))
  dplyr::select(Combined)

## # A tibble: 2 x 1
## # Rowwise:
##   Combined
##   <chr>
## 1 ARG = aph(3)-IIIa_1_AF330699 , SIMPER = 0.070 , p-value = 0.025
## 2 ARG = aph(3)-III_1_M26832 , SIMPER = 0.025 , p-value = 0.016

#AB
#simper.pretty(otu_table(Rps), metrics = sample_data(Rps), interesting = c("AB"), perc_cutoff=1, low_cu
#Rps_AB = data.frame(read.csv("Rps_AB_clean_simper.csv"))

#kruskal.pretty(otu_table(Rps), metrics = sample_data(Rps), csv = Rps_AB, interesting = c('AB'), output
#Kruskal's test for each comparison

KW_Rps_AB = data.frame(read.csv("Rps_AB_krusk_simper.csv"))
KW_Rps_AB = KW_Rps_AB[KW_Rps_AB$fdr_krusk_p.val < 0.05,] # filter out non-significant results, based on
KW_Rps_AB = KW_Rps_AB[with(KW_Rps_AB, order(SIMPER, decreasing = TRUE)),]
KW_Rps_AB$OTU = as.factor(KW_Rps_AB$OTU)

KW_Rps_AB %>% mutate_if(is.numeric, format.pval, 2) %>% dplyr::select("SIMPER", "OTU", "fdr_krusk_p.val")
  rowwise() %>% mutate(Combined = paste("ARG =", OTU, ", SIMPER =", SIMPER, ", p-value =", fdr_krusk_p.val))
  dplyr::select(Combined)

## # A tibble: 2 x 1
## # Rowwise:
##   Combined
##   <chr>
## 1 ARG = tet(44)_2_FN594949 , SIMPER = 0.020 , p-value = 0.0391
## 2 ARG = tet(Q)_1_L33696 , SIMPER = 0.013 , p-value = 0.0028

#Farms - too many comparisons so maybe too extensive for report

#simper.pretty(otu_table(Rps), metrics = sample_data(Rps), interesting = c("Farm2"), perc_cutoff=1, low_
#Rps_Farm = data.frame(read.csv("Rps_Farm_clean_simper.csv"))

#kruskal.pretty(otu_table(Rps), metrics = sample_data(Rps), csv = Rps_Farm, interesting = c('Farm2'), output
#Kruskal's test for each comparison

KW_Rps_Farm = data.frame(read.csv("Rps_Farm_krusk_simper.csv"))
KW_Rps_Farm = KW_Rps_Farm[KW_Rps_Farm$fdr_krusk_p.val < 0.05,] # filter out non-significant results, based on
KW_Rps_Farm = KW_Rps_Farm[with(KW_Rps_Farm, order(SIMPER, decreasing = TRUE)),]
KW_Rps_Farm$OTU = as.factor(KW_Rps_Farm$OTU)

KW_Rps_Farm %>% mutate_if(is.numeric, format.pval, 2) %>% dplyr::select("Comparison", "SIMPER", "OTU", "fdr_krusk_p.val")
  rowwise() %>% mutate(Combined = paste(Comparison, "ARG =", OTU, ", SIMPER =", SIMPER, ", p-value =", fdr_krusk_p.val))
  dplyr::select(Combined)

```

```

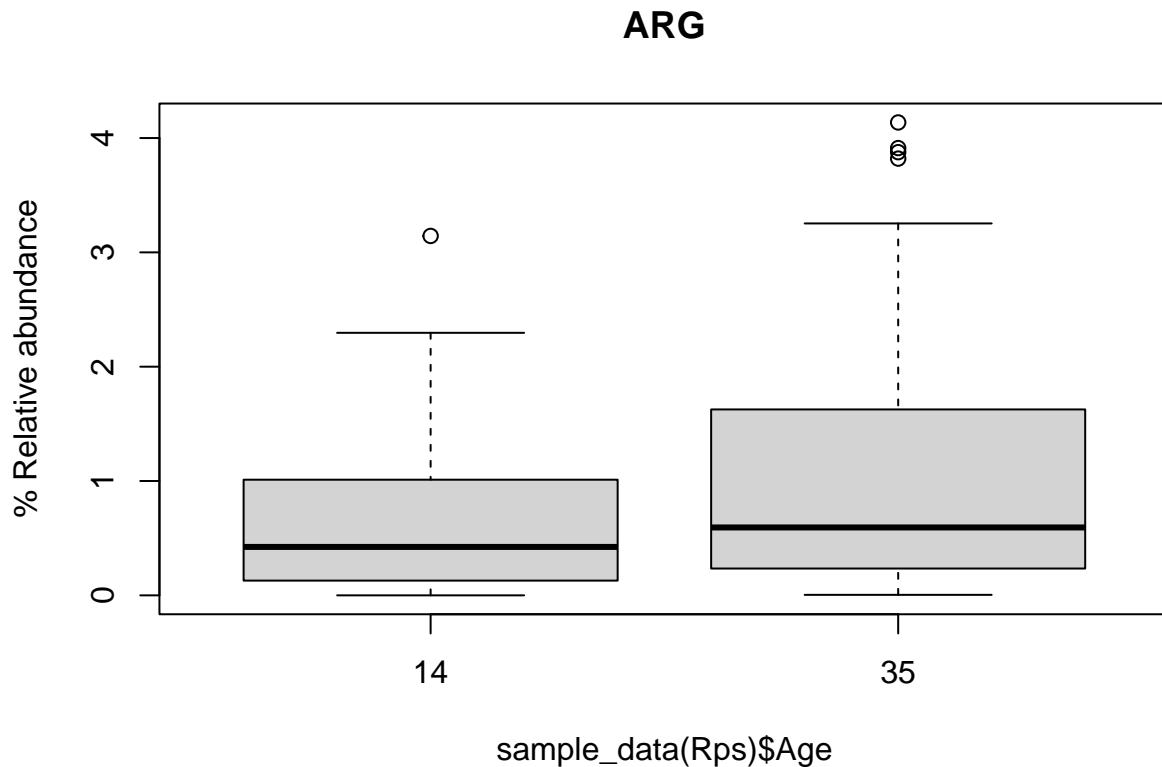
## # A tibble: 67 x 1
## # Rowwise:
##   Combined
##   <chr>
## 1 Farm3_Farm4 ARG = tet(W)_4_FN396364 , SIMPER = 0.120 , p-value = 0.00088
## 2 Farm3_Farm4 ARG = tet(W)_1_DQ060146 , SIMPER = 0.109 , p-value = 4.3e-05
## 3 Farm3_Farm1 ARG = tet(W)_4_FN396364 , SIMPER = 0.106 , p-value = 0.04519
## 4 Farm2_Farm1 ARG = lnu(C)_1_AY928180 , SIMPER = 0.102 , p-value = 8.7e-08
## 5 Farm3_Farm2 ARG = tet(W)_1_DQ060146 , SIMPER = 0.100 , p-value = 0.02920
## 6 Farm3_Farm1 ARG = tet(W)_1_DQ060146 , SIMPER = 0.094 , p-value = 0.00368
## 7 Farm3_Farm2 ARG = ant(6)-Ia_3_KF864551 , SIMPER = 0.094 , p-value = 0.01371
## 8 Farm4_Farm1 ARG = lnu(C)_1_AY928180 , SIMPER = 0.085 , p-value = 1.7e-05
## 9 Farm4_Farm2 ARG = tet(W)_1_DQ060146 , SIMPER = 0.084 , p-value = 0.02971
## 10 Farm2_Farm1 ARG = ant(6)-Ia_3_KF864551 , SIMPER = 0.080 , p-value = 0.00045
## # i 57 more rows

```

```

# plots to look at specific ARGs (age)
abund = otu_table(Rps)/rowSums(otu_table(Rps))*100
boxplot(unlist(data.frame(abund["tet(44)_2_FN594949"]))) ~ sample_data(Rps)$Age, ylab="% Relative abundance"

```



```

# specific test
kruskal.test(unlist(data.frame(otu_table(Rps)["tet(0/32/0)_5_FP929050"])), use.names = FALSE) ~ sample_data(Rps)$Age

##
## Kruskal-Wallis rank sum test

```

```

##  

## data: unlist(data.frame(otu_table(Rps) ["tet(0/32/0)_5_FP929050"])), use.names = FALSE) by sample_data  

## Kruskal-Wallis chi-squared = 5.1174, df = 1, p-value = 0.02369

# Trying out different distances, aggregation methods and indices for finding optimal number of clusters

tse = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)

tse <- transformCounts(tse, method = "relabundance")

assay <- t(assay(tse, "relabundance"))

diss_jaccard <- vegdist(assay, method = "jaccard")

# different aggregation methods and indices will grant different amount of clusters
NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "mcclain")$Best.nc # two clusters

##  

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be in matrix form

## Number_clusters      Value_Index
##          2.0000        0.0091

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "frey")$Best.nc # 3 clusters

##  

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be in matrix form

## Number_clusters      Value_Index
##          3.0000        2.1883

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "cindex")$Best.nc # 14 clusters

##  

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be in matrix form

## Number_clusters      Value_Index
##          14.0000       0.2002

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "silhouette")$Best.nc # two clusters

##  

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be in matrix form

## Number_clusters      Value_Index
##          2.0000        0.4678

NbClust(diss = diss_jaccard, distance = NULL, method = "complete", index = "dunn")$Best.nc # 2 clusters

```

```

##  

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m  

## Number_clusters      Value_Index  

##                 2.00          0.95  

NbClust(diss = diss_jaccard, distance = NULL, method = "ward.D2", index = "silhouette")$Best.nc # 2 clu  

##  

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m  

## Number_clusters      Value_Index  

##                 2.0000        0.2287  

NbClust(diss = diss_jaccard, distance = NULL, method = "ward.D", index = "silhouette")$Best.nc # 2 clu  

##  

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m  

## Number_clusters      Value_Index  

##                 2.000          0.246  

NbClust(diss = diss_jaccard, distance = NULL, method = "single", index = "silhouette")$Best.nc # 2 clu  

##  

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m  

## Number_clusters      Value_Index  

##                 2.0000        0.4678  

NbClust(diss = diss_jaccard, distance = NULL, method = "average", index = "silhouette")$Best.nc # 2 clu  

##  

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m  

## Number_clusters      Value_Index  

##                 2.0000        0.4678  

NbClust(diss = diss_jaccard, distance = NULL, method = "mcquitty", index = "silhouette")$Best.nc # 2 clu  

##  

## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m  

## Number_clusters      Value_Index  

##                 2.0000        0.4678

```

```

NbClust(diss = diss_jaccard, distance = NULL, method = "median", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##          2.0000        0.4678

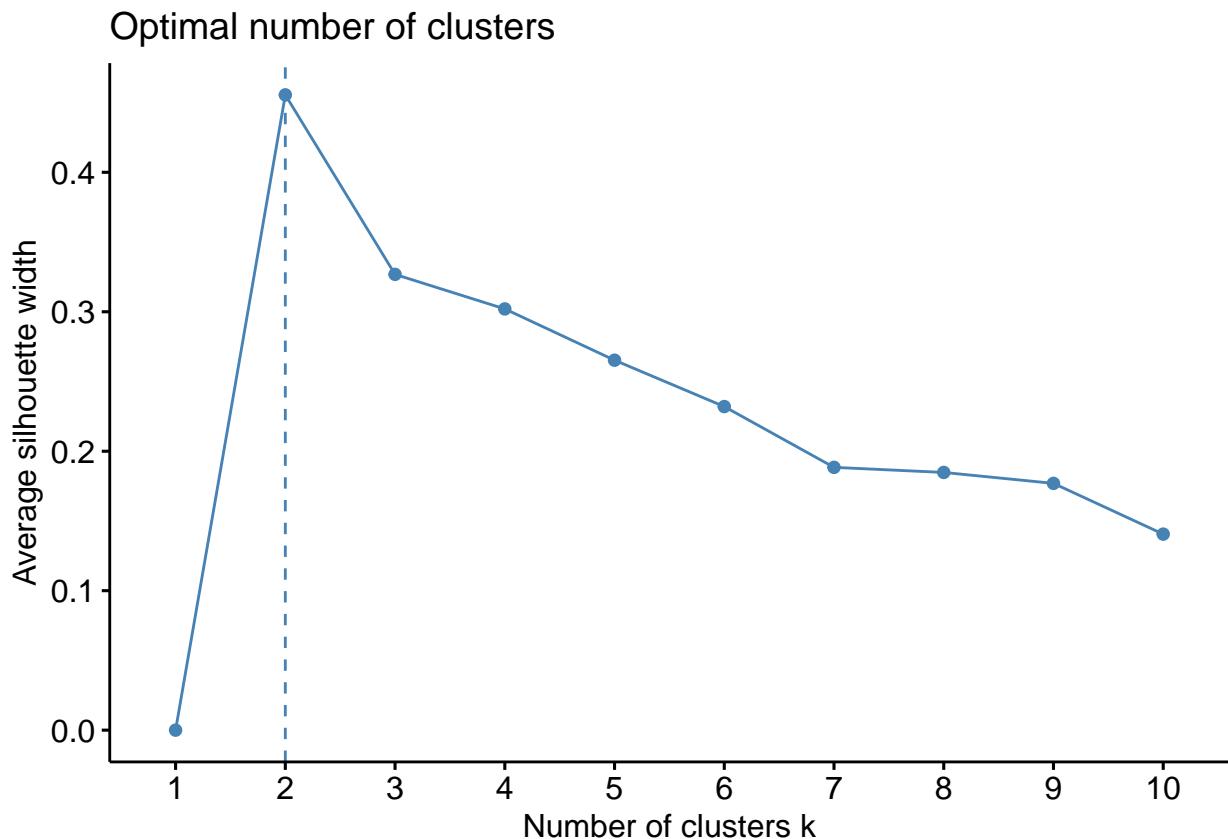
NbClust(diss = diss_jaccard, distance = NULL, method = "centroid", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

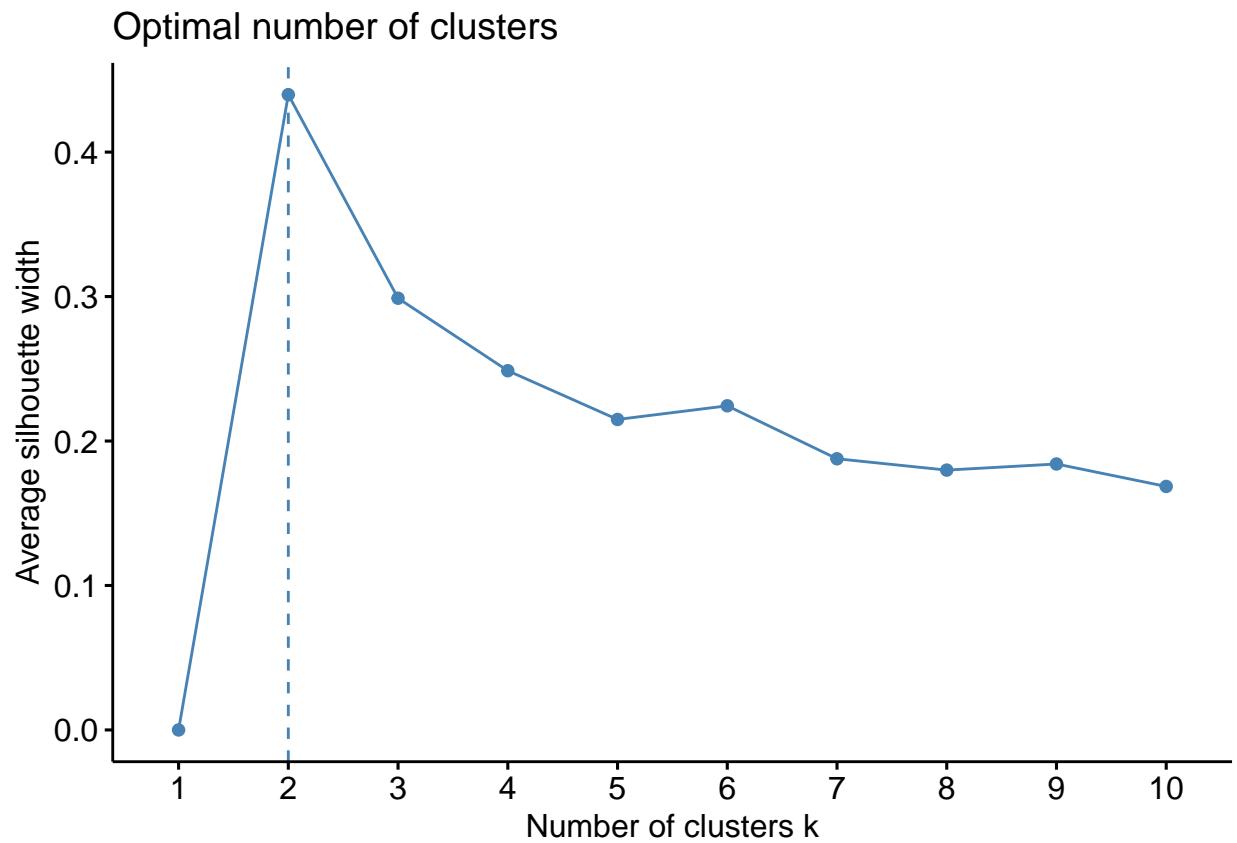
## Number_clusters      Value_Index
##          2.0000        0.4678

# silhouette (ASW), different clustering methods
diss_jaccard <- as.matrix(diss_jaccard)
fviz_nbclust(diss_jaccard, kmeans, method = "silhouette") # 2 seems optimal for k-means

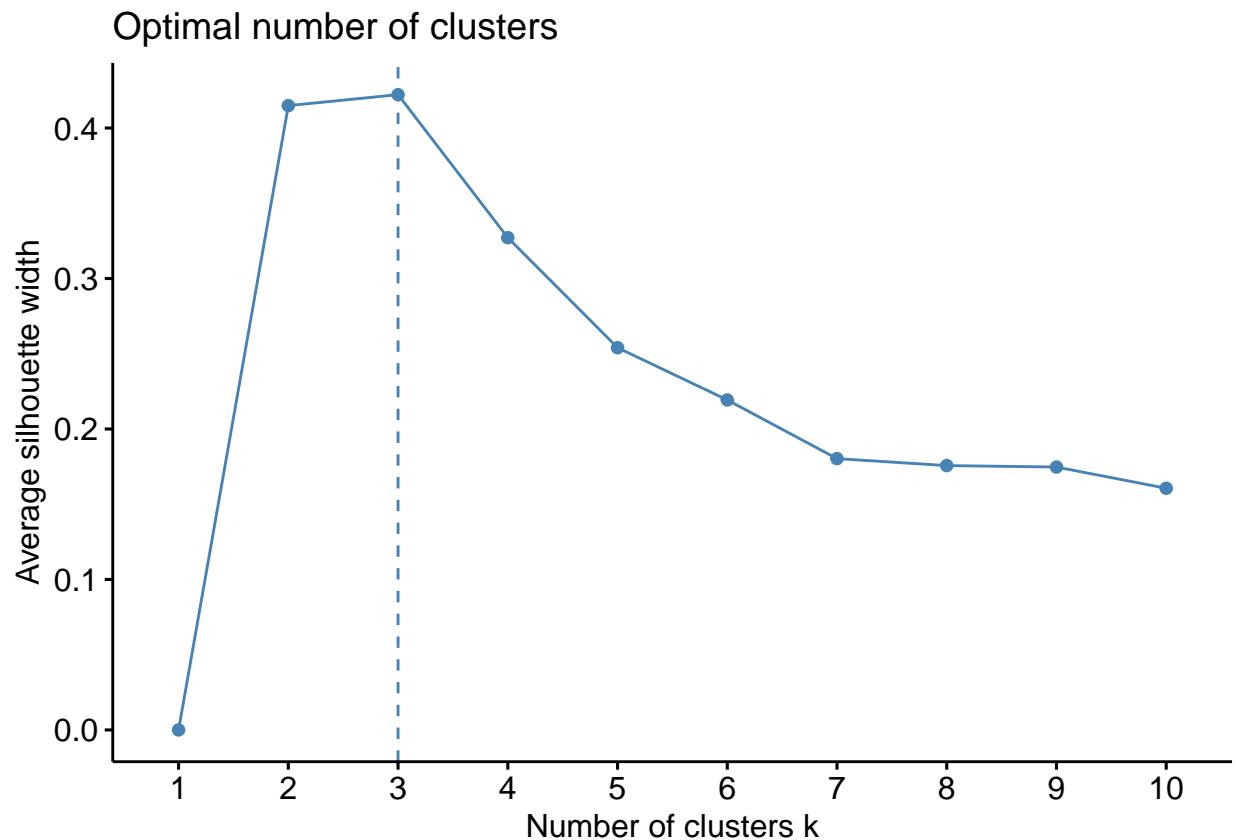
```



```
fviz_nbclust(diss_jaccard, cluster::pam, method = "silhouette") # 2 seems optimal for PAM
```

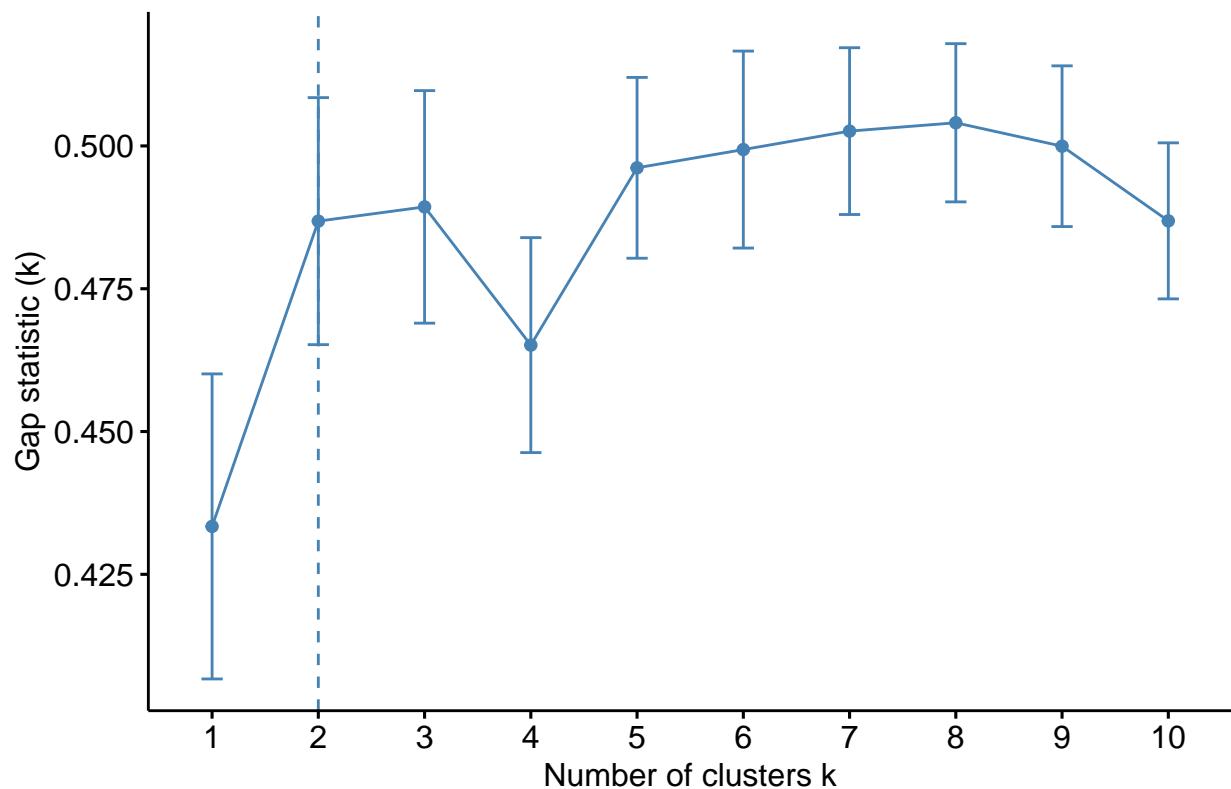


```
fviz_nbclust(diss_jaccard, hc, method = "silhouette") # 3 seems optimal for hc
```



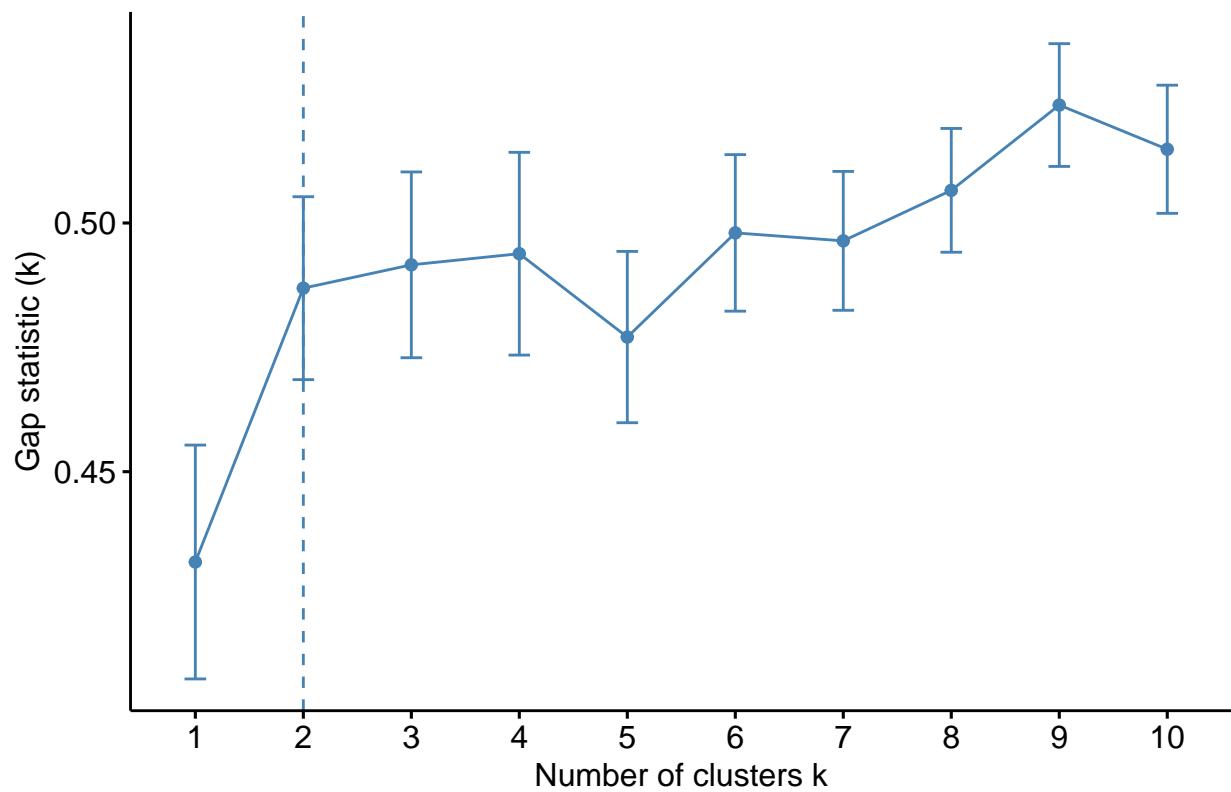
```
fviz_nbclust(diss_jaccard, kmeans, method = "gap_stat") # 2 seems optimal for k-means gap stat
```

Optimal number of clusters



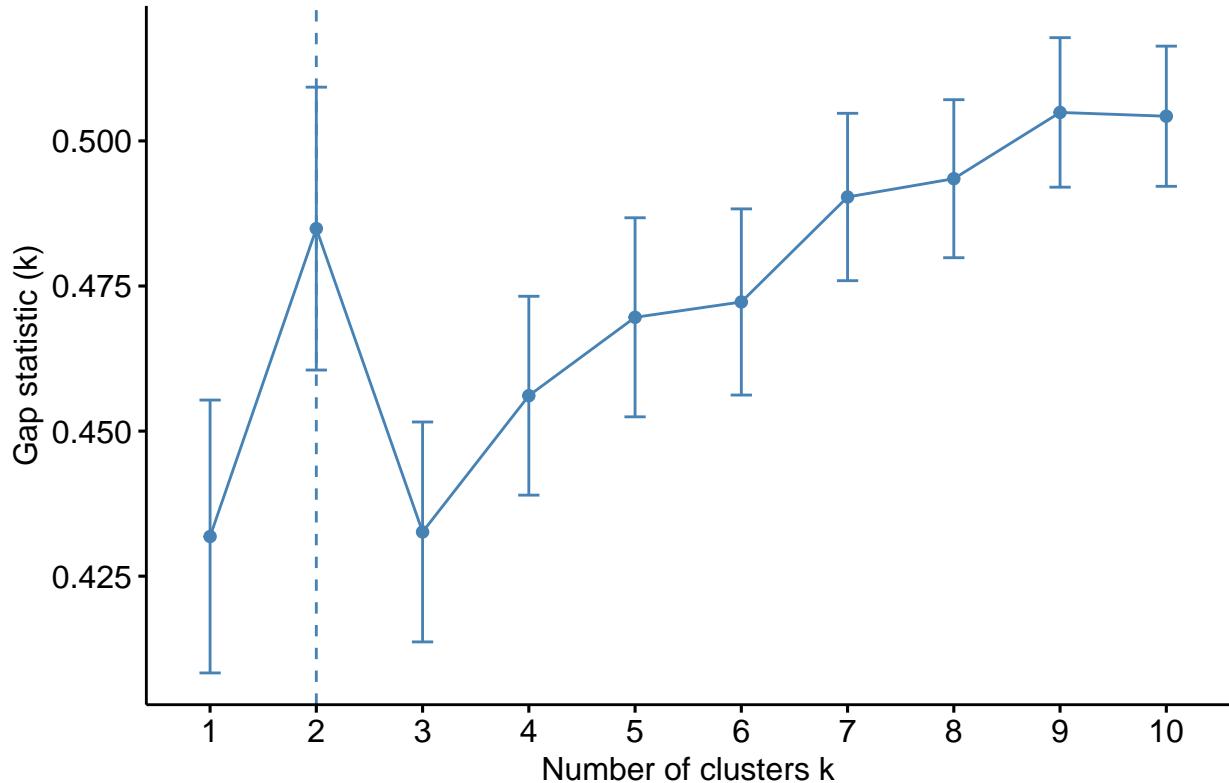
```
fviz_nbclust(diss_jaccard, cluster::pam, method = "gap_stat") # 2 seems optimal for PAM gap stat
```

Optimal number of clusters



```
fviz_nbclust(diss_jaccard, hc, method = "gap_stat") # 2 seems optimal for hc cut gap stat
```

Optimal number of clusters



```
# now, let's repeat this for BC

diss_bray <- vegdist(assay, method = "bray")

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "mcclain")$Best.nc # two clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##           2.000          0.007

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "frey")$Best.nc # 3 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m

## Number_clusters      Value_Index
##           3.0000         2.5708

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "cindex")$Best.nc # 14 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data m
```

```

## Number_clusters      Value_Index
##                 14.0000      0.1205

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "silhouette")$Best.nc # two clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.5904

NbClust(diss = diss_bray, distance = NULL, method = "complete", index = "dunn")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.9208

NbClust(diss = diss_bray, distance = NULL, method = "ward.D2", index = "silhouette")$Best.nc # 3 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 3.0000      0.3222

NbClust(diss = diss_bray, distance = NULL, method = "ward.D", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.2986

NbClust(diss = diss_bray, distance = NULL, method = "single", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.5904

NbClust(diss = diss_bray, distance = NULL, method = "average", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##                 2.0000      0.5904

```

```

NbClust(diss = diss_bray, distance = NULL, method = "mcquitty", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##              2.0000        0.5904

NbClust(diss = diss_bray, distance = NULL, method = "median", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##              2.0000        0.5904

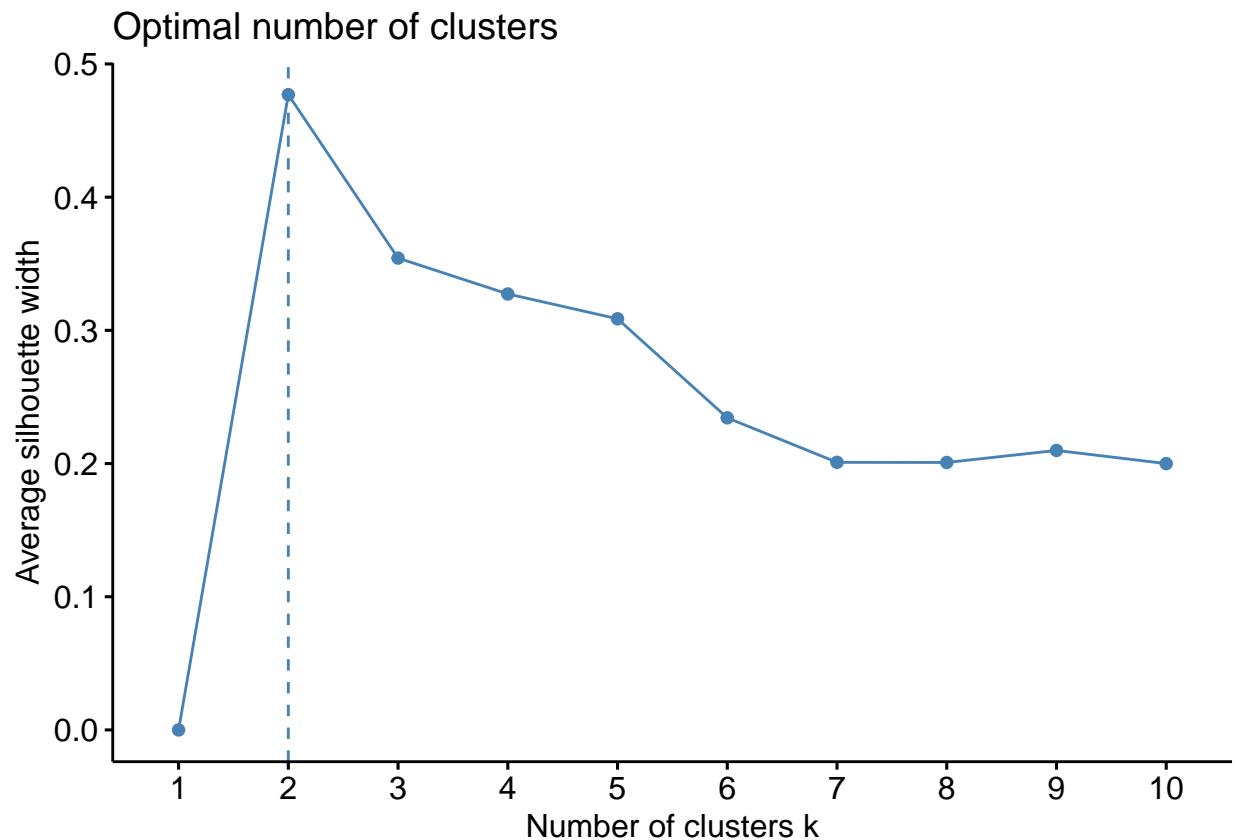
NbClust(diss = diss_bray, distance = NULL, method = "centroid", index = "silhouette")$Best.nc # 2 clusters

##
## Only frey, mcclain, cindex, sihouette and dunn can be computed. To compute the other indices, data must be numerical

## Number_clusters      Value_Index
##              2.0000        0.5904

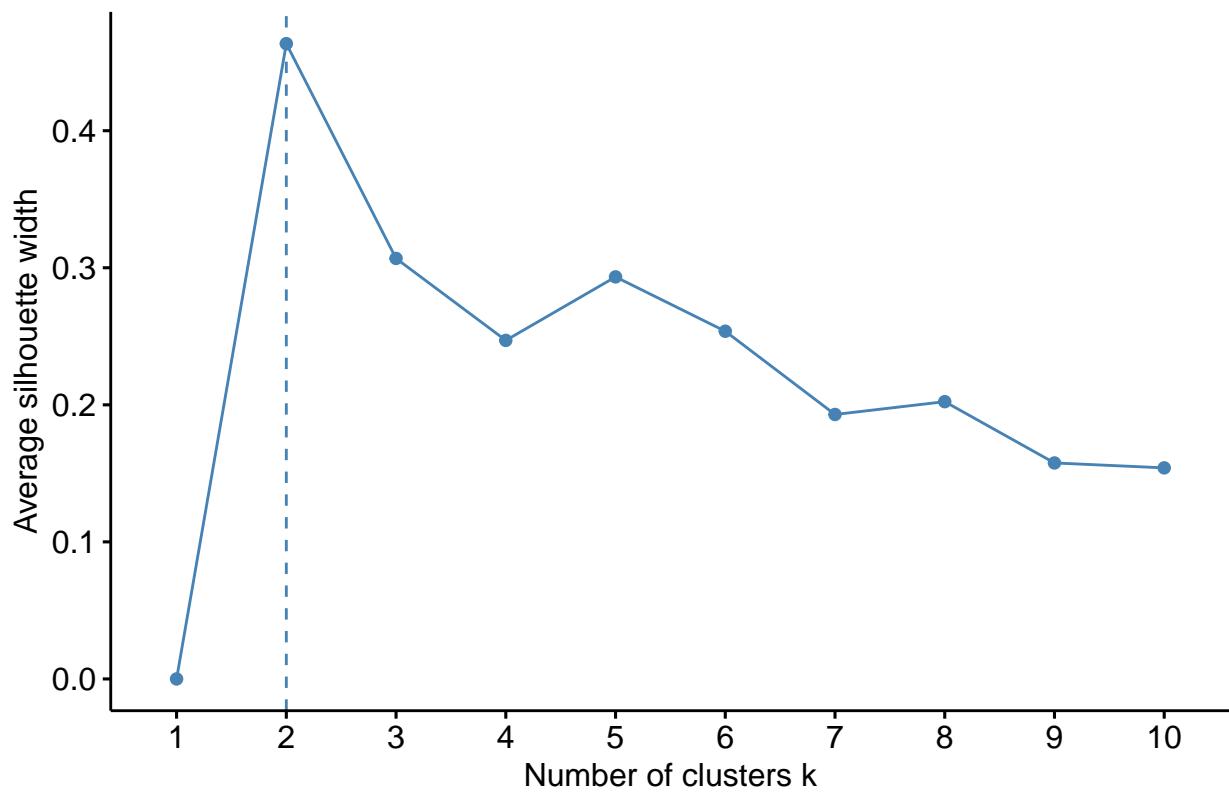
# silhouette (ASW)
diss_bray <- as.matrix(diss_bray)
fviz_nbclust(diss_bray, kmeans, method = "silhouette") # 2 seems optimal

```



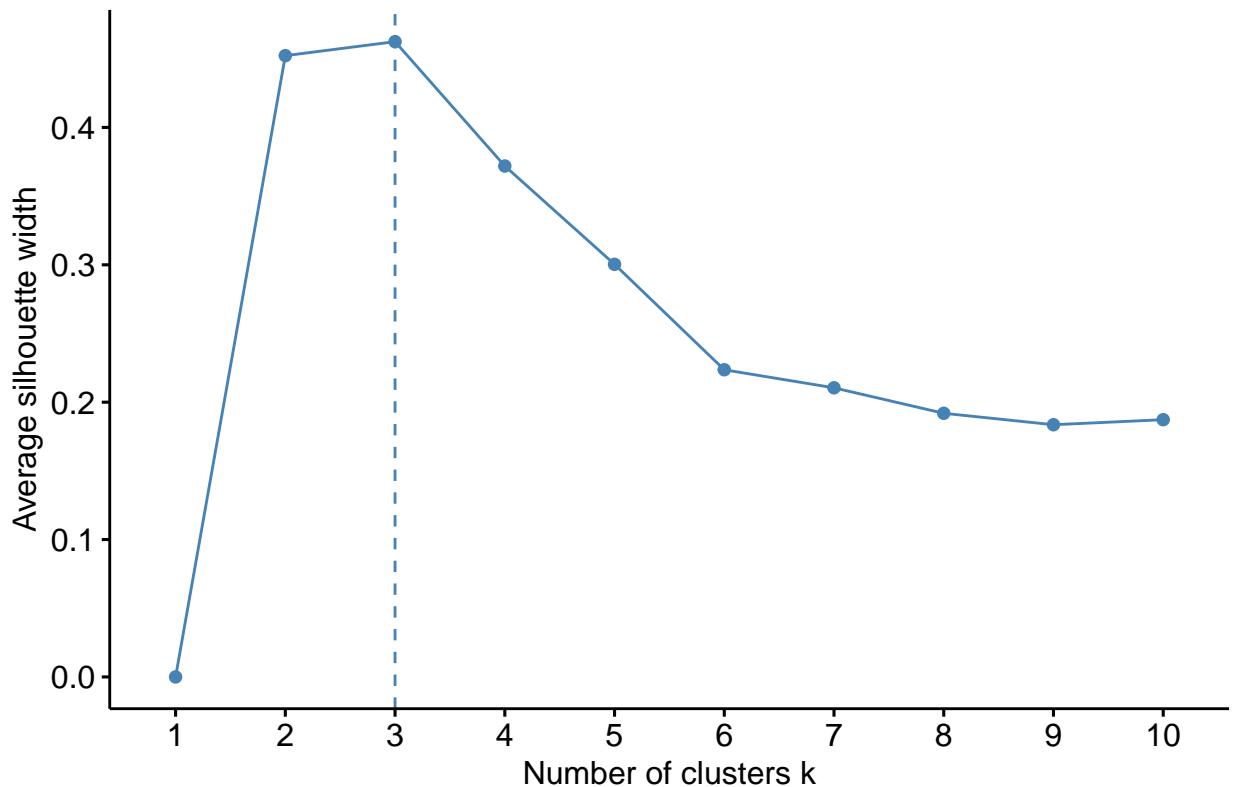
```
fviz_nbclust(diss_bray, cluster::pam, method = "silhouette") # 2 seems optimal for PAM
```

Optimal number of clusters

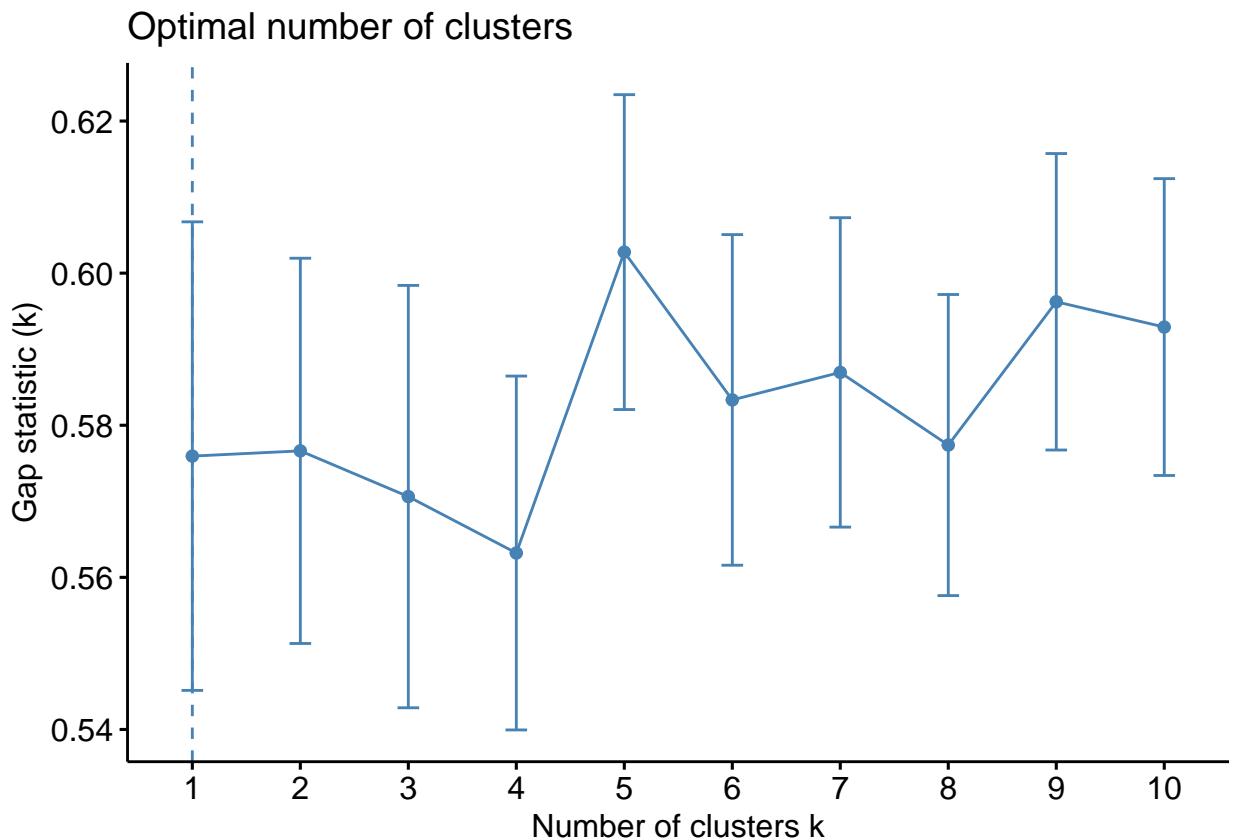


```
fviz_nbclust(diss_bray, hcut, method = "silhouette") # 3 seems optimal for hcut
```

Optimal number of clusters

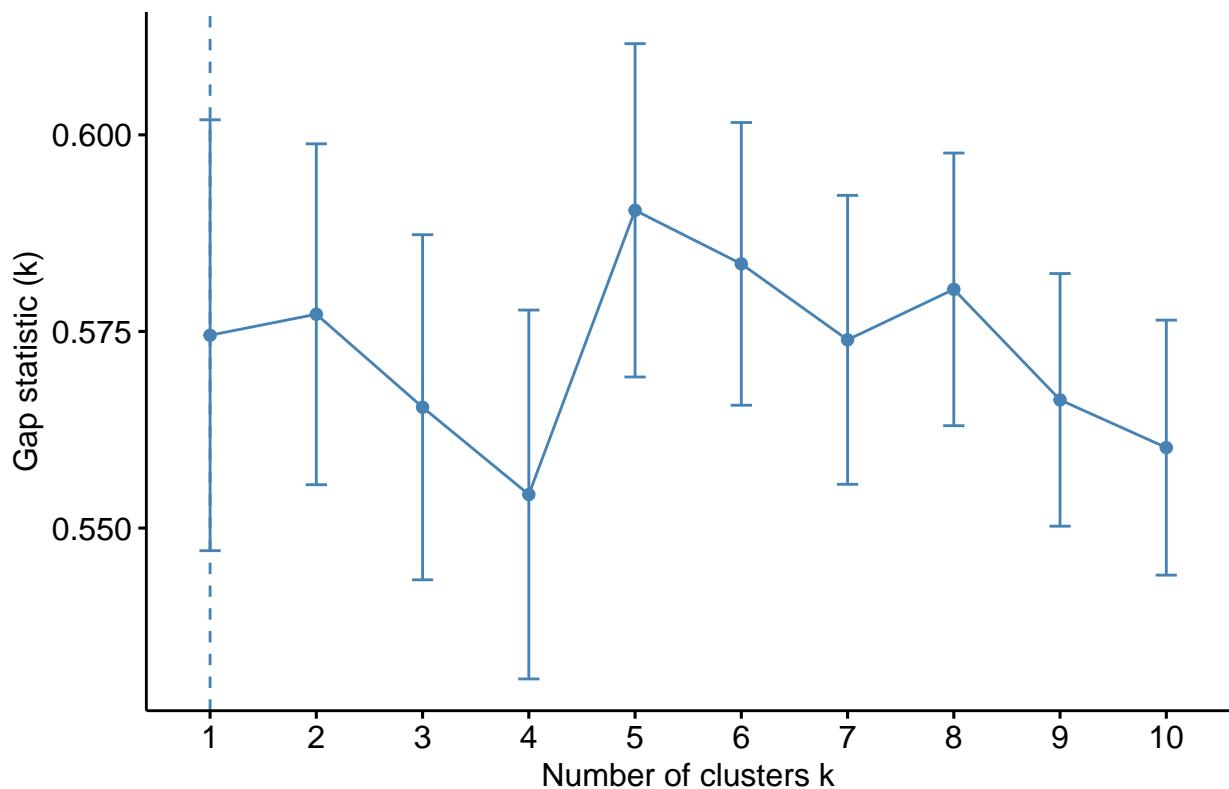


```
fviz_nbclust(diss_bray, kmeans, method = "gap_stat") # 1 seems optimal for k-means gap stat
```

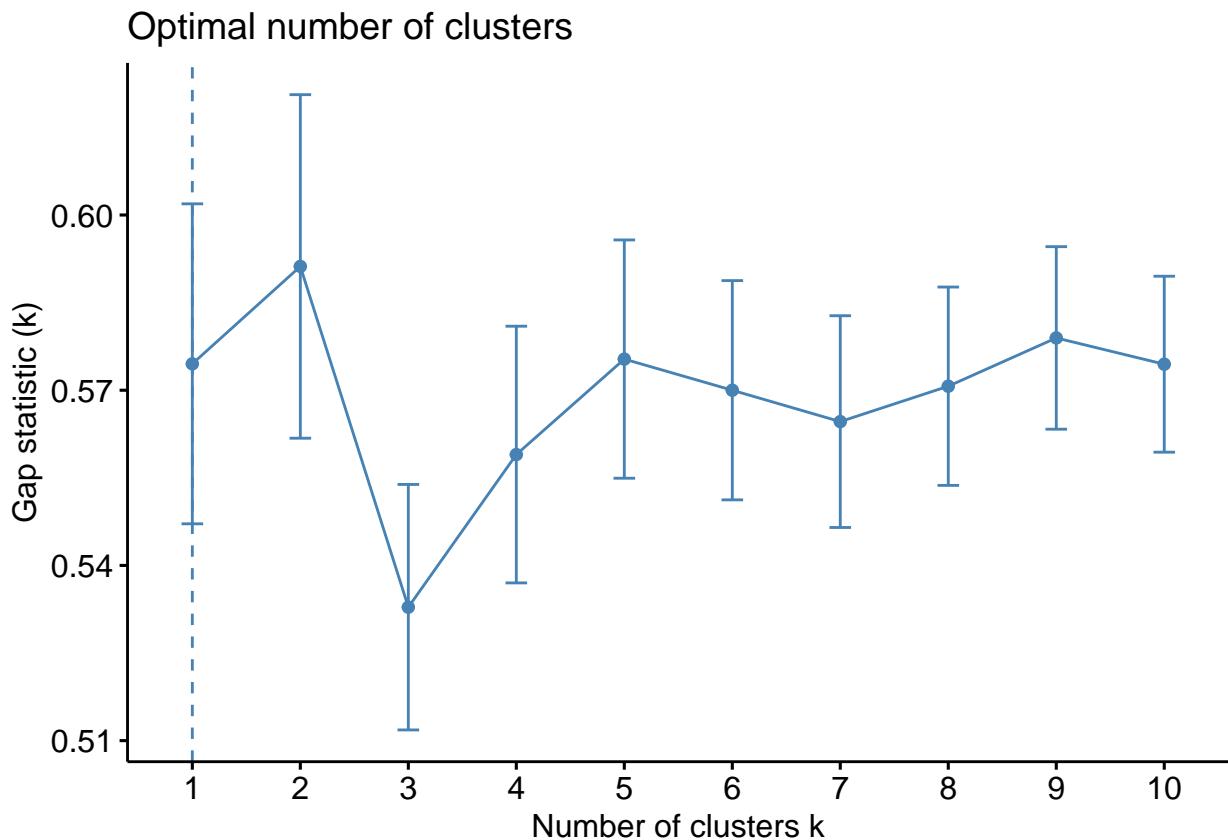


```
fviz_nbclust(diss_bray, cluster::pam, method = "gap_stat") # 1 seems optimal for PAM gap stat
```

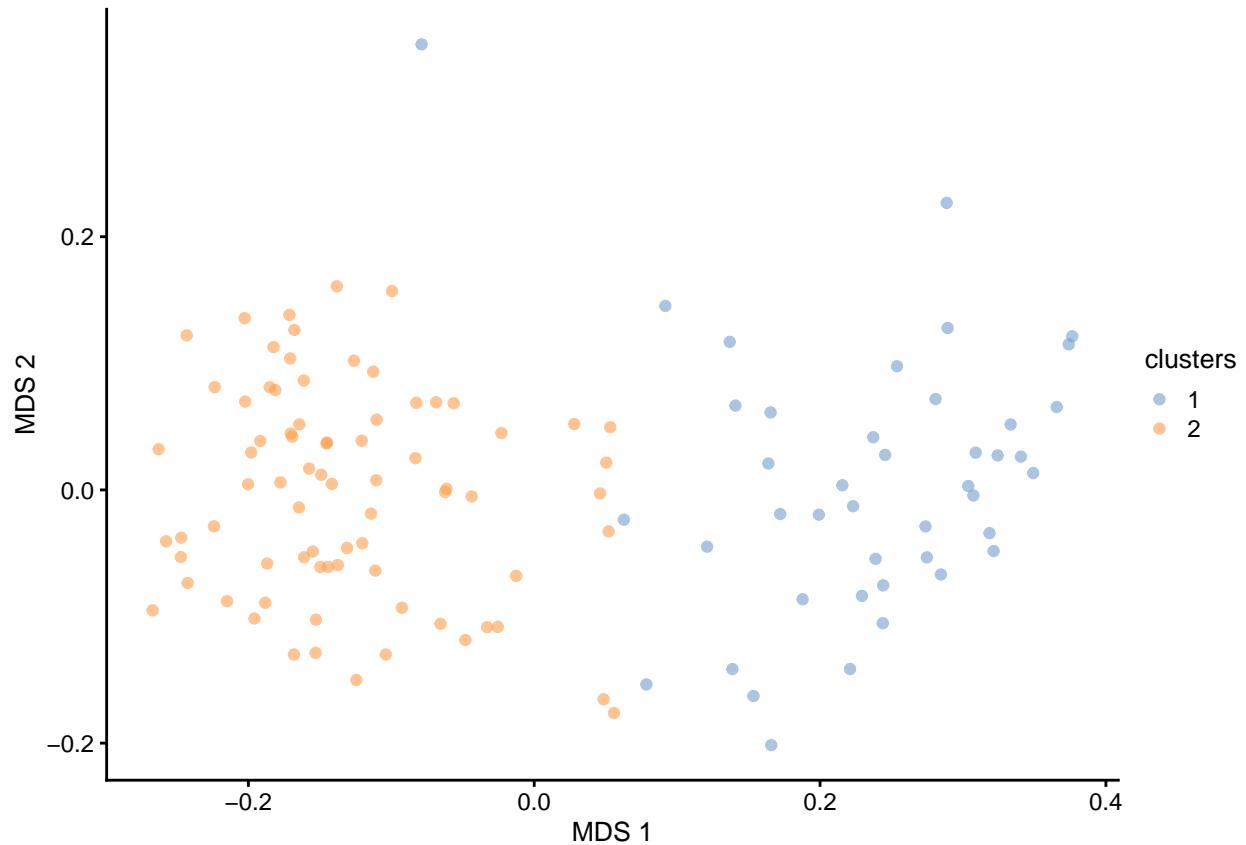
Optimal number of clusters



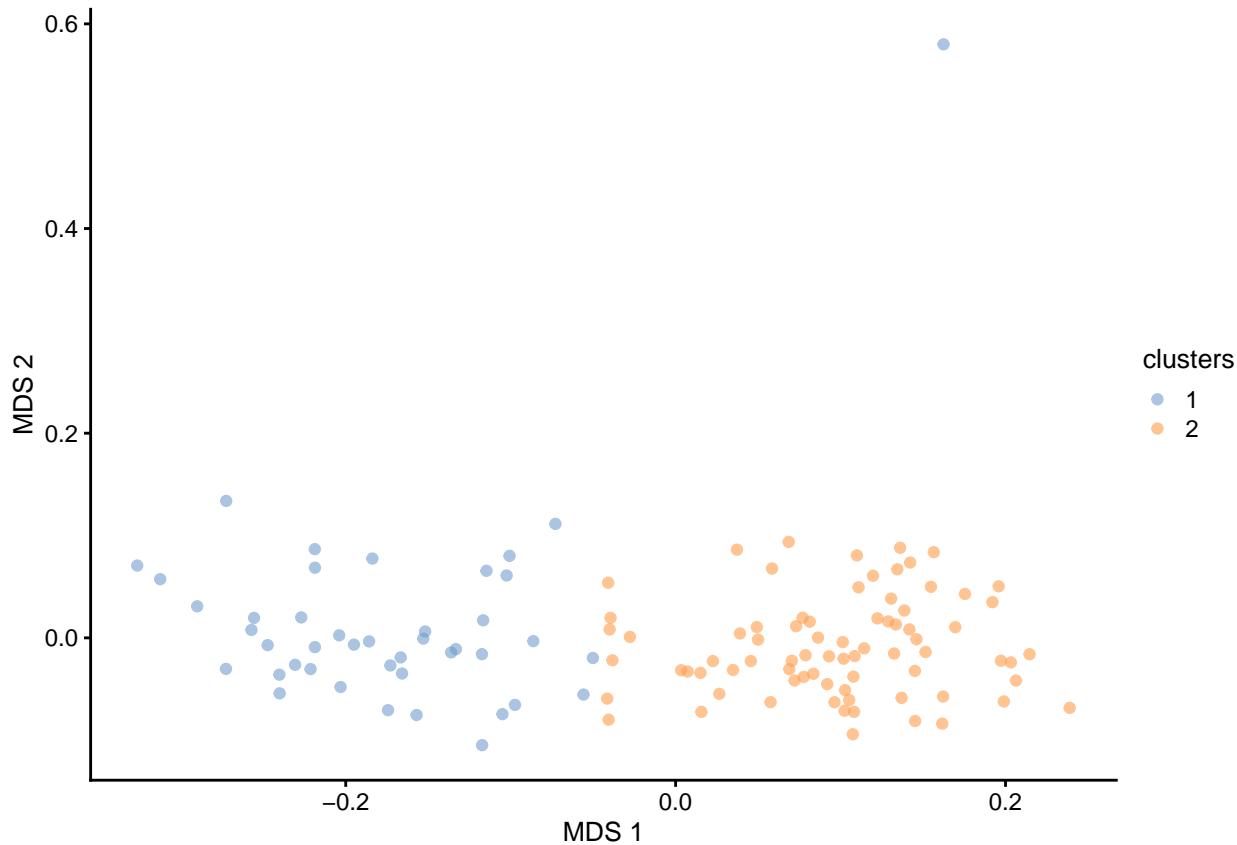
```
fviz_nbclust(diss_bray, hc, method = "gap_stat") # 1 seems optimal for hc cut gap stat
```



```
# k-means jaccard clusters
set.seed(1337)
km <- kmeans(diss_jaccard, 2, nstart = 25)
colData(tse)$clusters <- as.factor(km$cluster)
tse <- runMDS(tse, assay.type = "relabundance", FUN = vegan::vegdist, method = "jaccard")
plotReducedDim(tse, "MDS", colour_by = "clusters")
```



```
# k-means bray clusters
set.seed(1337)
km <- kmeans(diss_bray, 2, nstart = 25)
colData(tse)$clusters <- as.factor(km$cluster)
tse <- runMDS(tse, assay.type = "relabundance", FUN = vegan::vegdist, method = "bray")
plotReducedDim(tse, "MDS", colour_by = "clusters")
```



DMM (Laplace approximation) - ARG level

```
tse = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)
tse_dmn <- mia::runDMN(tse, name = "DMN", k = 1:7) # calculate most likely number of clusters from 1 to 7
tse_dmn
```

```
## class: TreeSummarizedExperiment
## dim: 662 120
## metadata(1): DMN
## assays(1): counts
## rownames(662): blaTEM-210_1_KJ484630 blaSHV-183_1_HG934764 ...
##   dfrA1_7_AJ400733 qnrS4_1_FJ418153
## rowData names(4): Phylum Class Order Family
## colnames(120): 10_1 10_10 ... 9_38 9_39
## colData names(33): SampleIdentifier ResCap ... ReadTot Stables
## reducedDimNames(0):
## mainExpName: NULL
## altExpNames(0):
## rowLinks: a LinkDataFrame (662 rows)
## rowTree: 1 phylo tree(s) (662 leaves)
## colLinks: NULL
## colTree: NULL
```

```
getDMN(tse_dmn)
```

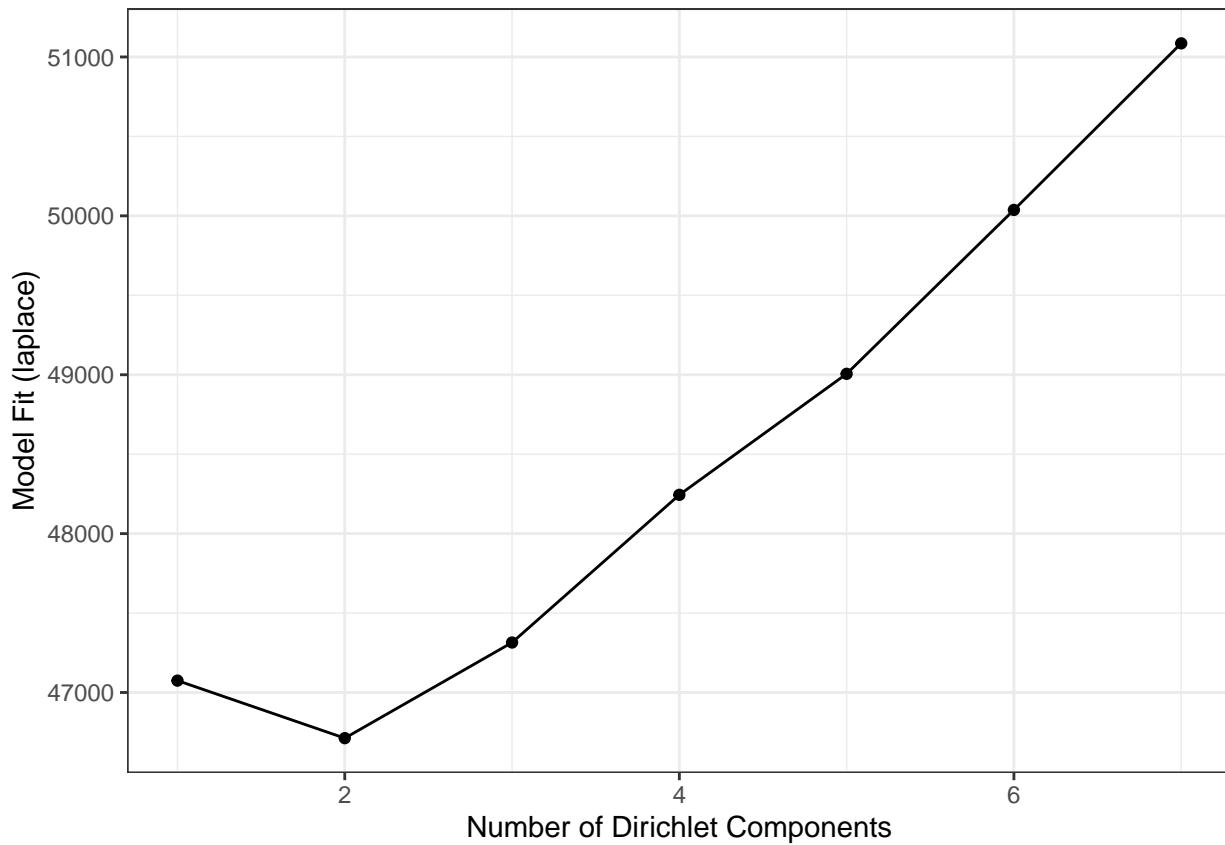
```
## [[1]]
```

```

## class: DMN
## k: 1
## samples x taxa: 120 x 662
## Laplace: 47074.83 BIC: 49311.81 AIC: 48389.15
##
## [[2]]
## class: DMN
## k: 2
## samples x taxa: 120 x 662
## Laplace: 46712.67 BIC: 51421.46 AIC: 49574.75
##
## [[3]]
## class: DMN
## k: 3
## samples x taxa: 120 x 662
## Laplace: 47314.57 BIC: 54431.83 AIC: 51661.06
##
## [[4]]
## class: DMN
## k: 4
## samples x taxa: 120 x 662
## Laplace: 48244.04 BIC: 57672.15 AIC: 53977.33
##
## [[5]]
## class: DMN
## k: 5
## samples x taxa: 120 x 662
## Laplace: 49005.73 BIC: 60761.76 AIC: 56142.89
##
## [[6]]
## class: DMN
## k: 6
## samples x taxa: 120 x 662
## Laplace: 50037.22 BIC: 64021.86 AIC: 58478.94
##
## [[7]]
## class: DMN
## k: 7
## samples x taxa: 120 x 662
## Laplace: 51085.62 BIC: 67258.59 AIC: 60791.61

miaViz::plotDMNFit(tse_dmn, type = "laplace")

```



```

getBestDMNFit(tse_dmn, type = "laplace") # 2 again

## class: DMN
## k: 2
## samples x taxa: 120 x 662
## Laplace: 46712.67 BIC: 51421.46 AIC: 49574.75

# genus level

tse = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)
tse_genus <- agglomerateByRank(tse, rank = "Class", agglomerateTree = TRUE)
tse_dmn <- mia::runDMN(tse_genus, name = "DMN", k = 1:7) # calculate most likely number of clusters from tse_dmn

## class: TreeSummarizedExperiment
## dim: 19 120
## metadata(2): agglomerated_by_rank DMN
## assays(1): counts
## rownames(19): Phylum:Beta-lactam Phylum:Aminoglycoside ...
##   Class:Aminoglycoside Phylum:Fosfomycin
## rowData names(4): Phylum Class Order Family
## colnames(120): 10_1 10_10 ... 9_38 9_39
## colData names(33): SampleIdentifier ResCap ... ReadTot Stables
## reducedDimNames(0):
## mainExpName: NULL

```

```

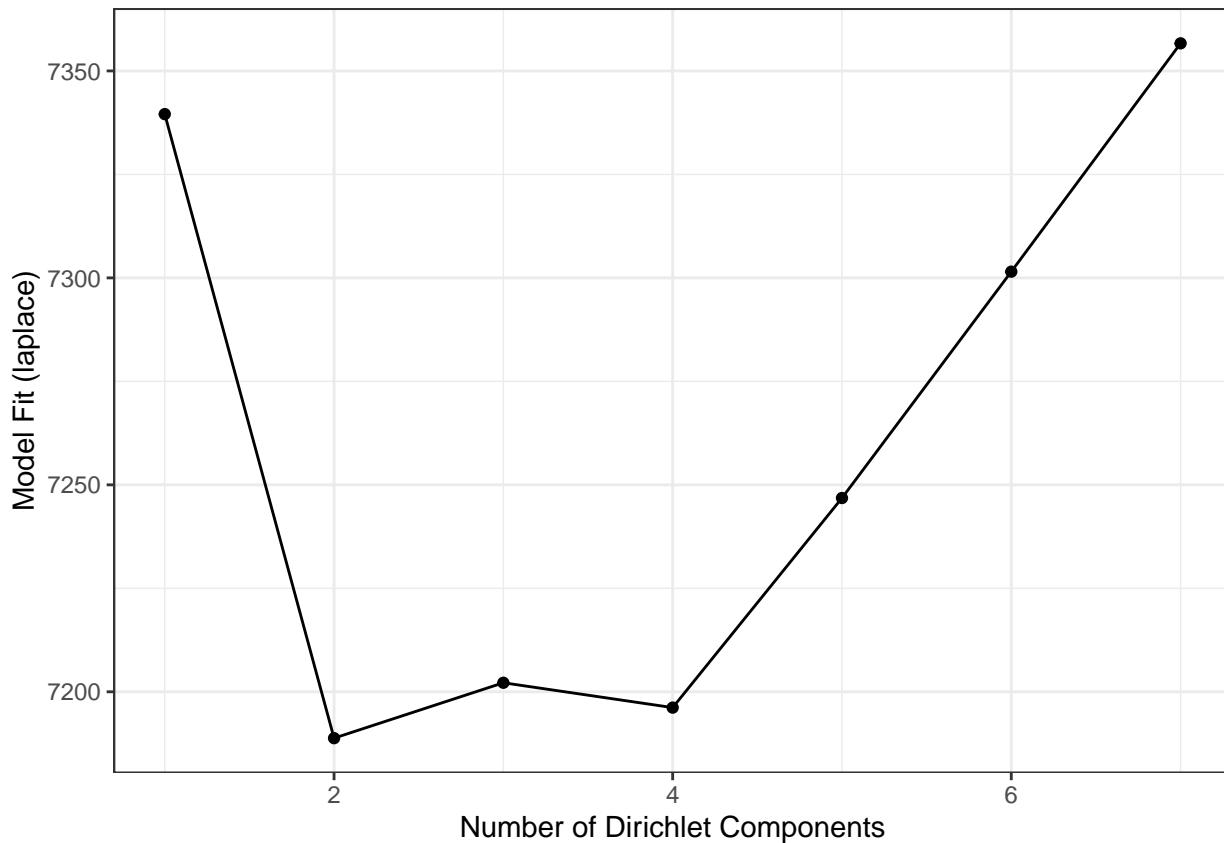
## altExpNames(0):
## rowLinks: a LinkDataFrame (19 rows)
## rowTree: 1 phylo tree(s) (16 leaves)
## collLinks: NULL
## colTree: NULL

getDMN(tse_dmn)

## [[1]]
## class: DMN
## k: 1
## samples x taxa: 120 x 19
## Laplace: 7339.569 BIC: 7361.579 AIC: 7335.098
##
## [[2]]
## class: DMN
## k: 2
## samples x taxa: 120 x 19
## Laplace: 7188.804 BIC: 7244.675 AIC: 7190.319
##
## [[3]]
## class: DMN
## k: 3
## samples x taxa: 120 x 19
## Laplace: 7202.187 BIC: 7294.175 AIC: 7211.944
##
## [[4]]
## class: DMN
## k: 4
## samples x taxa: 120 x 19
## Laplace: 7196.188 BIC: 7314.047 AIC: 7203.941
##
## [[5]]
## class: DMN
## k: 5
## samples x taxa: 120 x 19
## Laplace: 7246.821 BIC: 7383.342 AIC: 7245.361
##
## [[6]]
## class: DMN
## k: 6
## samples x taxa: 120 x 19
## Laplace: 7301.508 BIC: 7465.323 AIC: 7299.468
##
## [[7]]
## class: DMN
## k: 7
## samples x taxa: 120 x 19
## Laplace: 7356.665 BIC: 7544.576 AIC: 7350.845

miaViz::plotDMNFit(tse_dmn, type = "laplace")

```



```
getBestDMNFit(tse_dmn, type = "laplace") # Gives 3! as best fit for genus level data
```

```
## class: DMN
## k: 2
## samples x taxa: 120 x 19
## Laplace: 7188.804 BIC: 7244.675 AIC: 7190.319
```

```
# phylum level
```

```
tse = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)
tse_phylum <- agglomerateByRank(tse, rank = "Phylum", agglomerateTree = TRUE)
tse_dmn <- mia::runDMN(tse_phylum, name = "DMN", k = 1:7) # calculate most likely number of clusters fr
tse_dmn
```

```
## class: TreeSummarizedExperiment
## dim: 14 120
## metadata(2): agglomerated_by_rank DMN
## assays(1): counts
## rownames(14): Phylum:Beta-lactam Phylum:Aminoglycoside ...
##   Phylum:Oxazolidinone Phylum:Fosfomycin
## rowData names(4): Phylum Class Order Family
## colnames(120): 10_1 10_10 ... 9_38 9_39
## colData names(33): SampleIdentifier ResCap ... ReadTot Stables
## reducedDimNames(0):
## mainExpName: NULL
```

```

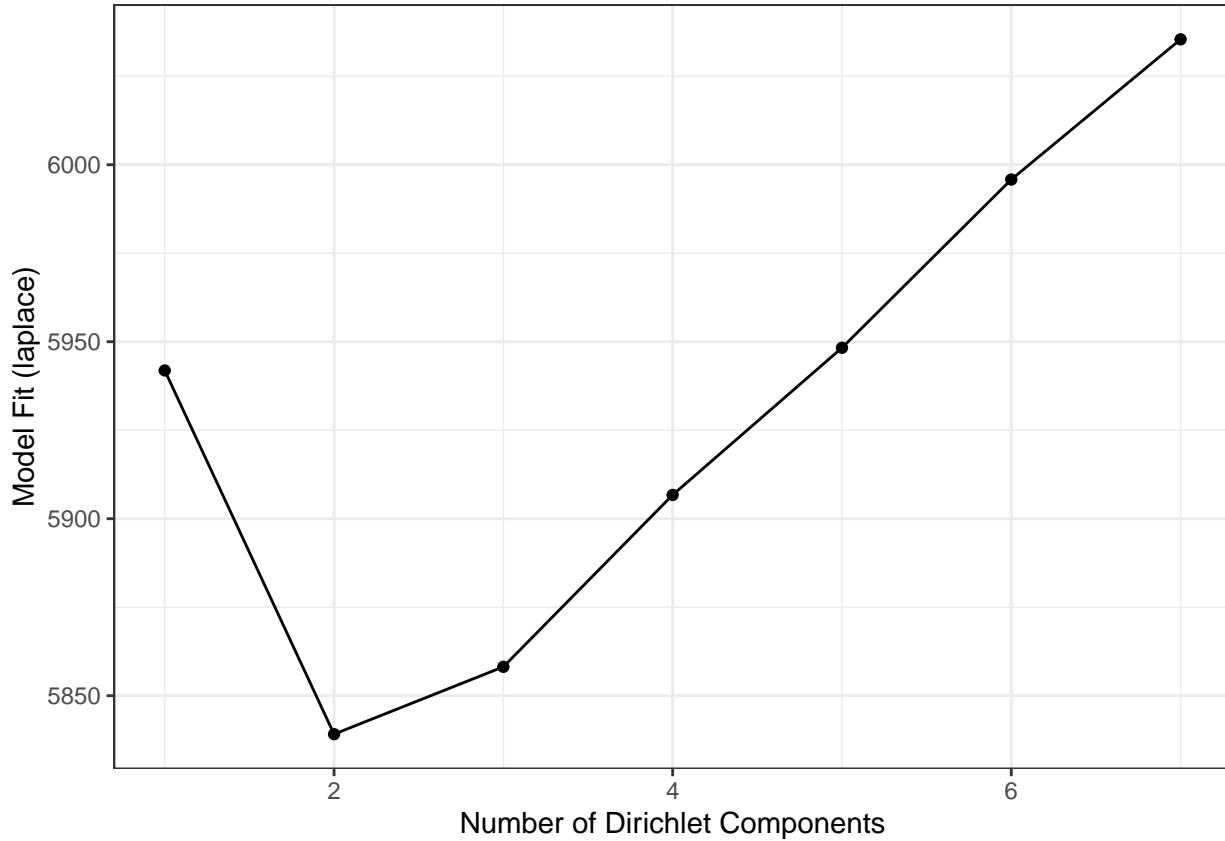
## altExpNames(0):
## rowLinks: a LinkDataFrame (14 rows)
## rowTree: 1 phylo tree(s) (14 leaves)
## collLinks: NULL
## colTree: NULL

getDMN(tse_dmn)

## [[1]]
## class: DMN
## k: 1
## samples x taxa: 120 x 14
## Laplace: 5941.881 BIC: 5957.522 AIC: 5938.01
##
## [[2]]
## class: DMN
## k: 2
## samples x taxa: 120 x 14
## Laplace: 5839.133 BIC: 5877.64 AIC: 5837.222
##
## [[3]]
## class: DMN
## k: 3
## samples x taxa: 120 x 14
## Laplace: 5858.158 BIC: 5917.767 AIC: 5856.442
##
## [[4]]
## class: DMN
## k: 4
## samples x taxa: 120 x 14
## Laplace: 5906.717 BIC: 5973.287 AIC: 5891.056
##
## [[5]]
## class: DMN
## k: 5
## samples x taxa: 120 x 14
## Laplace: 5948.292 BIC: 6032.688 AIC: 5929.551
##
## [[6]]
## class: DMN
## k: 6
## samples x taxa: 120 x 14
## Laplace: 5995.828 BIC: 6088.131 AIC: 5964.088
##
## [[7]]
## class: DMN
## k: 7
## samples x taxa: 120 x 14
## Laplace: 6035.429 BIC: 6140.726 AIC: 5995.777

miaViz::plotDMNFit(tse_dmn, type = "laplace")

```



```
getBestDMNFit(tse_dmn, type = "laplace") # Gives 2 as best fit for phylum level data
```

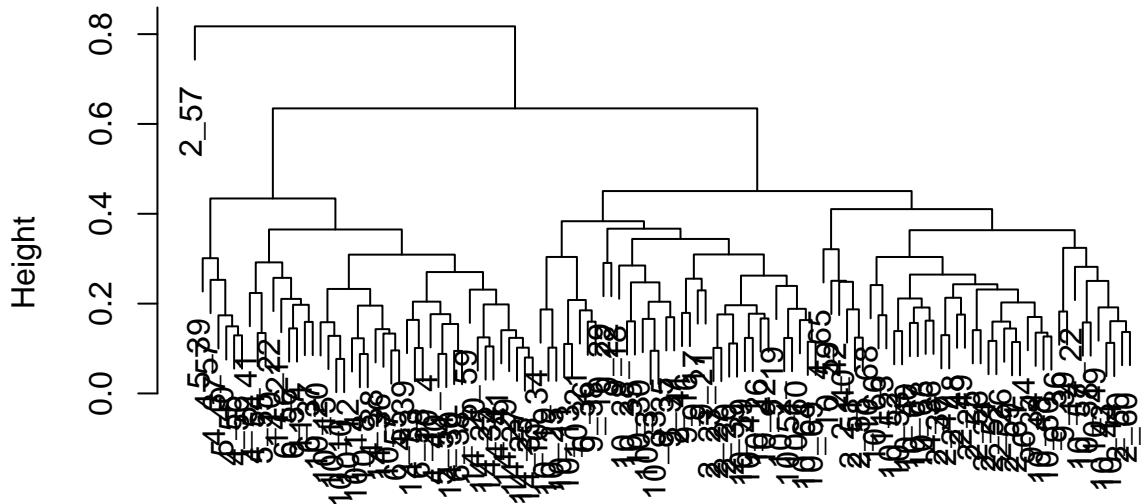
```
## class: DMN
## k: 2
## samples x taxa: 120 x 14
## Laplace: 5839.133 BIC: 5877.64 AIC: 5837.222
```

Hierachal clustering BC ARG

```
tse = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)
tse <- transformCounts(tse, method = "relabundance")
tse <- runMDS(tse,
               assay.type = "relabundance",
               FUN = vegan::vegdist,
               method = "bray")

hc_bray <- hclust(vegdist(t(assay(tse, "relabundance"))), method = "bray"), method = "complete")
plot(hc_bray)
```

Cluster Dendrogram



```
vegdist(t(assay(tse, "relabundance")), method = "bray")
hclust (*, "complete")
```

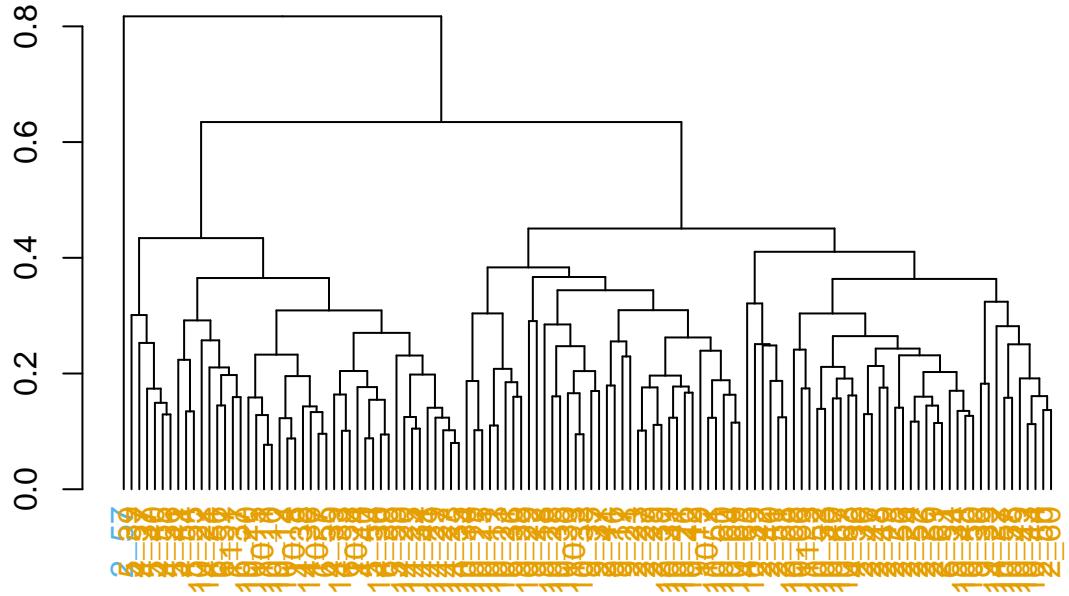
```
hcd = as.dendrogram(hc_bray)

cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442",
               "#0072B2", "#D55E00", "#CC79A7")

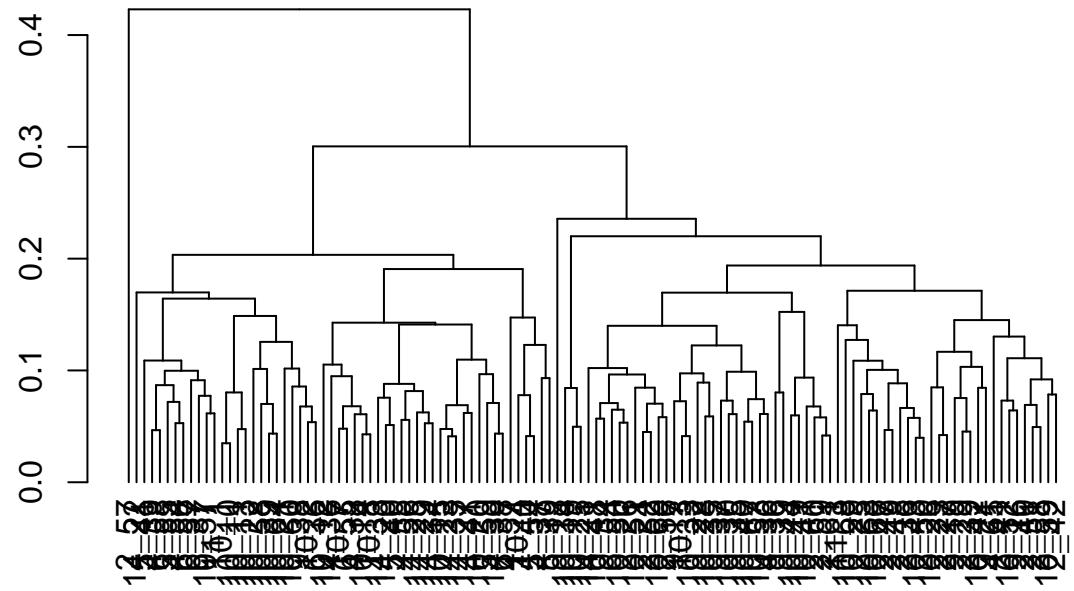
colorCode <- c(Control=cbPalette[2], CRC = cbPalette[3])

grouping = cutree(hc_bray, k = 2) # most methods gave 2 clusters, based on cuttree

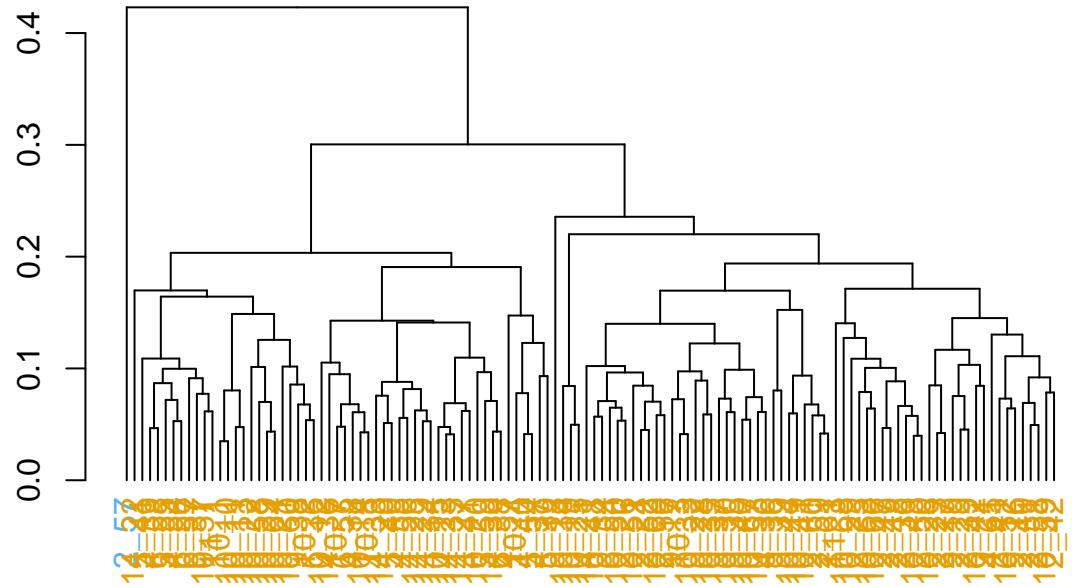
labels_colors(hcd) <- colorCode[grouping][order.dendrogram(hcd)]
plot(hcd)
```



```
hclust.out <- clusterRows(assay, HclustParam(method = "complete"), full = TRUE) # cutting based on comp
colData(tse)$clusters <- hclust.out$clusters
dendro <- as.dendrogram(hclust.out$objects$hclust)
plot(dendro)
```



```
labels_colors(dendro) <- colorCode[grouping][order.dendrogram(dendro)]
plot(dendro)
```

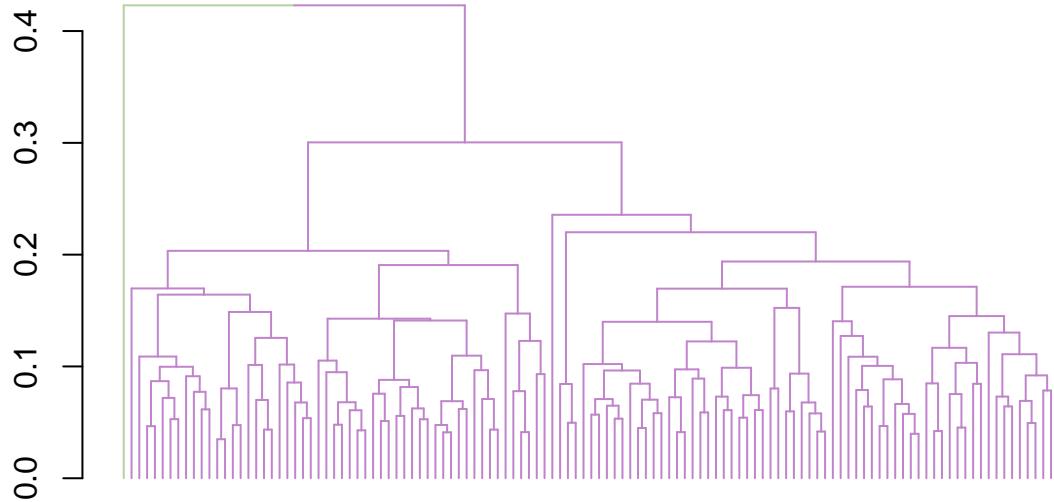


```

col_val_map <- randomcoloR::distinctColorPalette("2") %>%
  as.list() %>%
  setNames(paste0("clust_", seq("2")))

dend <- color_branches(dendro, k = 2, col = unlist(col_val_map))
labels(dend) <- NULL
plot(dend) # based on all three visualisations, only a few samples are clustered distinctly, based on s

```



```
# PAM clustering
tse = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)

tse <- transformCounts(tse, method = "relabundance")

pam.out <- clusterCells(tse,
                         assay.type = "relabundance",
                         BLUSPARAM = PamParam(centers = 2))

pam.out
```

```
## 10_1 10_10 10_11 10_12 10_13 10_14 10_15 10_19 10_2 10_20 10_21 10_22 10_25
## 1 1 1 1 1 1 1 2 2 1 2 2 2
## 10_26 10_28 10_29 10_3 10_30 10_33 10_34 10_35 10_39 10_4 10_40 10_41 10_42
## 2 2 2 2 2 2 2 2 2 1 2 2 2
## 10_43 10_44 10_48 10_49 10_50 10_51 10_52 10_53 10_57 10_58 10_59 10_60 10_63
## 2 2 2 2 2 2 2 2 1 1 2 2 2
## 10_64 10_66 10_67 10_68 10_69 10_7 10_8 11_1 11_3 14_20 14_21 14_22 14_23
## 2 2 2 2 2 1 1 1 2 1 2 1 1
## 14_25 14_27 14_29 14_30 14_33 14_34 14_35 14_36 2_23 2_24 2_25 2_26 2_27
## 1 1 1 1 1 1 1 1 2 2 2 2 2
## 2_29 2_36 4_65 2_39 2_40 2_41 2_42 2_47 2_48 2_49 2_50 2_51 2_52
## 2 2 2 2 2 2 2 2 2 2 2 2 2
## 2_56 2_57 2_58 2_59 2_60 2_61 4_36 4_37 4_38 4_39 4_40 4_41 4_54
## 2 2 2 2 2 2 2 1 1 1 1 1 1
## 4_55 4_56 4_57 5_39 5_40 5_41 5_54 5_55 5_59 6_36 6_37 6_38 6_54
```

```

##      1      1      1      1      1      2      1      1      1      1      1      1      1      1      2
## 6_55  6_56  6_57  6_58  9_16  9_17  9_18  9_19  9_21  9_22  9_34  9_35  9_36
##      1      1      1      1      2      2      2      2      2      2      2      2      2      2
## 9_37  9_38  9_39
##      2      2      2
## Levels: 1 2

n_iterations <- 1000
previous_cluster_assignment <- NULL
cluster_assignments <- list()

# loop that runs PAM clusterings X times and stores the results in a list, additionally checks if any changes occur
for (i in 1:n_iterations) {
  result <- clusterCells(tse, assay.type = "relabundance", BLUSPARAM = PamParam(centers = 2))
  cluster_assignments[[i]] <- result

  # Check if cluster assignments have changed
  if (!is.null(previous_cluster_assignment)) {
    samples_changed <- which(result != previous_cluster_assignment)
    if (length(samples_changed) > 0) {
      cat(sprintf("In iteration %d, the following samples changed clusters: %s\n", i, paste(samples_changed)))
    }
  }
  previous_cluster_assignment <- result
}

# To see if all of the clusters are the same or not
if (all(sapply(cluster_assignments, identical, cluster_assignments[[1]]))) {
  cat("All cluster assignments are the same across iterations.\n")
} else {
  cat("Cluster assignments vary across iterations.\n")
}

## All cluster assignments are the same across iterations.

# There are no differences in clusters when run 1000 times

```

Original PCoA plots, BC

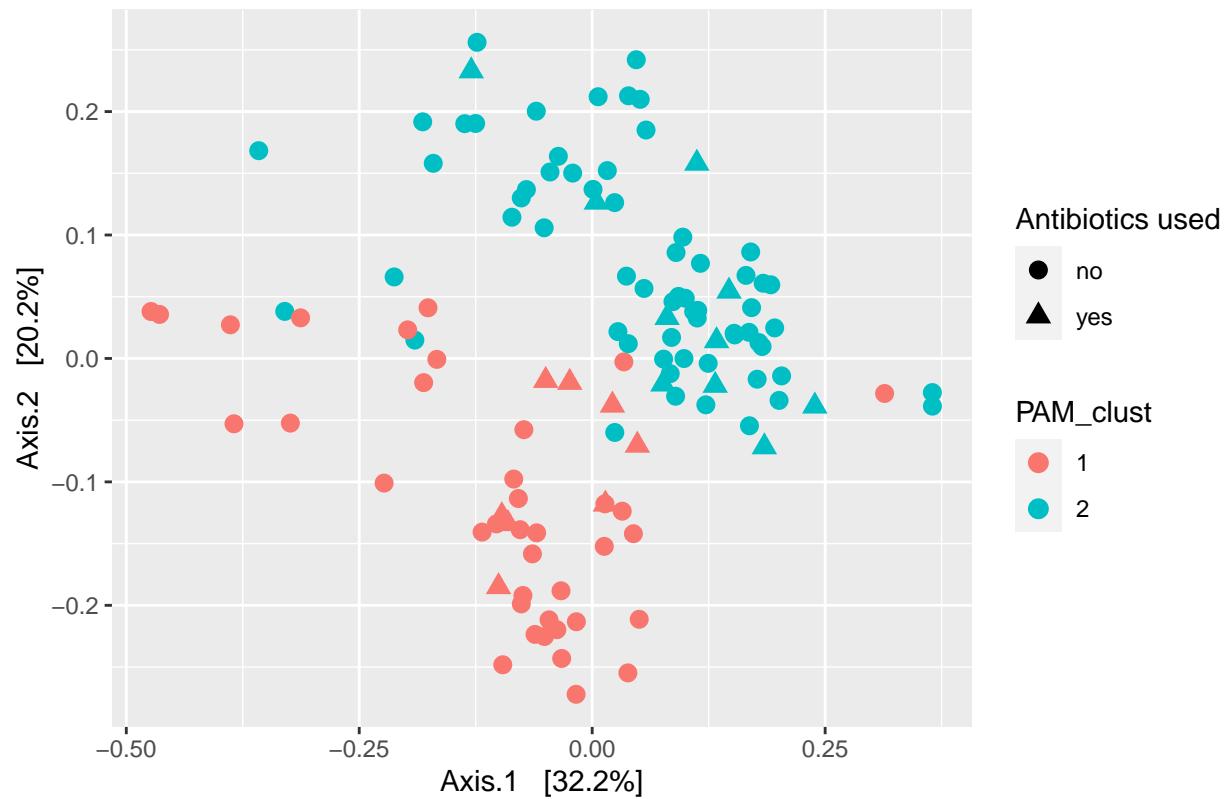
```

Rps@sam_data$PAM_clust = pam.out
sample_data(Rps)$PAM_clust = as.factor(sample_data(Rps)$PAM_clust)
pcoa_bc = ordinate(Rps, "PCoA", "bray")

plot_pcoa_ordination(Rps, pcoa_bc, "PAM_clust", "PCoA Bray Curtis")

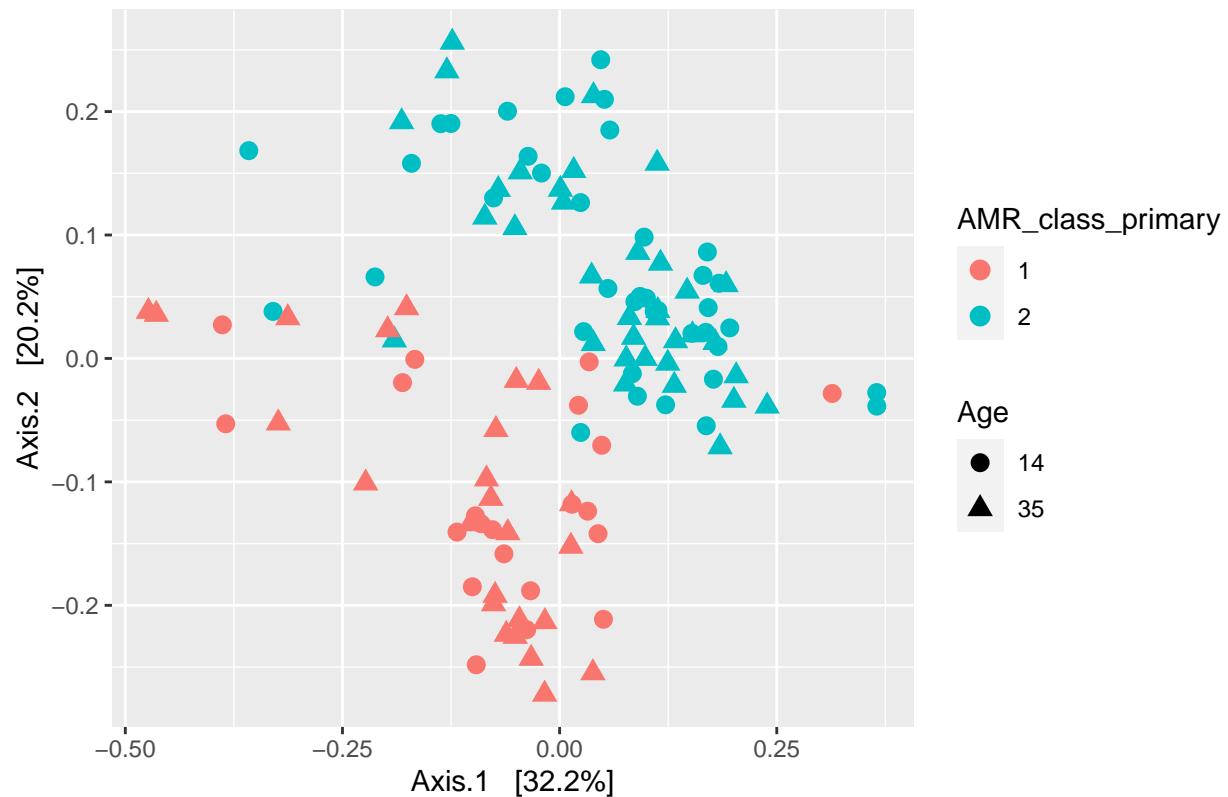
```

PCoA Bray Curtis



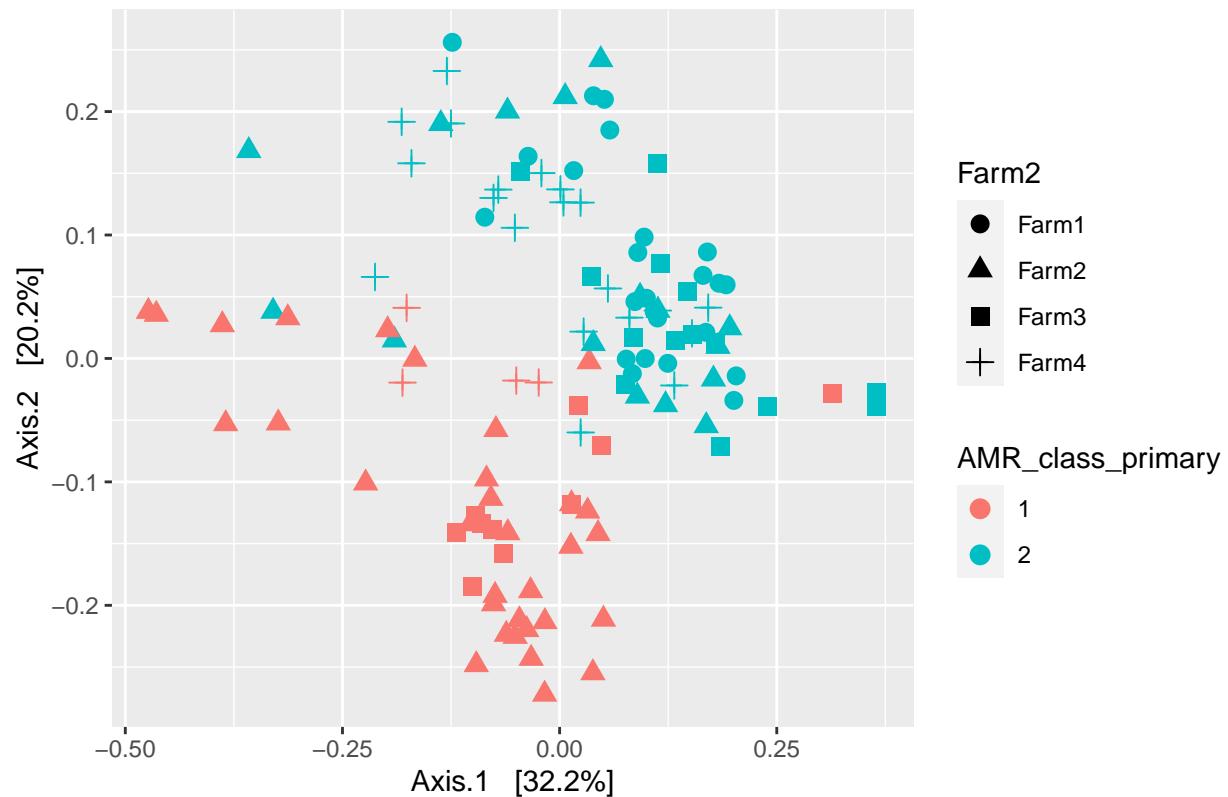
```
# change shape to different variables, age
plot_ordination(Rps, pcoa_bc, color = "PAM_clust", shape = "Age") +
  geom_point(size = 3) + labs(title = "PCoA Bray curtis", color = "AMR_class_primary")
```

PCoA Bray curtis



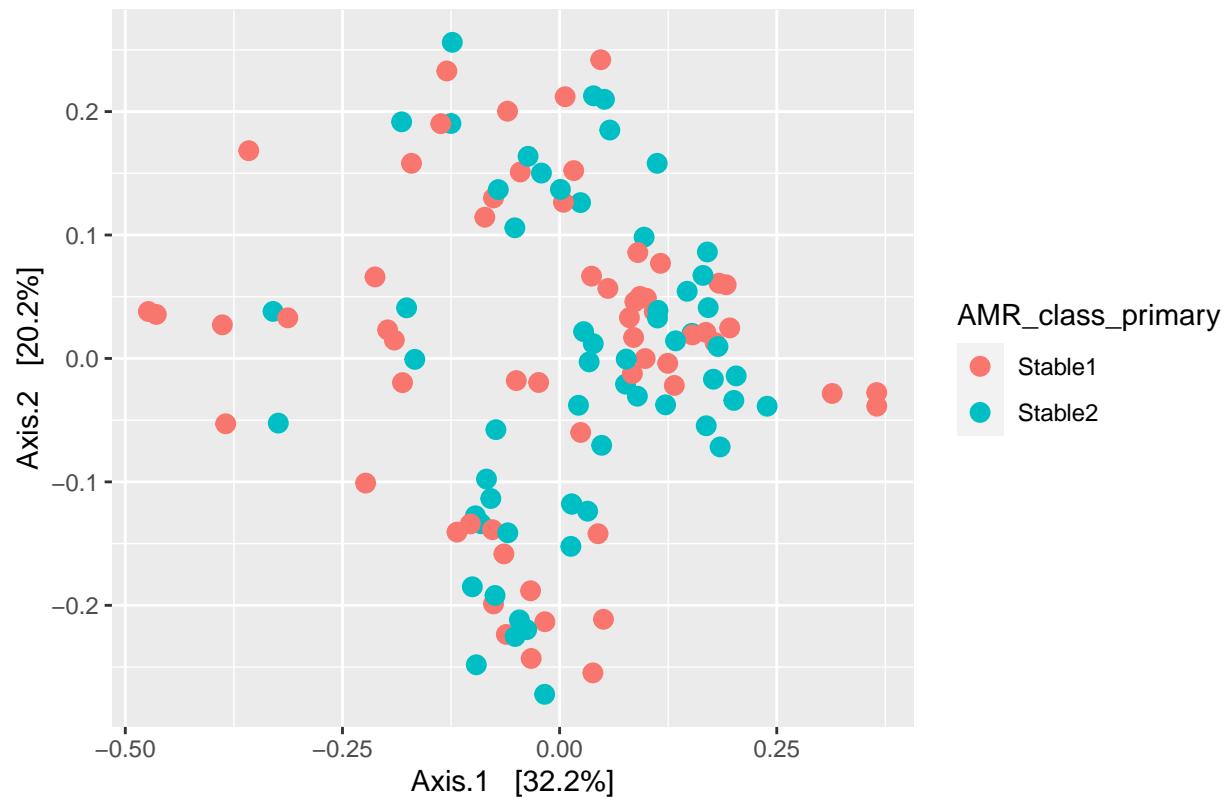
```
# change shape to different variables, farm
plot_ordination(Rps, pcoa_bc, color = "PAM_clust", shape = "Farm2") +
  geom_point(size = 3) + labs(title = "PCoA Bray curtis", color = "AMR_class_primary")
```

PCoA Bray curtis



```
# change shape to different variables, stable
plot_ordination(Rps, pcoa_bc, color = "Stable", shape = "PAM") +
  geom_point(size = 3) + labs(title = "PCoA Bray curtis", color = "AMR_class_primary")
```

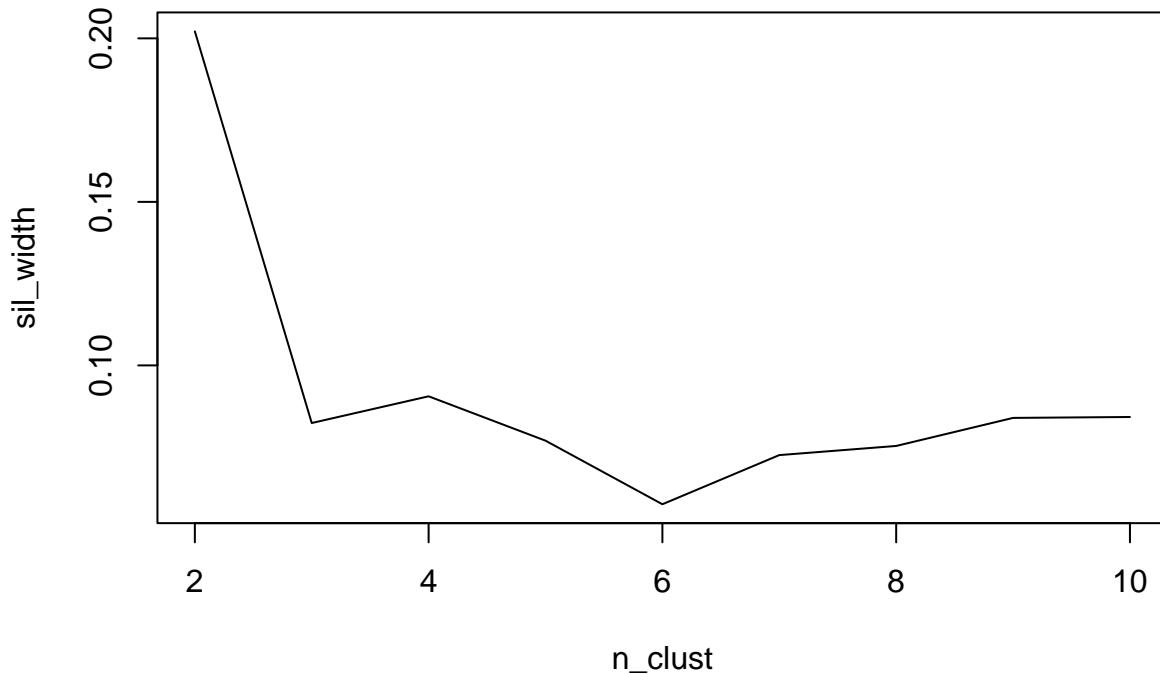
PCoA Bray curtis



Unifrac PAM plots

```
# Create PAM PCoA - from 2 to 10 clusters
phy_rel <- transform_sample_counts(Rps, function(x) log10(x+1/sum(x+1)))
UF <- UniFrac(phy_rel, weighted = TRUE)
n_clust <- 2:10
pam_list <- lapply(n_clust, function(x) pam(UF, k = x))

sil_width <- lapply(pam_list, function(x) mean(x$silinfo$widths[, "sil_width"]))
plot(n_clust, sil_width, type="l")
```



```

pcoa_data <- cmdscale(UF, eig = TRUE)
pcoa_df <- data.frame(PC1 = c(pcoa_data$points[,1]),
                       PC2 = c(pcoa_data$points[,2]),
                       Sample = rownames(pcoa_data$points))

# Add sample data
Samp <- data.frame(sample_data(Rps))
Samp$Sample <- sample_names(Rps)

pcoa_df <- merge(pcoa_df, Samp, by = "Sample")

pcoa_df

```

	Sample	PC1	PC2	SampleIdentifier	ResCap
## 1	10_1	0.0308617487	-2.147741e-03	Sample_61_1.F5.S1.CA.31.8.17	No
## 2	10_10	-0.0122466392	-2.032828e-04	Sample_62_1.F5.S2.CA.31.8.17	No
## 3	10_11	-0.0149253919	6.706921e-03	Sample_63_2.F5.S2.CA.31.8.17	No
## 4	10_12	-0.0169604545	-9.581893e-03	Sample_64_3.F5.S2.CA.31.8.17	No
## 5	10_13	-0.0122929116	-6.253164e-03	Sample_65_4.F5.S2.CA.31.8.17	No
## 6	10_14	-0.0186959094	-5.996277e-04	Sample_66_5.F5.S2.CA.31.8.17	No
## 7	10_15	-0.0166161460	3.759103e-03	Sample_67_6.F5.S2.CA.31.8.17	No
## 8	10_19	-0.0057788890	-2.935107e-03	Sample_68_1.F6.S1.CA.31.8.17	No
## 9	10_2	0.0432821491	-9.104968e-03	Sample_69_2.F5.S1.CA.31.8.17	No
## 10	10_20	-0.0159480245	-1.258393e-04	Sample_70_2.F6.S1.CA.31.8.17	No
## 11	10_21	-0.0046542929	-3.307018e-03	Sample_71_3.F6.S1.CA.31.8.17	No

## 12	10_22	-0.0155912428	2.107003e-03	Sample_72_4.F6.S1.CA.31.8.17	No
## 13	10_25	-0.0117946597	-2.206280e-03	Sample_73_7.F6.S1.CA.31.8.17	No
## 14	10_26	-0.0170055344	5.297607e-03	Sample_74_8.F6.S1.CA.31.8.17	No
## 15	10_28	0.0030973927	3.769991e-03	Sample_75_1.F6.S2.CA.31.8.17	No
## 16	10_29	0.0195691224	-1.292769e-02	Sample_76_2.F6.S2.CA.31.8.17	No
## 17	10_3	0.0437942727	-3.277766e-04	Sample_77_3.F5.S1.CA.31.8.17	No
## 18	10_30	0.0119184068	2.019109e-03	Sample_78_3.F6.S2.CA.31.8.17	No
## 19	10_33	0.0044997536	-4.324216e-03	Sample_79_4.F6.S2.CA.31.8.17	No
## 20	10_34	0.0034132875	-1.126630e-02	Sample_80_5.F6.S2.CA.31.8.17	No
## 21	10_35	0.0193204727	-1.374972e-02	Sample_81_6.F6.S2.CA.31.8.17	No
## 22	10_39	0.0163038886	7.718486e-03	Sample_82_1.F5.S1.CA.21.9.17	No
## 23	10_4	-0.0223795541	-2.993245e-03	Sample_83_4.F5.S1.CA.31.8.17	No
## 24	10_40	0.0090695431	1.841675e-02	Sample_84_2.F5.S1.CA.21.9.17	No
## 25	10_41	-0.0081940866	1.583016e-02	Sample_85_3.F5.S1.CA.21.9.17	No
## 26	10_42	-0.0027185570	1.778675e-02	Sample_86_4.F5.S1.CA.21.9.17	No
## 27	10_43	0.0120664520	1.541842e-02	Sample_87_5.F5.S1.CA.21.9.17	No
## 28	10_44	0.0100571408	1.668105e-02	Sample_88_6.F5.S1.CA.21.9.17	No
## 29	10_48	-0.0044704260	1.764435e-02	Sample_89_1.F5.S2.CA.21.9.17	Yes
## 30	10_49	-0.0117645403	1.448768e-02	Sample_90_2.F5.S2.CA.21.9.17	Yes
## 31	10_50	-0.0074897651	1.878814e-02	Sample_91_3.F5.S2.CA.21.9.17	Yes
## 32	10_51	-0.0133004277	6.557298e-03	Sample_92_4.F5.S2.CA.21.9.17	Yes
## 33	10_52	0.0128655730	2.558055e-03	Sample_93_5.F5.S2.CA.21.9.17	Yes
## 34	10_53	-0.0077769343	1.761318e-02	Sample_94_6.F5.S2.CA.21.9.17	Yes
## 35	10_57	0.0245279661	-1.223155e-02	Sample_95_1.F6.S1.CA.21.9.17	Yes
## 36	10_58	0.0245780146	-8.566694e-03	Sample_96_2.F6.S1.CA.21.9.17	Yes
## 37	10_59	0.0310343219	-1.510178e-02	Sample_97_3.F6.S1.CA.21.9.17	Yes
## 38	10_60	0.0127951812	-1.039044e-02	Sample_98_4.F6.S1.CA.21.9.17	Yes
## 39	10_63	0.0258947687	-8.605685e-03	Sample_99_7.F6.S1.CA.21.9.17	Yes
## 40	10_64	0.0078046000	3.276288e-03	Sample_100_8.F6.S1.CA.21.9.17	Yes
## 41	10_66	0.0203296970	-6.961970e-03	Sample_101_1.F6.S2.CA.21.9.17	No
## 42	10_67	-0.0028547592	-3.974910e-04	Sample_102_2.F6.S2.CA.21.9.17	No
## 43	10_68	-0.0002130074	-1.265577e-03	Sample_103_3.F6.S2.CA.21.9.17	No
## 44	10_69	0.0028873378	7.632360e-03	Sample_104_4.F6.S2.CA.21.9.17	No
## 45	10_7	-0.0198112200	-2.933875e-03	Sample_105_7.F5.S1.CA.31.8.17	No
## 46	10_8	-0.0187237062	-8.810983e-03	Sample_106_8.F5.S1.CA.31.8.17	No
## 47	11_1	0.0033057595	-1.650120e-02	Sample_107_5.F6.S2.CA.21.9.17	No
## 48	11_3	-0.0069311805	2.594544e-03	Sample_108_7.F6.S2.CA.21.9.17	No
## 49	14_20	-0.0185757202	-3.915569e-03	Sample_109_1.F4.S1.CA.29.8.2017	No
## 50	14_21	-0.0163542642	-1.069977e-02	Sample_110_2.F4.S1.CA.29.8.2017	No
## 51	14_22	-0.0219177406	-7.459273e-03	Sample_111_3.F4.S1.CA.29.8.2017	No
## 52	14_23	-0.0166222879	-5.914784e-03	Sample_112_4.F4.S1.CA.29.8.2017	No
## 53	14_25	-0.0145793598	-1.121395e-02	Sample_113_6.F4.S1.CA.29.8.2017	No
## 54	14_27	-0.0160025394	-7.610781e-03	Sample_114_8.F4.S1.CA.29.8.2017	No
## 55	14_29	-0.0278047841	-3.102545e-03	Sample_115_1.F4.S2.CA.29.8.2017	No
## 56	14_30	-0.0177008474	-6.686362e-03	Sample_116_2.F4.S2.CA.29.8.2017	No
## 57	14_33	-0.0228217976	-1.698313e-02	Sample_117_3.F4.S2.CA.29.8.2017	No
## 58	14_34	-0.0229524444	-1.103602e-03	Sample_118_4.F4.S2.CA.29.8.2017	No
## 59	14_35	-0.0192336573	2.080233e-03	Sample_119_5.F4.S2.CA.29.8.2017	No
## 60	14_36	-0.0213318912	-2.178582e-04	Sample_120_6.F4.S2.CA.29.8.2017	No
## 61	2_23	-0.0103004466	9.532955e-03	Sample_1_2.F1S1.31.08	No
## 62	2_24	0.0195619837	4.183424e-03	Sample_2_6.F1S1.31.08	No
## 63	2_25	-0.0056905267	8.716847e-03	Sample_3_10.F1S1.31.08	No
## 64	2_26	0.0026558124	7.222759e-03	Sample_4_14.F1S1.31.08	No
## 65	2_27	-0.0004842214	1.199354e-02	Sample_5_20.F1S1.31.08	No

## 66	2_29	-0.0102999410	1.311853e-02	Sample_6_28.F1S1.31.08	No
## 67	2_36	0.0074798774	6.626096e-03	Sample_7_9.F1S2.31.08	No
## 68	2_39	0.0143076658	1.376078e-03	Sample_10_21.F1S2.31.08	No
## 69	2_40	0.0050669076	-2.585201e-03	Sample_11_27.F1S2.31.08	No
## 70	2_41	-0.0144021601	1.310041e-02	Sample_12_31.F1S2.31.08	No
## 71	2_42	0.0147255231	-4.618657e-03	Sample_8_35.F1S2.31.08	No
## 72	2_47	0.0192113764	2.220345e-02	Sample_13_4.F1S1.21.09	No
## 73	2_48	0.0324390067	5.534507e-05	Sample_14_6.F1S1.21.09	No
## 74	2_49	0.0191935197	1.191188e-02	Sample_15_12.F1S1.21.09	No
## 75	2_50	0.0191777646	5.515603e-03	Sample_16_14.F1S1.21.09	No
## 76	2_51	0.0367150598	5.601008e-03	Sample_17_20.F1S1.21.09	No
## 77	2_52	0.0313383440	3.063639e-03	Sample_18_22.F1S1.21.09	No
## 78	2_56	-0.0094505990	1.122701e-02	Sample_19_1.F1S2.21.09	No
## 79	2_57	0.0227529992	-1.831178e-02	Sample_20_7.F1S2.21.09	No
## 80	2_58	0.0315919873	1.006555e-02	Sample_21_11.F1S2.21.09	No
## 81	2_59	0.0294812530	1.802285e-02	Sample_22_15.F1S2.21.09	No
## 82	2_60	0.0112989361	1.169577e-02	Sample_23_19.F1S2.21.09	No
## 83	2_61	-0.0014658745	7.867918e-03	Sample_24_21.F1S2.21.09	No
## 84	4_36	0.0037214380	-6.883059e-04	Sample_25_2.F2S1.20.06	Yes
## 85	4_37	-0.0084216386	-8.739740e-03	Sample_26_4.F2S1.20.06	Yes
## 86	4_38	-0.0090529302	9.560368e-03	Sample_27_8.F2S1.20.06	Yes
## 87	4_39	-0.0174488119	1.960027e-03	Sample_28_10.F2S1.20.06	Yes
## 88	4_40	-0.0009670481	1.457474e-03	Sample_29_14.F2S1.20.06	Yes
## 89	4_41	-0.0130975272	3.789811e-04	Sample_30_18.F2S1.20.06	Yes
## 90	4_54	-0.0022860245	-8.197761e-03	Sample_31_1.F2S1.11.07	Yes
## 91	4_55	-0.0237396094	-2.851137e-03	Sample_32_4.F2S1.11.07	Yes
## 92	4_56	0.0124244408	-1.309224e-02	Sample_33_9.F2S1.11.07	Yes
## 93	4_57	-0.0071986371	-1.042837e-02	Sample_34_10.F2S1.11.07	Yes
## 94	4_65	0.0036376625	2.393695e-03	Sample_9_5.F1S2.31.08	No
## 95	5_39	-0.0051576568	-8.777351e-03	Sample_35_2.F2S2.20.06	No
## 96	5_40	-0.0119068454	-5.195954e-03	Sample_36_5.F2S2.20.06	No
## 97	5_41	-0.0102167799	7.058579e-04	Sample_37_9.F2S2.20.06	No
## 98	5_54	0.0064843239	3.026791e-03	Sample_38_20.F2S1.11.07	Yes
## 99	5_55	0.0015755180	-2.233736e-03	Sample_39_24.F2S1.11.07	Yes
## 100	5_59	0.0176274650	-8.657869e-03	Sample_40_7.F2S2.11.07	No
## 101	6_36	0.0008945552	-4.293446e-03	Sample_41_12.F2S2.20.06	No
## 102	6_37	-0.0227060078	-8.824719e-03	Sample_42_14.F2S2.20.06	No
## 103	6_38	-0.0115484992	-4.478479e-03	Sample_43_17.F2S2.20.06	No
## 104	6_54	-0.0084493857	1.336277e-02	Sample_44_10.F2S2.11.07	No
## 105	6_55	-0.0237929368	2.222219e-03	Sample_45_13.F2S2.11.07	No
## 106	6_56	-0.0030509394	-1.025878e-02	Sample_46_18.F2S2.11.07	No
## 107	6_57	-0.0006466478	-3.644848e-03	Sample_47_19.F2S2.11.07	No
## 108	6_58	-0.0101634168	-6.463974e-03	Sample_48_22.F2S2.11.07	No
## 109	9_16	0.0154290001	-6.075002e-03	Sample_49_1.F4.S1.CA.8.8.2017	No
## 110	9_17	0.0134239463	-1.422878e-02	Sample_50_2.F4.S1.CA.8.8.2017	No
## 111	9_18	0.0228290799	-6.700340e-04	Sample_51_3.F4.S1.CA.8.8.2017	No
## 112	9_19	0.0056358205	-1.940892e-03	Sample_52_4.F4.S1.CA.8.8.2017	No
## 113	9_21	-0.0013024887	-5.847235e-03	Sample_53_6.F4.S1.CA.8.8.2017	No
## 114	9_22	0.0043434387	-8.363062e-03	Sample_54_7.F4.S1.CA.8.8.2017	No
## 115	9_34	-0.0122393877	-1.634027e-03	Sample_55_1.F4.S2.CA.8.8.2017	No
## 116	9_35	-0.0126957986	1.602112e-03	Sample_56_2.F4.S2.CA.8.8.2017	No
## 117	9_36	0.0056642706	-7.589723e-04	Sample_57_3.F4.S2.CA.8.8.2017	No
## 118	9_37	-0.0061223160	-4.956885e-03	Sample_58_4.F4.S2.CA.8.8.2017	No
## 119	9_38	-0.0112595573	-5.927578e-03	Sample_59_5.F4.S2.CA.8.8.2017	No

```

## 120 9_39 -0.0133655723 2.959105e-03 Sample_60_6.F4.S2.CA.8.8.2017 No
## Conc...ng..pl. SampleID LibraryNumber Sample_Unique LibraryName
## 1 73.33500 1.F5.S1.CA.31.8.17 10 10_1 2017_0053
## 2 72.29580 1.F5.S2.CA.31.8.17 10 10_10 2017_0053
## 3 63.80300 2.F5.S2.CA.31.8.17 10 10_11 2017_0053
## 4 60.42300 3.F5.S2.CA.31.8.17 10 10_12 2017_0053
## 5 79.59467 4.F5.S2.CA.31.8.17 10 10_13 2017_0053
## 6 69.94600 5.F5.S2.CA.31.8.17 10 10_14 2017_0053
## 7 69.52307 6.F5.S2.CA.31.8.17 10 10_15 2017_0053
## 8 72.32983 1.F6.S1.CA.31.8.17 10 10_19 2017_0053
## 9 72.93300 2.F5.S1.CA.31.8.17 10 10_2 2017_0053
## 10 76.16580 2.F6.S1.CA.31.8.17 10 10_20 2017_0053
## 11 69.75523 3.F6.S1.CA.31.8.17 10 10_21 2017_0053
## 12 66.00067 4.F6.S1.CA.31.8.17 10 10_22 2017_0053
## 13 64.12267 7.F6.S1.CA.31.8.17 10 10_25 2017_0053
## 14 77.33790 8.F6.S1.CA.31.8.17 10 10_26 2017_0053
## 15 71.92313 1.F6.S2.CA.31.8.17 10 10_28 2017_0053
## 16 82.66720 2.F6.S2.CA.31.8.17 10 10_29 2017_0053
## 17 77.25233 3.F5.S1.CA.31.8.17 10 10_3 2017_0053
## 18 67.56417 3.F6.S2.CA.31.8.17 10 10_30 2017_0053
## 19 64.64467 4.F6.S2.CA.31.8.17 10 10_33 2017_0053
## 20 71.48300 5.F6.S2.CA.31.8.17 10 10_34 2017_0053
## 21 73.63020 6.F6.S2.CA.31.8.17 10 10_35 2017_0053
## 22 76.03517 1.F5.S1.CA.21.9.17 10 10_39 2017_0053
## 23 69.57433 4.F5.S1.CA.31.8.17 10 10_4 2017_0053
## 24 73.50420 2.F5.S1.CA.21.9.17 10 10_40 2017_0053
## 25 72.34367 3.F5.S1.CA.21.9.17 10 10_41 2017_0053
## 26 67.43267 4.F5.S1.CA.21.9.17 10 10_42 2017_0053
## 27 68.10107 5.F5.S1.CA.21.9.17 10 10_43 2017_0053
## 28 74.28150 6.F5.S1.CA.21.9.17 10 10_44 2017_0053
## 29 69.71200 1.F5.S2.CA.21.9.17 10 10_48 2017_0053
## 30 73.99490 2.F5.S2.CA.21.9.17 10 10_49 2017_0053
## 31 79.25733 3.F5.S2.CA.21.9.17 10 10_50 2017_0053
## 32 82.94253 4.F5.S2.CA.21.9.17 10 10_51 2017_0053
## 33 89.19880 5.F5.S2.CA.21.9.17 10 10_52 2017_0053
## 34 78.57480 6.F5.S2.CA.21.9.17 10 10_53 2017_0053
## 35 76.82853 1.F6.S1.CA.21.9.17 10 10_57 2017_0053
## 36 67.72533 2.F6.S1.CA.21.9.17 10 10_58 2017_0053
## 37 74.09880 3.F6.S1.CA.21.9.17 10 10_59 2017_0053
## 38 64.87040 4.F6.S1.CA.21.9.17 10 10_60 2017_0053
## 39 77.01563 7.F6.S1.CA.21.9.17 10 10_63 2017_0053
## 40 48.77200 8.F6.S1.CA.21.9.17 10 10_64 2017_0053
## 41 77.41880 1.F6.S2.CA.21.9.17 10 10_66 2017_0053
## 42 64.98707 2.F6.S2.CA.21.9.17 10 10_67 2017_0053
## 43 81.37103 3.F6.S2.CA.21.9.17 10 10_68 2017_0053
## 44 58.31600 4.F6.S2.CA.21.9.17 10 10_69 2017_0053
## 45 84.45150 7.F5.S1.CA.31.8.17 10 10_7 2017_0053
## 46 79.01577 8.F5.S1.CA.31.8.17 10 10_8 2017_0053
## 47 73.03933 5.F6.S2.CA.21.9.17 11 11_1 2017_0054
## 48 77.95520 7.F6.S2.CA.21.9.17 11 11_3 2017_0054
## 49 84.25480 1.F4.S1.CA.29.8.2017 14 14_20 2018_0010
## 50 68.90347 2.F4.S1.CA.29.8.2017 14 14_21 2018_0010
## 51 62.67573 3.F4.S1.CA.29.8.2017 14 14_22 2018_0010
## 52 79.13413 4.F4.S1.CA.29.8.2017 14 14_23 2018_0010

```

## 53	73.48297	6.F4.S1.CA.29.8.2017	14	14_25	2018_0010
## 54	90.94703	8.F4.S1.CA.29.8.2017	14	14_27	2018_0010
## 55	70.74400	1.F4.S2.CA.29.8.2017	14	14_29	2018_0010
## 56	72.05500	2.F4.S2.CA.29.8.2017	14	14_30	2018_0010
## 57	68.78233	3.F4.S2.CA.29.8.2017	14	14_33	2018_0010
## 58	67.81027	4.F4.S2.CA.29.8.2017	14	14_34	2018_0010
## 59	71.63660	5.F4.S2.CA.29.8.2017	14	14_35	2018_0010
## 60	80.74540	6.F4.S2.CA.29.8.2017	14	14_36	2018_0010
## 61	50.83700	2.F1S1.31.08	2	2_23	2017_0005
## 62	25.61300	6.F1S1.31.08	2	2_24	2017_0005
## 63	53.77000	10.F1S1.31.08	2	2_25	2017_0005
## 64	81.22200	14.F1S1.31.08	2	2_26	2017_0005
## 65	44.94700	20.F1S1.31.08	2	2_27	2017_0005
## 66	64.64400	28.F1S1.31.08	2	2_29	2017_0005
## 67	24.72100	9.F1S2.31.08	2	2_36	2017_0005
## 68	22.32200	21.F1S2.31.08	2	2_39	2017_0005
## 69	39.18500	27.F1S2.31.08	2	2_40	2017_0005
## 70	50.09800	31.F1S2.31.08	2	2_41	2017_0005
## 71	104.39700	35.F1S2.31.08	2	2_42	2017_0005
## 72	91.13600	4.F1S1.21.09	2	2_47	2017_0005
## 73	52.57400	6.F1S1.21.09	2	2_48	2017_0005
## 74	39.98300	12.F1S1.21.09	2	2_49	2017_0005
## 75	56.80000	14.F1S1.21.09	2	2_50	2017_0005
## 76	141.90600	20.F1S1.21.09	2	2_51	2017_0005
## 77	81.55400	22.F1S1.21.09	2	2_52	2017_0005
## 78	65.46300	1.F1S2.21.09	2	2_56	2017_0005
## 79	61.03200	7.F1S2.21.09	2	2_57	2017_0005
## 80	113.21600	11.F1S2.21.09	2	2_58	2017_0005
## 81	61.96000	15.F1S2.21.09	2	2_59	2017_0005
## 82	93.91200	19.F1S2.21.09	2	2_60	2017_0005
## 83	36.29700	21.F1S2.21.09	2	2_61	2017_0005
## 84	12.12100	2.F2S1.20.06	4	4_36	2017_0020
## 85	3.51200	4.F2S1.20.06	4	4_37	2017_0020
## 86	5.50800	8.F2S1.20.06	4	4_38	2017_0020
## 87	6.58900	10.F2S1.20.06	4	4_39	2017_0020
## 88	5.51000	14.F2S1.20.06	4	4_40	2017_0020
## 89	6.64700	18.F2S1.20.06	4	4_41	2017_0020
## 90	4.20100	1.F2S1.11.07	4	4_54	2017_0020
## 91	5.79000	4.F2S1.11.07	4	4_55	2017_0020
## 92	4.27900	9.F2S1.11.07	4	4_56	2017_0020
## 93	4.15800	10.F2S1.11.07	4	4_57	2017_0020
## 94	59.70000	5.F1S2.31.08	4	4_65	2017_0020
## 95	6.85800	2.F2S2.20.06	5	5_39	2017_0021
## 96	3.32300	5.F2S2.20.06	5	5_40	2017_0021
## 97	4.81800	9.F2S2.20.06	5	5_41	2017_0021
## 98	3.60300	20.F2S1.11.07	5	5_54	2017_0021
## 99	5.27500	24.F2S1.11.07	5	5_55	2017_0021
## 100	4.32300	7.F2S2.11.07	5	5_59	2017_0021
## 101	78.02533	12.F2S2.20.06	6	6_36	2017_0022
## 102	55.90383	14.F2S2.20.06	6	6_37	2017_0022
## 103	56.30460	17.F2S2.20.06	6	6_38	2017_0022
## 104	58.89543	10.F2S2.11.07	6	6_54	2017_0022
## 105	64.67200	13.F2S2.11.07	6	6_55	2017_0022
## 106	67.05417	18.F2S2.11.07	6	6_56	2017_0022

## 107	68.20467	19.F2S2.11.07	6	6_57	2017_0022	
## 108	65.37080	22.F2S2.11.07	6	6_58	2017_0022	
## 109	66.42850	1.F4.S1.CA.8.8.2017	9	9_16	2017_0045	
## 110	70.57377	2.F4.S1.CA.8.8.2017	9	9_17	2017_0045	
## 111	97.43933	3.F4.S1.CA.8.8.2017	9	9_18	2017_0045	
## 112	78.36967	4.F4.S1.CA.8.8.2017	9	9_19	2017_0045	
## 113	70.48433	6.F4.S1.CA.8.8.2017	9	9_21	2017_0045	
## 114	72.95400	7.F4.S1.CA.8.8.2017	9	9_22	2017_0045	
## 115	87.33893	1.F4.S2.CA.8.8.2017	9	9_34	2017_0045	
## 116	89.91280	2.F4.S2.CA.8.8.2017	9	9_35	2017_0045	
## 117	81.12347	3.F4.S2.CA.8.8.2017	9	9_36	2017_0045	
## 118	84.23263	4.F4.S2.CA.8.8.2017	9	9_37	2017_0045	
## 119	74.69257	5.F4.S2.CA.8.8.2017	9	9_38	2017_0045	
## 120	61.72093	6.F4.S2.CA.8.8.2017	9	9_39	2017_0045	
##	Farm	Farm2	Stable	FarmRoundStable	Days Age	Sname
## 1	Farm5	Farm3	Stable1	Farm3R1S1	Day14 14	1_F5_S1_CA_31.8.17
## 2	Farm5	Farm3	Stable2	Farm3R1S2	Day14 14	1_F5_S2_CA_31.8.17
## 3	Farm5	Farm3	Stable2	Farm3R1S2	Day14 14	2_F5_S2_CA_31.8.17
## 4	Farm5	Farm3	Stable2	Farm3R1S2	Day14 14	3_F5_S2_CA_31.8.17
## 5	Farm5	Farm3	Stable2	Farm3R1S2	Day14 14	4_F5_S2_CA_31.8.17
## 6	Farm5	Farm3	Stable2	Farm3R1S2	Day14 14	5_F5_S2_CA_31.8.17
## 7	Farm5	Farm3	Stable2	Farm3R1S2	Day14 14	6_F5_S2_CA_31.8.17
## 8	Farm6	Farm4	Stable1	Farm4R1S1	Day13 14	1_F6_S1_CA_31.8.17
## 9	Farm5	Farm3	Stable1	Farm3R1S1	Day14 14	2_F5_S1_CA_31.8.17
## 10	Farm6	Farm4	Stable1	Farm4R1S1	Day13 14	2_F6_S1_CA_31.8.17
## 11	Farm6	Farm4	Stable1	Farm4R1S1	Day13 14	3_F6_S1_CA_31.8.17
## 12	Farm6	Farm4	Stable1	Farm4R1S1	Day13 14	4_F6_S1_CA_31.8.17
## 13	Farm6	Farm4	Stable1	Farm4R1S1	Day13 14	7_F6_S1_CA_31.8.17
## 14	Farm6	Farm4	Stable1	Farm4R1S1	Day13 14	8_F6_S1_CA_31.8.17
## 15	Farm6	Farm4	Stable2	Farm4R1S2	Day13 14	1_F6_S2_CA_31.8.17
## 16	Farm6	Farm4	Stable2	Farm4R1S2	Day13 14	2_F6_S2_CA_31.8.17
## 17	Farm5	Farm3	Stable1	Farm3R1S1	Day14 14	3_F5_S1_CA_31.8.17
## 18	Farm6	Farm4	Stable2	Farm4R1S2	Day13 14	3_F6_S2_CA_31.8.17
## 19	Farm6	Farm4	Stable2	Farm4R1S2	Day13 14	4_F6_S2_CA_31.8.17
## 20	Farm6	Farm4	Stable2	Farm4R1S2	Day13 14	5_F6_S2_CA_31.8.17
## 21	Farm6	Farm4	Stable2	Farm4R1S2	Day13 14	6_F6_S2_CA_31.8.17
## 22	Farm5	Farm3	Stable1	Farm3R1S1	Day35 35	1_F5_S1_CA_21.9.17
## 23	Farm5	Farm3	Stable1	Farm3R1S1	Day14 14	4_F5_S1_CA_31.8.17
## 24	Farm5	Farm3	Stable1	Farm3R1S1	Day35 35	2_F5_S1_CA_21.9.17
## 25	Farm5	Farm3	Stable1	Farm3R1S1	Day35 35	3_F5_S1_CA_21.9.17
## 26	Farm5	Farm3	Stable1	Farm3R1S1	Day35 35	4_F5_S1_CA_21.9.17
## 27	Farm5	Farm3	Stable1	Farm3R1S1	Day35 35	5_F5_S1_CA_21.9.17
## 28	Farm5	Farm3	Stable1	Farm3R1S1	Day35 35	6_F5_S1_CA_21.9.17
## 29	Farm5	Farm3	Stable2	Farm3R1S2	Day35 35	1_F5_S2_CA_21.9.17
## 30	Farm5	Farm3	Stable2	Farm3R1S2	Day35 35	2_F5_S2_CA_21.9.17
## 31	Farm5	Farm3	Stable2	Farm3R1S2	Day35 35	3_F5_S2_CA_21.9.17
## 32	Farm5	Farm3	Stable2	Farm3R1S2	Day35 35	4_F5_S2_CA_21.9.17
## 33	Farm5	Farm3	Stable2	Farm3R1S2	Day35 35	5_F5_S2_CA_21.9.17
## 34	Farm5	Farm3	Stable2	Farm3R1S2	Day35 35	6_F5_S2_CA_21.9.17
## 35	Farm6	Farm4	Stable1	Farm4R1S1	Day34 35	1_F6_S1_CA_21.9.17
## 36	Farm6	Farm4	Stable1	Farm4R1S1	Day34 35	2_F6_S1_CA_21.9.17
## 37	Farm6	Farm4	Stable1	Farm4R1S1	Day34 35	3_F6_S1_CA_21.9.17
## 38	Farm6	Farm4	Stable1	Farm4R1S1	Day34 35	4_F6_S1_CA_21.9.17
## 39	Farm6	Farm4	Stable1	Farm4R1S1	Day34 35	7_F6_S1_CA_21.9.17

## 40	Farm6	Farm4	Stable1	Farm4R1S1	Day34	35	8_F6_S1_CA_21.9.17
## 41	Farm6	Farm4	Stable2	Farm4R1S2	Day34	35	1_F6_S2_CA_21.9.17
## 42	Farm6	Farm4	Stable2	Farm4R1S2	Day34	35	2_F6_S2_CA_21.9.17
## 43	Farm6	Farm4	Stable2	Farm4R1S2	Day34	35	3_F6_S2_CA_21.9.17
## 44	Farm6	Farm4	Stable2	Farm4R1S2	Day34	35	4_F6_S2_CA_21.9.17
## 45	Farm5	Farm3	Stable1	Farm3R1S1	Day14	14	7_F5_S1_CA_31.8.17
## 46	Farm5	Farm3	Stable1	Farm3R1S1	Day14	14	8_F5_S1_CA_31.8.17
## 47	Farm6	Farm4	Stable2	Farm4R1S2	Day34	35	5_F6_S2_CA_21.9.17
## 48	Farm6	Farm4	Stable2	Farm4R1S2	Day34	35	7_F6_S2_CA_21.9.17
## 49	Farm4	Farm2	Stable1	Farm2R2S1	Day35	35	1.F4.S1.CA.29.8.2017
## 50	Farm4	Farm2	Stable1	Farm2R2S1	Day35	35	2.F4.S1.CA.29.8.2017
## 51	Farm4	Farm2	Stable1	Farm2R2S1	Day35	35	3.F4.S1.CA.29.8.2017
## 52	Farm4	Farm2	Stable1	Farm2R2S1	Day35	35	4.F4.S1.CA.29.8.2017
## 53	Farm4	Farm2	Stable1	Farm2R2S1	Day35	35	6.F4.S1.CA.29.8.2017
## 54	Farm4	Farm2	Stable1	Farm2R2S1	Day35	35	8.F4.S1.CA.29.8.2017
## 55	Farm4	Farm2	Stable2	Farm2R2S2	Day35	35	1.F4.S2.CA.29.8.2017
## 56	Farm4	Farm2	Stable2	Farm2R2S2	Day35	35	2.F4.S2.CA.29.8.2017
## 57	Farm4	Farm2	Stable2	Farm2R2S2	Day35	35	3.F4.S2.CA.29.8.2017
## 58	Farm4	Farm2	Stable2	Farm2R2S2	Day35	35	4.F4.S2.CA.29.8.2017
## 59	Farm4	Farm2	Stable2	Farm2R2S2	Day35	35	5.F4.S2.CA.29.8.2017
## 60	Farm4	Farm2	Stable2	Farm2R2S2	Day35	35	6.F4.S2.CA.29.8.2017
## 61	Farm1	Farm1	Stable1	Farm1R1S1	Day14	14	2.F1S1.31.08
## 62	Farm1	Farm1	Stable1	Farm1R1S1	Day14	14	6.F1S1.31.08
## 63	Farm1	Farm1	Stable1	Farm1R1S1	Day14	14	10.F1S1.31.08
## 64	Farm1	Farm1	Stable1	Farm1R1S1	Day14	14	14.F1S1.31.08
## 65	Farm1	Farm1	Stable1	Farm1R1S1	Day14	14	20.F1S1.31.08
## 66	Farm1	Farm1	Stable1	Farm1R1S1	Day14	14	28.F1S1.31.08
## 67	Farm1	Farm1	Stable2	Farm1R1S2	Day14	14	9.F1S2.31.08
## 68	Farm1	Farm1	Stable2	Farm1R1S2	Day14	14	21.F1S2.31.08
## 69	Farm1	Farm1	Stable2	Farm1R1S2	Day14	14	27.F1S2.31.08
## 70	Farm1	Farm1	Stable2	Farm1R1S2	Day14	14	31.F1S2.31.08
## 71	Farm1	Farm1	Stable2	Farm1R1S2	Day14	14	35.F1S2.31.08
## 72	Farm1	Farm1	Stable1	Farm1R1S1	Day35	35	4.F1S1.21.09
## 73	Farm1	Farm1	Stable1	Farm1R1S1	Day35	35	6.F1S1.21.09
## 74	Farm1	Farm1	Stable1	Farm1R1S1	Day35	35	12.F1S1.21.09
## 75	Farm1	Farm1	Stable1	Farm1R1S1	Day35	35	14.F1S1.21.09
## 76	Farm1	Farm1	Stable1	Farm1R1S1	Day35	35	20.F1S1.21.09
## 77	Farm1	Farm1	Stable1	Farm1R1S1	Day35	35	22.F1S1.21.09
## 78	Farm1	Farm1	Stable2	Farm1R1S2	Day35	35	1.F1S2.21.09
## 79	Farm1	Farm1	Stable2	Farm1R1S2	Day35	35	7.F1S2.21.09
## 80	Farm1	Farm1	Stable2	Farm1R1S2	Day35	35	11.F1S2.21.09
## 81	Farm1	Farm1	Stable2	Farm1R1S2	Day35	35	15.F1S2.21.09
## 82	Farm1	Farm1	Stable2	Farm1R1S2	Day35	35	19.F1S2.21.09
## 83	Farm1	Farm1	Stable2	Farm1R1S2	Day35	35	21.F1S2.21.09
## 84	Farm2	Farm2	Stable1	Farm2R1S1	Day14	14	2.F2S1.20.06
## 85	Farm2	Farm2	Stable1	Farm2R1S1	Day14	14	4.F2S1.20.06
## 86	Farm2	Farm2	Stable1	Farm2R1S1	Day14	14	8.F2S1.20.06
## 87	Farm2	Farm2	Stable1	Farm2R1S1	Day14	14	10.F2S1.20.06
## 88	Farm2	Farm2	Stable1	Farm2R1S1	Day14	14	14.F2S1.20.06
## 89	Farm2	Farm2	Stable1	Farm2R1S1	Day14	14	18.F2S1.20.06
## 90	Farm2	Farm2	Stable1	Farm2R1S1	Day35	35	1.F2S1.11.07
## 91	Farm2	Farm2	Stable1	Farm2R1S1	Day35	35	4.F2S1.11.07
## 92	Farm2	Farm2	Stable1	Farm2R1S1	Day35	35	9.F2S1.11.07
## 93	Farm2	Farm2	Stable1	Farm2R1S1	Day35	35	10.F2S1.11.07

## 94	Farm1	Farm1	Stable2	Farm1R1S2	Day14	14	5.F1S2.31.08			
## 95	Farm2	Farm2	Stable2	Farm2R1S2	Day14	14	2.F2S2.20.06			
## 96	Farm2	Farm2	Stable2	Farm2R1S2	Day14	14	5.F2S2.20.06			
## 97	Farm2	Farm2	Stable2	Farm2R1S2	Day14	14	9.F2S2.20.06			
## 98	Farm2	Farm2	Stable1	Farm2R1S1	Day35	35	20.F2S1.11.07			
## 99	Farm2	Farm2	Stable1	Farm2R1S1	Day35	35	24.F2S1.11.07			
## 100	Farm2	Farm2	Stable2	Farm2R1S2	Day35	35	7.F2S2.11.07			
## 101	Farm2	Farm2	Stable2	Farm2R1S2	Day14	14	12.F2S2.20.06			
## 102	Farm2	Farm2	Stable2	Farm2R1S2	Day14	14	14.F2S2.20.06			
## 103	Farm2	Farm2	Stable2	Farm2R1S2	Day14	14	17.F2S2.20.06			
## 104	Farm2	Farm2	Stable2	Farm2R1S2	Day35	35	10.F2S2.11.07			
## 105	Farm2	Farm2	Stable2	Farm2R1S2	Day35	35	13.F2S2.11.07			
## 106	Farm2	Farm2	Stable2	Farm2R1S2	Day35	35	18.F2S2.11.07			
## 107	Farm2	Farm2	Stable2	Farm2R1S2	Day35	35	19.F2S2.11.07			
## 108	Farm2	Farm2	Stable2	Farm2R1S2	Day35	35	22.F2S2.11.07			
## 109	Farm4	Farm2	Stable1	Farm2R2S1	Day14	14	1.F4.S1.CA.8.8.2017			
## 110	Farm4	Farm2	Stable1	Farm2R2S1	Day14	14	2.F4.S1.CA.8.8.2017			
## 111	Farm4	Farm2	Stable1	Farm2R2S1	Day14	14	3.F4.S1.CA.8.8.2017			
## 112	Farm4	Farm2	Stable1	Farm2R2S1	Day14	14	4.F4.S1.CA.8.8.2017			
## 113	Farm4	Farm2	Stable1	Farm2R2S1	Day14	14	6.F4.S1.CA.8.8.2017			
## 114	Farm4	Farm2	Stable1	Farm2R2S1	Day14	14	7.F4.S1.CA.8.8.2017			
## 115	Farm4	Farm2	Stable2	Farm2R2S2	Day14	14	1.F4.S2.CA.8.8.2017			
## 116	Farm4	Farm2	Stable2	Farm2R2S2	Day14	14	2.F4.S2.CA.8.8.2017			
## 117	Farm4	Farm2	Stable2	Farm2R2S2	Day14	14	3.F4.S2.CA.8.8.2017			
## 118	Farm4	Farm2	Stable2	Farm2R2S2	Day14	14	4.F4.S2.CA.8.8.2017			
## 119	Farm4	Farm2	Stable2	Farm2R2S2	Day14	14	5.F4.S2.CA.8.8.2017			
## 120	Farm4	Farm2	Stable2	Farm2R2S2	Day14	14	6.F4.S2.CA.8.8.2017			
##	Weight	Animal	Gender	Age	ParentStock	Hatchery	Researcher	AB	Abday	FlockSize
## 1	471	F		49	HatcheryC		AK	no	NA	32200
## 2	515	M		49	HatcheryC		AK	yes	2	38400
## 3	501	F		49	HatcheryC		AK	yes	2	38400
## 4	460	F		49	HatcheryC		AK	yes	2	38400
## 5	416	F		49	HatcheryC		AK	yes	2	38400
## 6	543	M		49	HatcheryC		AK	yes	2	38400
## 7	390	M		49	HatcheryC		AK	yes	2	38400
## 8	422	F		54	HatcheryD		AK	no	NA	30500
## 9	472	M		49	HatcheryC		AK	no	NA	32200
## 10	489	M		54	HatcheryD		AK	no	NA	30500
## 11	489	F		54	HatcheryD		AK	no	NA	30500
## 12	506	M		54	HatcheryD		AK	no	NA	30500
## 13	491	F		54	HatcheryD		AK	no	NA	30500
## 14	482	M		54	HatcheryD		AK	no	NA	30500
## 15	488	M		54	HatcheryD		AK	no	NA	39000
## 16	492	F		54	HatcheryD		AK	no	NA	39000
## 17	458	M		49	HatcheryC		AK	no	NA	32200
## 18	487	F		54	HatcheryD		AK	no	NA	39000
## 19	500	F		54	HatcheryD		AK	no	NA	39000
## 20	508	M		54	HatcheryD		AK	no	NA	39000
## 21	540	M		54	HatcheryD		AK	no	NA	39000
## 22	2110	F		49	HatcheryC		AK	no	NA	32200
## 23	485	M		49	HatcheryC		AK	no	NA	32200
## 24	1987	M		49	HatcheryC		AK	no	NA	32200
## 25	2819	F		49	HatcheryC		AK	no	NA	32200
## 26	1957	F		49	HatcheryC		AK	no	NA	32200

## 27	2222	M	49 HatcheryC	AK no	NA	32200
## 28	2182	M	49 HatcheryC	AK no	NA	32200
## 29	1960	F	49 HatcheryC	AK yes	2	38400
## 30	2320	M	49 HatcheryC	AK yes	2	38400
## 31	2308	F	49 HatcheryC	AK yes	2	38400
## 32	2456	M	49 HatcheryC	AK yes	2	38400
## 33	1978	F	49 HatcheryC	AK yes	2	38400
## 34	2298	M	49 HatcheryC	AK yes	2	38400
## 35	2266	M	54 HatcheryD	AK yes	21	30500
## 36	2630	M	54 HatcheryD	AK yes	21	30500
## 37	2233	F	54 HatcheryD	AK yes	21	30500
## 38	2630	M	54 HatcheryD	AK yes	21	30500
## 39	2050	F	54 HatcheryD	AK yes	21	30500
## 40	2150	F	54 HatcheryD	AK yes	21	30500
## 41	1770	F	54 HatcheryD	AK no	NA	39000
## 42	2420	M	54 HatcheryD	AK no	NA	39000
## 43	2436	M	54 HatcheryD	AK no	NA	39000
## 44	2060	F	54 HatcheryD	AK no	NA	39000
## 45	516	F	49 HatcheryC	AK no	NA	32200
## 46	548	M	49 HatcheryC	AK no	NA	32200
## 47	2432	M	54 HatcheryD	AK no	NA	39000
## 48	1883	F	54 HatcheryD	AK no	NA	39000
## 49	2225	F	42 HatcheryB	AK no	NA	23500
## 50	2663	M	42 HatcheryB	AK no	NA	23500
## 51	1674	F	42 HatcheryB	AK no	NA	23500
## 52	2202	F	42 HatcheryB	AK no	NA	23500
## 53	2529	M	42 HatcheryB	AK no	NA	23500
## 54	2651	M	42 HatcheryB	AK no	NA	23500
## 55	2841	M	42 HatcheryB	AK no	NA	23500
## 56	2661	M	42 HatcheryB	AK no	NA	23500
## 57	2717	M	42 HatcheryB	AK no	NA	23500
## 58	2348	F	42 HatcheryB	AK no	NA	23500
## 59	2506	F	42 HatcheryB	AK no	NA	23500
## 60	2005	F	42 HatcheryB	AK no	NA	23500
## 61	456	F	55 HatcheryA	AK no	NA	28000
## 62	568	M	55 HatcheryA	AK no	NA	28000
## 63	525	F	55 HatcheryA	AK no	NA	28000
## 64	408	M	55 HatcheryA	AK no	NA	28000
## 65	561	M	55 HatcheryA	AK no	NA	28000
## 66	520	F	55 HatcheryA	TM no	NA	28000
## 67	580	M	55 HatcheryA	TM no	NA	28000
## 68	497	F	55 HatcheryA	TM no	NA	28000
## 69	496	F	55 HatcheryA	TM no	NA	28000
## 70	608	M	55 HatcheryA	TM no	NA	28000
## 71	499	F	55 HatcheryA	TM no	NA	28000
## 72	1722	F	55 HatcheryA	AK no	NA	28000
## 73	1956	M	55 HatcheryA	AK no	NA	28000
## 74	1881	M	55 HatcheryA	AK no	NA	28000
## 75	2180	F	55 HatcheryA	AK no	NA	28000
## 76	2300	M	55 HatcheryA	AK no	NA	28000
## 77	1665	F	55 HatcheryA	AK no	NA	28000
## 78	2655	M	55 HatcheryA	FV no	NA	28000
## 79	2119	F	55 HatcheryA	AK no	NA	28000
## 80	2089	F	55 HatcheryA	FV no	NA	28000

## 81	1961	M	55 HatcheryA	FV	no	NA	28000
## 82	1972	F	55 HatcheryA	FV	no	NA	28000
## 83	2605	M	55 HatcheryA	FV	no	NA	28000
## 84	586	M	35 HatcheryB	AM	no	NA	23500
## 85	565	M	35 HatcheryB	AK	no	NA	23500
## 86	504	F	35 HatcheryB	AK	no	NA	23500
## 87	514	M	35 HatcheryB	AK	no	NA	23500
## 88	502	F	35 HatcheryB	AM	no	NA	23500
## 89	517	F	35 HatcheryB	AK	no	NA	23500
## 90	1957	F	35 HatcheryB	KR	no	NA	23500
## 91	2708	M	35 HatcheryB	AK	no	NA	23500
## 92	2479	M	35 HatcheryB	AK	no	NA	23500
## 93	2072	F	35 HatcheryB	KR	no	NA	23500
## 94	513	M	55 HatcheryA	TM	no	NA	28000
## 95	548	M	35 HatcheryB	AK	no	NA	23500
## 96	554	F	35 HatcheryB	AK	no	NA	23500
## 97	430	F	35 HatcheryB	AM	no	NA	23500
## 98	2457	F	35 HatcheryB	AK	no	NA	23500
## 99	2998	M	35 HatcheryB	KR	no	NA	23500
## 100	2736	M	35 HatcheryB	AK	no	NA	23500
## 101	529	F	35 HatcheryB	AM	no	NA	23500
## 102	569	M	35 HatcheryB	AM	no	NA	23500
## 103	536	M	35 HatcheryB	AK	no	NA	23500
## 104	2563	M	35 HatcheryB	KR	no	NA	23500
## 105	2663	M	35 HatcheryB	AK	no	NA	23500
## 106	2303	F	35 HatcheryB	KR	no	NA	23500
## 107	2197	F	35 HatcheryB	AK	no	NA	23500
## 108	2374	M	35 HatcheryB	AK	no	NA	23500
## 109	581	M	42 HatcheryB	AK	no	NA	23500
## 110	525	F	42 HatcheryB	AK	no	NA	23500
## 111	541	M	42 HatcheryB	AK	no	NA	23500
## 112	568	M	42 HatcheryB	AK	no	NA	23500
## 113	531	F	42 HatcheryB	AK	no	NA	23500
## 114	513	F	42 HatcheryB	AK	no	NA	23500
## 115	508	F	42 HatcheryB	AK	no	NA	23500
## 116	599	M	42 HatcheryB	AK	no	NA	23500
## 117	502	M	42 HatcheryB	AK	no	NA	23500
## 118	570	F	42 HatcheryB	AK	no	NA	23500
## 119	338	M	42 HatcheryB	AK	no	NA	23500
## 120	552	F	42 HatcheryB	AK	no	NA	23500
##	FeedF	FeedType	FeedProducent	Cox	OPG	Cluster	LitterType
## 1	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 2	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 3	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 4	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 5	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 6	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 7	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 8	F6.F2	GrowFeed1	SupplierB	Sacox	NA	2	Strawpellets
## 9	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 10	F6.F2	GrowFeed1	SupplierB	Sacox	NA	2	Strawpellets
## 11	F6.F2	GrowFeed1	SupplierB	Sacox	NA	2	Strawpellets
## 12	F6.F2	GrowFeed1	SupplierB	Sacox	NA	2	Strawpellets
## 13	F6.F2	GrowFeed1	SupplierB	Sacox	NA	2	Strawpellets

## 14	F6.F2	GrowFeed1	SupplierB	Sacox	NA	1	Strawpellets
## 15	F6.F2	GrowFeed1	SupplierB	Sacox	NA	2	Strawpellets
## 16	F6.F2	GrowFeed1	SupplierB	Sacox	NA	1	Strawpellets
## 17	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 18	F6.F2	GrowFeed1	SupplierB	Sacox	NA	1	Strawpellets
## 19	F6.F2	GrowFeed1	SupplierB	Sacox	NA	1	Strawpellets
## 20	F6.F2	GrowFeed1	SupplierB	Sacox	NA	2	Strawpellets
## 21	F6.F2	GrowFeed1	SupplierB	Sacox	NA	2	Strawpellets
## 22	F5.F4	FinalFeed	SupplierC	None	111	1	Strawpellets
## 23	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 24	F5.F4	FinalFeed	SupplierC	None	111	1	Strawpellets
## 25	F5.F4	FinalFeed	SupplierC	None	111	1	Strawpellets
## 26	F5.F4	FinalFeed	SupplierC	None	111	1	Strawpellets
## 27	F5.F4	FinalFeed	SupplierC	None	111	1	Strawpellets
## 28	F5.F4	FinalFeed	SupplierC	None	111	1	Strawpellets
## 29	F5.F4	FinalFeed	SupplierC	None	92213	1	Strawpellets
## 30	F5.F4	FinalFeed	SupplierC	None	92213	1	Strawpellets
## 31	F5.F4	FinalFeed	SupplierC	None	92213	1	Strawpellets
## 32	F5.F4	FinalFeed	SupplierC	None	92213	1	Strawpellets
## 33	F5.F4	FinalFeed	SupplierC	None	92213	1	Strawpellets
## 34	F5.F4	FinalFeed	SupplierC	None	92213	1	Strawpellets
## 35	F6.F3	FinalFeed	SupplierB	None	98879	1	Strawpellets
## 36	F6.F3	FinalFeed	SupplierB	None	98879	1	Strawpellets
## 37	F6.F3	FinalFeed	SupplierB	None	98879	1	Strawpellets
## 38	F6.F3	FinalFeed	SupplierB	None	98879	1	Strawpellets
## 39	F6.F3	FinalFeed	SupplierB	None	98879	1	Strawpellets
## 40	F6.F3	FinalFeed	SupplierB	None	98879	1	Strawpellets
## 41	F6.F3	FinalFeed	SupplierB	None	101101	1	Strawpellets
## 42	F6.F3	FinalFeed	SupplierB	None	101101	1	Strawpellets
## 43	F6.F3	FinalFeed	SupplierB	None	101101	1	Strawpellets
## 44	F6.F3	FinalFeed	SupplierB	None	101101	1	Strawpellets
## 45	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 46	F5.F2	GrowFeed1	SupplierC	Maxiban	NA	1	Strawpellets
## 47	F6.F3	FinalFeed	SupplierB	None	101101	1	Strawpellets
## 48	F6.F3	FinalFeed	SupplierB	None	101101	1	Strawpellets
## 49	F2.F4	FinalFeed	SupplierB	None	68771	1	Peat
## 50	F2.F4	FinalFeed	SupplierB	None	68771	2	Peat
## 51	F2.F4	FinalFeed	SupplierB	None	68771	1	Peat
## 52	F2.F4	FinalFeed	SupplierB	None	68771	1	Peat
## 53	F2.F4	FinalFeed	SupplierB	None	68771	1	Peat
## 54	F2.F4	FinalFeed	SupplierB	None	68771	1	Peat
## 55	F2.F4	FinalFeed	SupplierB	None	107767	1	Peat
## 56	F2.F4	FinalFeed	SupplierB	None	107767	1	Peat
## 57	F2.F4	FinalFeed	SupplierB	None	107767	1	Peat
## 58	F2.F4	FinalFeed	SupplierB	None	107767	1	Peat
## 59	F2.F4	FinalFeed	SupplierB	None	107767	1	Peat
## 60	F2.F4	FinalFeed	SupplierB	None	107767	1	Peat
## 61	F1.F2	GrowFeed1	SupplierA	Maxiban	0	1	Woodshaving
## 62	F1.F2	GrowFeed1	SupplierA	Maxiban	0	1	Woodshaving
## 63	F1.F2	GrowFeed1	SupplierA	Maxiban	0	2	Woodshaving
## 64	F1.F2	GrowFeed1	SupplierA	Maxiban	0	1	Woodshaving
## 65	F1.F2	GrowFeed1	SupplierA	Maxiban	0	1	Woodshaving
## 66	F1.F2	GrowFeed1	SupplierA	Maxiban	0	2	Woodshaving
## 67	F1.F2	GrowFeed1	SupplierA	Maxiban	0	1	Woodshaving

```

## 68 F1.F2 GrowFeed1 SupplierA Maxiban 0 2 Woodshaving
## 69 F1.F2 GrowFeed1 SupplierA Maxiban 0 1 Woodshaving
## 70 F1.F2 GrowFeed1 SupplierA Maxiban 0 2 Woodshaving
## 71 F1.F2 GrowFeed1 SupplierA Maxiban 0 2 Woodshaving
## 72 F1.F4 FinalFeed SupplierA Monteban 12110 1 Woodshaving
## 73 F1.F4 FinalFeed SupplierA Monteban 12110 1 Woodshaving
## 74 F1.F4 FinalFeed SupplierA Monteban 12110 1 Woodshaving
## 75 F1.F4 FinalFeed SupplierA Monteban 12110 1 Woodshaving
## 76 F1.F4 FinalFeed SupplierA Monteban 12110 1 Woodshaving
## 77 F1.F4 FinalFeed SupplierA Monteban 12110 1 Woodshaving
## 78 F1.F4 FinalFeed SupplierA Monteban 112322 1 Woodshaving
## 79 F1.F4 FinalFeed SupplierA Monteban 112322 2 Woodshaving
## 80 F1.F4 FinalFeed SupplierA Monteban 112322 1 Woodshaving
## 81 F1.F4 FinalFeed SupplierA Monteban 112322 1 Woodshaving
## 82 F1.F4 FinalFeed SupplierA Monteban 112322 1 Woodshaving
## 83 F1.F4 FinalFeed SupplierA Monteban 112322 1 Woodshaving
## 84 F2.F2 GrowFeed1 SupplierB Maxiban 0 1 Peat
## 85 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 86 F2.F2 GrowFeed1 SupplierB Maxiban 0 1 Peat
## 87 F2.F2 GrowFeed1 SupplierB Maxiban 0 1 Peat
## 88 F2.F2 GrowFeed1 SupplierB Maxiban 0 1 Peat
## 89 F2.F2 GrowFeed1 SupplierB Maxiban 0 1 Peat
## 90 F2.F4 FinalFeed SupplierB None 77326 1 Peat
## 91 F2.F4 FinalFeed SupplierB None 77326 1 Peat
## 92 F2.F4 FinalFeed SupplierB None 77326 1 Peat
## 93 F2.F4 FinalFeed SupplierB None 77326 1 Peat
## 94 F1.F2 GrowFeed1 SupplierA Monteban 0 2 Woodshaving
## 95 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 96 F2.F2 GrowFeed1 SupplierB Maxiban 0 1 Peat
## 97 F2.F2 GrowFeed1 SupplierB Maxiban 0 1 Peat
## 98 F2.F4 FinalFeed SupplierB None 77326 1 Peat
## 99 F2.F4 FinalFeed SupplierB None 77326 1 Peat
## 100 F2.F4 FinalFeed SupplierB None 1555 1 Peat
## 101 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 102 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 103 F2.F2 GrowFeed1 SupplierB Maxiban 0 1 Peat
## 104 F2.F4 FinalFeed SupplierB None 1555 1 Peat
## 105 F2.F4 FinalFeed SupplierB None 1555 1 Peat
## 106 F2.F4 FinalFeed SupplierB None 1555 1 Peat
## 107 F2.F4 FinalFeed SupplierB None 1555 1 Peat
## 108 F2.F4 FinalFeed SupplierB None 1555 1 Peat
## 109 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 110 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 111 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 112 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 113 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 114 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 115 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 116 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 117 F2.F2 GrowFeed1 SupplierB Maxiban 0 1 Peat
## 118 F2.F2 GrowFeed1 SupplierB Maxiban 0 1 Peat
## 119 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
## 120 F2.F2 GrowFeed1 SupplierB Maxiban 0 2 Peat
##     Metagenomics ReadPerc ReadTot Stables PAM_clust

```

## 1	yes	13.44	1662806	Stable7	1
## 2	yes	13.37	1567066	Stable8	1
## 3	yes	17.04	1668626	Stable8	1
## 4	yes	12.83	1454542	Stable8	1
## 5	yes	13.62	1747744	Stable8	1
## 6	yes	14.91	2136942	Stable8	1
## 7	yes	16.28	1703404	Stable8	1
## 8	yes	19.55	2154168	Stable9	2
## 9	yes	16.72	3945097	Stable7	2
## 10	yes	26.18	2064718	Stable9	1
## 11	yes	14.46	2139188	Stable9	2
## 12	yes	21.88	2184513	Stable9	2
## 13	yes	18.67	2012514	Stable9	2
## 14	yes	24.20	2384128	Stable9	2
## 15	yes	12.20	1223856	Stable10	2
## 16	yes	23.87	2780666	Stable10	2
## 17	yes	19.74	2424890	Stable7	2
## 18	yes	13.99	1677382	Stable10	2
## 19	yes	12.47	1411064	Stable10	2
## 20	yes	18.54	2034267	Stable10	2
## 21	yes	13.83	4051027	Stable10	2
## 22	yes	13.79	3369118	Stable7	2
## 23	yes	10.91	1016067	Stable7	1
## 24	yes	18.66	5137505	Stable7	2
## 25	yes	11.70	2578713	Stable7	2
## 26	yes	12.99	3499270	Stable7	2
## 27	yes	11.66	2813795	Stable7	2
## 28	yes	10.64	2676779	Stable7	2
## 29	yes	10.64	2589994	Stable8	2
## 30	yes	10.73	2901855	Stable8	2
## 31	yes	10.05	2635070	Stable8	2
## 32	yes	11.93	2889482	Stable8	2
## 33	yes	16.32	3250744	Stable8	2
## 34	yes	12.65	3556716	Stable8	2
## 35	yes	18.11	5135439	Stable9	1
## 36	yes	16.64	4221198	Stable9	1
## 37	yes	25.33	6528583	Stable9	2
## 38	yes	14.10	3568621	Stable9	2
## 39	yes	18.56	5538190	Stable9	2
## 40	yes	15.76	3388263	Stable9	2
## 41	yes	14.38	3798960	Stable10	2
## 42	yes	18.30	1612234	Stable10	2
## 43	yes	20.58	1741029	Stable10	2
## 44	yes	21.90	2327327	Stable10	2
## 45	yes	13.96	1383070	Stable7	1
## 46	yes	12.87	1591294	Stable7	1
## 47	yes	18.55	1821741	Stable10	1
## 48	yes	18.69	1786656	Stable10	2
## 49	yes	11.16	1283157	Stable5	1
## 50	yes	11.87	3140951	Stable5	2
## 51	yes	13.21	1231876	Stable5	1
## 52	yes	11.65	1151790	Stable5	1
## 53	yes	12.57	1184840	Stable5	1
## 54	yes	12.36	1390633	Stable5	1

## 55	yes	11.55	1336765	Stable6	1
## 56	yes	10.91	1060836	Stable6	1
## 57	yes	12.43	1300671	Stable6	1
## 58	yes	21.25	2229969	Stable6	1
## 59	yes	14.72	4003979	Stable6	1
## 60	yes	13.57	3115534	Stable6	1
## 61	yes	8.23	1557432	Stable1	2
## 62	yes	14.17	3612753	Stable1	2
## 63	yes	13.95	4750422	Stable1	2
## 64	yes	12.99	2896229	Stable1	2
## 65	yes	14.67	3430725	Stable1	2
## 66	yes	14.70	3278076	Stable1	2
## 67	yes	16.06	3801686	Stable2	2
## 68	yes	13.18	3277649	Stable2	2
## 69	yes	18.47	4377753	Stable2	2
## 70	yes	17.27	5318096	Stable2	2
## 71	yes	15.62	1507738	Stable2	2
## 72	yes	15.68	4941732	Stable1	2
## 73	yes	20.09	5204030	Stable1	2
## 74	yes	19.68	1688928	Stable1	2
## 75	yes	13.82	3043306	Stable1	2
## 76	yes	16.72	5149789	Stable1	2
## 77	yes	12.90	2703844	Stable1	2
## 78	yes	13.59	4714831	Stable2	2
## 79	yes	55.90	6211649	Stable2	2
## 80	yes	20.94	5877051	Stable2	2
## 81	yes	14.07	4761024	Stable2	2
## 82	yes	12.34	3612417	Stable2	2
## 83	yes	10.90	4109934	Stable2	2
## 84	yes	13.82	3706286	Stable3	2
## 85	yes	21.10	5948176	Stable3	1
## 86	yes	11.72	3274554	Stable3	1
## 87	yes	14.90	4109367	Stable3	1
## 88	yes	12.42	3570790	Stable3	1
## 89	yes	11.63	2901594	Stable3	1
## 90	yes	12.27	4232553	Stable3	1
## 91	yes	22.02	5297224	Stable3	1
## 92	yes	12.22	3077281	Stable3	1
## 93	yes	15.31	3609169	Stable3	1
## 94	yes	19.73	4809921	Stable2	2
## 95	yes	14.87	4455669	Stable4	1
## 96	yes	11.19	3393401	Stable4	1
## 97	yes	11.16	2678330	Stable4	2
## 98	yes	11.73	2893488	Stable3	1
## 99	yes	17.99	4023523	Stable3	1
## 100	yes	14.92	3819432	Stable4	1
## 101	yes	15.84	4635450	Stable4	1
## 102	yes	16.48	3568806	Stable4	1
## 103	yes	13.60	4736433	Stable4	1
## 104	yes	15.80	3321072	Stable4	2
## 105	yes	12.79	1394348	Stable4	1
## 106	yes	12.89	1151808	Stable4	1
## 107	yes	17.96	6299555	Stable4	1
## 108	yes	11.48	2037978	Stable4	1

```

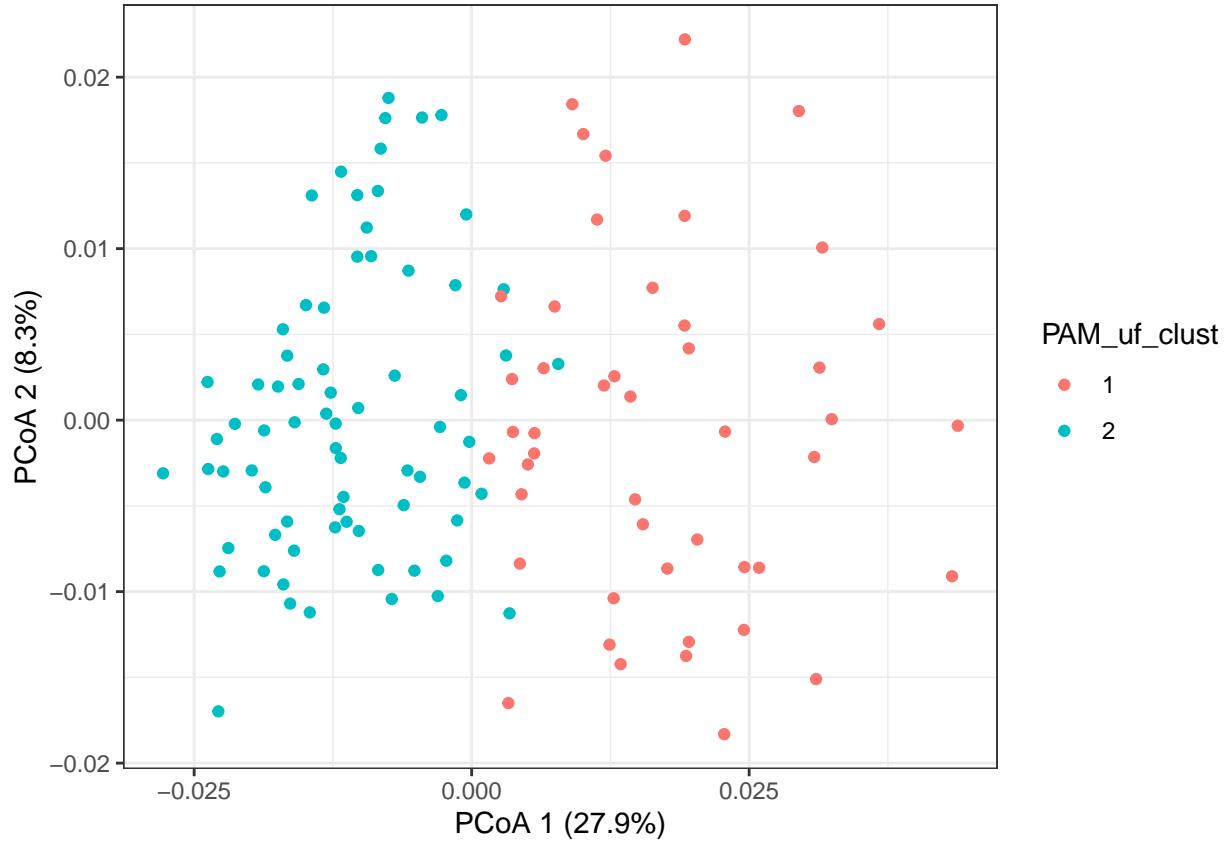
## 109      yes  21.17 5963566 Stable5      2
## 110      yes  23.76 4260260 Stable5      2
## 111      yes  11.64 3473215 Stable5      2
## 112      yes  16.35 3412142 Stable5      2
## 113      yes  16.98 5113248 Stable5      2
## 114      yes  21.18 4060597 Stable5      2
## 115      yes  11.86 3644560 Stable6      2
## 116      yes  12.81 1097955 Stable6      2
## 117      yes  14.02 1150637 Stable6      2
## 118      yes  14.23 1118009 Stable6      2
## 119      yes  13.49 1442372 Stable6      2
## 120      yes  12.52 1120381 Stable6      2

# Add cluster information
clusters <- factor(pam_list[[which.max(sil_width)]]$clustering)
pcoa_df <- merge(pcoa_df, clusters, by.x = "Sample", by.y = "row.names")
colnames(pcoa_df)[ncol(pcoa_df)] <- "PAM_uf_clust"

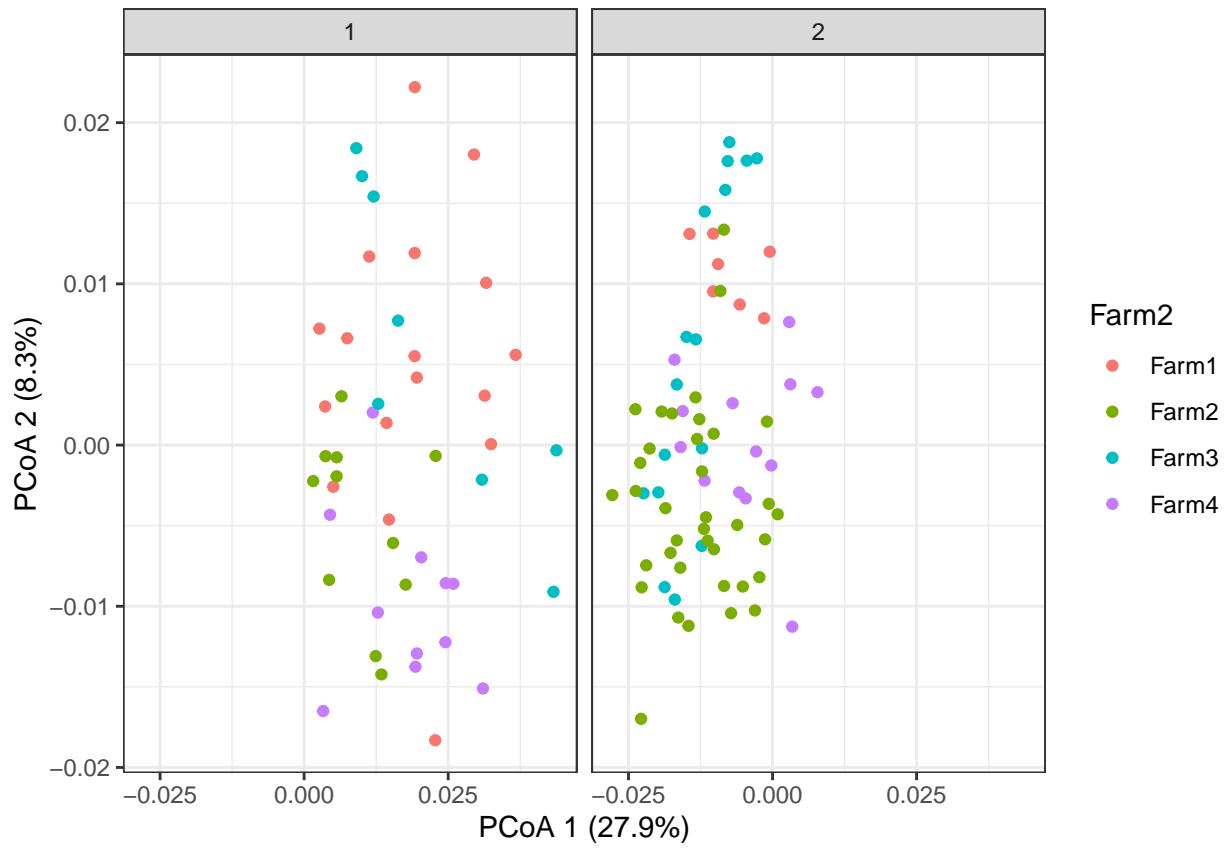
# Variance explained
ve <- pcoa_data$eig/sum(pcoa_data$eig)

# Plot
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = PAM_uf_clust)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)"))

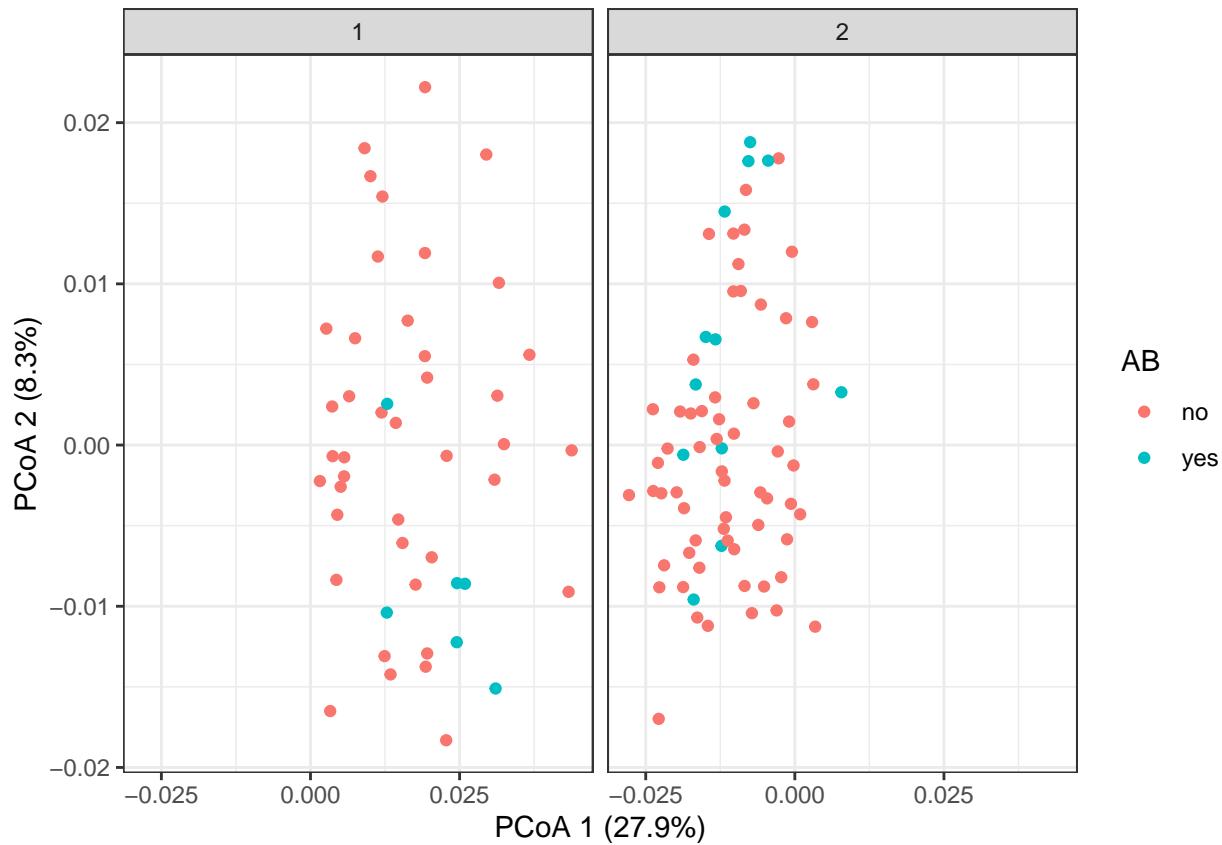
```



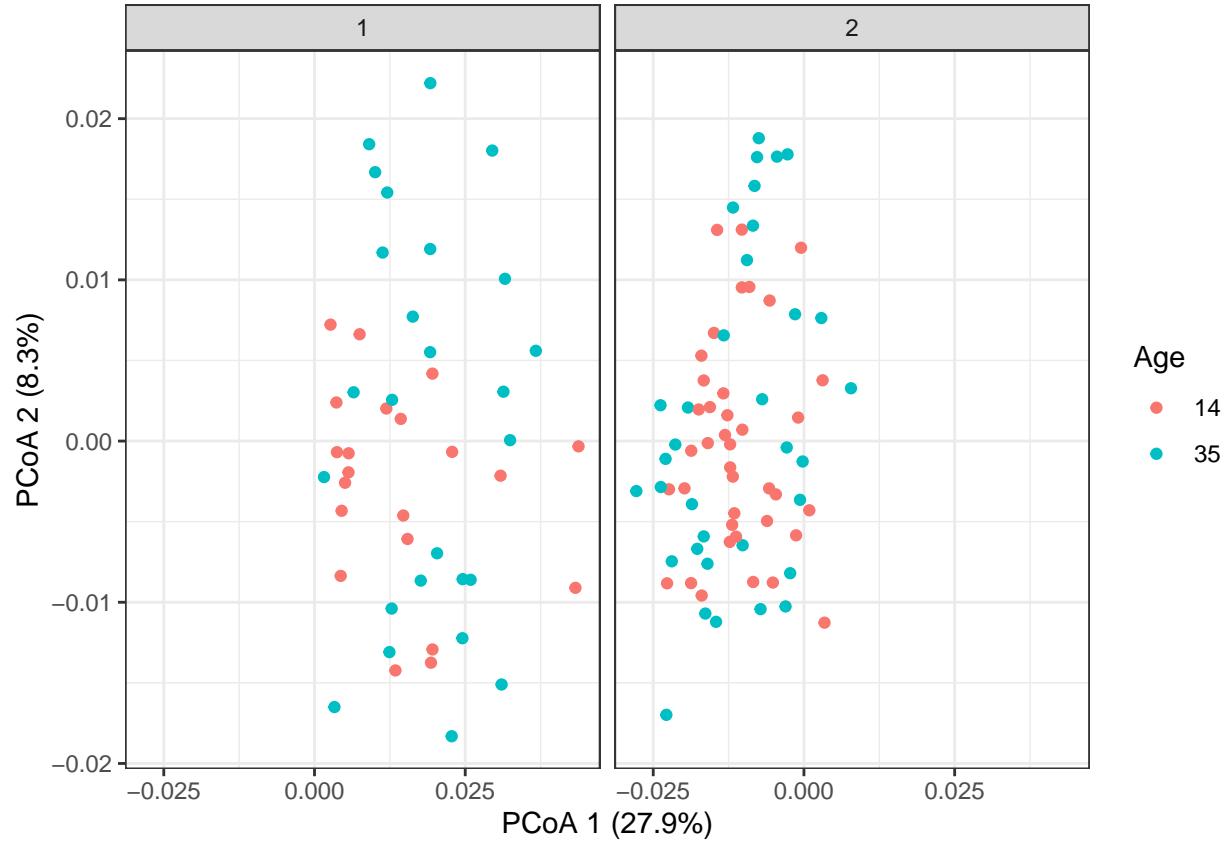
```
# facet by clusters and colour by farm
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = Farm2)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM_uf_clust)
```



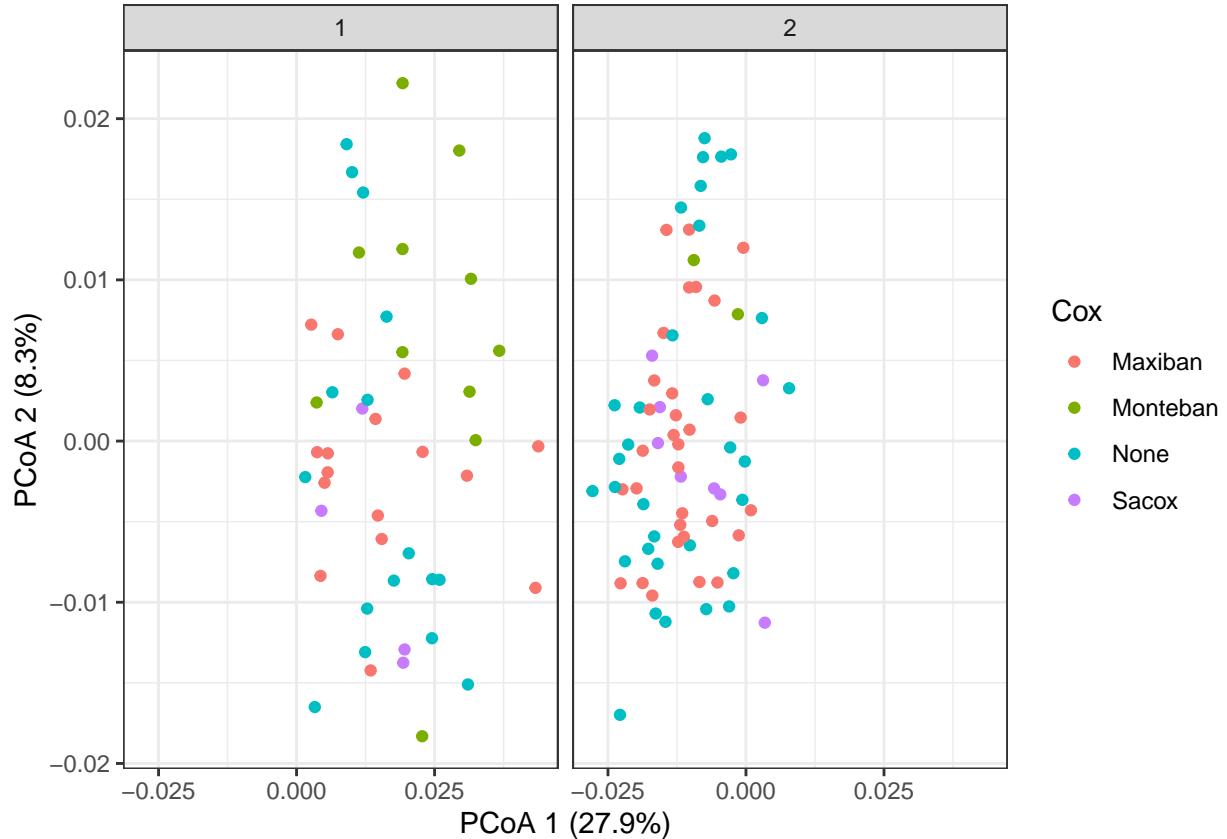
```
# facet by clusters and colour by AB
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = AB)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (", round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (", round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM_uf_clust)
```



```
# facet by clusters and colour by Age
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = Age)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (" ,round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (" ,round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM_uf_clust)
```



```
# facet by clusters and colour by Agent
ggplot(pcoa_df, aes(x = PC1, y = PC2, color = Cox)) +
  theme_bw() +
  geom_point() +
  xlab(paste0("PCoA 1 (" , round(ve[1]*100,1), "%)")) +
  ylab(paste0("PCoA 2 (" , round(ve[2]*100,1), "%)")) +
  facet_wrap(~PAM_uf_clust)
```



PCoA for Phylum data, BC with DMM, euclidian (tse_dmn can be set to different taxonomic level)

```
dmn_group <- calculateDMNgroup(tse_dmn,
                                variable = "Age", assay.type = "counts",
                                k = 2, seed = .Machine$integer.max)

dmn_group <- calculateDMNgroup(tse_dmn,
                                variable = "Farm2", assay.type = "counts",
                                k = 2, seed = .Machine$integer.max)
dmn_group <- calculateDMNgroup(tse_dmn,
                                variable = "AB", assay.type = "counts",
                                k = 2, seed = .Machine$integer.max)
dmn_group
```

```
## class: DMNGroup
## summary:
##      k samples taxa      NLE     LogDet     Laplace      BIC      AIC
## no   2      102   14 4965.8738 109.39160 4993.9204 5032.9359 4994.8738
## yes  2       18   14  905.6743  60.15666  909.1034  947.5847  934.6743
```

```
DirichletMultinomial::mixturewt(getBestDMNFit(tse_dmn)) # measure weights
```

```
##      pi      theta
## 1 0.6184112 68.11193
## 2 0.3815888 79.34569
```

```

head(DirichletMultinomial::mixture(getBestDMNFit(tse_dmn))) # sample-cluster assignment probabilities

##          [,1]      [,2]
## 10_1  3.174022e-03 0.9968260
## 10_10 7.765007e-04 0.9992235
## 10_11 5.052618e-02 0.9494738
## 10_12 5.708221e-09 1.0000000
## 10_13 2.822658e-01 0.7177342
## 10_14 2.100166e-06 0.9999979

head(DirichletMultinomial::fitted(getBestDMNFit(tse_dmn))) # taxa contribution

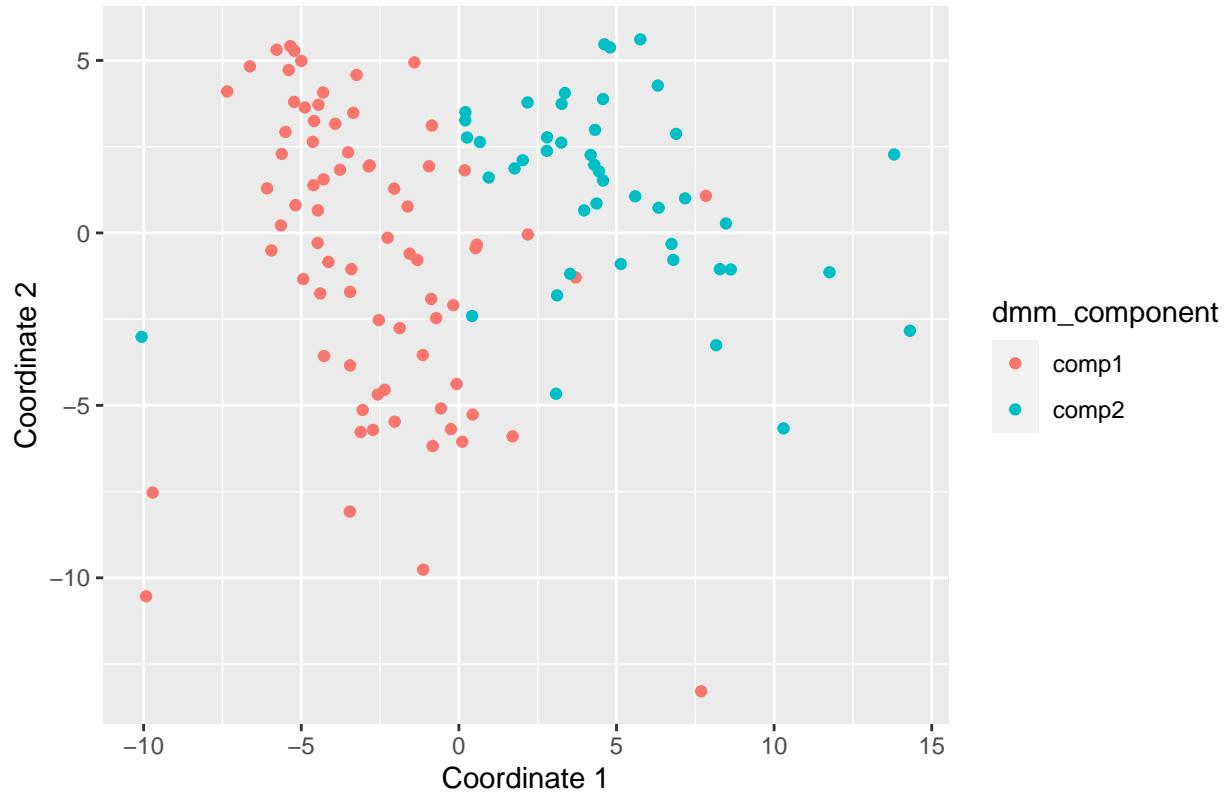
##          [,1]      [,2]
## Phylum:Beta-lactam    0.2937271 0.2589380
## Phylum:Aminoglycoside 18.6352086 13.4812404
## Phylum:Not determined 0.4246984 0.2477229
## Phylum:Tetracycline   34.6943099 59.2030324
## Phylum:Trimethoprim   0.3952342 0.2015072
## Phylum:Lincosamide    5.2963892 2.0300996

prob <- DirichletMultinomial::mixture(getBestDMNFit(tse_dmn))
colnames(prob) <- c("comp1", "comp2")
vec <- colnames(prob)[max.col(prob, ties.method = "first")]
assay(tse, "pseudo") <- assay(tse, "counts") + 1
tse <- transformCounts(tse, assay.type = "pseudo", method = "relabundance")
tse <- transformCounts(tse, "relabundance", method = "clr")
df <- calculateMDS(tse, assay.type = "clr", method = "euclidean")
euclidean_pcoa_df <- data.frame(
  pcoa1 = df[, 1],
  pcoa2 = df[, 2])
euclidean_dmm_pcoa_df <- cbind(euclidean_pcoa_df,
                                   dmm_component = vec)

ggplot(data = euclidean_dmm_pcoa_df, aes(x = pcoa1, y = pcoa2, color = dmm_component)) +
  geom_point() +
  labs(x = "Coordinate 1",
       y = "Coordinate 2",
       title = "PCoA with Aitchison distances")

```

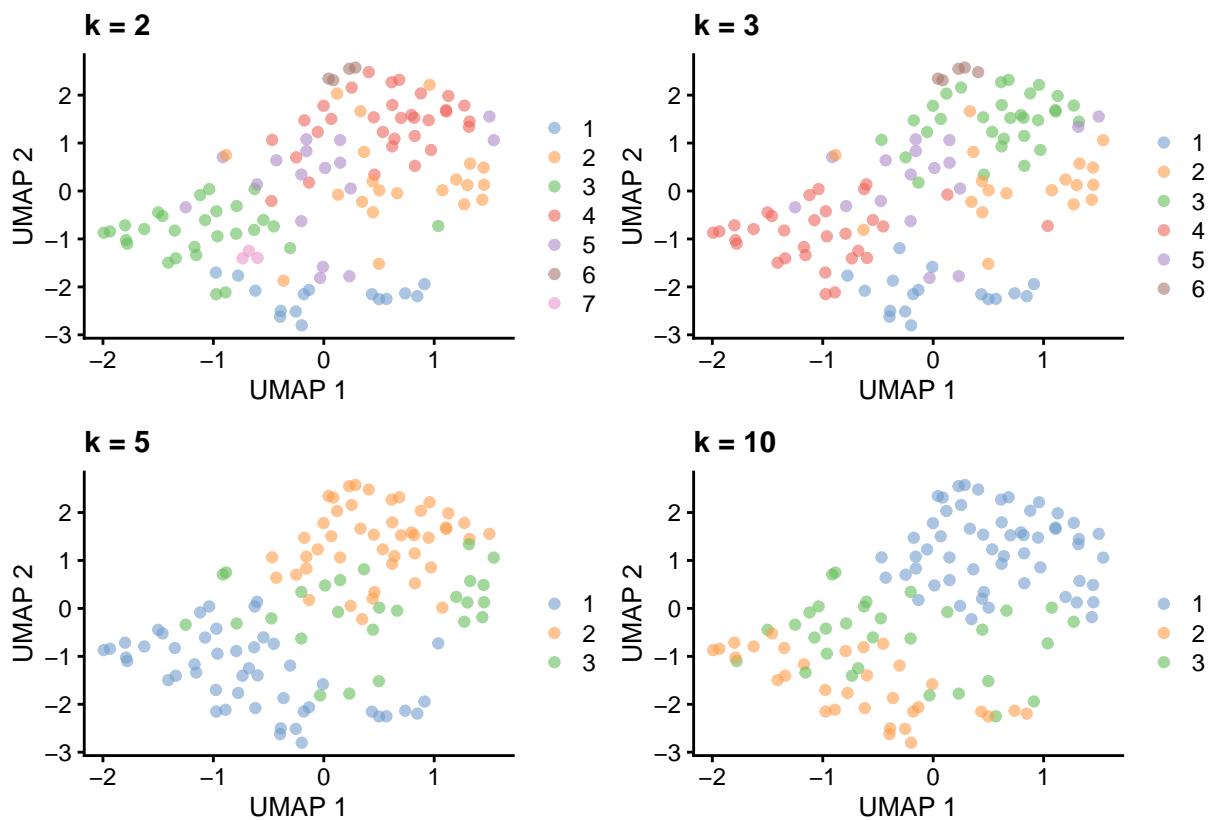
PCoA with Aitchison distances



UMAP with different ks

```
tse = makeTreeSummarizedExperimentFromPhyloseq(Rps_copy)
tse <- transformCounts(tse, method = "rclr")
tse <- runUMAP(tse, name = "UMAP", assay.type = "rclr")
k <- c(2, 3, 5, 10)
ClustAndPlot <- function(x) {
  # Creating the graph and running the short random walks algorithm
  graph_clusters <- clusterRows(t(assays(tse)$rclr), NNGraphParam(k = x))

  # Results of the clustering as a color for each sample
  plotUMAP(tse, colour_by = I(graph_clusters)) +
    labs(title = paste0("k = ", x))
}
plots <- lapply(k, ClustAndPlot)
(plots[[1]] + plots[[2]]) / (plots[[3]] + plots[[4]])
```



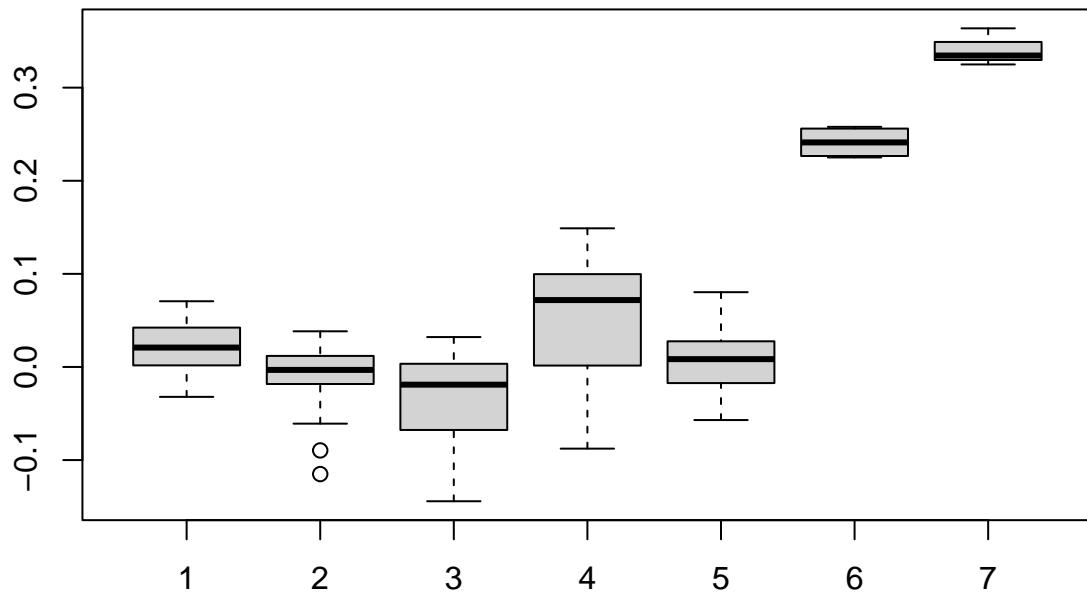
```
# boxplots
ClustDiagPlot <- function(x) {
  # Getting the clustering results
  graph_clusters <- clusterRows(t(assays(tse)$rclr), NNGraphParam(k = x))

  # Computing the diagnostic info
  sil <- approxSilhouette(t(assays(tse)$rclr), graph_clusters)

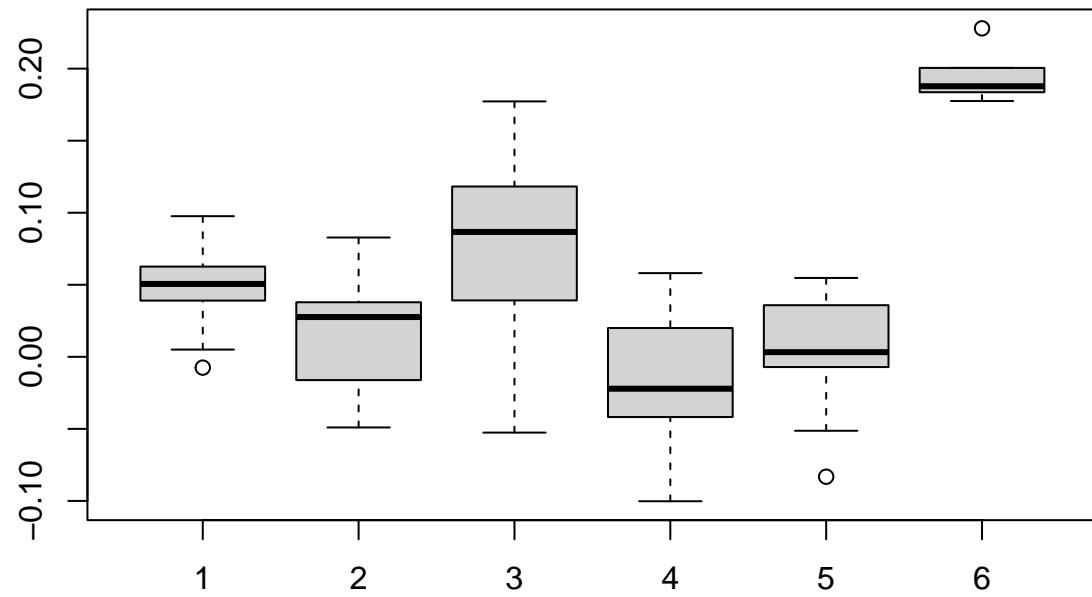
  # Plotting as a boxplot to observe cluster separation
  boxplot(split(sil$width, graph_clusters), main = paste0("k = ", x))
}

# Applying the function for different k values
res <- lapply(k, ClustDiagPlot)
```

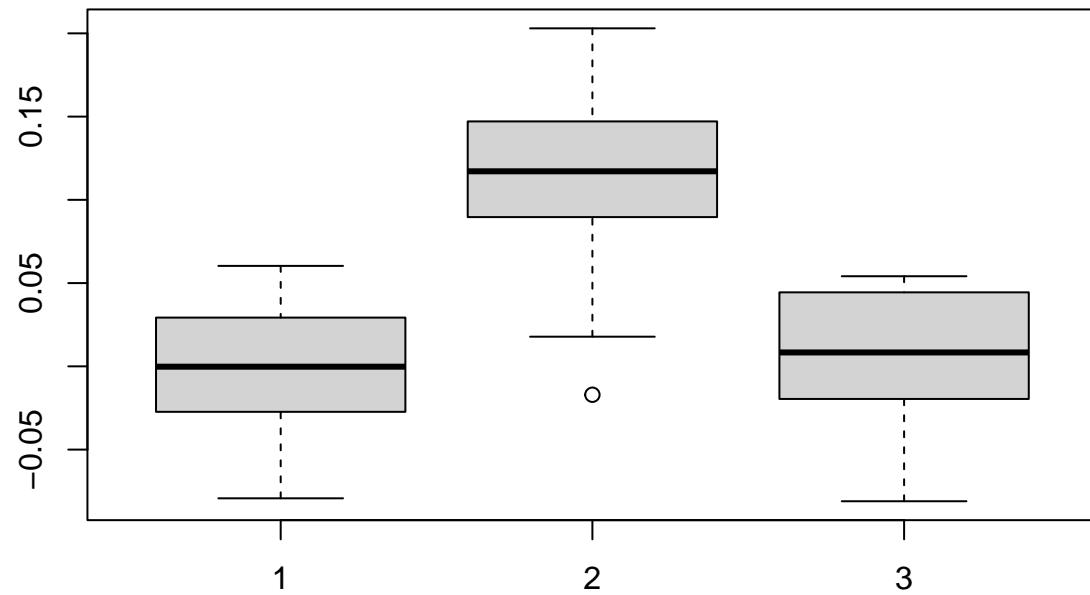
k = 2



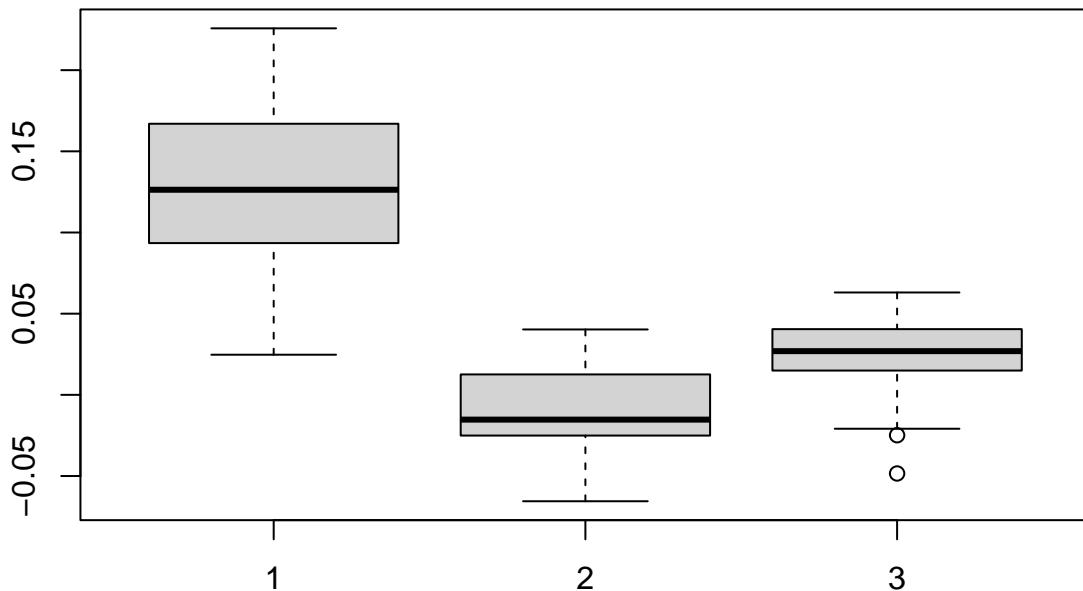
k = 3



k = 5



k = 10

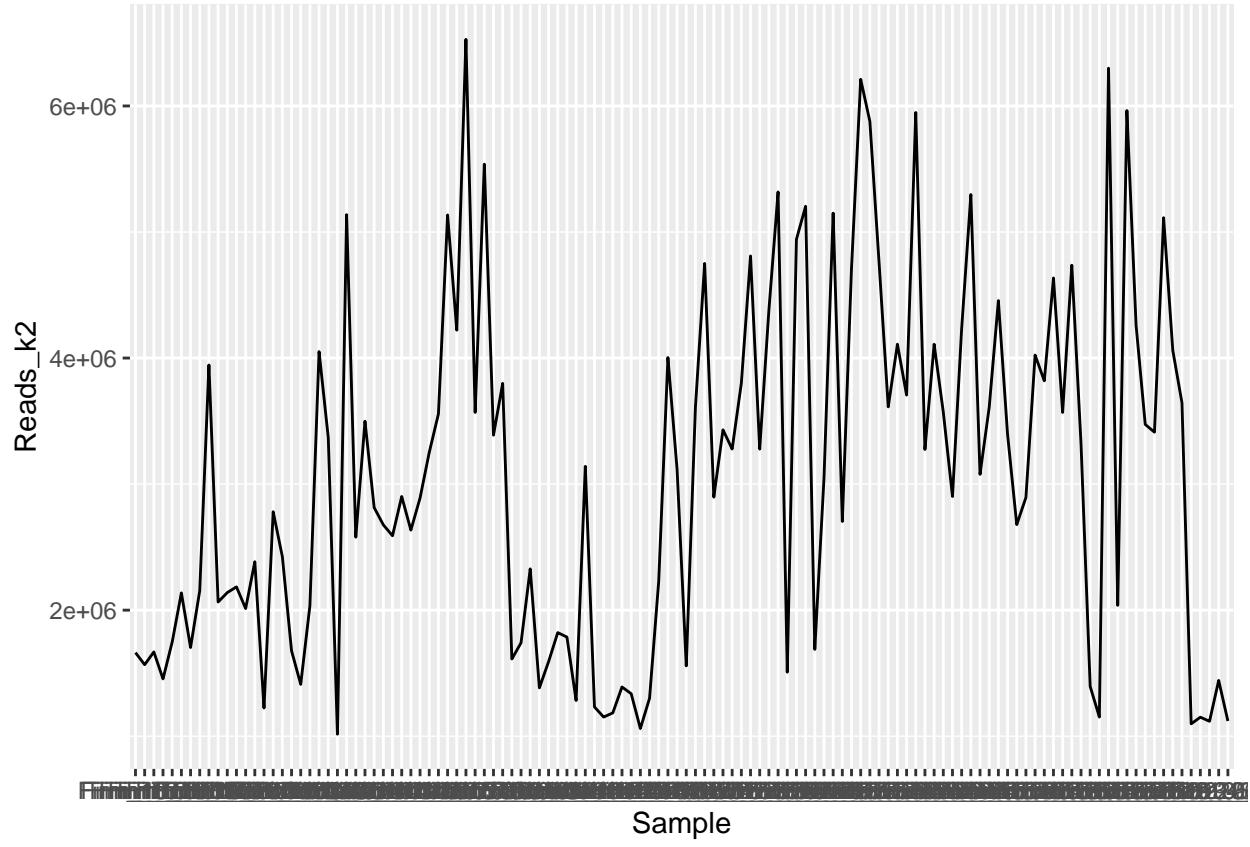


Full comparison of Kraken 2 and MetaPhlAn

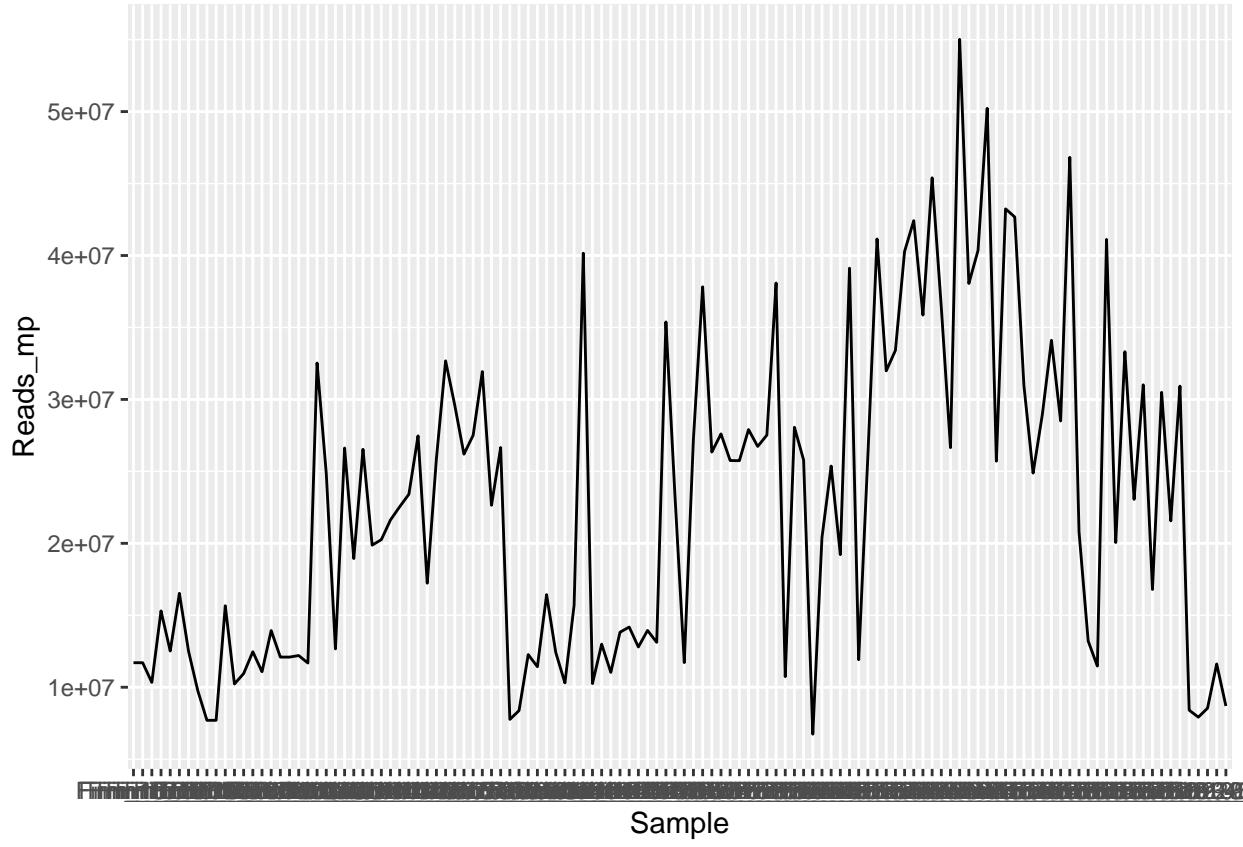
```
reads_k2 = read.table("total_mapped_reads_kraken2.tab", sep = "\t", header = FALSE)
reads_mp = read.table("total_mapped_reads_metaphlan.tab", sep = "\t", header = FALSE)
colnames(reads_k2) <- c("Sample", "Reads_k2")
colnames(reads_mp) <- c("Sample", "Reads_mp")

mapped_reads = dplyr::left_join(reads_k2, reads_mp, by="Sample")

ggplot(mapped_reads, aes(x = Sample, y = Reads_k2, group = 1)) + geom_line()
```



```
ggplot(mapped_reads, aes(x = Sample, y = Reads_mp, group = 1)) + geom_line()
```



```
t.test(reads_k2$Reads_k2, y = reads_mp$Reads_mp, paired = TRUE)
```

```
##  
##  Paired t-test  
##  
## data:  reads_k2$Reads_k2 and reads_mp$Reads_mp  
## t = -21.055, df = 119, p-value < 2.2e-16  
## alternative hypothesis: true mean difference is not equal to 0  
## 95 percent confidence interval:  
## -21521420 -17821457  
## sample estimates:  
## mean difference  
##          -19671439
```

```
mapped_reads$ratio = mapped_reads$Reads_k2 / mapped_reads$Reads_mp  
print(mapped_reads$ratio)
```

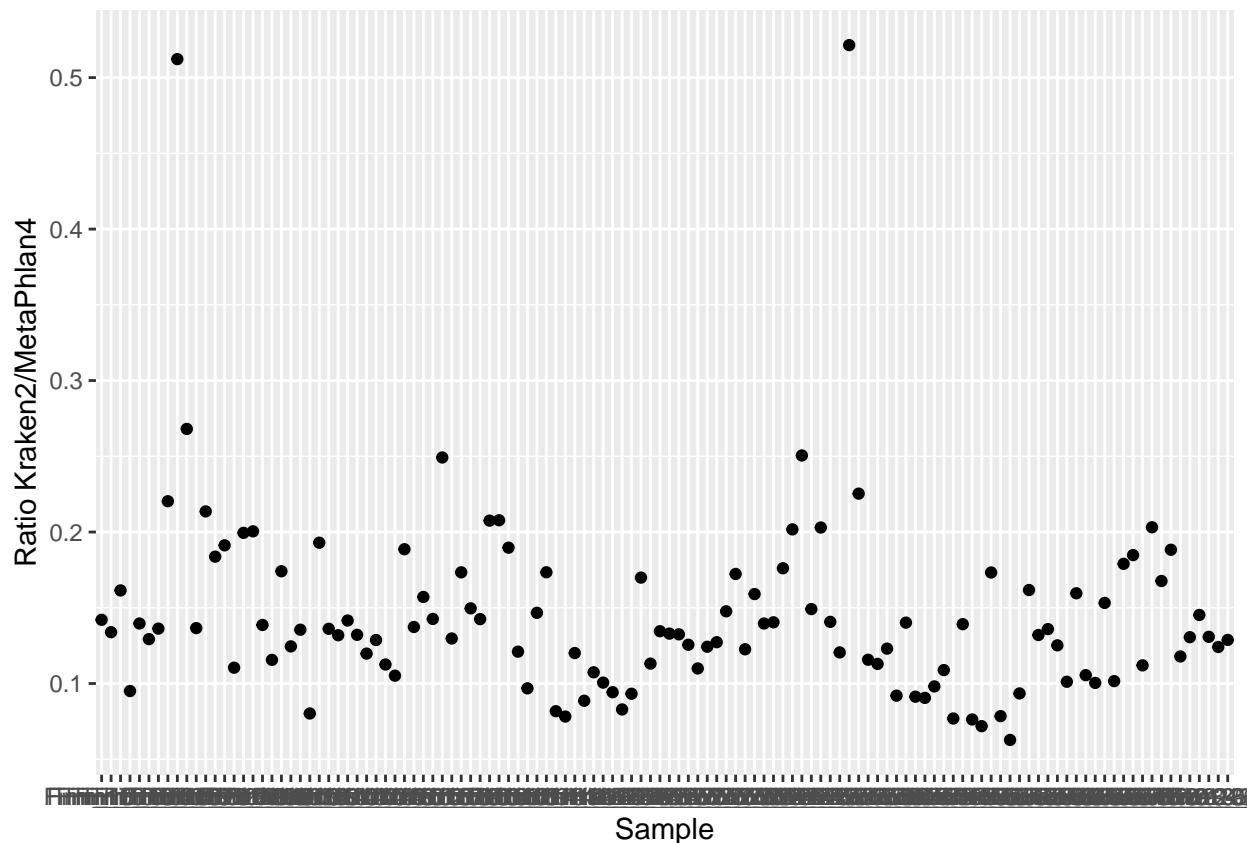
```
## [1] 0.13388931 0.16145344 0.09503424 0.13962290 0.12930943 0.13625680  
## [7] 0.22033884 0.14206928 0.26806822 0.13658300 0.21362082 0.18375995  
## [13] 0.19122746 0.11046122 0.19946427 0.51220318 0.13866611 0.11562124  
## [19] 0.17411403 0.12453024 0.13552939 0.20046124 0.19297084 0.13612535  
## [25] 0.13187980 0.14159567 0.13215316 0.11975510 0.12872011 0.08023767  
## [31] 0.11255947 0.10519333 0.18865355 0.13732116 0.15714201 0.14263595  
## [37] 0.24922515 0.12972898 0.17341579 0.14961961 0.14251771 0.20754219
```

```

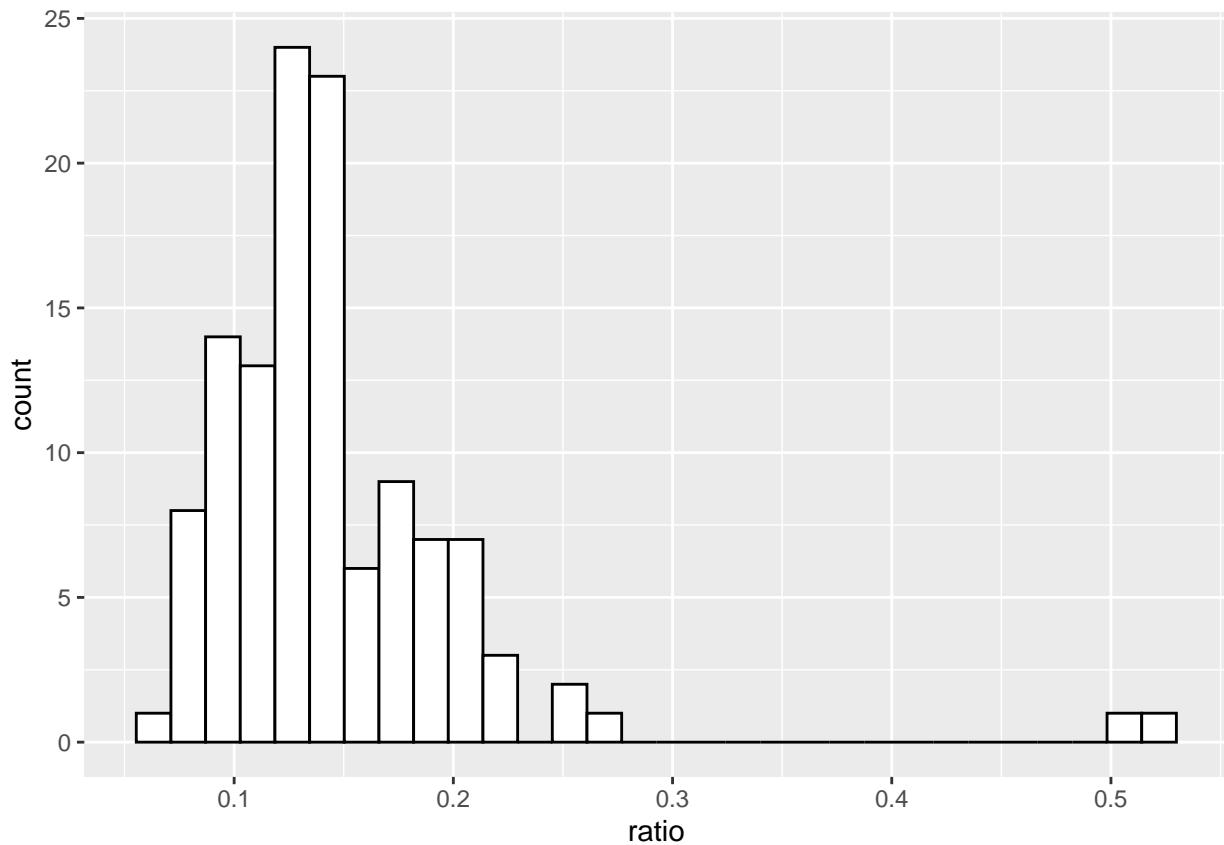
## [43] 0.20777959 0.18968406 0.12104243 0.09682082 0.14669207 0.17345196
## [49] 0.08173042 0.07820878 0.12011718 0.08863443 0.10740829 0.10059399
## [55] 0.09426055 0.08288753 0.09325292 0.16992961 0.11315995 0.13454060
## [61] 0.13294942 0.13246895 0.12559685 0.10995297 0.12428694 0.12727975
## [67] 0.14767614 0.17236200 0.12260245 0.15906699 0.13962649 0.14040581
## [73] 0.17608552 0.20175144 0.25055662 0.14914360 0.20297812 0.14071703
## [79] 0.12052654 0.52143468 0.22536379 0.11570317 0.11294362 0.12302942
## [85] 0.09199905 0.14021749 0.09131010 0.09051779 0.09808928 0.10887689
## [91] 0.07692750 0.13919600 0.07626631 0.07185770 0.17328895 0.07847888
## [97] 0.06275841 0.09344922 0.16172379 0.13204308 0.13591007 0.12520825
## [103] 0.10116236 0.15957307 0.10553484 0.10042689 0.15321317 0.10159465
## [109] 0.17900004 0.18477522 0.11198318 0.20313838 0.16769023 0.18832729
## [115] 0.11786187 0.13057510 0.14525410 0.13084379 0.12413730 0.12878225

```

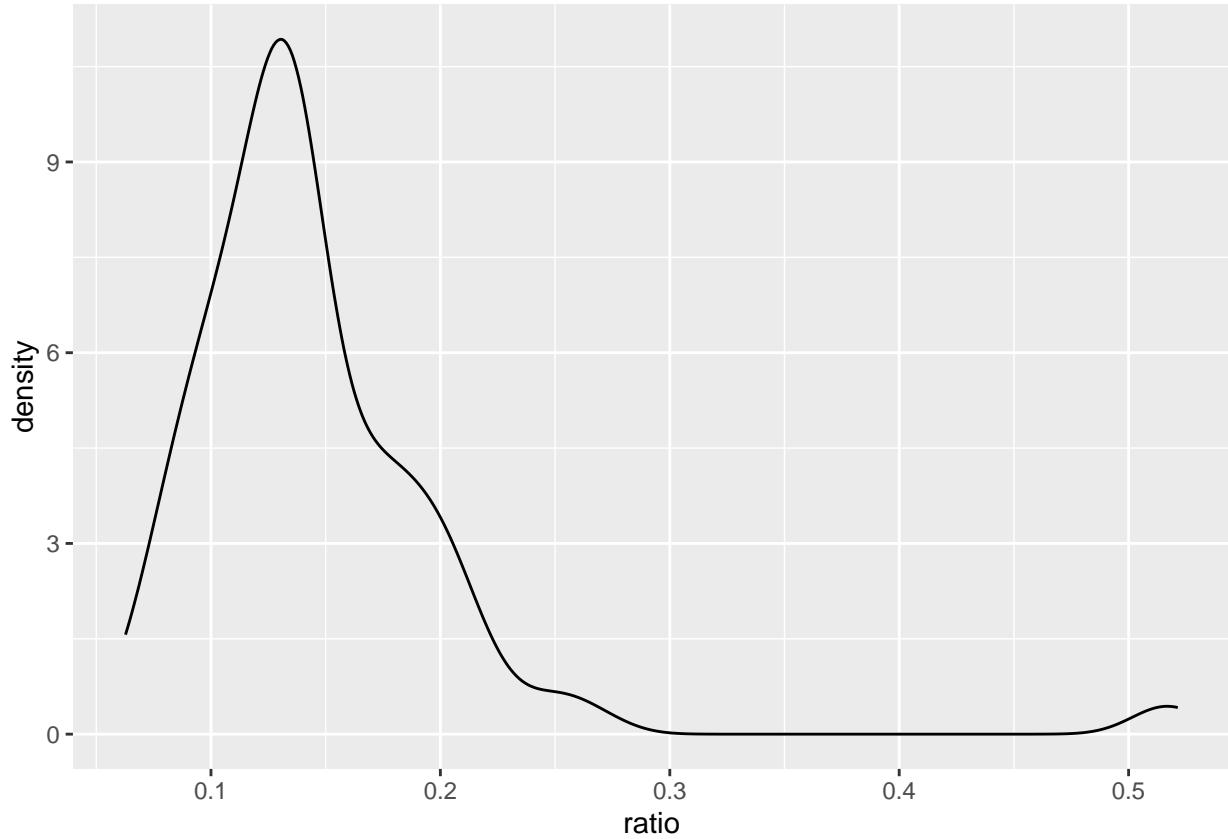
```
ggplot(mapped_reads, aes(x = Sample, y = ratio)) + geom_point() + ylab("Ratio Kraken2/MetaPhlan4")
```



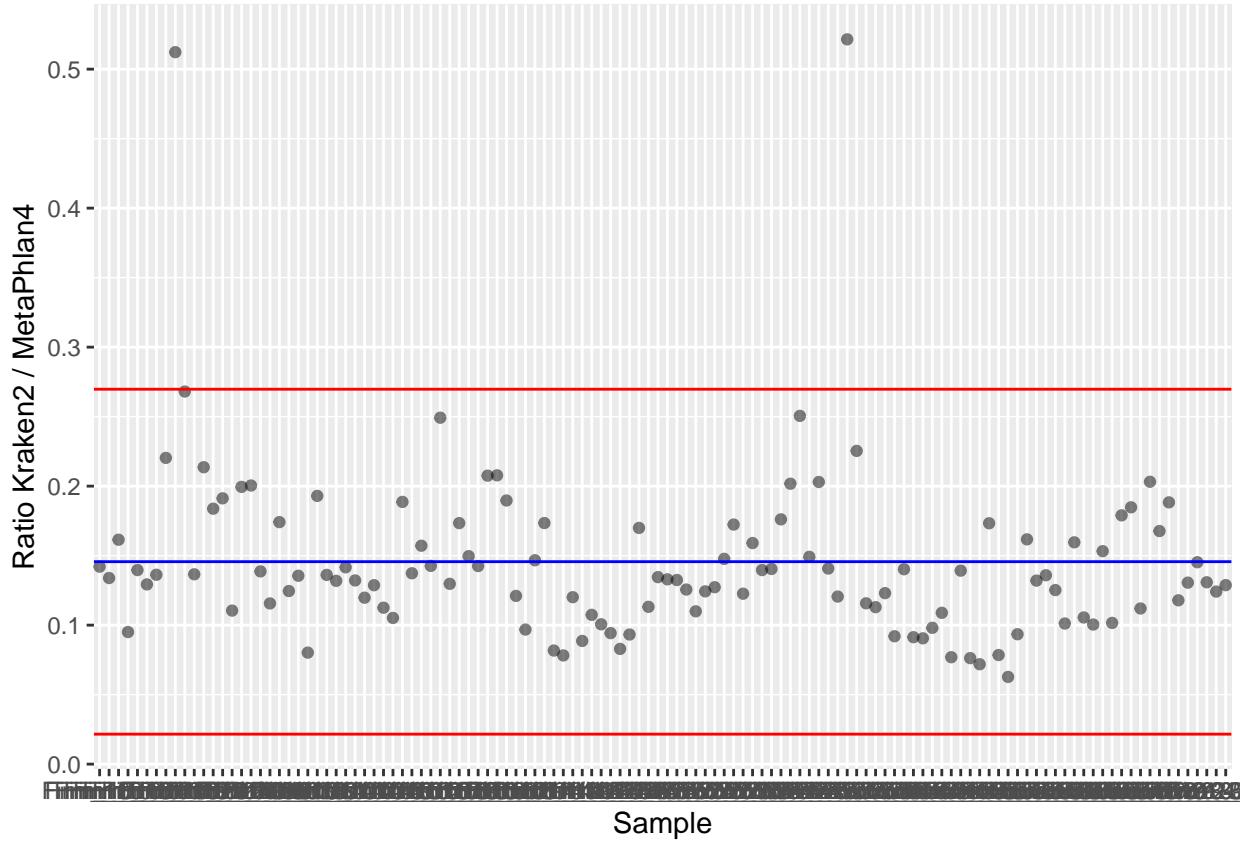
```
ggplot(mapped_reads, aes(x = ratio)) + geom_histogram(color="black", fill="white")
```



```
ggplot(mapped_reads, aes(x = ratio)) + geom_density()
```



```
# bland - altman plot
ggplot(mapped_reads, aes(x = Sample, y = ratio, label = Sample)) +
  geom_point(alpha = 0.5) +
  geom_hline(yintercept = mean(mapped_reads$ratio), colour = "blue", size = 0.5) +
  geom_hline(yintercept = mean(mapped_reads$ratio) - (1.96 * sd(mapped_reads$ratio)), colour = "red", s
  geom_hline(yintercept = mean(mapped_reads$ratio) + (1.96 * sd(mapped_reads$ratio)), colour = "red", s
  ylab("Ratio Kraken2 / MetaPhlan4")
```



```
## Procrustes plots
```

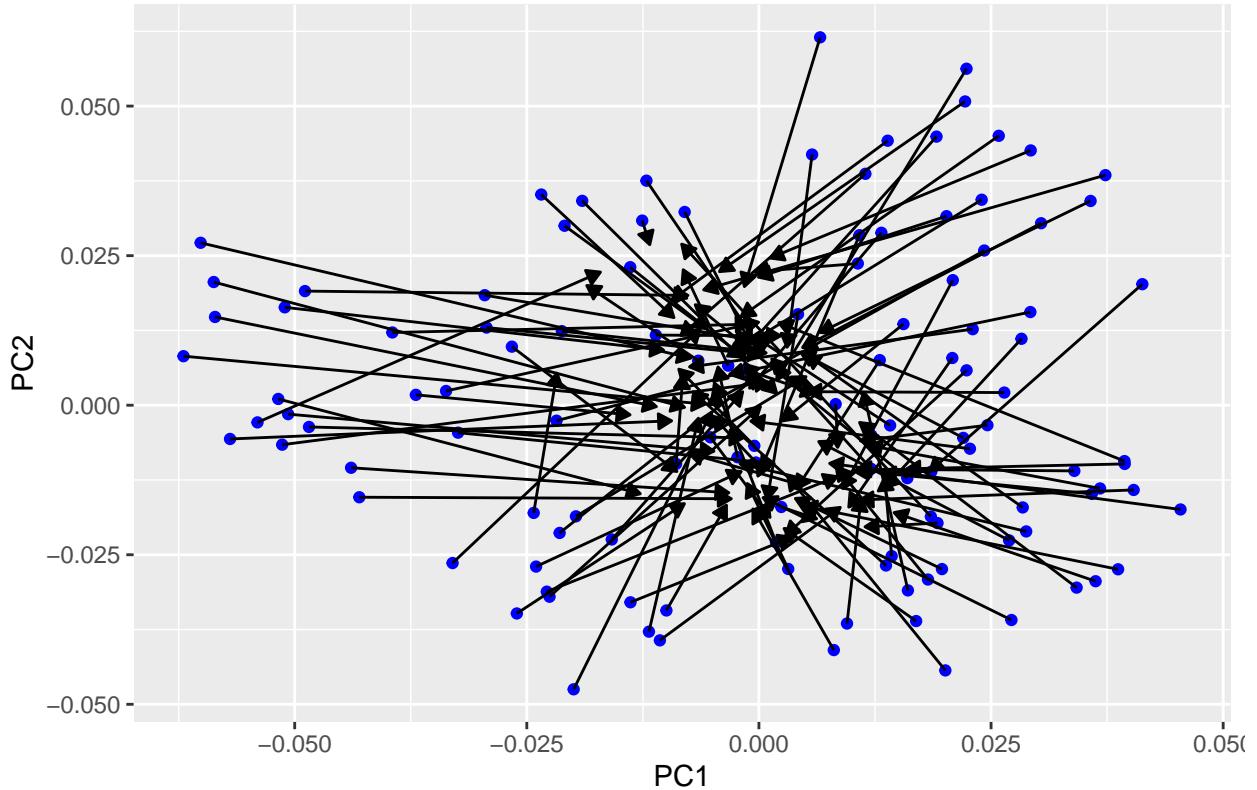
```
#Procrustes analyses
copy16S = subset16S
# get the samples in the same order
sample_names(copy16S) = sample_names(subsetMG)

PCoA_BC_16s = ordinate(copy16S, "PCoA")
PCoA_BC_MG = ordinate(subsetMG, "PCoA")
procrustes = protest(PCoA_BC_16s$vectors, PCoA_BC_MG$vectors)

plot_data <- data.frame(
  MT_PC1 = procrustes$X[, 1],
  MT_PC2 = procrustes$X[, 2],
  MG_PC1 = procrustes$Yrot[, 1],
  MG_PC2 = procrustes$Yrot[, 2])

# with arrows pointing from MG to 16S
ggplot(plot_data) +
  geom_point(aes(x=MT_PC1, y=MT_PC2), color = "blue") +
  geom_segment(aes(x=MT_PC1,y=MT_PC2,xend=MG_PC1,yend=MG_PC2),arrow=arrow(type = "closed", length=unit(10, "mm")))
  labs(title = "Procrustes Plot metagenomic vs. metataxonomic", x = "PC1", y = "PC2") +
  scale_color_manual(values = c("16S" = "black", "MG" = "blue"))
```

Procrustes Plot metagenomic vs. metataxonomic



```

# plot with both points
#ggplot(plot_data) +
#  geom_point(aes(x=MT_PC1, y=MT_PC2), color = "green") +
#  geom_point(aes(x=MG_PC1, y=MG_PC2), color = "blue") +
#  geom_segment(aes(x = MT_PC1, xend = MG_PC1, y = MT_PC2, yend = MG_PC2), linetype = "solid") +
#  labs(title = "Procrustes Plot Metataxonomic vs metagenomic")

# Resistome vs MG
copyRps = Rps
# get the samples in the same order
sample_names(copyRps) = sample_names(subsetMG)

PCoA_BC_Rps = ordinate(copyRps, "PCoA")
PCoA_BC_MG = ordinate(subsetMG, "PCoA")
procrustes = protest(PCoA_BC_Rps$vectors, PCoA_BC_MG$vectors)

plot_data <- data.frame(
  R_PC1 = procrustes$X[, 1],
  R_PC2 = procrustes$X[, 2],
  MG_PC1 = procrustes$Yrot[, 1],
  MG_PC2 = procrustes$Yrot[, 2])

# resistome (k2) points to MG
ggplot(plot_data) +
  geom_point(aes(x=R_PC1, y=R_PC2), color = "blue") +
  geom_segment(aes(x=R_PC1,y=R_PC2,xend=MG_PC1,yend=MG_PC2),arrow=arrow(type = "closed", length=unit(0.1, "cm")))

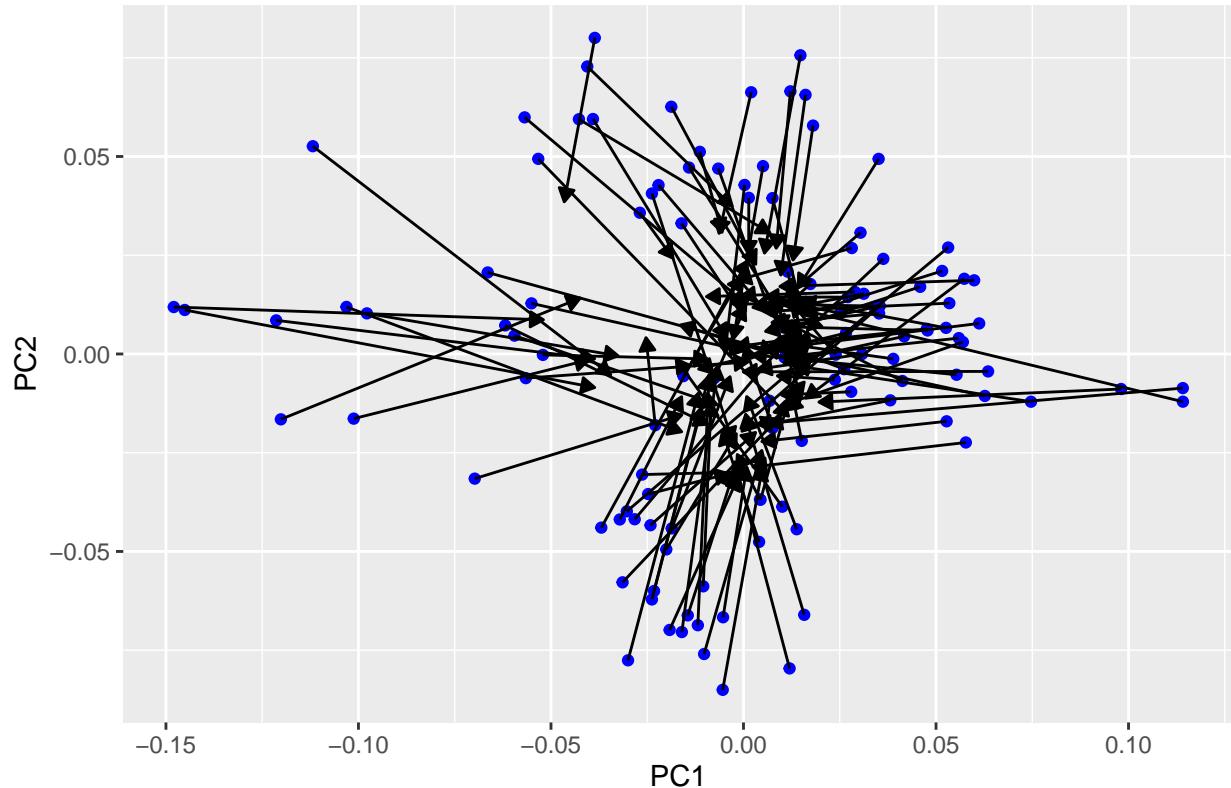
```

```

scale_color_manual(values = c("16S" = "black", "MG" = "blue")) +
guides(color = guide_legend(title = "Data Type")) +
labs(title = "Procrustes Plot resistomic vs. metagenomic", x = "PC1", y = "PC2")

```

Procrustes Plot resistomic vs. metagenomic

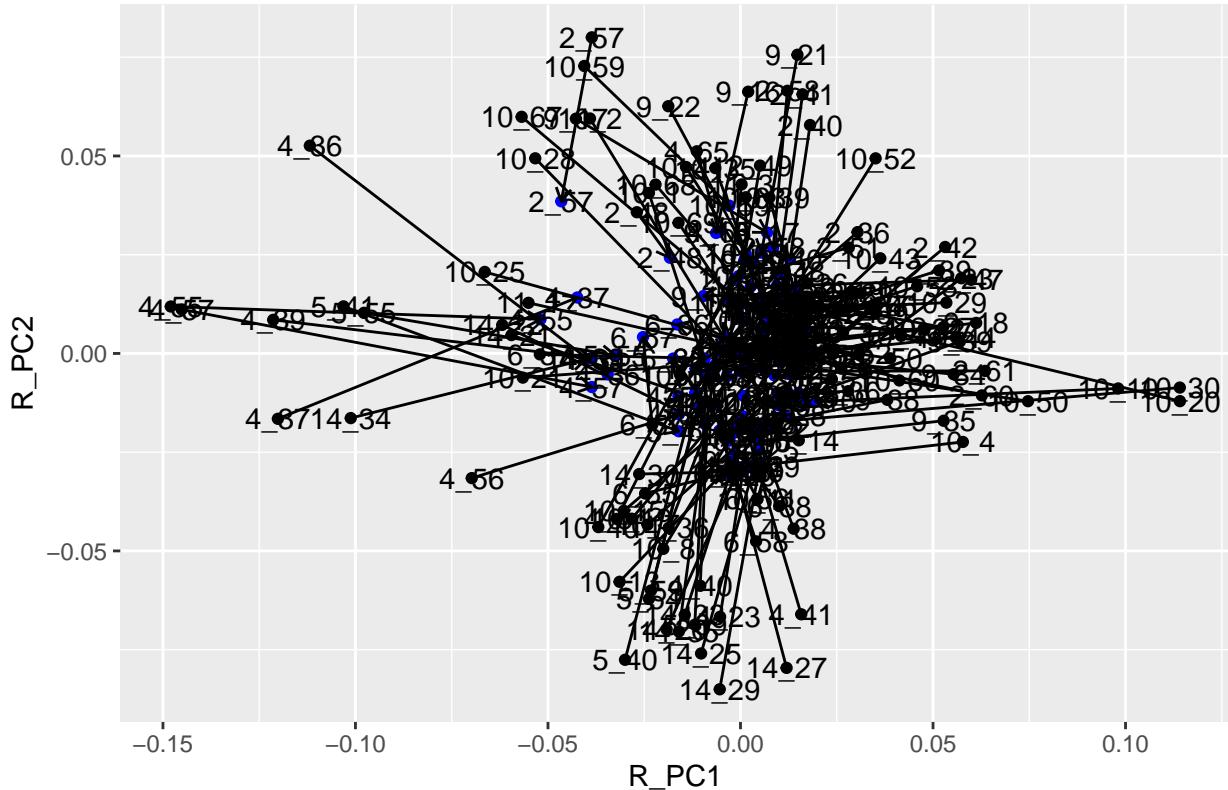


```

# adds labels to see if samples line up
ggplot(plot_data) +
  geom_point(aes(x=R_PC1, y=R_PC2)) +
  geom_point(aes(x=MG_PC1, y=MG_PC2), color = "blue")+
  geom_segment(aes(x=R_PC1,y=R_PC2,xend=MG_PC1,yend=MG_PC2),arrow=arrow(length=unit(0.2,"cm")))+ 
  geom_text(aes(x = MG_PC1, y = MG_PC2, label = rownames(plot_data))) +
  geom_text(aes(x = R_PC1, y = R_PC2, label = rownames(plot_data))) +
  labs(title = "Procrustes Plot")

```

Procrustes Plot



```

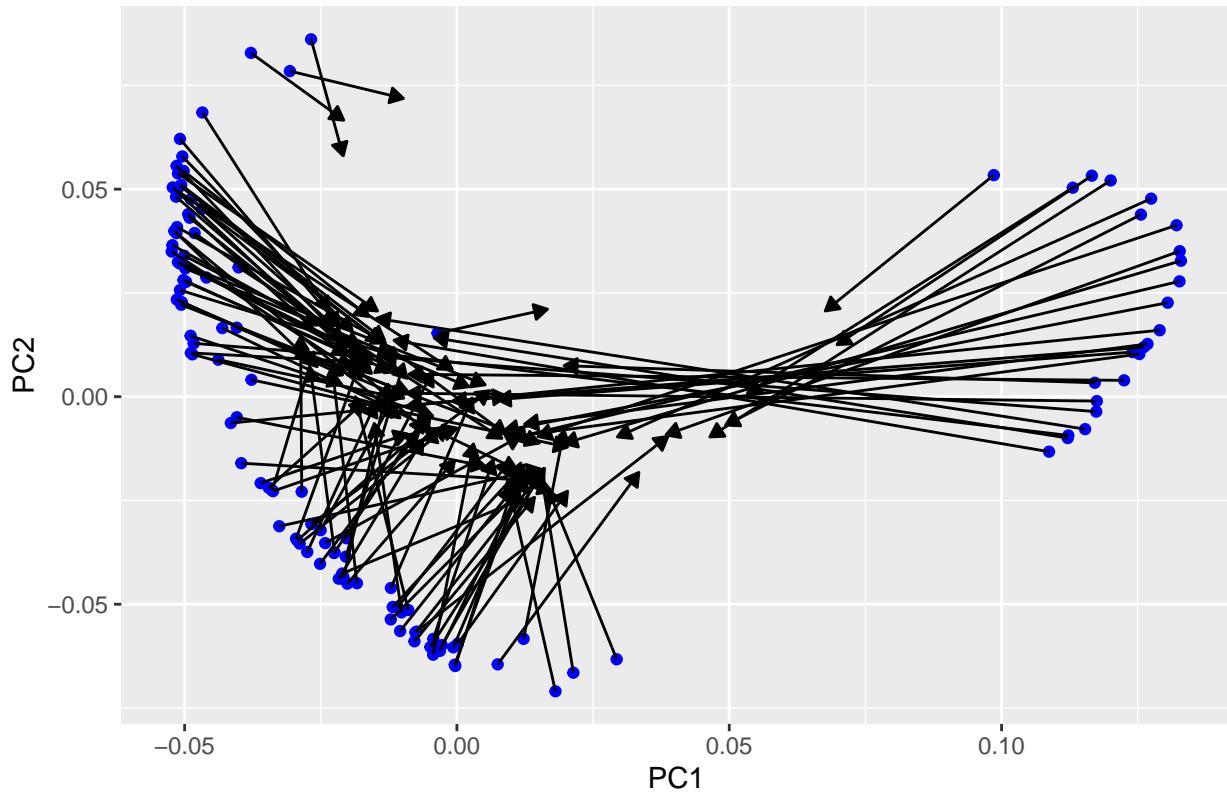
# MP vs k2
PCoA_BC_MP = ordinate(Rps_mp, "PCoA")
PCoA_BC_k2 = ordinate(Rps, "PCoA")
procrustes = protest(PCoA_BC_MP$vectors, PCoA_BC_k2$vectors)

plot_data <- data.frame(
  MP_PC1 = procrustes$X[, 1],
  MP_PC2 = procrustes$X[, 2],
  k2_PC1 = procrustes$Yrot[, 1],
  k2_PC2 = procrustes$Yrot[, 2])

ggplot(plot_data) +
  geom_point(aes(x=MP_PC1, y=MP_PC2), color = "blue") +
  geom_segment(aes(x=MP_PC1,y=MP_PC2,xend=k2_PC1,yend=k2_PC2),arrow=arrow(type = "closed", length=unit(5,"mm"))) +
  labs(title = "Procrustes Plot MetaPhlAn vs Kraken 2", x = "PC1", y = "PC2") +
  scale_color_manual(values = c("16S" = "black", "MG" = "blue")) +
  guides(color = guide_legend(title = "Data Type"))

```

Procrustes Plot MetaPhiAn vs Kraken 2



Comparison of different confidence settings

```
prebuilt = import_biom("orig.biom")
custom_conf = import_biom("pb_conf.biom")
custom_no_conf = import_biom("pb_no_conf.biom")

prebuilt

## phyloseq-class experiment-level object
## otu_table()    OTU Table:      [ 1782 taxa and 5 samples ]
## sample_data() Sample Data:    [ 5 samples by 1 sample variables ]
## tax_table()   Taxonomy Table: [ 1782 taxa by 7 taxonomic ranks ]

custom_conf

## phyloseq-class experiment-level object
## otu_table()    OTU Table:      [ 869 taxa and 5 samples ]
## sample_data() Sample Data:    [ 5 samples by 1 sample variables ]
## tax_table()   Taxonomy Table: [ 869 taxa by 7 taxonomic ranks ]

custom_no_conf
```

```

## phyloseq-class experiment-level object
## otu_table()    OTU Table:          [ 9891 taxa and 5 samples ]
## sample_data() Sample Data:        [ 5 samples by 1 sample variables ]
## tax_table()   Taxonomy Table:     [ 9891 taxa by 7 taxonomic ranks ]

# set rank names
colnames(tax_table(prebuilt)) = c("Domain", "Phylum", "Class", "Order", "Family", "Genus", "Species")
colnames(tax_table(custom_conf)) = c("Domain", "Phylum", "Class", "Order", "Family", "Genus", "Species")
colnames(tax_table(custom_no_conf)) = c("Domain", "Phylum", "Class", "Order", "Family", "Genus", "Species")

# filtering
prebuilt <- subset_taxa(prebuilt, Domain!="k__Archaea")
prebuilt <- subset_taxa(prebuilt, Domain!="k__Viruses")
prebuilt <- subset_taxa(prebuilt, Domain!="k__Eukaryota")

custom_conf <- subset_taxa(custom_conf, Domain!="k__Archaea")
custom_conf <- subset_taxa(custom_conf, Domain!="k__Viruses")
custom_conf <- subset_taxa(custom_conf, Domain!="k__Eukaryota")

custom_no_conf <- subset_taxa(custom_no_conf, Domain!="k__Archaea")
custom_no_conf <- subset_taxa(custom_no_conf, Domain!="k__Viruses")
custom_no_conf <- subset_taxa(custom_no_conf, Domain!="k__Eukaryota")

prebuilt

## phyloseq-class experiment-level object
## otu_table()    OTU Table:          [ 1619 taxa and 5 samples ]
## sample_data() Sample Data:        [ 5 samples by 1 sample variables ]
## tax_table()   Taxonomy Table:     [ 1619 taxa by 7 taxonomic ranks ]

custom_conf

## phyloseq-class experiment-level object
## otu_table()    OTU Table:          [ 869 taxa and 5 samples ]
## sample_data() Sample Data:        [ 5 samples by 1 sample variables ]
## tax_table()   Taxonomy Table:     [ 869 taxa by 7 taxonomic ranks ]

custom_no_conf

## phyloseq-class experiment-level object
## otu_table()    OTU Table:          [ 9891 taxa and 5 samples ]
## sample_data() Sample Data:        [ 5 samples by 1 sample variables ]
## tax_table()   Taxonomy Table:     [ 9891 taxa by 7 taxonomic ranks ]

plot_taxa_prevalence(custom_no_conf, "Phylum")

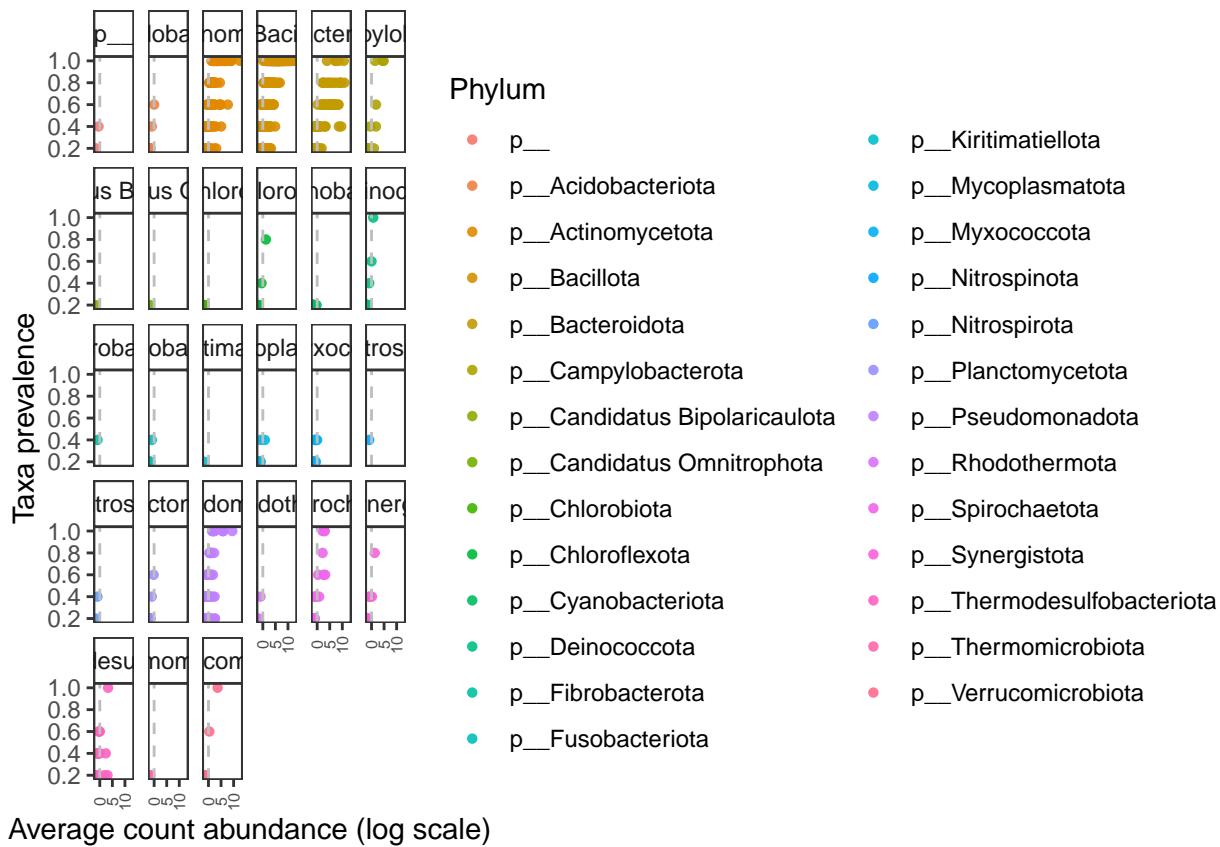
```

Phylum

- p__
- p__Acidobacteriota
- p__Actinomycetota
- p__Aquificota
- p__Armatimonadota
- p__Atribacterota
- p__Bacillota
- p__Bacteroidota
- p__Balneolota
- p__Bdellovibrionota
- p__Caldicericota
- p__Calditrichota
- p__Campylobacterota
- p__Candidatus Absconditabacteria
- p__Candidatus Bipolaricaulota
- p__Candidatus Cloacimonadota
- p__Candidatus Fervidibacteria
- p__Candidatus Omnitrophota
- p__Candidatus Saccharibacteria
- p__Chlamydiota
- p__Chlorobiota
- p__Chloroflexota
- p__Chrysigenota
- p__Coprothermobacterota
- p__Cyanobacteriota
- p__Deferrribacterota
- p__Deinococcota
- p__Dictyoglomota
- p__Elusimicrobiota
- p__Fibrobacterota
- p__Fusobacteriota
- p__Gemmatimonadota
- p__Ignavibacteriota
- p__Kiritimatiellota
- p__Lentisphaerota
- p__Mycoplasmatota
- p__Myxococcota
- p__Nitrospinota
- p__Nitrospiota
- p__Planctomycetota
- p__Pseudomonadota
- p__Rhodothermota
- p__Spirochaetota
- p__Synergistota
- p__Thermodesulfob
- p__Thermomicrobio
- p__Thermotogota
- p__Verrucomicrobio

undance (log scale)

```
plot_taxa_prevalence(prepbuilt, "Phylum")
```



Comparison of Kraken 2 runs on internal and external servers

```
### loading two subsets of metagenomic data into phyloseq format
externalps = import_biom("extern.biom")
internalps = import_biom("intern.biom")

externalps

## phyloseq-class experiment-level object
## otu_table()    OTU Table:          [ 2434 taxa and 10 samples ]
## sample_data()  Sample Data:       [ 10 samples by 1 sample variables ]
## tax_table()    Taxonomy Table:    [ 2434 taxa by 7 taxonomic ranks ]

internalps # internal has 102 less taxa, but this is unfiltered

## phyloseq-class experiment-level object
## otu_table()    OTU Table:          [ 2331 taxa and 10 samples ]
## sample_data()  Sample Data:       [ 10 samples by 1 sample variables ]
## tax_table()    Taxonomy Table:    [ 2331 taxa by 7 taxonomic ranks ]

# set Rank names
colnames(tax_table(externalps)) = c("Domain", "Phylum", "Class", "Order", "Family", "Genus", "Species")
```

```

colnames(tax_table(internalps)) = c("Domain", "Phylum", "Class", "Order", "Family", "Genus", "Species")

# filtering
externalps <- subset_taxa(externalps, Domain!="k__Archaea")
externalps <- subset_taxa(externalps, Domain!="k__Viruses")
externalps <- subset_taxa(externalps, Domain!="k__Eukaryota")

internalps <- subset_taxa(internalps, Domain!="k__Archaea")
internalps <- subset_taxa(internalps, Domain!="k__Viruses")
internalps <- subset_taxa(internalps, Domain!="k__Eukaryota")

externalps

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 2187 taxa and 10 samples ]
## sample_data() Sample Data: [ 10 samples by 1 sample variables ]
## tax_table() Taxonomy Table: [ 2187 taxa by 7 taxonomic ranks ]

internalps # now, there are 17 more taxa in internal

## phyloseq-class experiment-level object
## otu_table() OTU Table: [ 2204 taxa and 10 samples ]
## sample_data() Sample Data: [ 10 samples by 1 sample variables ]
## tax_table() Taxonomy Table: [ 2204 taxa by 7 taxonomic ranks ]

datatable(tax_table(externalps))

sample_names(internalps) = c("Firm_2_42", "Firm_2_47", "Firm_2_48", "Firm_2_49", "Firm_2_50", "Firm_2_51",
                            "Firm_2_56", "Firm_2_57", "Firm_2_58")

# Visualize in the same plot
dataset1 = ps_filter(externalps)
dataset2 = ps_filter(internalps)

dataset1 %>% ps_mutate(dataset = "external")
dataset2 %>% ps_mutate(dataset = "internal")

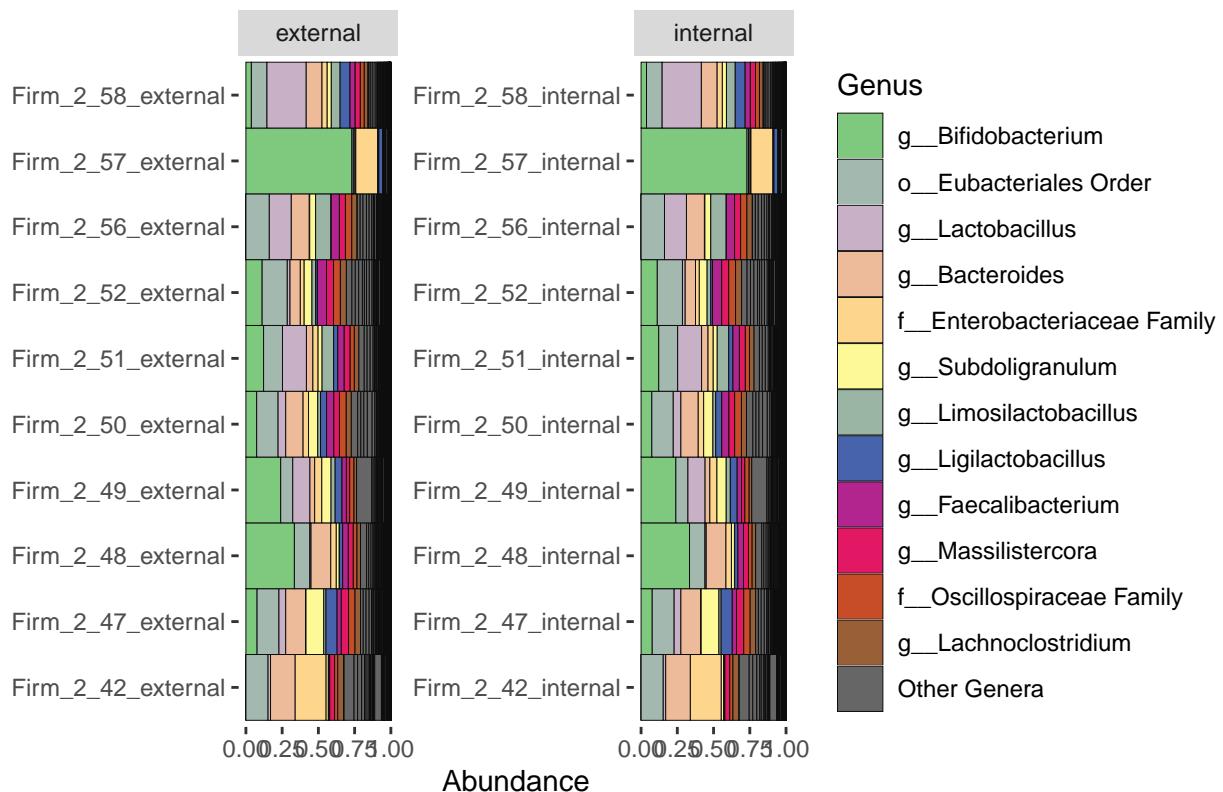
sample_names(dataset1) <- paste(sample_names(dataset1), "external", sep="_")
sample_names(dataset2) <- paste(sample_names(dataset2), "internal", sep="_")

combined <- phyloseq::merge_phyloseq(
  dataset1 %>% tax_fix(unknowns = c("g_")) %>% tax_agg("Genus") %>% ps_get(),
  dataset2 %>% tax_fix(unknowns = c("g_")) %>% tax_agg("Genus") %>% ps_get()
)

combined %>% tax_fix(unknowns = c("g_")) %>%
  comp_barplot("Genus", facet_by = "dataset", n_taxa = 12, palette = colorRampPalette(brewer.pal(8,"Accent")),
               other_name = "Other Genera", merge_other = F, sample_order = "asis") +
  coord_flip() + ggtitle("External vs internal dataset")

```

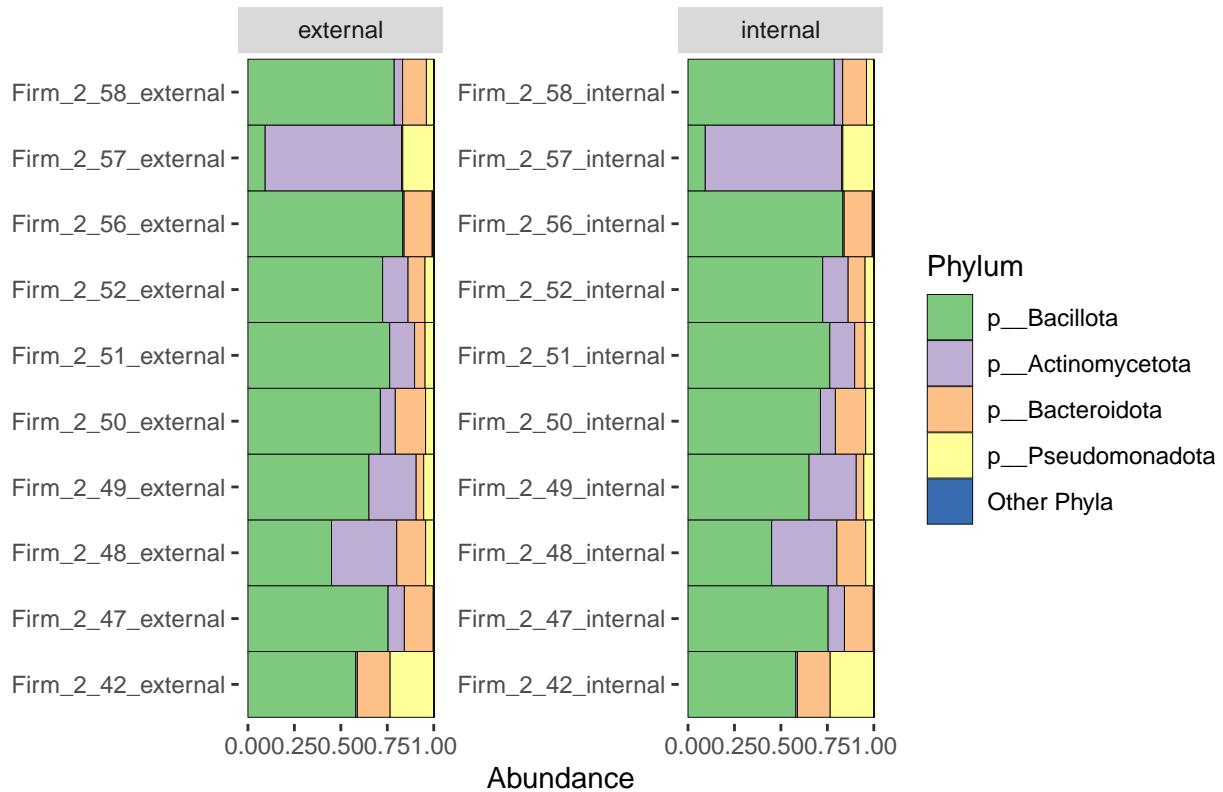
External vs internal dataset



```
# phylum
combined <- phyloseq::merge_phyloseq(
  dataset1 %>% tax_aggr("Phylum") %>% ps_get(),
  dataset2 %>% tax_aggr("Phylum") %>% ps_get()
)

combined %>% tax_fix(unknowns = c("g_"))
comp_barplot("Phylum", facet_by = "dataset", n_taxa = 4, palette = colorRampPalette(brewer.pal(5,"Accent")),
             other_name = "Other Phyla", merge_other = F, sample_order = "asis") +
  coord_flip() + ggtitle("External vs internal dataset")
```

External vs internal dataset



#Procrustes analysis

```

PCoA_BC_int = ordinate(internalps, "PCoA")
PCoA_BC_ext = ordinate(externalps, "PCoA")
procrustes = protest(PCoA_BC_int$vectors, PCoA_BC_ext$vectors)

plot_data <- data.frame(
  int_PC1 = procrustes$X[, 1],
  int_PC2 = procrustes$X[, 2],
  ext_PC1 = procrustes$Yrot[, 1],
  ext_PC2 = procrustes$Yrot[, 2]
)

sample_pairs = data.frame("int_sample"=rownames(PCoA_BC_int$vectors), "ext_sample"=rownames(PCoA_BC_ext$vectors))

ggplot(plot_data) +
  geom_point(aes(x=ext_PC1, y=ext_PC2), color = "green") +
  geom_point(aes(x=int_PC1, y=int_PC2), color = "blue") +
  geom_segment(aes(x=int_PC1,y=int_PC2,xend=ext_PC1,yend=ext_PC2),arrow=arrow(type = "closed", length=unit(5, "mm")))
  labs(title = "Procrustes Plot internal vs external server") # there is complete overlap, only one color
  
```

Procrustes Plot internal vs external server

