Final_Project

Pankaj Gaonkar

2025-04-10

Contents

ABOUT

This script is designed for the analysis of the Shotgun_POULTRY microbiome data This script contains the statistical analysis, exploratory plots and publication plots. Each step of the analysis is discussed at my best capabilities.For more detailed understanding please refer the Readme file. Click Here: Readme File

NOTE: This document is for class purpose and is intended to be updated

Getting ready

Loading packages

```
# Loading Packages
# Note: Uncomment and install/load the packages if required

# if(!requireNamespace("BiocManager")){
#   install.packages("BiocManager")
# }
# BiocManager::install("phyloseq")
# BiocManager::install("microViz")
# BiocManager::install("microbiomeMarker")
#
# install.packages("remotes")
# remotes::install_github("david-barnett/microViz")
# library(microViz)
#
# #Install pairwiseAdonis package if not already installed
# if (!requireNamespace("devtools", quietly = TRUE)) {
#   install.packages("devtools")
# }
# devtools::install_github("pmartinezarbizu/pairwiseAdonis/pairwiseAdonis")

# Load the package
# library(pairwiseAdonis)


library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
library(svglite)
library(scales)
library(tibble)
library(reshape2)
library(Polychrome)
library(RColorBrewer)
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.4.3
```

```
library(tidyverse)
```

```
## Warning: package 'purrr' was built under R version 4.4.2
```

```
## Warning: package 'lubridate' was built under R version 4.4.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(ggpubr)

library(phyloseq)
library(vegan)
```

```
## Warning: package 'vegan' was built under R version 4.4.2
```

```
## Loading required package: permute
## Loading required package: lattice
```

```
library(microViz)
```

```
## microViz version 0.12.6 - Copyright (C) 2021-2024 David Barnett
## Attaching package: 'microViz'
## The following object is masked from 'package:ggpubr':
##
## stat_chull
## ! Website: https://david-barnett.github.io/microViz
## v Useful?  For citation details, run: `citation("microViz")`
## x Silence? `suppressPackageStartupMessages(library(microViz))`
```

```
library(microbiomeMarker)
```

```
## Registered S3 method overwritten by 'gplots':
##   method         from
##   reorder.factor DescTools
##
## Attaching package: 'microbiomeMarker'
##
## The following object is masked from 'package:phyloseq':
##
##     plot_heatmap
```

```
library(dplyr)
library(tidyr)

library(VennDiagram)
```

```
## Loading required package: grid
## Loading required package: futile.logger
##
## Attaching package: 'VennDiagram'
##
## The following object is masked from 'package:ggpubr':
##
##     rotate
```

```
library(grid)
```

Loading required input files

```
# Load the microbiome realtive abundance data, sample metadata, taxanomy metadata.
# Note: All files are present in project folder and can be uploaded relatively.

tbl_otu <- read.delim("Microbiome_RelAb.tsv") %>%
  column_to_rownames(var = "Taxon")

mt_smpl <- read.delim("Sample_metadata.tsv",
                      #row.names=1,
                      stringsAsFactors=TRUE) %>%
           #Duplicating sample column for rowname
           mutate(sample = Sample) %>%
           column_to_rownames(var = "sample")

# Reordering factors
# We have two fain groups: Farm types and Sample types. Within main group we have respective subgroups.

mt_smpl$Farm_type <- factor(mt_smpl$Farm_type,
                            levels = c("Pullet", "Breeder", "Broiler", "Processing_plant")
                            )
mt_smpl$Sample_type <- factor(mt_smpl$Sample_type,
                              levels = c("Litter", "Soil")
                              )
#unique command helps to avoid duplicates, arrange levels as per biological relevance, here based on po

mt_smpl$Sample_group <- factor(mt_smpl$Sample_group,
                               levels = unique(c("C2F4L","C2F4L","C2F4L","C2F1L","C3F10L","C3F14L","C2F4


mt_smpl$Sample <- factor(mt_smpl$Sample,
                               levels = c("C2F4L1","C2F4L2","C2F4L3","C2F1L","C3F10L","C3F14L","C2F4S1"

# Loading taxonomy metadata file
```

```
mt_tax <- read.delim("Taxonomy_metadata.tsv",
                     #row.names=1,
                     stringsAsFactors=TRUE) %>%
          column_to_rownames(var = "Taxon")
```

Setting colors

It is important to follow a consistent coloring scheme throughout the plots and paper. The color palette for the taxa will be created using the package "Polychrome".

```
# Creating color vectors

# Sample colors
col_sample <- c("#D02C2C", "#5BBCD6", "#F2AD00", "#F98400","#00A08A")

# Source colors
col_farm <- c("#E0BD48", "#319dc8", "#735794", "darkgrey" )

col_phy <- c ("#0DFFCA","#DDE996","#FB6CE0","#71BDA3","#FFAE8D","#A90040","#C2C8FE","#72722A","#D1EE0D"

col_tax <- c("#b2df8a","#f781bf","#a65628","#ffff33","#984ea3","#4daf4a","#ff7f00","#377eb8","#e41a1c",

# We can use the following as our coloring scheme (uncomment if needed below)
#swatch(col_sample)
#swatch(col_farm)
#swatch(col_tax)
#swatch(col_phy)
```

Print dataset structure

```
as_tibble(mt_smpl)
```

```
## # A tibble: 75 x 7
##    Sample Farm_Num Sample_group Farm_type Sample_type Company TotalReads
##    <fct>  <fct>    <fct>        <fct>     <fct>       <fct>        <int>
##  1 C2F1L  FN1      C2F1L        Pullet    Litter      A         78823314
##  2 C2F1S  FN1      C2F1S        Pullet    Soil        A         89536814
##  3 C2F4F1 FN2      C2F4F        Pullet    <NA>        A         41022956
##  4 C2F4F2 FN2      C2F4F        Pullet    <NA>        A         37557920
##  5 C2F4F3 FN2      C2F4F        Pullet    <NA>        A         49042293
##  6 C2F4L1 FN2      C2F4L        Pullet    Litter      A         48994993
##  7 C2F4L2 FN2      C2F4L        Pullet    Litter      A         53297079
##  8 C2F4L3 FN2      C2F4L        Pullet    Litter      A         40989190
##  9 C2F4S1 FN2      C2F4S        Pullet    Soil        A         37288480
## 10 C2F4S2 FN2      C2F4S        Pullet    Soil        A         49869876
## # i 65 more rows
```

```
#Note the NA values belong to the Fecal samples and Processing plant samples, which are not included in
```

Metaphlan and phyloseq

- Metaphlan outputs a table of relative abundances. However, phyloseq *only accepts absolute abundances for alpha-diversity analysis.*
- In order to use phyloseq the relative abundance data will be coerced into natural counts. This will done by multiplying the relative abundance by the number of mapped reads/sample (included in the table mt_smpl and obtained from metaphlan output), then rounding the table, thus creating a dummy table of natural counts. This table preserves the relative abundances relationships between SGBs and allows phyloseq to run.

Creating natural count table

```
#Creating a vector containing the the total number of reads in the same order as the columns in tbl_otu

nreads <- mt_smpl$TotalReads
#nreads   #Uncomment to see output

#Dividing all values in tbl_otu to shrink proportions to 0-1

tbl_otu <- tbl_otu[,1:ncol(tbl_otu)]/100

#Multiplying all values (x) of column ith (i) by the ith element of the nreads vector (i'), such as i(x

tbl_otu_ecount <- as.data.frame(t(t(tbl_otu)*nreads)) %>%
                  dplyr::mutate(across(is.numeric, round))
```

```
## Warning: There was 1 warning in `dplyr::mutate()`.
## i In argument: `across(is.numeric, round)`.
## Caused by warning:
## ! Use of bare predicate functions was deprecated in tidyselect 1.1.0.
## i Please use wrap predicates in `where()` instead.
##   # Was:
##   data %>% select(is.numeric)
##
##   # Now:
##   data %>% select(where(is.numeric))
```

```
#The table is being transposed so that every row (i) matches with the vector entry (i'), then performin
```

The count data is ready to be imported into phyloseq

For our analysis, we will focus only on the bacterial community.

```
#Importing files to phyloseq

OTUcount = otu_table(tbl_otu_ecount,
                taxa_are_rows = TRUE)
      #Needs to be parsed as matrix.
TAX = tax_table(as.matrix(mt_tax)
```

```
                )
      #Needs to be parsed as matrix.
sampledata = sample_data(mt_smpl)

#Creating phyloseq objects
phycount_e = phyloseq(OTUcount, TAX, sampledata)

#Excluding Unclassified, Archaea and Dropping other groups that needs to be excluded
phycount_e <- phycount_e %>%
  subset_taxa(!Phylum == "UNCLASSIFIED") #%>%
```

Phyloseq summary

```
phycount_e
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 3535 taxa and 73 samples ]
## sample_data() Sample Data:       [ 73 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 3535 taxa by 7 taxonomic ranks ]
```

```
print("")
```

```
## [1] ""
```

```
print("Unique Phyla in phycount_e Object")
```

```
## [1] "Unique Phyla in phycount_e Object"
```

```
get_taxa_unique(phycount_e, "Phylum")
```

```
##  [1] "Firmicutes"                  "Actinobacteria"
##  [3] "Bacteroidetes"               "Proteobacteria"
##  [5] "Euryarchaeota"               "Cyanobacteria"
##  [7] "Bacteria_unclassified"       "Deinococcus_Thermus"
##  [9] "Thaumarchaeota"              "Acidobacteria"
## [11] "Planctomycetes"              "Chloroflexi"
## [13] "Gemmatimonadetes"            "Nitrospirae"
## [15] "Ascomycota"                  "Verrucomicrobia"
## [17] "Candidatus_Thermoplasmatota" "Fusobacteria"
## [19] "Candidatus_Melainabacteria"  "Lentisphaerae"
## [21] "Synergistetes"               "Chlamydiae"
## [23] "Spirochaetes"                "Rhodothermaeota"
## [25] "Deferribacteres"             "Candidatus_Adlerbacteria"
## [27] "Candidatus_Saccharibacteria"
```

```
print("Sample Sources in phycount_e object")
```

```
## [1] "Sample Sources in phycount_e object"
```

```
levels(sample_data(phycount_e)$Sample_type)
```

```
## [1] "Litter" "Soil"
```

Alpha Diversity

Calculating alpha-diversity indexes

```
phycount_div <- estimate_richness(phycount_e,
                                  split = TRUE,
                                  measures = c("Observed", "Simpson", "Shannon")
                                  )
```

```
## Warning in estimate_richness(phycount_e, split = TRUE, measures = c("Observed", : The data you have p
## any singletons. This is highly suspicious. Results of richness
## estimates (for example) are probably unreliable, or wrong, if you have already
## trimmed low-abundance taxa from the data.
##
## We recommended that you find the un-trimmed data and retry.
```

```
#Adding sample data to table

phycount_div <- left_join(rownames_to_column(phycount_div, "Sample"),
                          mt_smpl,
                          by = "Sample")

#Calculating average and standard deviation

phycount_div <- phycount_div %>%
  group_by(Sample_group) %>%
  mutate(m_Observed = mean(Observed),
         sd_Observed = sd(Observed),
         m_Shannon = mean(Shannon),
         sd_Shannon = sd(Shannon),
         m_Simpson = mean(Simpson),
         sd_Simpson = sd(Simpson)
         )

phycount_div
```

```
## # A tibble: 73 x 16
## # Groups:   Sample_group [55]
##    Sample  Observed Shannon Simpson Farm_Num Sample_group Farm_type Sample_type
##    <chr>      <dbl>   <dbl>   <dbl> <fct>    <fct>        <fct>     <fct>
##  1 C12FCHL      320    3.85   0.961 FN21     C12FCHL      Pullet    Litter
##  2 C12FCHS      503    5.14   0.990 FN21     C12FCHS      Pullet    Soil
##  3 C13FJBL      219    3.40   0.927 FN22     C13FJBL      Pullet    Litter
##  4 C13FJBS      600    5.06   0.984 FN22     C13FJBS      Pullet    Soil
##  5 C15FNSL      299    2.91   0.808 FN24     C15FNSL      Breeder   Litter
```

```
##  6 C15FNSS      385    4.82   0.986 FN24     C15FNSS      Breeder   Soil
##  7 C15FTCL      398    3.34   0.884 FN25     C15FTCL      Breeder   Litter
##  8 C15FTCS      318    3.40   0.849 FN25     C15FTCS      Breeder   Soil
##  9 C16FHWL      390    3.90   0.952 FN26     C16FHWL      Breeder   Litter
## 10 C16FHWS      457    3.80   0.939 FN26     C16FHWS      Breeder   Soil
## # i 63 more rows
## # i 8 more variables: Company <fct>, TotalReads <int>, m_Observed <dbl>,
## #   sd_Observed <dbl>, m_Shannon <dbl>, sd_Shannon <dbl>, m_Simpson <dbl>,
## #   sd_Simpson <dbl>
```

```
#Write the alpha diversity values for each sample and average the replicates
write.csv(phycount_div, "Phycount_div_each_sample.csv")


## Table for alpha diversity

Alpha_descriptive_stats <- phycount_div %>%
  filter(Sample_type %in% c("Litter", "Soil")) %>%
  group_by(Farm_type, Sample_type) %>%
  summarise(
         m_Observed = mean(Observed),
         sd_Observed = sd(Observed),
         m_Simpson = mean(Simpson),
         sd_Simpson = sd(Simpson),
         m_Shannon = mean(Shannon),
         sd_Shannon = sd(Shannon),

  ) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Farm_type'. You can override using the
## `.groups` argument.
```

```
# Print the table
print(Alpha_descriptive_stats)
```

```
## # A tibble: 6 x 8
##   Farm_type Sample_type m_Observed sd_Observed m_Simpson sd_Simpson m_Shannon
##   <fct>     <fct>            <dbl>       <dbl>     <dbl>      <dbl>     <dbl>
## 1 Pullet    Litter            140.       105.      0.892    0.0767       2.99
## 2 Pullet    Soil              340.       168.      0.979    0.00657      4.54
## 3 Breeder   Litter            244.       105.      0.896    0.0614       3.18
## 4 Breeder   Soil              328.       159.      0.947    0.0413       4.07
## 5 Broiler   Litter            240.        58.5     0.919    0.0291       3.17
## 6 Broiler   Soil              367.       176.      0.961    0.0190       4.19
## # i 1 more variable: sd_Shannon <dbl>
```

```
write.csv(Alpha_descriptive_stats, "Alpha_Divertsity_descriptive_stats.csv")
```

Alpha diversity: t-test comparision between Litter and Soil within each farm types

```
# Using t.test to compare between the two groups

##Comparing Observed Richness
print("RICHNESS PULLET")
```

```
## [1] "RICHNESS PULLET"
```

```
phycount_div %>%
  filter(Sample_type %in% c("Soil", "Litter"),
         Farm_type == "Pullet") %>%
  t.test(Observed ~ Sample_type,
       data = .
       )
```

```
##
##  Welch Two Sample t-test
##
## data:  Observed by Sample_type
## t = -2.4673, df = 8.4047, p-value = 0.0375
## alternative hypothesis: true difference in means between group Litter and group Soil is not equal to
## 95 percent confidence interval:
##  -384.08631  -14.58035
## sample estimates:
## mean in group Litter   mean in group Soil
##             140.3333             339.6667
```

```
print("RICHNESS BREEDER")
```

```
## [1] "RICHNESS BREEDER"
```

```
phycount_div %>%
  filter(Sample_type %in% c("Soil", "Litter"),
         Farm_type == "Breeder") %>%
  t.test(Observed ~ Sample_type,
       data = .
       )
```

```
##
##  Welch Two Sample t-test
##
## data:  Observed by Sample_type
## t = -1.3218, df = 13.876, p-value = 0.2076
## alternative hypothesis: true difference in means between group Litter and group Soil is not equal to
## 95 percent confidence interval:
##  -221.00037   52.55593
## sample estimates:
## mean in group Litter   mean in group Soil
##             243.5556             327.7778
```

```r
print("RICHNESS BROILER")
```

```
## [1] "RICHNESS BROILER"
```

```r
phycount_div %>%
  filter(Sample_type %in% c("Soil", "Litter"),
         Farm_type == "Broiler") %>%
  t.test(Observed ~ Sample_type,
       data = .
       )
```

```
##
##  Welch Two Sample t-test
##
## data:  Observed by Sample_type
## t = -2.7454, df = 18.28, p-value = 0.01317
## alternative hypothesis: true difference in means between group Litter and group Soil is not equal to
## 95 percent confidence interval:
##  -224.30191  -29.94809
## sample estimates:
## mean in group Litter   mean in group Soil
##              240.125              367.250
```

```r
##Comparing Shannon index
print("SHANNON PULLET")
```

```
## [1] "SHANNON PULLET"
```

```r
phycount_div %>%
  filter(Sample_type %in% c("Soil", "Litter"),
         Farm_type == "Pullet") %>%
  t.test(Shannon ~ Sample_type,
       data = .
       )
```

```
##
##  Welch Two Sample t-test
##
## data:  Shannon by Sample_type
## t = -5.2641, df = 9.4962, p-value = 0.0004341
## alternative hypothesis: true difference in means between group Litter and group Soil is not equal to
## 95 percent confidence interval:
##  -2.2161807 -0.8913371
## sample estimates:
## mean in group Litter   mean in group Soil
##             2.986119             4.539878
```

```r
print("SHANNON BREEDER")
```

```
## [1] "SHANNON BREEDER"
```

```
phycount_div %>%
  filter(Sample_type %in% c("Soil", "Litter"),
         Farm_type == "Breeder") %>%
  t.test(Shannon ~ Sample_type,
       data = .
       )
```

```
##
##  Welch Two Sample t-test
##
## data:  Shannon by Sample_type
## t = -4.6001, df = 15.972, p-value = 0.000297
## alternative hypothesis: true difference in means between group Litter and group Soil is not equal to
## 95 percent confidence interval:
##  -1.3030532 -0.4808508
## sample estimates:
## mean in group Litter   mean in group Soil
##             3.177859             4.069811
```

```
print("SHANNON BROILER")
```

```
## [1] "SHANNON BROILER"
```

```
phycount_div %>%
  filter(Sample_type %in% c("Soil", "Litter"),
         Farm_type == "Broiler") %>%
  t.test(Shannon ~ Sample_type,
       data = .
       )
```

```
##
##  Welch Two Sample t-test
##
## data:  Shannon by Sample_type
## t = -8.0914, df = 19.293, p-value = 1.268e-07
## alternative hypothesis: true difference in means between group Litter and group Soil is not equal to
## 95 percent confidence interval:
##  -1.2823304 -0.7556906
## sample estimates:
## mean in group Litter   mean in group Soil
##             3.173157             4.192168
```

```
##Comparing Simpson index
print("SIMPSON PULLET")
```

```
## [1] "SIMPSON PULLET"
```

```
phycount_div %>%
  filter(Sample_type %in% c("Soil", "Litter"),
         Farm_type == "Pullet") %>%
  t.test(Simpson ~ Sample_type,
       data = .
       )
```

```
##
##  Welch Two Sample t-test
##
## data:  Simpson by Sample_type
## t = -2.7854, df = 5.0735, p-value = 0.03804
## alternative hypothesis: true difference in means between group Litter and group Soil is not equal to
## 95 percent confidence interval:
##  -0.167894531 -0.007098203
## sample estimates:
## mean in group Litter   mean in group Soil
##            0.8919210            0.9794174
```

```
print("SIMPSON BREEDER")
```

```
## [1] "SIMPSON BREEDER"
```

```
phycount_div %>%
  filter(Sample_type %in% c("Soil", "Litter"),
         Farm_type == "Breeder") %>%
  t.test(Simpson ~ Sample_type,
       data = .
       )
```

```
##
##  Welch Two Sample t-test
##
## data:  Simpson by Sample_type
## t = -2.0496, df = 14.013, p-value = 0.0596
## alternative hypothesis: true difference in means between group Litter and group Soil is not equal to
## 95 percent confidence interval:
##  -0.103409053  0.002342238
## sample estimates:
## mean in group Litter   mean in group Soil
##            0.8960792            0.9466126
```

```
print("SIMPSON BROILER")
```

```
## [1] "SIMPSON BROILER"
```

```
phycount_div %>%
  filter(Sample_type %in% c("Soil", "Litter"),
         Farm_type == "Broiler") %>%
  t.test(Observed ~ Sample_type,
       data = .
       )
```

```
##
##  Welch Two Sample t-test
##
## data:  Observed by Sample_type
## t = -2.7454, df = 18.28, p-value = 0.01317
```

```
## alternative hypothesis: true difference in means between group Litter and group Soil is not equal to
## 95 percent confidence interval:
##  -224.30191  -29.94809
## sample estimates:
## mean in group Litter   mean in group Soil
##              240.125              367.250
```

Writing alpha diversity t-test to file

```
## Sink command, it print everything between sinks()

set.seed(43)
sink("AlphaDiversity_ttest.txt")

   ##Comparing OBserved Richness
   print("RICHNESS PULLET")
   phycount_div %>%
     filter(Sample_type %in% c("Soil", "Litter"),
            Farm_type == "Pullet") %>%
     t.test(Observed ~ Sample_type,
          data = .
          )
   print("RICHNESS BREEDER")
   phycount_div %>%
     filter(Sample_type %in% c("Soil", "Litter"),
            Farm_type == "Breeder") %>%
     t.test(Observed ~ Sample_type,
          data = .
          )
   print("RICHNESS BROILER")
   phycount_div %>%
     filter(Sample_type %in% c("Soil", "Litter"),
            Farm_type == "Broiler") %>%
     t.test(Observed ~ Sample_type,
          data = .
          )


   ##Comparing Shannon index
   print("SHANNON PULLET")
   phycount_div %>%
     filter(Sample_type %in% c("Soil", "Litter"),
            Farm_type == "Pullet") %>%
     t.test(Shannon ~ Sample_type,
          data = .
          )
   print("SHANNON BREEDER")
   phycount_div %>%
     filter(Sample_type %in% c("Soil", "Litter"),
            Farm_type == "Breeder") %>%
    t.test(Shannon ~ Sample_type,
         data = .
         )
  print("SHANNON BROILER")
```

```
  phycount_div %>%
    filter(Sample_type %in% c("Soil", "Litter"),
           Farm_type == "Broiler") %>%
    t.test(Shannon ~ Sample_type,
         data = .
         )

  ##Comparing Simpson index
  print("SIMPSON PULLET")
  phycount_div %>%
    filter(Sample_type %in% c("Soil", "Litter"),
           Farm_type == "Pullet") %>%
    t.test(Simpson ~ Sample_type,
         data = .
         )
  print("SIMPSON BREEDER")
  phycount_div %>%
    filter(Sample_type %in% c("Soil", "Litter"),
           Farm_type == "Breeder") %>%
    t.test(Simpson ~ Sample_type,
         data = .
         )
  print("SIMPSON BROILER")
  phycount_div %>%
    filter(Sample_type %in% c("Soil", "Litter"),
           Farm_type == "Broiler") %>%
    t.test(Observed ~ Sample_type,
         data = .
         )

sink()
```

Drawing Richness and Diversity plots

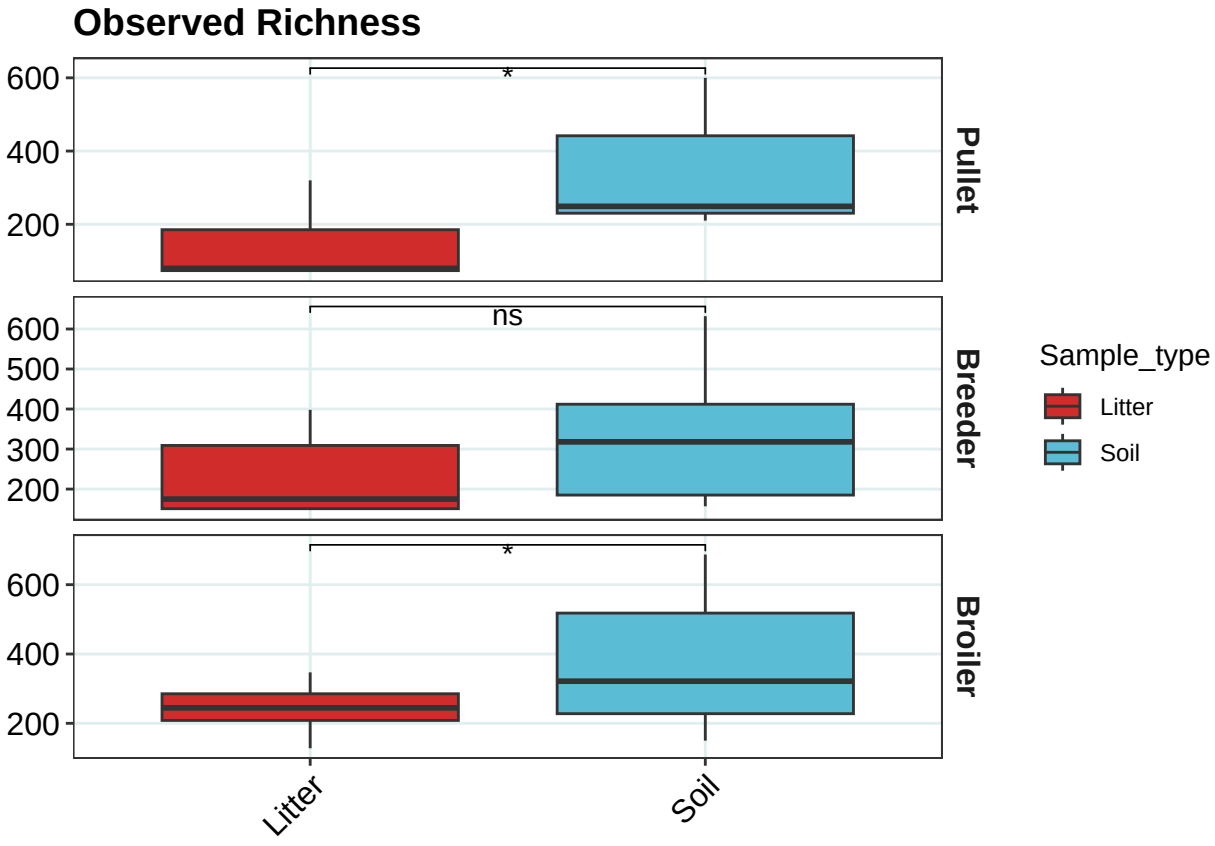
```
#Creating vector for comparing samples
 complist <- list(c("Soil", "Litter"))

#Creating vector with new names
 Farm_name <- c("Pullet", "Breeder", "Broiler", "Processing\nPlant")
names(Farm_name) <- c("Pullet", "Breeder", "Broiler", "Processing_plant")

#Drawing Observed Richness plot
bp_rc <-
      phycount_div %>%
  # filter(Sample_type == "Litter" | Sample_type == "Soil") %>%
  filter(Farm_type != "Processing_plant" & Sample_type != "Fecal") %>%
  ggplot(aes(x = Sample_type,
             y = Observed,
             fill = Sample_type
             )
         ) +
  geom_boxplot() +
  ggtitle("Observed Richness") +
```

```
  theme_bw() +
  theme(axis.text = element_text(size = 12,
                                 color = "black",
                                 hjust = 0.5
                                   ),
        axis.text.x = element_text(angle = 45,
                                   vjust = 1,
                                   hjust = 1),
        axis.title = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid.major = element_line(color = "azure2"),
        plot.title = element_text(size = 14, face = "bold"),
        strip.background = element_rect(fill = "transparent",
                                        color = "transparent"),
        strip.text = element_text(size = 12,
                                  face = "bold"
                                  )
        ) +
  stat_compare_means(comparisons = complist,
                     method = "t.test",
                     label = "p.signif",
                     vjust = 1
                     ) +
  scale_fill_manual(values = col_sample) +
  #facet_grid(Farm_type ~ Company,
   facet_grid(Farm_type ~ .,
              labeller = labeller(Farm_type = Farm_name),
              scales = "free",
             #space = "free_x"
              )


#Drawing Shannon diversity index plot
bp_sn <-
    phycount_div %>%
  # filter(Sample_type == "Litter" | Sample_type == "Soil") %>%
  filter(Farm_type != "Processing_plant" & Sample_type != "Fecal") %>%
  ggplot(aes(x = Sample_type,
             y = Shannon,
             fill = Sample_type
             )
         ) +
  geom_boxplot() +
  ggtitle("Shannon Diversity Index") +
  theme_bw() +
  theme(axis.text = element_text(size = 12,
                                 color = "black",
                                 hjust = 0.5
                                   ),
        axis.text.x = element_text(angle = 45,
                                   vjust = 1,
                                   hjust = 1),
        axis.title = element_blank(),
```

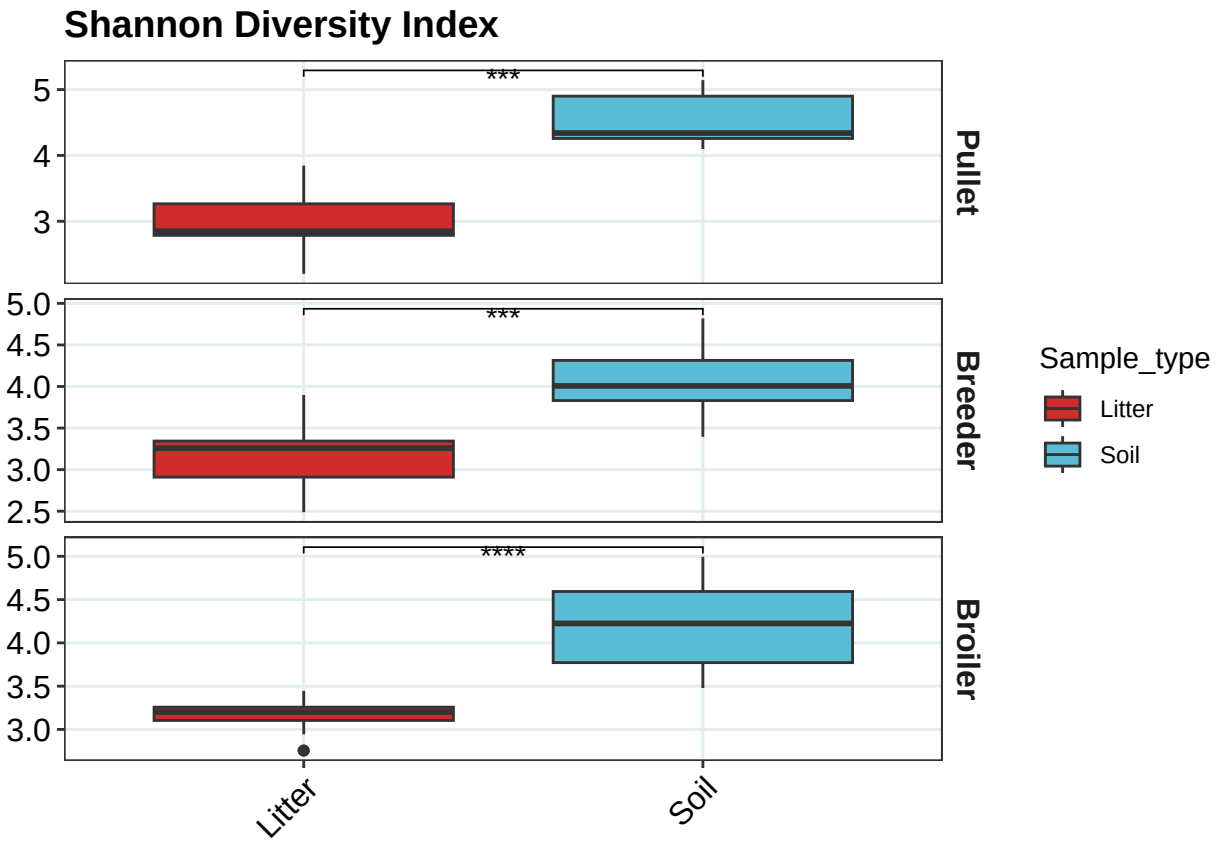
```
        panel.grid.minor = element_blank(),
        panel.grid.major = element_line(color = "azure2"),
        plot.title = element_text(size = 14, face = "bold"),
        strip.background = element_rect(fill = "transparent",
                                        color = "transparent"),
        strip.text = element_text(size = 12,
                                  face = "bold"
                                  )
        ) +
  stat_compare_means(comparisons = complist,
                     method = "t.test",
                     label = "p.signif",
                     vjust = 1
                     ) +
  scale_fill_manual(values = col_sample) +
  #facet_grid(Farm_type ~ Company,
  facet_grid(Farm_type ~ .,
             labeller = labeller(Farm_type = Farm_name),
             scales = "free",
            space = "free_x"
             )


#Drawing Simpson diversity index plot
bp_sp <-
  phycount_div %>%
  # filter(Sample_type == "Litter" | Sample_type == "Soil") %>%
  filter(Farm_type != "Processing_plant" & Sample_type != "Fecal") %>%
  ggplot(aes(x = Sample_type,
             y = Simpson,
             fill = Sample_type
                    )
                ) +
  geom_boxplot() +
  ggtitle("Simpson Diversity Index") +
  theme_bw() +
  theme(axis.text = element_text(size = 12,
                                 color = "black",
                                 hjust = 0.5
                                   ),
        axis.text.x = element_text(angle = 45,
                                   vjust = 1,
                                   hjust = 1),
        axis.title = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid.major = element_line(color = "azure2"),
        plot.title = element_text(size = 14, face = "bold"),
        strip.background = element_rect(fill = "transparent",
                                        color = "transparent"),
        strip.text = element_text(size = 12,
                                  face = "bold"
                                  )
        ) +
```

```
  stat_compare_means(comparisons = complist,
                     method = "t.test",
                     label = "p.signif",
                     vjust = 1
                     ) +
  scale_fill_manual(values = col_sample) +
  #facet_grid(Farm_type ~ Company,
  facet_grid(Farm_type ~ .,
             labeller = labeller(Farm_type = Farm_name),
             scales = "free",
            space = "free_x"
             )
```

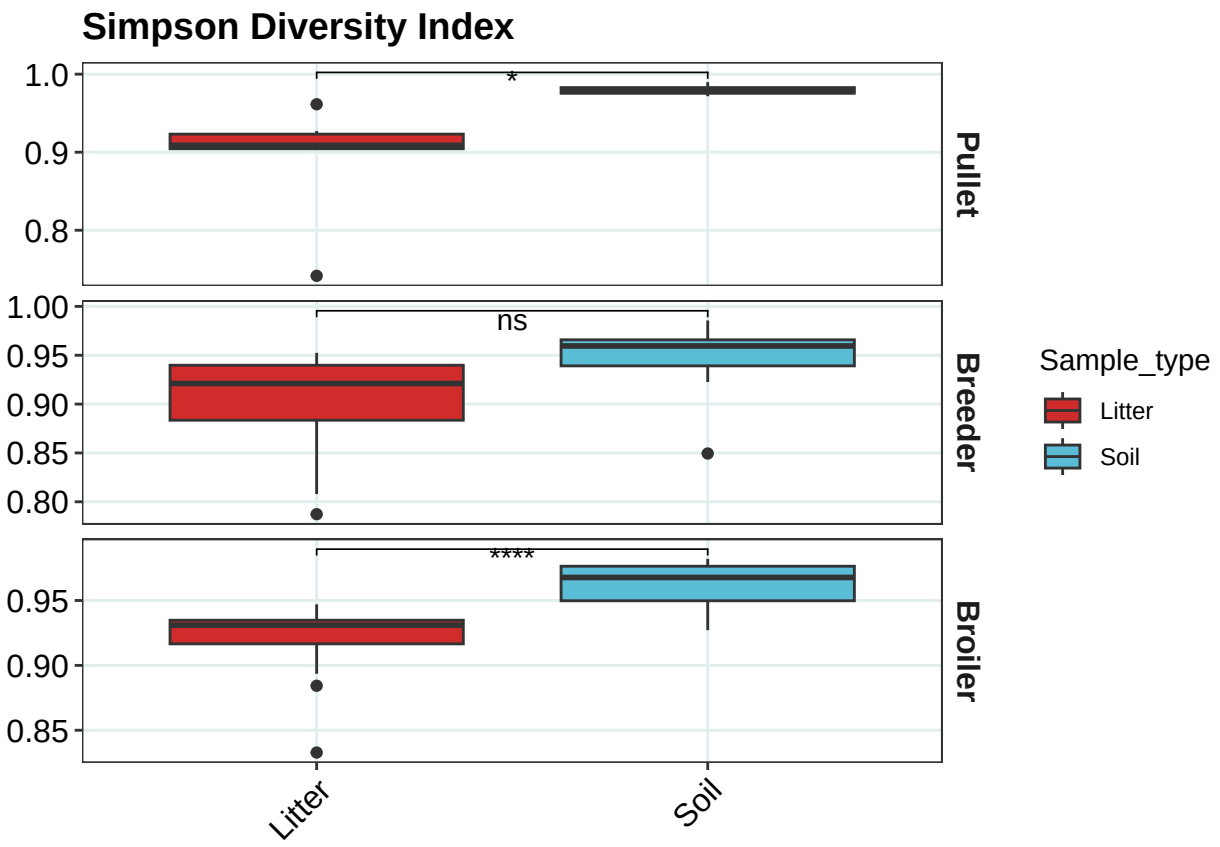
```
bp_rc
```




```
bp_sn
```


Shannon Diversity Index
***
***
****
Pullet
Breeder
Broiler
Sample_type
Litter
Soil
Litter
Soil

```
bp_sp
```



Saving combined alpha diversity plots

```
#Saving plots

#Combined plots
combined_plot_Alpha_Diversity <- ggarrange(bp_rc, bp_sn, bp_sp,
                          ncol = 3,
                          nrow = 1,
                          common.legend = TRUE
)


#Saving plots
ggsave(filename = "Alpha_Diversity_Plots.svg",
       plot = combined_plot_Alpha_Diversity,
       device = "svg",
       units = "mm",
       width = 300,
       height = 250)
```

Alpha diversity result explanation:

In our study, observed richness and Simpson diversity index were higher in the soil during the pullet and broiler stages, whereas no significant differences were found between litter and soil in the breeder stage. Moreover, the Shannon diversity index was higher in soil samples than in litter across all production stages.

These findings indicate that soil contains a more diverse and evenly distributed microbiome, whereas litter is a more selective environment with limited microbes that make up its composition.

Exploring the microbial community

Lets check which phylum are overall present in our dataset

```
table(tax_table(phycount_e)[,"Phylum"])
```

```
##
##               Acidobacteria             Actinobacteria
##                          10                        964
##                  Ascomycota      Bacteria_unclassified
##                           3                          2
##               Bacteroidetes   Candidatus_Adlerbacteria
##                         398                          1
##  Candidatus_Melainabacteria Candidatus_Saccharibacteria
##                           6                          2
## Candidatus_Thermoplasmatota                 Chlamydiae
##                           3                          4
##                 Chloroflexi              Cyanobacteria
##                           6                         54
##             Deferribacteres        Deinococcus_Thermus
##                           1                         21
##               Euryarchaeota                Firmicutes
##                          15                        742
##                Fusobacteria           Gemmatimonadetes
##                           2                          2
##               Lentisphaerae               Nitrospirae
##                           4                          4
##              Planctomycetes             Proteobacteria
##                          20                       1246
##             Rhodothermaeota               Spirochaetes
##                           1                          3
##               Synergistetes             Thaumarchaeota
##                           1                          8
##             Verrucomicrobia
##                          12
```

Microbial community bar plots

First step in visualizing the community structure. In order to create the barplots, the data needs to be transformed from wide to long. This will also allow all the metadata to be incorporated into the table

```
#Melting table

tbl_mstr_pc <- melt(rownames_to_column(tbl_otu, var = "Taxon"),
                  id.vars = c("Taxon"),
                  variable.name = "Sample",
                  value.name = "Count") %>%
```

```
#Adding Metadata
            left_join(mt_smpl,
                      by = "Sample") %>%
            left_join(rownames_to_column(mt_tax, var = "Taxon"),
                      by = "Taxon")
```

Re-ordering factors so that the main colors of the swatch correspond to the most abundant phyla across:

```
#Summarizing counts by phyla

tbl_mstr_pc %>%
  group_by(Phylum) %>%
  summarise(Count = sum(Count)) %>%
  arrange(Count) %>%
  select(Phylum) %>%
  ungroup()
```

```
## # A tibble: 28 x 1
##    Phylum
##    <fct>
##  1 Deferribacteres
##  2 Rhodothermaeota
##  3 Candidatus_Adlerbacteria
##  4 Candidatus_Saccharibacteria
##  5 Lentisphaerae
##  6 Spirochaetes
##  7 Ascomycota
##  8 Chlamydiae
##  9 Synergistetes
## 10 Verrucomicrobia
## # i 18 more rows
```

```
#Reordering factors in Phylum column according to most abundant.
#NOTE: Phyla will be reordered in ascending order (from least to most). This allows us to drop some lev

tbl_mstr_pc$Phylum <- factor(tbl_mstr_pc$Phylum,
                             levels = c("Deferribacteres","Rhodothermaeota","Candidatus_Adlerbacteria","Ca
```

Creating summarised table for barplots

```
tbl_mstr_bp <- tbl_mstr_pc %>%
  group_by(Sample, Phylum) %>%
  summarise(Count = sum(Count)
            ) %>%
  left_join(mt_smpl, by = "Sample") %>%
  distinct() %>%
  droplevels() %>%
  ungroup
```

```
## `summarise()` has grouped output by 'Sample'. You can override using the
## `.groups` argument.
```

creating averaged table

```
# Averaging taxons by Sample_group

tbl_mstr_av <- tbl_mstr_pc %>%
  group_by(Taxon, Sample_group) %>%
  summarise(Count = mean(Count)) %>%
  left_join(select(mt_smpl, !c(Sample, TotalReads)), by = "Sample_group") %>%
  left_join(rownames_to_column(mt_tax, var = "Taxon"), by = "Taxon") %>%
  distinct() %>%
  droplevels() %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Taxon'. You can override using the
## `.groups` argument.
```

```
## Warning in left_join(., select(mt_smpl, !c(Sample, TotalReads)), by = "Sample_group"): Detected an u
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 6 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
##Summarising by Phylum count

tbl_mstr_av_bp <- tbl_mstr_av %>%
  group_by(Phylum, Sample_group) %>%
  summarise(Count = sum(Count)) %>%
  left_join(select(mt_smpl, !c(Sample, TotalReads)), by = "Sample_group") %>%
  distinct() %>%
  droplevels() %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'Phylum'. You can override using the
## `.groups` argument.
```

```
## Warning in left_join(., select(mt_smpl, !c(Sample, TotalReads)), by = "Sample_group"): Detected an u
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 6 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
#Reordering Phylum factors (left join brought in factors with disordered levels)

tbl_mstr_av$Phylum <- factor(tbl_mstr_av$Phylum,
                             levels = c("Deferribacteres","Rhodothermaeota","Candidatus_Adlerbacteria","Ca




tbl_mstr_av_bp$Phylum <- factor(tbl_mstr_av_bp$Phylum,
                              levels = c("Deferribacteres","Rhodothermaeota","Candidatus_Adlerbacteria","C
```

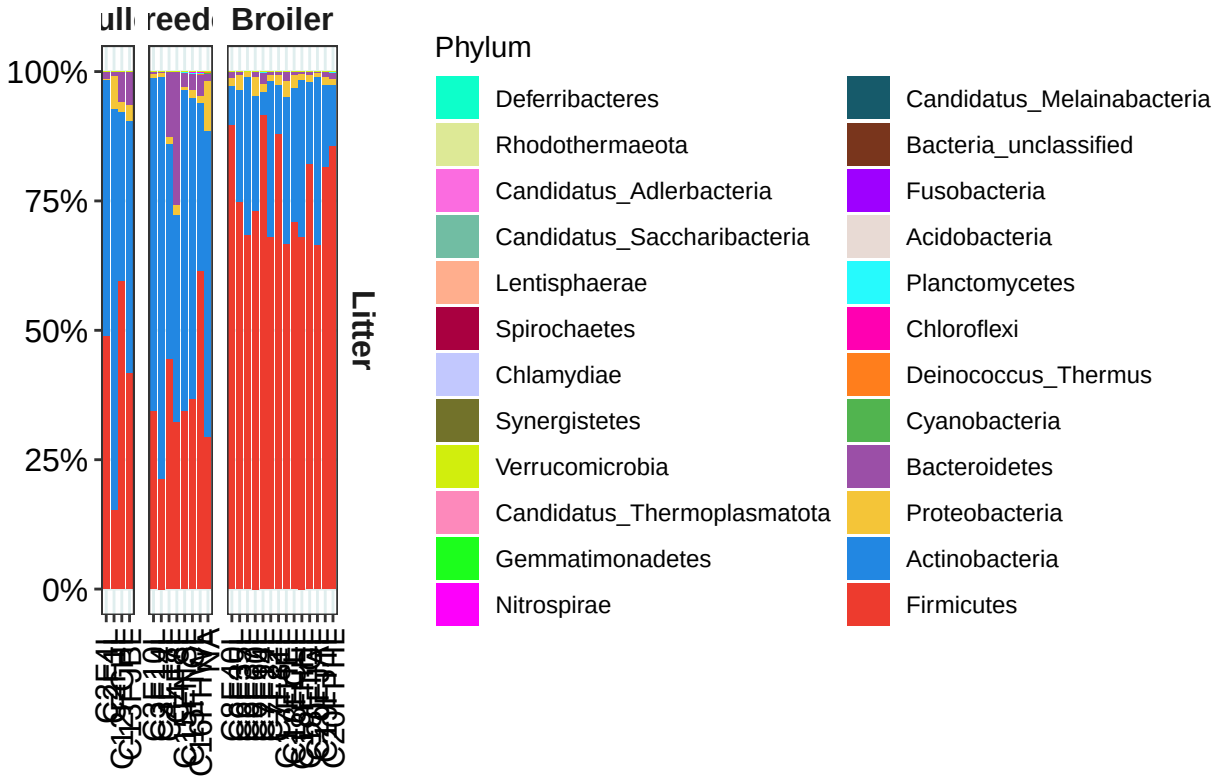
Drawing AVERAGED microbiome bar plots

```
####Litter bacterial

relbp_av_bac_Litter <-
tbl_mstr_av_bp %>%
  subset(Phylum!="UNCLASSIFIED" & Phylum!="Thaumarchaeota" & Phylum!= "Euryarchaeota" & Phylum!= "Ascomy
  ggplot(aes(x = Sample_group,
             y = Count,
             fill = Phylum
             )
         ) +
  geom_bar(position="fill", stat= "identity") +
  scale_y_continuous(labels = percent) +
  scale_fill_manual(values = col_phy) +
  theme_bw() +
  labs(title = "Microbiome Composition - Bacterial") +
  ylab("Relative Abundance") +
  theme_bw() +
  theme(axis.text = element_text(size = 12,
                                 color = "black"),
        axis.text.x = element_text(hjust = 1,
                                   vjust = 0.5,
                                   angle = 90),
        axis.title = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid.major = element_line(color = "azure2"),
        plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
        strip.background = element_rect(fill = "transparent",
                                        color = "transparent"),
        strip.text = element_text(size = 12,
                                  face = "bold"
                                  )
        ) +
  facet_grid(Sample_type ~ Farm_type, scale = "free_x", space = "free_x")



####Soil bacterial

relbp_av_bac_Soil <-
tbl_mstr_av_bp %>%
  subset(Phylum!="UNCLASSIFIED" & Phylum!="Thaumarchaeota" & Phylum!= "Euryarchaeota" & Phylum!= "Ascomy
  ggplot(aes(x = Sample_group,
             y = Count,
             fill = Phylum
             )
         ) +
  geom_bar(position="fill", stat= "identity") +
  scale_y_continuous(labels = percent) +
  scale_fill_manual(values = col_phy) +
  theme_bw() +
  ylab("Relative Abundance") +
  theme_bw() +
```

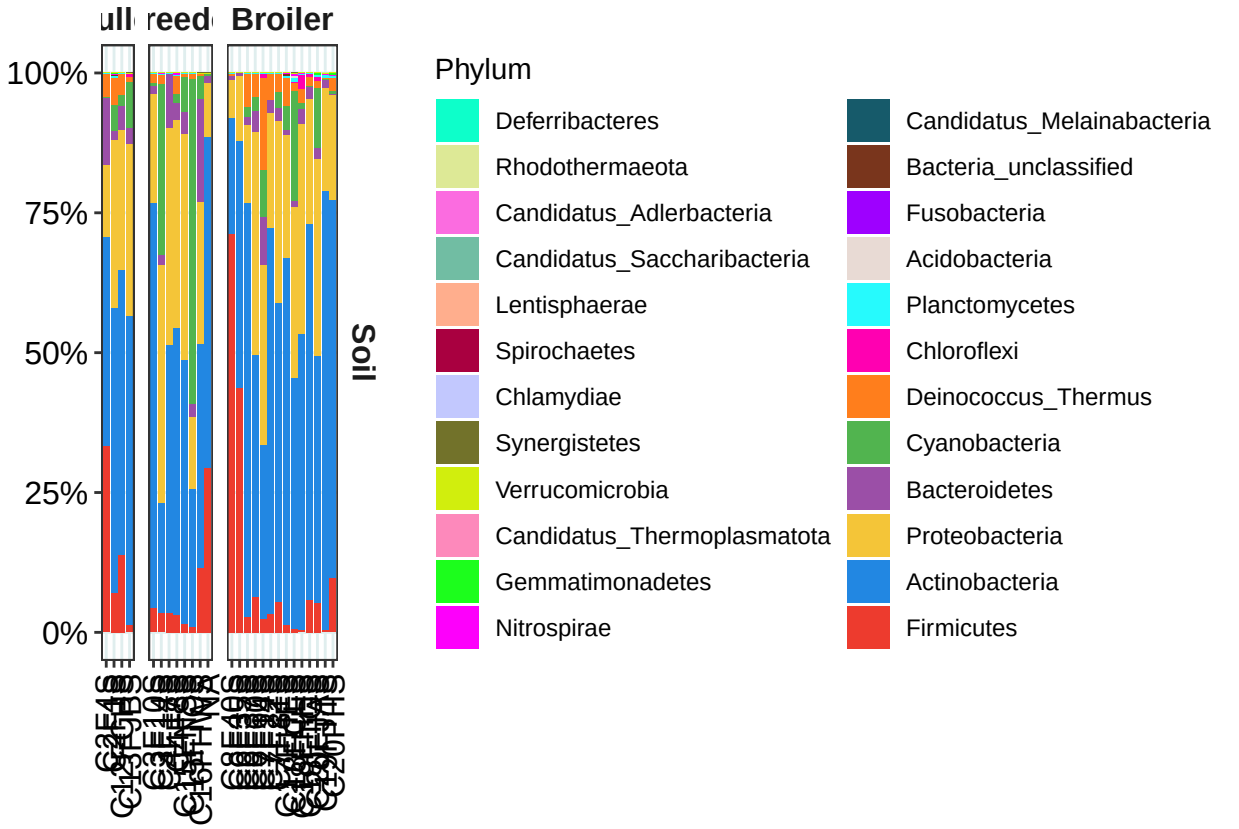
```
  theme(axis.text = element_text(size = 12,
                                 color = "black"),
        axis.text.x = element_text(hjust = 1,
                                   vjust = 0.5,
                                   angle = 90),
        axis.title = element_blank(),
        panel.grid.minor = element_blank(),
        panel.grid.major = element_line(color = "azure2"),
        strip.background = element_rect(fill = "transparent",
                                        color = "transparent"),
        strip.text = element_text(size = 12,
                                  face = "bold"
                                  )
        ) +
  facet_grid(Sample_type ~ Farm_type, scale = "free_x", space = "free_x")
```

```
relbp_av_bac_Litter
```

biome Composition − Bacterial




```
relbp_av_bac_Soil
```



Saving combined relative abundance plots

```
#Saving plots

#Combined plots
combined_plot_relative_abundance_bacteria <- ggarrange(relbp_av_bac_Litter, relbp_av_bac_Soil,
                          ncol = 1,
                          nrow = 2,
                          common.legend = TRUE,
                          legend = "right")

#Saving plots
ggsave(filename = "Relative_abundance_Plots.svg",
       plot = combined_plot_relative_abundance_bacteria,
       device = "svg",
       units = "mm",
       width = 350,
       height = 220)
```

Exploring Beta-Diversity

If alpha-diversity is the diversity *within* a community, beta-diversity is the diversity across communities. In beta-diversity we are concerned with comparing and contrasting different profiles.

Communities can be statistically compared by producing distance matrices. On compositional data we will use robust Aitchison distances, which relies on the centered log ratio transform. Once these distances are

calculated, profiles can be statistically compared using PERMANOVA and visualized using ordination plots, such as PCoA or PCA.

In order to perform these analysis the abundances will be again exported into phyloseq but this time as relative abundances. The phyloseq object will be used as input into two different packages: –> Vegan: where PERMANOVA and PERMAdisp will be calculates –> MicrobiomeMarker: where Ordinations and normalization will be performed

Creating relative abundance phyloseq object

```
OTUcount = otu_table(tbl_otu,
                taxa_are_rows = TRUE)
      #Needs to be parsed as matrix.
TAX = tax_table(as.matrix(mt_tax)
                )
      #Needs to be parsed as matrix.
sampledata = sample_data(mt_smpl)

#Creating phyloseq objects
phycount = phyloseq(OTUcount, TAX, sampledata)


#Excluding Unclassified, Archaea and Dropping incomplete sample groups: Larvae and Pollen
phycount <- phycount %>%
  subset_taxa(!Phylum == "UNCLASSIFIED" & !Kingdom == "Archaea" & !Kingdom == "Eukaryota") %>%
  subset_samples(!Sample_type == "Fecal" & !Farm_type == "Processing_plant")
```

```
# printing phyloseq summary

phycount
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 3506 taxa and 62 samples ]
## sample_data() Sample Data:       [ 62 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 3506 taxa by 7 taxonomic ranks ]
```

```
print("")
```

```
## [1] ""
```

```
print("Unique Phyla in phycount Object")
```

```
## [1] "Unique Phyla in phycount Object"
```

```
get_taxa_unique(phycount, "Phylum")
```

```
##  [1] "Firmicutes"                "Actinobacteria"
##  [3] "Bacteroidetes"             "Proteobacteria"
##  [5] "Cyanobacteria"             "Bacteria_unclassified"
##  [7] "Deinococcus_Thermus"       "Acidobacteria"
##  [9] "Planctomycetes"            "Chloroflexi"
## [11] "Gemmatimonadetes"          "Nitrospirae"
```

```
## [13] "Verrucomicrobia"             "Fusobacteria"
## [15] "Candidatus_Melainabacteria"  "Lentisphaerae"
## [17] "Synergistetes"               "Chlamydiae"
## [19] "Spirochaetes"                "Rhodothermaeota"
## [21] "Deferribacteres"             "Candidatus_Adlerbacteria"
## [23] "Candidatus_Saccharibacteria"
```

```
print("Sample Sources in phycount object")
```

```
## [1] "Sample Sources in phycount object"
```

```
levels(sample_data(phycount)$Sample_type)
```

```
## [1] "Litter" "Soil"
```

```
levels(sample_data(phycount)$Farm_type)
```

```
## [1] "Pullet"  "Breeder" "Broiler"
```

##Performing PERMANOVAS

- PERMANOVA, (permutational multivariate ANOVA), is a non-parametric alternative to MANOVA, or multivariate ANOVA test.
- PERMANOVAS are calculated using the adonis functions in the package Vegan. To use Vegan, phyloseq objects need to be turned into Vegan objects.
- Next functions convert physeq format into Vegan.

creating pssd2veg function

```
pssd2veg <- function(physeq) {
  sd <- sample_data(physeq)
  return(as(sd,"data.frame"))
}
```

Creating psotu2veg function

```
psotu2veg <- function(physeq) {
  OTU <- otu_table(physeq)
  if (taxa_are_rows(OTU)) {
    OTU <- t(OTU)
  }
  return(as(OTU, "matrix"))
}
```

Importing Phyloseq data into teh package Vegan

```
vegan_count <- pssd2veg(phycount)
vegan_otu <- psotu2veg(phycount)
```

Calculating distance matrix

```
##Calculating distance matrix using Robust Aitchinson (Euclidean distance of the CLR transform of non-z
veg_count_raitch <- vegdist(vegan_otu, "robust.aitchison")
```

RUNNING PERMANOVAs

```
#Writing the overall PERMANOVA results in text file
sink("PERMANOVA_Overall_Results.txt")


set.seed(43)

print("Comparing profiles by PERMANOVA",quote = FALSE, justify = "centre")
print("Grouping by Sample_group (Sample_type*Farm_type*Company",quote = FALSE, justify = "centre")

adonis2(veg_count_raitch ~ Sample_type*Farm_type*Company, data = vegan_count, permutations = 999)
print("",quote = FALSE, justify = "centre")
print("Grouping by Sample_type",quote = FALSE, justify = "centre")

adonis2(veg_count_raitch ~ Sample_type, data = vegan_count, permutations = 999)
print("",quote = FALSE, justify = "centre")
print("Grouping by Farm_type",quote = FALSE, justify = "centre")

adonis2(veg_count_raitch ~ Farm_type, data = vegan_count, permutations = 999)
print("",quote = FALSE, justify = "centre")
print("Grouping by Company",quote = FALSE, justify = "centre")

adonis2(veg_count_raitch ~ Company, data = vegan_count, permutations = 999)

sink()
```

Specific adonis pariwise comparision

```
#Writing the specific pariwise comparission of interest in text file
sink("specific_Pairwise_Comparissions_PERMANOVA.txt")

## Install pairwiseAdonis package if not already installed
#if (!requireNamespace("devtools", quietly = TRUE)) {
# install.packages("devtools")
#}
#devtools::install_github("pmartinezarbizu/pairwiseAdonis/pairwiseAdonis")

# Load the package
library(pairwiseAdonis)
```

```
## Loading required package: cluster
```

```
# Perform pairwise PERMANOVA for Sample_type
pairwise.adonis2(veg_count_raitch ~ Sample_type, data = vegan_count, permutations = 999)

# Perform pairwise PERMANOVA for Farm_type
```

```
pairwise.adonis2(veg_count_raitch ~ Farm_type, data = vegan_count, permutations = 999)


sink()
```

ORDINATION WITH PCOA

Drawing PCA plots

```
#Sample type
pcoa_sp_clr_S <-
  phycount %>%
 tax_transform(rank = "Species",
               trans = "identity") %>%
 dist_calc(dist = "robust.aitchison") %>%
 ord_calc(method = "PCoA") %>%
 ord_plot(axes = c(1, 2),
          plot_taxa = 1:3,
          colour = "black",
          fill = "Sample_type",
          shape = "Farm_type",
          alpha = 0.8,
          size = 5
          ) +
  stat_ellipse(aes(colour = Sample_type), linewidth = 0.3) +
  scale_shape_girafe_filled() +
  ggtitle("PCoA Plot: Beta Diversity - Sample Types") +
   guides(fill = guide_legend(override.aes=list(shape = 21)),
          color = FALSE) +
  scale_fill_manual(values = col_sample) +
  # scale_color_manual(values = col_sample) +
  # scale_alpha_discrete(range = c(0.35, 1)) +
  theme_linedraw() +
  theme(panel.grid = element_blank(),
        panel.background = element_rect(fill = "#fdfdfd"),
        axis.text = element_text(size = 14,
                                 color = "black"),
        axis.title = element_text(size = 16,
                                  color = "black")) +
  geom_text(x = 2.5,
            y = -3.4,
            hjust =0,
            vjust = 1,
            label="Sample Type:\nPERMANOVA=0.001",
            size = 3.5,
            fontface = "plain")
```

```
## Warning: otu_table of counts is NOT available!
## Available otu_table contains 16736 values that are not non-negative integers
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
```

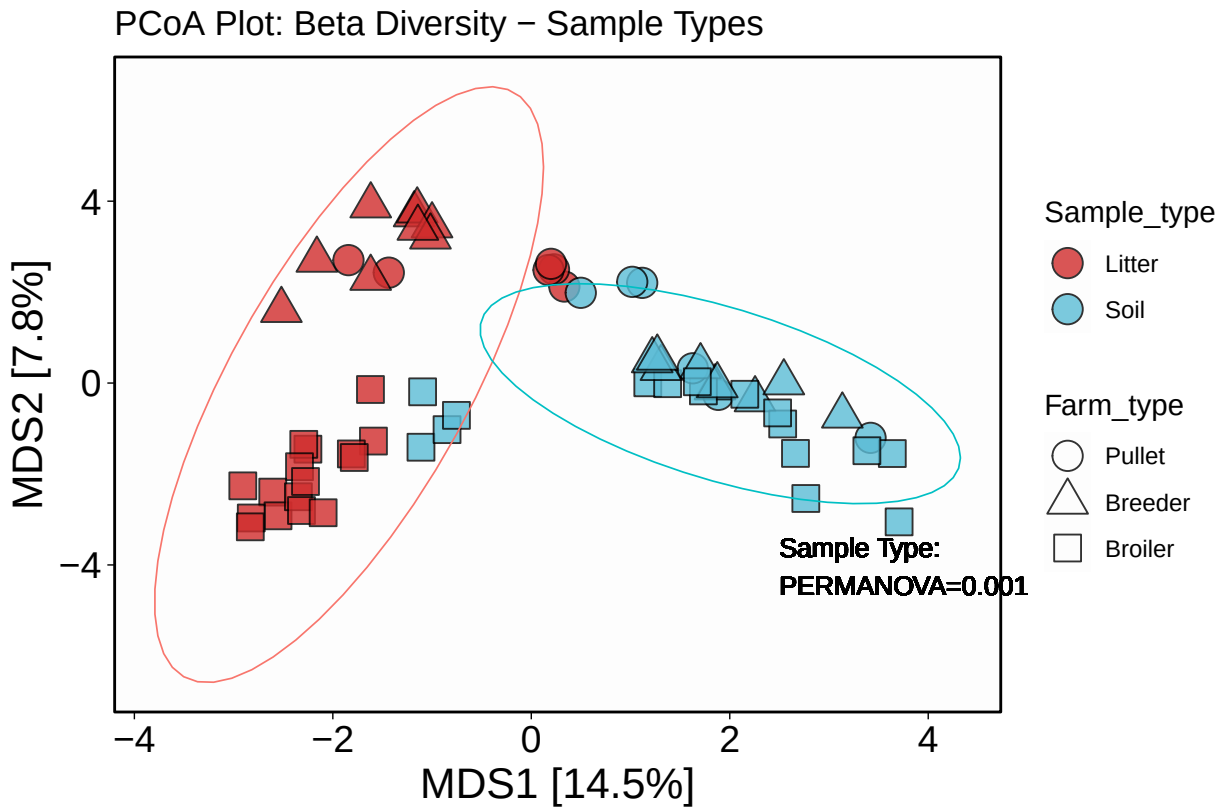
```
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
#Farm type
pcoa_sp_clr_farm <-
  phycount %>%
 tax_transform(rank = "Species",
               trans = "identity") %>%
 dist_calc(dist = "robust.aitchison") %>%
 ord_calc(method = "PCoA") %>%
 ord_plot(axes = c(1, 2),
          plot_taxa = 1:3,
          colour = "black",
          fill = "Farm_type",
          shape = "Sample_type",
          alpha = 0.8,
          size = 5
          ) +
  stat_ellipse(aes(colour = Farm_type), linewidth = 0.3) +
  scale_shape_girafe_filled() +
  ggtitle("PCoA Plot: Beta Diversity - Farm Types") +
   guides(fill = guide_legend(override.aes=list(shape = 21)),
          color = FALSE) +
  scale_fill_manual(values = col_farm) +
  scale_color_manual(values = col_farm) +
  # scale_alpha_discrete(range = c(0.35, 1)) +
  theme_linedraw() +
  theme(panel.grid = element_blank(),
        panel.background = element_rect(fill = "#fdfdfd"),
        axis.text = element_text(size = 14,
                                 color = "black"),
        axis.title = element_text(size = 16,
                                  color = "black")) +
  geom_text(x = 2.5,
            y = -3.4,
            hjust =0,
            vjust = 1,
            label="Farm Type:\nPERMANOVA=0.001",
            size = 3.5,
            fontface = "plain") #+
```

```
## Warning: otu_table of counts is NOT available!
## Available otu_table contains 16736 values that are not non-negative integers
```

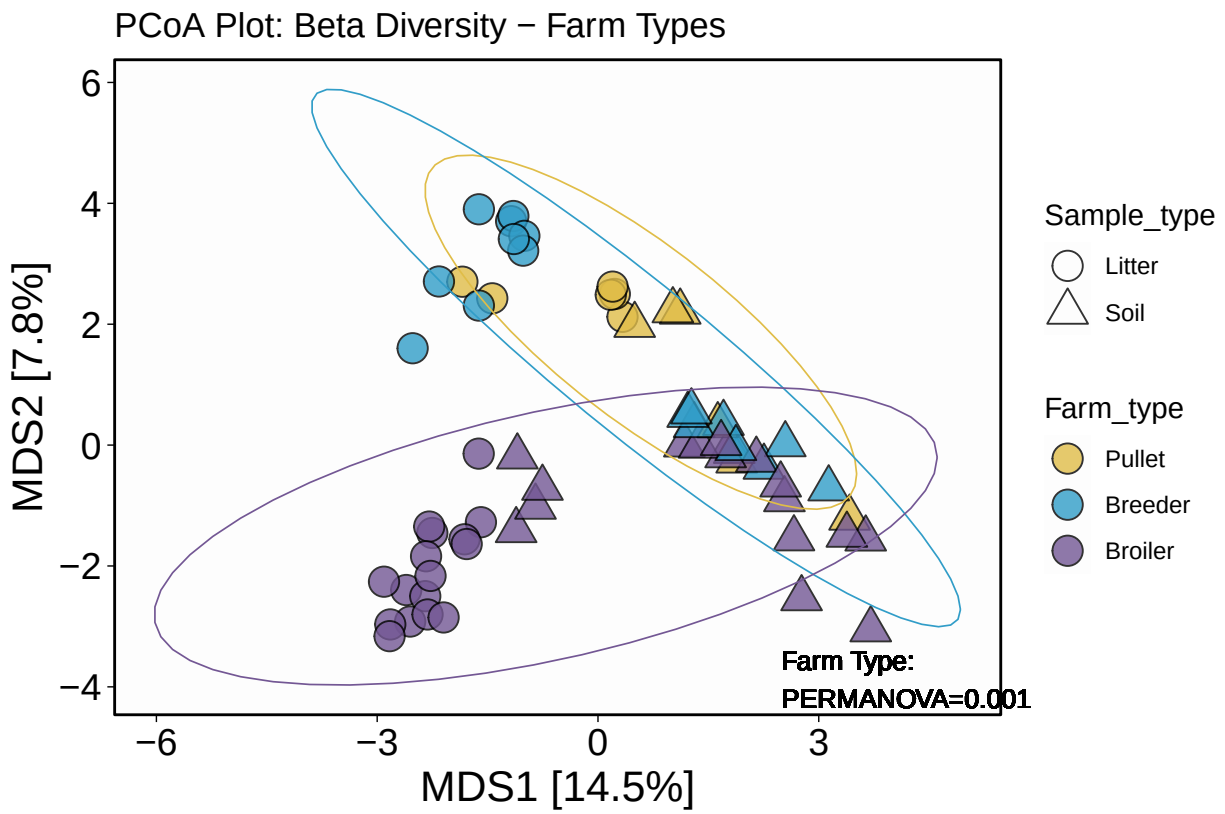
```
  #facet_grid(. ~ Company, scale = "free")
```

```
pcoa_sp_clr_S
```



62 samples & 3244 taxa (Species). PCoA tax_transform=identity dist=robust.aitchison

```
pcoa_sp_clr_farm
```



62 samples & 3244 taxa (Species). PCoA tax_transform=identity dist=robust.aitchison

Saving combined beta diversity - PCoA plots

```
#Combined plots
combined_PCoA <- ggarrange(pcoa_sp_clr_S, pcoa_sp_clr_farm,
                           ncol = 1,
                           nrow = 2)


ggsave(filename = "Combined_PCoA_plots_betadiversity.svg",
       plot = combined_PCoA,
       device = "svg",
       units = "mm",
       width = 250,
       height = 300)
```

Link to GitHub repository

Click here to go to GitHub repository