ITS of Wheat

2023-08-23

ITS of Wheat

ABC.....

Qiime2 to Phyloseq

Converting QIIME2 outcomes to the phyloseq object structure in R enables comprehensive analysis and visualization of microbial community profiles. The phyloseq package provides functions for organizing and manipulating data within the phyloseq object, allowing for diverse analyses such as diversity assessments, differential abundance testing, and taxonomic profile visualization. By leveraging R's capabilities, researchers can perform advanced statistical analysis, integrate with other omics data, and gain deeper insights into microbiome datasets.

```
# Download qiime2R from Github
# if (!requireNamespace("devtools", quietly = TRUE)){install.packages("devtools")}
# devtools::install_github("jbisanz/qiime2R")
library("qiime2R")

# Download phyloseq from CRAN
# if (!require("BiocManager", quietly = TRUE))
#   install.packages("BiocManager")
# BiocManager::install("phyloseq")
library("phyloseq")
```

```
# Convert qiime2 to phyloseq format
physeq.a <- qza_to_phyloseq(
  features = "table-its-fungi-with-phyla-no-mitochondria-no-chloroplast.qza", # table.qza
  tree = "phylogeny-align-to-tree-mafft-fasttree/rooted_tree.qza",
  taxonomy = "taxonomy-fungi.qza",
  metadata = "meta-table.txt"
)
physeq.a ## confirm the object
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 3904 taxa and 37 samples ]
## sample_data() Sample Data:       [ 37 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 3904 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 3904 tips and 3878 internal nodes ]
```

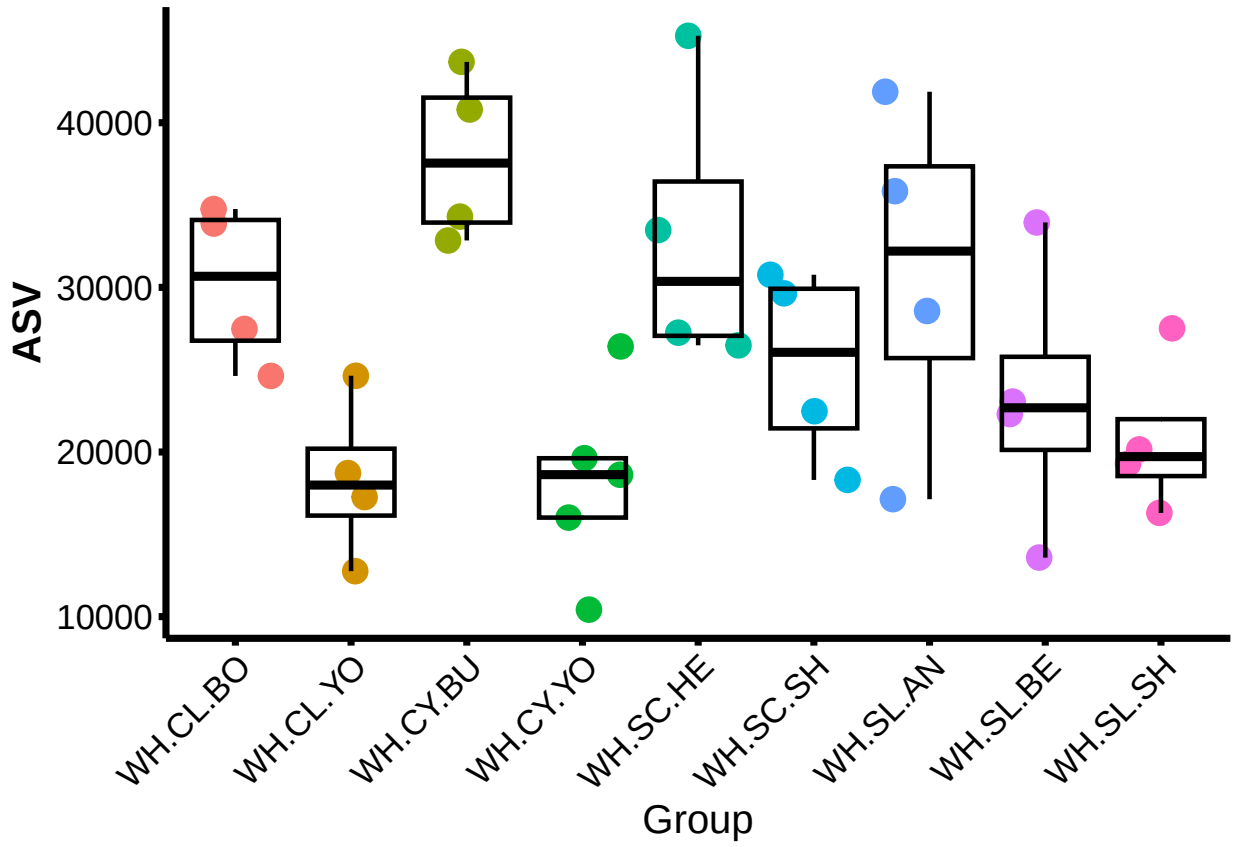
Assessing ASV Distribution (merged and filtered)

We evaluated the number of Amplicon Sequence Variants (ASVs) after merging and filtering, as well as visualized the raw read counts per sample within each group. This analysis provided insights into the

distribution of ASVs and the sequencing depth of each sample, allowing for a comprehensive assessment of data quality and quantity.

```
# install.packages("ggplot2")
library("ggplot2")
```

```
ASV <- sample_sums(physeq.a)
ASV <- as.data.frame(ASV)
ASV$Group <- physeq.a@sam_data$Group

ggplot(ASV, aes(x = Group, y = ASV,  colour = Group))+
  geom_point(alpha = 1, position = "jitter", size = 4) +
  geom_boxplot(alpha = 0, colour = "black", size = 0.8)+
  theme_classic() +
  theme(text = element_text(size=15, colour = "black"),
        axis.ticks = element_line(colour = "black", size = 1.25),
        axis.line = element_line(colour = 'black', size = 1.25),
        axis.text.x = element_text(angle=45, hjust=1, colour = "black", size = 13),
        axis.text.y = element_text(angle=0, hjust=0.5, colour = "black",size = 13),
        axis.title.y = element_text(color="black", size=15,face="bold"),
        legend.position = "none")
```



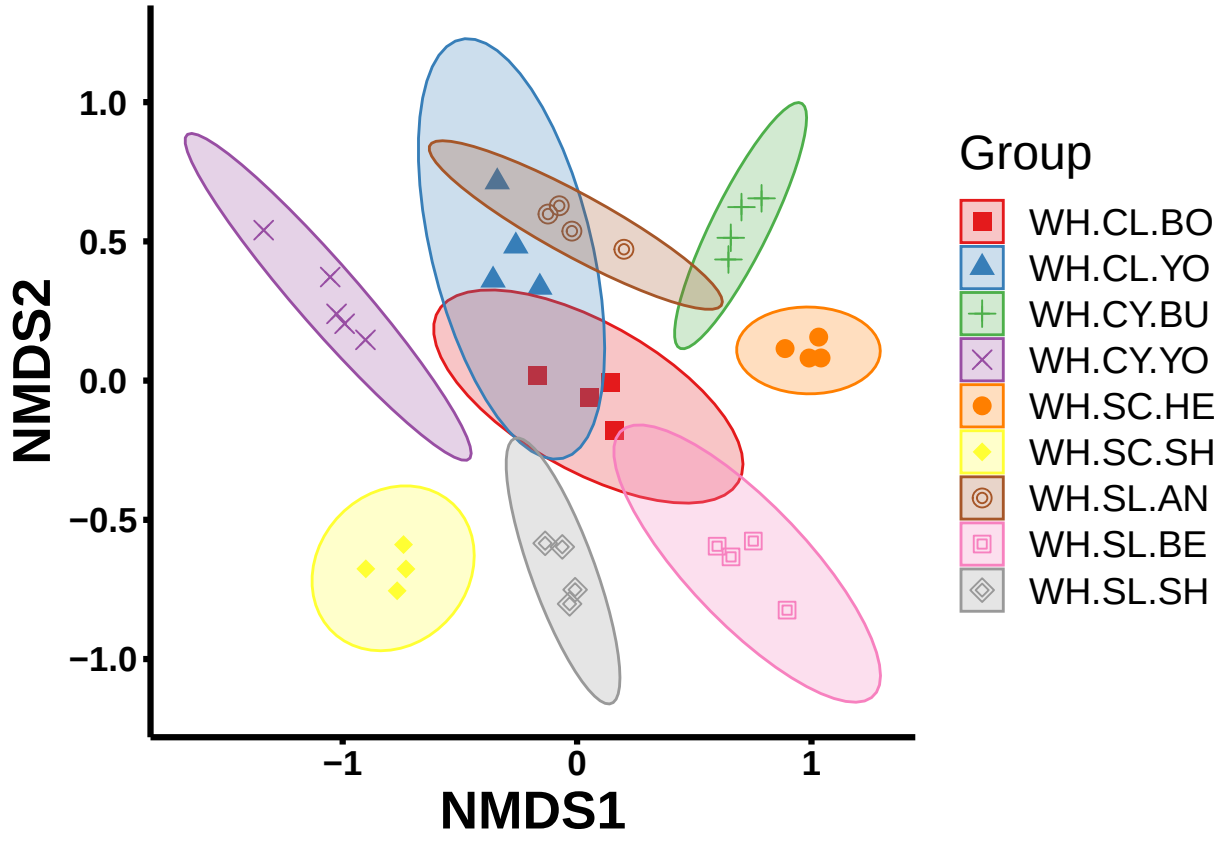
Beta diversity

Beta diversity is a metric used in microbial community studies to assess species composition dissimilarity between samples, providing insights into microbial community dynamics and their ecological significance. By quantifying variation in community structure and employing visualization techniques, beta diversity analysis allows comparisons across habitats or treatments, revealing key drivers of community variation and functional implications. It aids our understanding of microbial assemblages and their responses to environmental changes in ecology, environmental science, and human health research.

```
# method options: NMDS / PCoA
NMDS <- ordinate(physeq = physeq.a, method = "NMDS", distance = "bray")
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.196196
## Run 1 stress 0.196196
## ... Procrustes: rmse 4.397297e-06  max resid 1.073234e-05
## ... Similar to previous best
## Run 2 stress 0.2605251
## Run 3 stress 0.196196
## ... New best solution
## ... Procrustes: rmse 3.654596e-06  max resid 1.417036e-05
## ... Similar to previous best
## Run 4 stress 0.196196
## ... Procrustes: rmse 5.240983e-06  max resid 1.659334e-05
## ... Similar to previous best
## Run 5 stress 0.26306
## Run 6 stress 0.2301789
## Run 7 stress 0.2196345
## Run 8 stress 0.196196
## ... Procrustes: rmse 1.284066e-05  max resid 5.619169e-05
## ... Similar to previous best
## Run 9 stress 0.1929835
## ... New best solution
## ... Procrustes: rmse 0.1085369  max resid 0.2861591
## Run 10 stress 0.196196
## Run 11 stress 0.2095858
## Run 12 stress 0.1929835
## ... New best solution
## ... Procrustes: rmse 1.296255e-06  max resid 3.596235e-06
## ... Similar to previous best
## Run 13 stress 0.1929835
## ... Procrustes: rmse 2.109807e-06  max resid 3.777983e-06
## ... Similar to previous best
## Run 14 stress 0.196196
## Run 15 stress 0.223666
## Run 16 stress 0.2413254
## Run 17 stress 0.2819592
## Run 18 stress 0.196196
## Run 19 stress 0.2865937
## Run 20 stress 0.2110517
## *** Best solution repeated 2 times
```

```r
# Plot ordination
plot_ordination(
  physeq = physeq.a,
  ordination = NMDS,
  color = "Group",
  shape = "Group"
) +
  theme_classic() +
  geom_point(aes(color = Group), alpha = 1, size = 3) +
  theme(
    text = element_text(size = 18, colour = "black"),
    axis.ticks = element_line(colour = "black", size = 1.1),
    axis.line = element_line(colour = 'black', size = 1.1),
    axis.text.x = element_text(colour = "black", angle = 0, hjust = 0.5, size = 13, face = "bold"),
    axis.text.y = element_text(colour = "black", angle = 0, hjust = 0.5, size = 13, face = "bold"),
    axis.title.y = element_text(color = "black", size = 20, face = "bold"),
    axis.title.x = element_text(color = "black", size = 20, face = "bold")
  ) + stat_ellipse(geom = "polygon", type="norm", alpha=0.25, aes(fill = Group)) +
  scale_color_brewer(palette = "Set1") +
  scale_fill_brewer(palette = "Set1") +
  scale_shape_manual(values = c(15, 17, 3, 4, 16, 18, 21, 22, 23)) # Set custom shapes
```

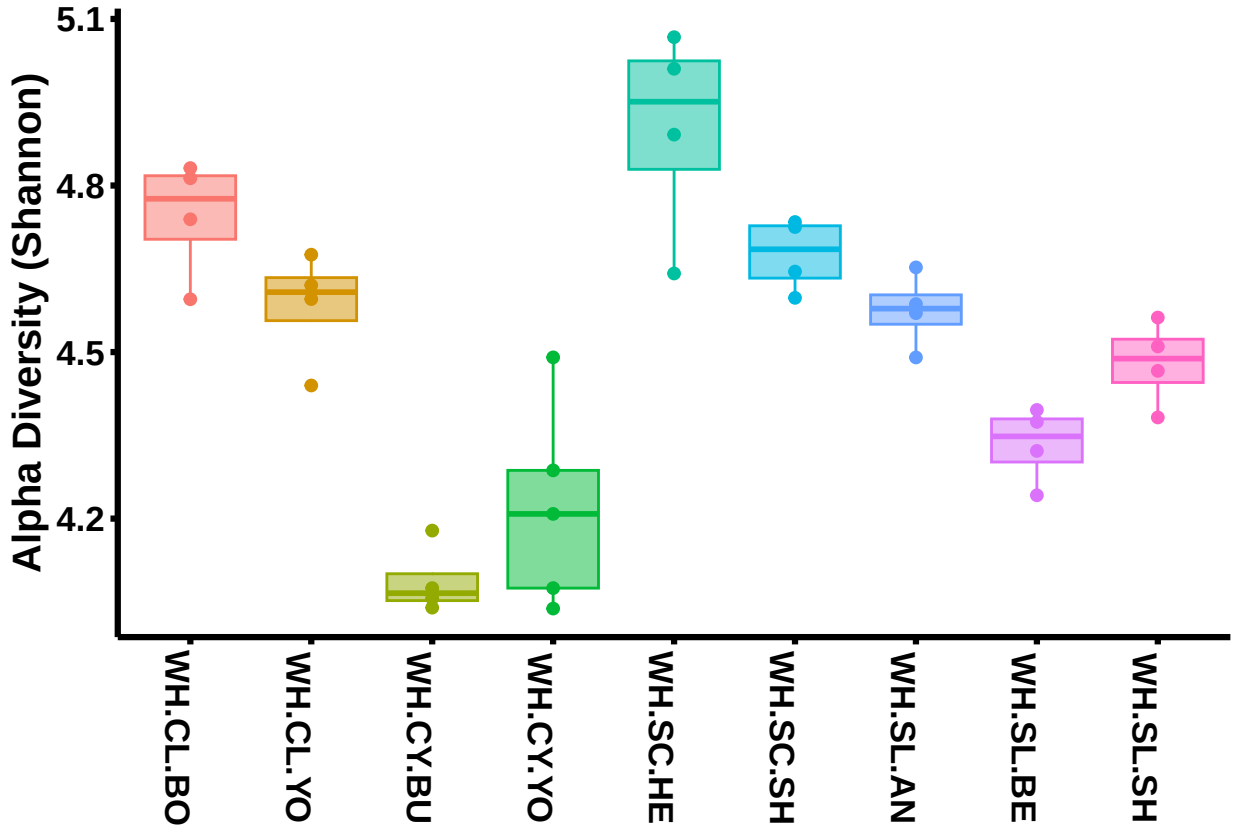



```r
# Clean up by removing objects that are no longer needed
rm(pcoa, physeq.W.B)
```

Alpha diversity

Alpha diversity is a fundamental concept in ecology and refers to the diversity or richness of species within a specific community or habitat. In the context of microbial ecology, alpha diversity represents the diversity of microorganisms within a given sample or microbiome. It provides insights into the variety and evenness of microbial species present in a particular environment. Common measures of alpha diversity include species richness, which counts the number of unique species, and evenness, which assesses the distribution of species abundances. Alpha diversity is crucial for understanding the stability, resilience, and functional potential of microbial communities. It can be influenced by various factors, including environmental conditions, host factors, and perturbations. By comparing alpha diversity across different samples or experimental groups, researchers can gain insights into the impact of factors such as disease, habitat changes, or interventions on microbial community structure.

```r
# available measurements [c("Observed", "Chao1", "ACE", "Shannon", "Simpson", "InvSimpson", "Fisher")]
alpha.object <- cbind(
  x = sample_data(physeq.a),
  y = estimate_richness(physeq.a, measures = 'Shannon')
)

# Create a plot for alpha diversity
ggplot(data = alpha.object, aes(x = x.Group, y = Shannon, color = x.Group, fill = x.Group)) +
  theme_classic() +
  labs(
    x = element_blank(),                    # No x-axis label
    y = "Alpha Diversity (Shannon)"         # y-axis label
  ) +
  geom_point(size = 1.75) +                 # Add points
  geom_boxplot(alpha = 0.5) +               # Add boxplot with transparency
  theme(
    text = element_text(size = 18, colour = "black"),
    axis.ticks = element_line(colour = "black", size = 1.1),
    axis.line = element_line(colour = 'black', size = 1.1),
    axis.text.x = element_text(colour = "black", angle = 270, size = 13, face = "bold"),
    axis.text.y = element_text(angle = 0, hjust = 0, colour = "black", size = 13, face = "bold"),
    axis.title.y = element_text(color = "black", size = 15, face = "bold"),
    legend.position = "none"                # Hide legend
  )
```




```
# Clean up by removing the alpha.object
rm(alpha.object)
```

Pairwise comparison using PERMANOVA

Pairwise PERMANOVA is a statistical method used to compare multiple groups or treatments in ecological and microbial community studies. It assesses dissimilarity between samples and provides a p-value to determine the significance of observed differences. This approach is valuable for targeted group comparisons, allowing researchers to investigate the effects of specific factors on microbial communities and uncover significant variations in community composition. By considering within- and between-group variation, pairwise PERMANOVA provides robust statistical analysis and insights into microbial community dynamics and functioning.

```
# devtools::install_github("pmartinezarbizu/pairwiseAdonis/pairwiseAdonis")
library("pairwiseAdonis")
```

```
metdat = as.data.frame(as.matrix(physeq.a@sam_data))
dat = as.data.frame(t(as.data.frame(physeq.a@otu_table)))
pairwise.adonis(dat, metdat$Group, sim.function = "vegdist",
                sim.method = "bray", p.adjust.m = "bonferroni",
                reduce = NULL, perm = 100000)
```

```
##                  pairs Df SumsOfSqs   F.Model        R2     p.value p.adjusted
## 1  WH.CL.BO vs WH.CL.YO  1 0.6836316  7.141412 0.5434281 0.028571429  1.0000000
```

```
## 2  WH.CL.BO vs WH.CY.BU  1 0.8424259 11.626774 0.6596087 0.028571429  1.0000000
## 3  WH.CL.BO vs WH.CY.YO  1 0.9402290  8.830922 0.5578274 0.007949921  0.2861971
## 4  WH.CL.BO vs WH.SC.HE  1 0.7165156  7.172596 0.5445089 0.028571429  1.0000000
## 5  WH.CL.BO vs WH.SC.SH  1 0.7061539  5.851479 0.4937341 0.028571429  1.0000000
## 6  WH.CL.BO vs WH.SL.AN  1 0.5239285  4.907670 0.4499284 0.028571429  1.0000000
## 7  WH.CL.BO vs WH.SL.BE  1 0.6106859  4.633587 0.4357501 0.028571429  1.0000000
## 8  WH.CL.BO vs WH.SL.SH  1 0.6328513  6.156368 0.5064315 0.028571429  1.0000000
## 9  WH.CL.YO vs WH.CY.BU  1 1.0094973 16.080161 0.7282629 0.028571429  1.0000000
## 10 WH.CL.YO vs WH.CY.YO  1 0.7288589  7.424016 0.5146983 0.008159918  0.2937571
## 11 WH.CL.YO vs WH.SC.HE  1 0.8085318  8.961816 0.5989792 0.028571429  1.0000000
## 12 WH.CL.YO vs WH.SC.SH  1 0.7401712  6.668033 0.5263669 0.028571429  1.0000000
## 13 WH.CL.YO vs WH.SL.AN  1 0.5871301  6.047873 0.5019868 0.028571429  1.0000000
## 14 WH.CL.YO vs WH.SL.BE  1 0.7148304  5.853562 0.4938230 0.028571429  1.0000000
## 15 WH.CL.YO vs WH.SL.SH  1 0.7150048  7.678351 0.5613506 0.028571429  1.0000000
## 16 WH.CY.BU vs WH.CY.YO  1 1.2795885 16.357105 0.7003053 0.007709923  0.2775572
## 17 WH.CY.BU vs WH.SC.HE  1 0.6135266  9.164290 0.6043336 0.028571429  1.0000000
## 18 WH.CY.BU vs WH.SC.SH  1 0.9171570 10.454216 0.6353518 0.028571429  1.0000000
## 19 WH.CY.BU vs WH.SL.AN  1 0.7116171  9.641420 0.6164031 0.028571429  1.0000000
## 20 WH.CY.BU vs WH.SL.BE  1 0.8752828  8.854946 0.5960941 0.028571429  1.0000000
## 21 WH.CY.BU vs WH.SL.SH  1 0.9093068 13.018466 0.6845171 0.028571429  1.0000000
## 22 WH.CY.YO vs WH.SC.HE  1 1.2661487 12.443868 0.6399893 0.008059919  0.2901571
## 23 WH.CY.YO vs WH.SC.SH  1 0.8514764  7.121568 0.5043043 0.008159918  0.2937571
## 24 WH.CY.YO vs WH.SL.AN  1 0.8690449  8.074413 0.5356370 0.008609914  0.3099569
## 25 WH.CY.YO vs WH.SL.BE  1 0.9576115  7.418112 0.5144995 0.007869921  0.2833172
## 26 WH.CY.YO vs WH.SL.SH  1 0.9647977  9.256035 0.5693907 0.007969920  0.2869171
## 27 WH.SC.HE vs WH.SC.SH  1 0.8233765  7.149139 0.5436964 0.028571429  1.0000000
## 28 WH.SC.HE vs WH.SL.AN  1 0.6725153  6.642198 0.5253990 0.028571429  1.0000000
## 29 WH.SC.HE vs WH.SL.BE  1 0.5757410  4.558976 0.4317631 0.028571429  1.0000000
## 30 WH.SC.HE vs WH.SL.SH  1 0.7799919  8.017346 0.5719589 0.028571429  1.0000000
## 31 WH.SC.SH vs WH.SL.AN  1 0.6679211  5.473318 0.4770475 0.028571429  1.0000000
## 32 WH.SC.SH vs WH.SL.BE  1 0.6649246  4.521124 0.4297187 0.028571429  1.0000000
## 33 WH.SC.SH vs WH.SL.SH  1 0.6153668  5.211822 0.4648506 0.028571429  1.0000000
## 34 WH.SL.AN vs WH.SL.BE  1 0.6718066  5.045557 0.4567952 0.028571429  1.0000000
## 35 WH.SL.AN vs WH.SL.SH  1 0.8360346  8.027306 0.5722628 0.028571429  1.0000000
## 36 WH.SL.BE vs WH.SL.SH  1 0.6155645  4.764900 0.4426330 0.028571429  1.0000000
##    sig
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
```

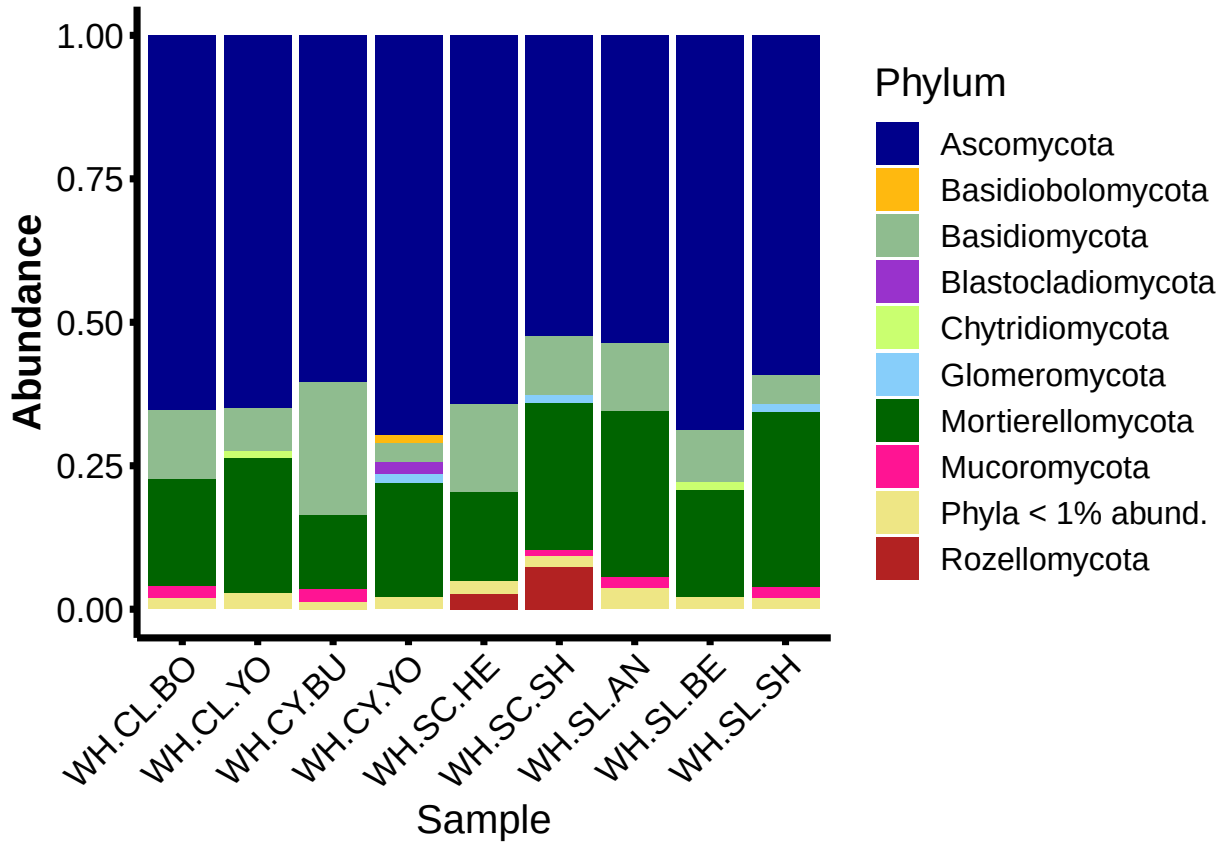
```
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30
## 31
## 32
## 33
## 34
## 35
## 36
```

Bar plot composition

```
# install.packages("plyr")
library("plyr")
```

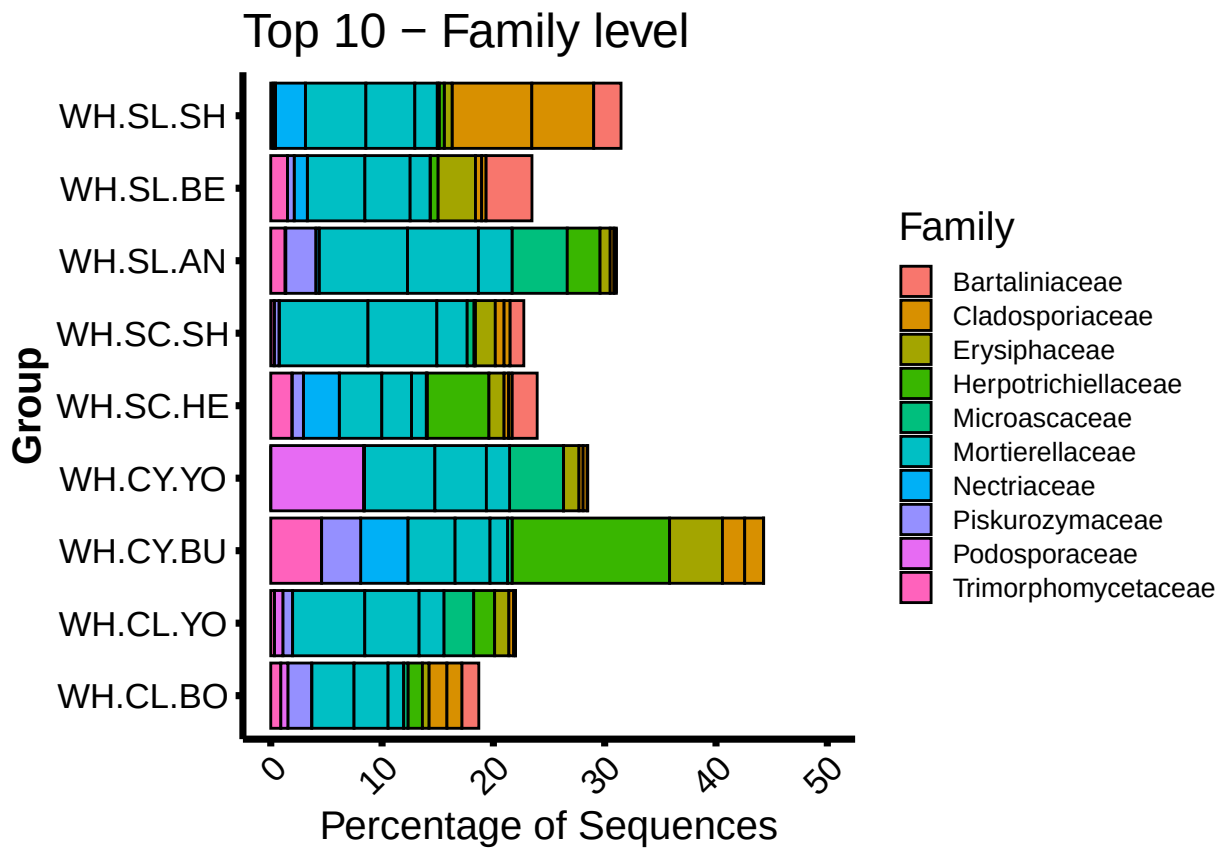
```
# https://github.com/joey711/phyloseq/issues/901

## (3A) Merge the replicate samples for each Group
physeq.a.group = merge_samples(physeq.a, "Group") # Sum between replicate samples

# (3B) repair factors in the sample metadata
sample_data(physeq.a.group)$Group <- levels(sample_data(physeq.a)$Group)[get_variable(physeq.a.group, "
sample_data(physeq.a.group)$Soil <- levels(sample_data(physeq.a)$Soil)[get_variable(physeq.a.group, "So

# Filter taxa with mean abundance > 0.1
physeq2 = filter_taxa(physeq.a.group, function(x) mean(x) > 0.1, TRUE)

# Transform sample counts to relative abundances
physeq3 = transform_sample_counts(physeq2, function(x) x / sum(x))

# Aggregate taxa at the Phylum level
glom <- tax_glom(physeq3, taxrank = 'Phylum')

# Convert the aggregated phyloseq object to a dataframe
data <- psmelt(glom)
# Convert Phylum column to character
data$Phylum <- as.character(data$Phylum)

# Rename phyla with abundance < 1% to "< 1% abund."
data$Phylum[data$Abundance < 0.01] <- "< 1% abund."

# Calculate medians of abundance for each Phylum
medians <- ddply(data, ~Phylum, function(x) c(median = median(x$Abundance)))
```

```
# Extract phyla with median abundance <= 0.01
remainder <- medians[medians$median <= 0.01, ]$Phylum

# Rename selected phyla to "Phyla < 1% abund."
data[data$Phylum %in% remainder, ]$Phylum <- "Phyla < 1% abund."

# Rename phyla with abundance < 1% to "Phyla < 1% abund."
data$Phylum[data$Abundance < 0.01] <- "Phyla < 1% abund."

# Plot the data with condensed phyla categories
p <- ggplot(data = data, aes(x = Sample, y = Abundance, fill = Phylum))
p + geom_bar(aes(), stat = "identity", position = "stack") +
  scale_fill_manual(values = c("darkblue", "darkgoldenrod1", "darkseagreen", "darkorchid", "darkolivegr
  theme(text = element_text(size=15, colour = "black"),
        axis.ticks = element_line(colour = "black", size = 1.25),
        axis.line = element_line(colour = 'black', size = 1.25),
        axis.text.x = element_text(angle=45, hjust=1, colour = "black", size = 13),
        axis.text.y = element_text(angle=0, hjust=0.5, colour = "black",size = 13),
        axis.title.y = element_text(color="black", size=15,face="bold"), legend.position = "right") # +
```



Plotting the top 10 taxa at family level

```r
## Transform normalised ASVs to proportions
proportions = transform_sample_counts(physeq.a.group, function(x) 100 * x/sum(x))

##
Top30ASVs = names(sort(taxa_sums(proportions), TRUE)[1:13])
Taxtab30 = cbind(tax_table(proportions), Family30 = NA)
Taxtab30[Top30ASVs, "Family30"] <- as(tax_table(proportions)[Top30ASVs, "Family"], "character")
tax_table(proportions) <- tax_table(Taxtab30)

Rgsm30 = prune_taxa(Top30ASVs, proportions)

# plotting
title = "Top 10 - Family level"
plot_bar(Rgsm30, "Group", fill = "Family", title = title) +
  coord_flip() +
  ylab("Percentage of Sequences") + ylim(0, 50) +  theme_classic() +
  theme(text = element_text(size=15, colour = "black"),
        axis.ticks = element_line(colour = "black", size = 1.25),
        axis.line = element_line(colour = 'black', size = 1.25),
        axis.text.x = element_text(angle=45, hjust=1, colour = "black", size = 13),
        axis.text.y = element_text(angle=0, hjust=0.5, colour = "black",size = 13),
        axis.title.y = element_text(color="black", size=15,face="bold"),
        legend.position = "right",
        legend.text = element_text(size = 9.5),
        legend.key.height= unit(0.45, 'cm'),
        legend.key.width= unit(0.45, 'cm')
        )
```




```
# Clean up by removing unnecessary objects
# rm(proportions, Top30ASVs, Taxtab30, Rgsm30, title)
```

Upset plot using UpsetR

When it comes to representing sets visually, the go-to option is usually a Venn diagram. These diagrams work well when dealing with up to five sets, providing a clear visualisation. However, as the dataset expands, such as when dealing with five sets, deriving the desired insights from the diagram becomes more complex. As a result, considering an UpSet graph for data visualisation becomes an appealing choice. UpSet graphs offer a more streamlined way to display intersections and complements, particularly when dealing with larger datasets or multiple sets. This option ensures a more intuitive and informative representation of the data.

```
library("UpSetR")
library("reshape2")
library("plyr")
library("dplyr")
# BiocManager::install("microbiome")
library("microbiome")
```

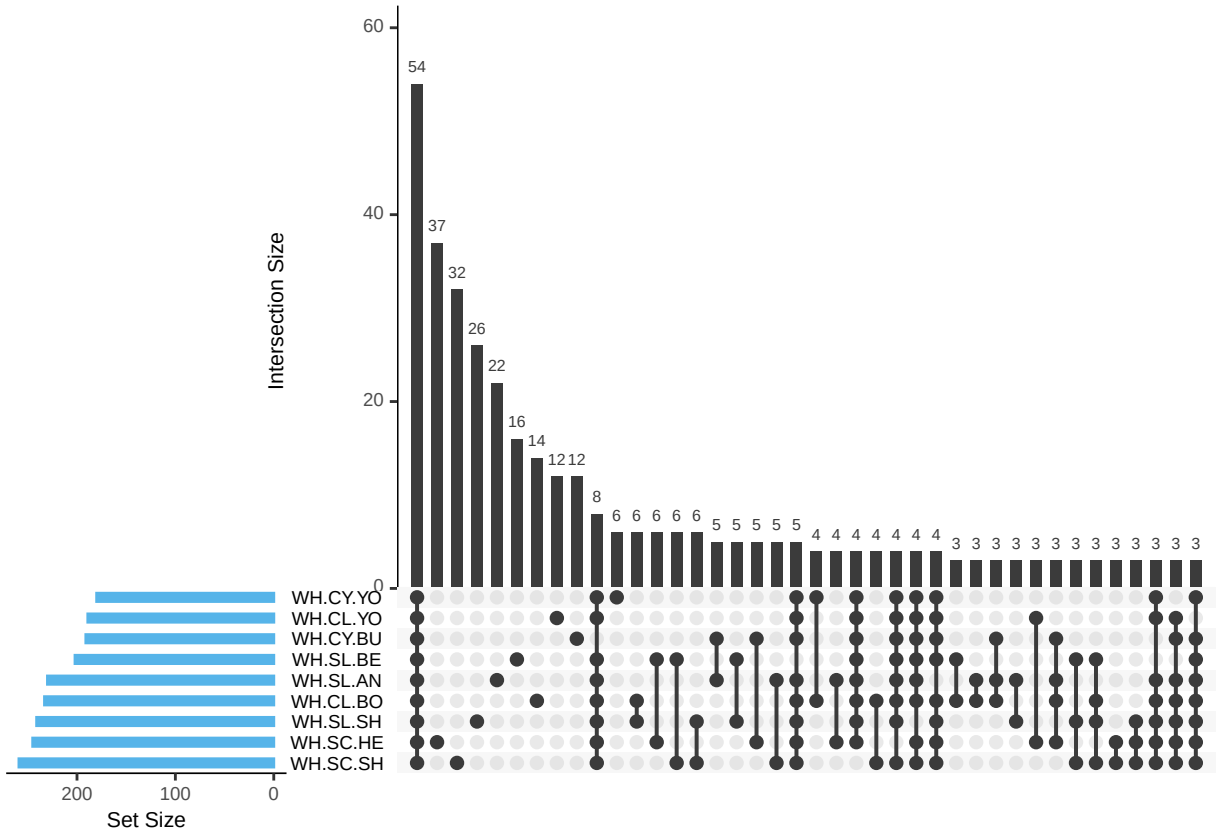
```
# Aggregate taxa at the genus level
B <- aggregate_taxa(physeq.a.group, "Genus", verbose = TRUE)
```

```
## [1] "Remove taxonomic information below the target level"
## [1] "Mark the potentially ambiguous taxa"
```

```
## [1] "-- split"
## [1] "-- sum"
## [1] "Create phyloseq object"
## [1] "Remove ambiguous levels"
## [1] "-- unique"
## [1] "-- Rename the lowest level"
## [1] "-- rownames"
## [1] "-- taxa"
## [1] "Convert to taxonomy table"
## [1] "Combine OTU and Taxon matrix into Phyloseq object"
## [1] "Add the metadata as is"
```

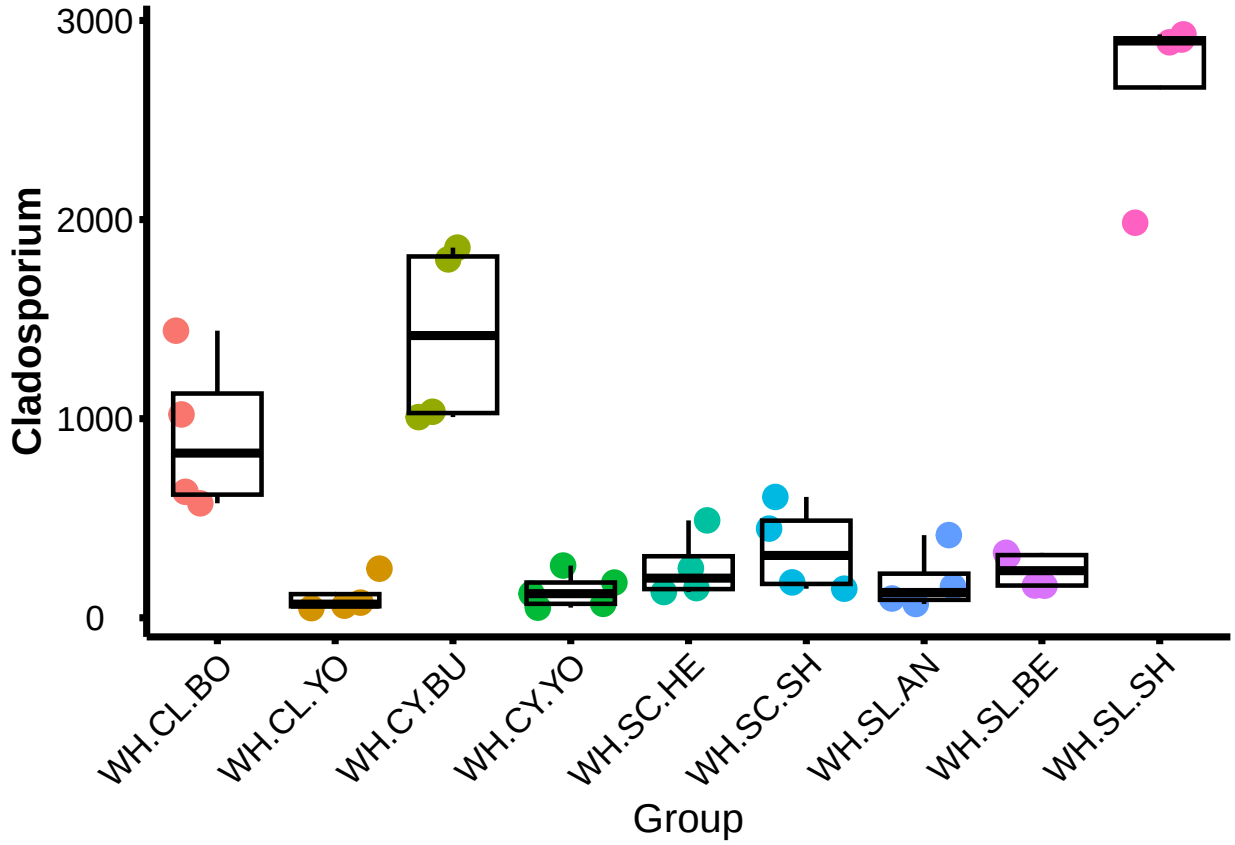
```r
# Remove undesired genera
# B2 <- subset_taxa(B, !get("Genus") %in% c("uncultured", "Unknown"))

# Remove unwanted taxon names
taxa_to_remove <- c("uncultured", "Unknown")
B2 <- subset_taxa(B, !get("Genus") %in% taxa_to_remove)

# Extract relevant data from the phyloseq object
sample_data <- sample_data(B2)
otu_table <- otu_table(B2)
abundance <- as.vector(otu_table)

# Create a tibble with the extracted data
D <- tibble(
  Sample = rep(sample_data$Group, each = nrow(otu_table)),
  ASV = rep(rownames(otu_table), times = ncol(otu_table)),
  Abundance = abundance
) %>%
  group_by(Sample) %>%
  mutate(rank = rank(desc(Abundance))) %>%
   filter(Abundance > 0) %>%
  ungroup() %>%
  select(Sample, Abundance, ASV)

# Remove the Abundance column
D$Abundance <- NULL

# Rename the second column to "ASV"
names(D)[2] <- "ASV"

# Convert data from long to wide format
E <- dcast(D, ASV ~ Sample)

# Define a binary function
binary_fun <- function(x) {
  x[is.na(x)] <- 0
  ifelse(x > 0, 1, 0)
}

# Apply the binary function to columns 2 to 10
temp_df <- apply(E[2:10], 2, binary_fun)
temp_df <- as.data.frame(temp_df)
```

```r
# Create an UpSet plot
upset_plot <- upset(temp_df, sets = colnames(temp_df), sets.bar.color = "#56B4E9",
                    order.by = "freq", empty.intersections = "on")

# Print the UpSet plot
print(upset_plot)
```




```r
# Plotting for the abundance of one specific fungi
# Subset the taxa to Genus from physeq.wheat
physeq.a.genus <- subset_taxa(physeq.a, Genus == "Cladosporium")

# Calculate the total abundance of Cladosporium for each sample
meta = physeq.a.genus@sam_data
otudf = as.data.frame(t(as.data.frame(physeq.a.genus@otu_table)))
meta$Cladosporium = rowSums(otudf)

# Plot a graph of the abundance of Cladosporium for each sample grouped by Group:
ggplot(subset(meta, Group %in% c("WH.CL.BO","WH.CL.YO",
                                 "WH.CY.BU","WH.CY.YO",
                                 "WH.SC.HE","WH.SC.SH",
                                 "WH.SL.AN","WH.SL.BE",
                                 "WH.SL.SH")),
       aes(x = Group, y = Cladosporium,  colour = interaction(Group))) +
  geom_point(alpha = 1, position = "jitter", size = 4) +
  geom_boxplot(alpha = 0, colour = "black", size = 0.8)+
```

```
  theme_classic() +
  theme(text = element_text(size=15, colour = "black"),
        axis.ticks = element_line(colour = "black", size = 1.25),
        axis.line = element_line(colour = 'black', size = 1.25),
        axis.text.x = element_text(angle=45, hjust=1, colour = "black", size = 13),
        axis.text.y = element_text(angle=0, hjust=0.5, colour = "black",size = 13),
        axis.title.y = element_text(color="black", size=15,face="bold"), legend.position = "none")
```



###Phylogenetic tree Phylogenetic trees are branching diagrams that show the evolutionary relationships between different species or other entities. They are constructed by comparing the organisms' DNA, proteins, or other features. The more similar the features, the more closely related the organisms are thought to be. It can be used to answer a variety of questions about evolution, such as how different species are related to each other, when different species diverged from each other, and what are the common ancestors of different species.

```
# Load required libraries
library(phyloseq)    # For handling phylogenetic sequencing data
library(ggtree)      # For tree visualization
library(scales)      # For scaling transformations
```

```
# Load the GlobalPatterns dataset and prune taxa
GP <- prune_taxa(taxa_sums(physeq.a) > 0, physeq.a)  # Remove taxa with zero sums
GP.chl <- subset_taxa(GP, Genus == "Cladosporium")   # Subset data based on Phylum value

# Create a ggtree plot
```

```
p <- ggtree(GP.chl, ladderize = FALSE) +
  geom_text2(aes(subset = !isTip, label = label), hjust = -0.5, size = 4) +
  geom_tiplab(aes(label = Genus), as_ylab=TRUE) +
  geom_point(aes(x = x + hjust, color = Group,
                 shape = Group, size = Abundance), na.rm = TRUE) +
  scale_size_continuous(trans = log_trans(10)) +
  scale_shape_manual(values = c(15, 17, 3, 4, 16, 18, 21, 22, 23)) + # Set custom
  theme(legend.position = "bottom") +
  ggtitle("Reproduce Phyloseq by ggtree")

# Print the ggtree plot
print(p)
```