PhyloseqMRAManuscript

Tricia

2024-05-25

##Load required packages

```
library(ggplot2)
library(phyloseq)
library(vegan)
library(dplyr)
library(plyr)
library(decontam)
library(ANCOMBC) #differential taxa expression
library(MicEco) #psvenn
library(BiMiCo) #rmnonbac
library(ggpubr) #statcomparemeans
```

##Load taxa and seqtab files to start here

```
load("RData/taxa.RData")
load("RData/seqtab.nochim.RData")
```

##import metadata

```
metadata<-read.csv("metadata.csv", header=TRUE, row.names = 1)
```

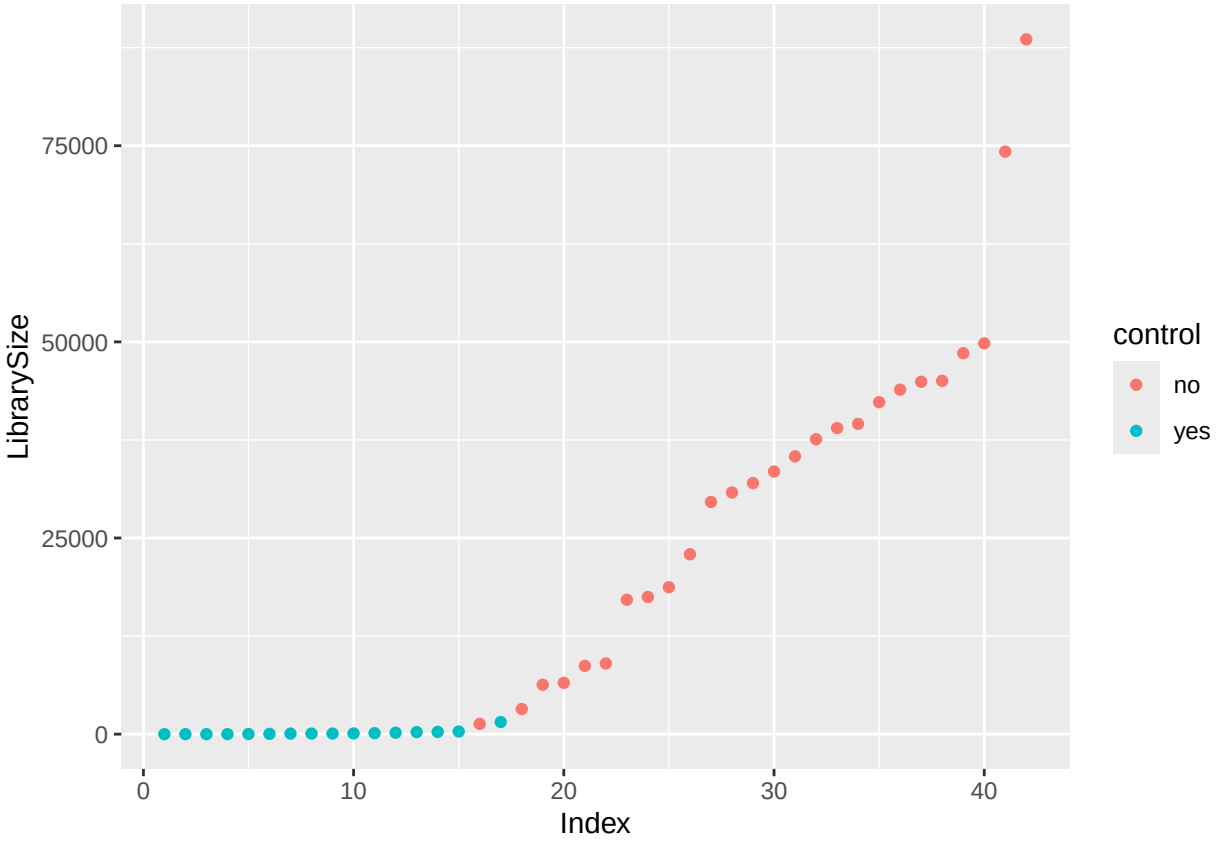
##Create phyloseq object

```
#make sure the seqtab.nochim and taxa objects are loaded
physeq <- phyloseq(otu_table(seqtab.nochim, taxa_are_rows=FALSE),
               sample_data(metadata),
               tax_table(taxa))
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 7525 taxa and 42 samples ]
## sample_data() Sample Data:       [ 42 samples by 6 sample variables ]
## tax_table()   Taxonomy Table:    [ 7525 taxa by 7 taxonomic ranks ]
```

##inspect library sizes

```
df <- as.data.frame(sample_data(physeq)) # Put sample_data into a ggplot-friendly data.frame
df$LibrarySize <- sample_sums(physeq)
df <- df[order(df$LibrarySize),]
df$Index <- seq(nrow(df))
ggplot(data=df, aes(x=Index, y=LibrarySize, color=control)) + geom_point()
```



##identify contaminants

```
sample_data(physeq)$is.neg <- sample_data(physeq)$control == "yes"
contamdf.prev <- isContaminant(physeq, method="prevalence", neg="is.neg", threshold=0.5) #identify cont
table(contamdf.prev$contaminant)
```

```
##
## FALSE  TRUE
##  7507    18
```

```
head(which(contamdf.prev$contaminant))
```

```
## [1]  68 100 118 320 393 503
```

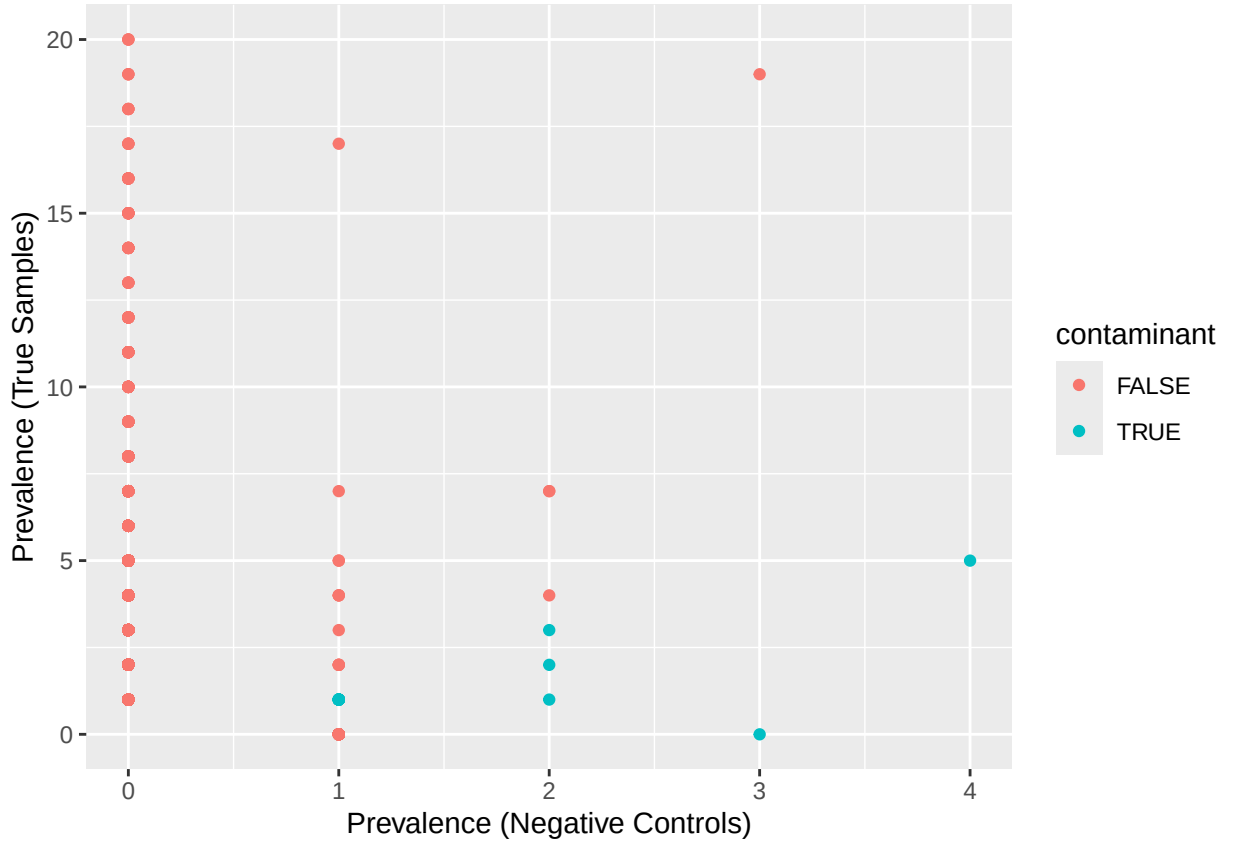
##remove control samples

```
# Make phyloseq object of presence-absence in negative controls and true samples
physeq.pa <- transform_sample_counts(physeq, function(abund) 1*(abund>0))
physeq.pa.neg <- prune_samples(sample_data(physeq.pa)$control == "yes", physeq.pa)
physeq <- prune_samples(sample_data(physeq.pa)$control == "no", physeq.pa) #this will contain positives
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 7525 taxa and 26 samples ]
## sample_data() Sample Data:       [ 26 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 7525 taxa by 7 taxonomic ranks ]
```

##graph controls

```
df.pa <- data.frame(pa.pos=taxa_sums(physeq), pa.neg=taxa_sums(physeq.pa.neg),
                    contaminant=contamdf.prev$contaminant)
ggplot(data=df.pa, aes(x=pa.neg, y=pa.pos, color=contaminant)) + geom_point() +
  xlab("Prevalence (Negative Controls)") + ylab("Prevalence (True Samples)")
```



##remove contaminants

```
physeq <- prune_taxa(!contamdf.prev$contaminant, physeq)
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 7507 taxa and 26 samples ]
## sample_data() Sample Data:       [ 26 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 7507 taxa by 7 taxonomic ranks ]
```

##Remove mock community

```
physeq <- subset_samples(physeq, mock != "yes")
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 7507 taxa and 22 samples ]
## sample_data() Sample Data:       [ 22 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 7507 taxa by 7 taxonomic ranks ]
```

##Remove the sequence itself and replace with ASV

```
dna <- Biostrings::DNAStringSet(taxa_names(physeq))
names(dna) <- taxa_names(physeq)
```

```
physeq <- merge_phyloseq(physeq, dna)
taxa_names(physeq) <- paste0("ASV", seq(ntaxa(physeq)))
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 7507 taxa and 22 samples ]
## sample_data() Sample Data:       [ 22 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 7507 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 7507 reference sequences ]
```

##remove mitochondria and chloroplast matches.

```
physeq <- physeq %>% subset_taxa( Family!= "Mitochondria" | is.na(Family) & Order!="Chloroplast" | is.na
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 6461 taxa and 22 samples ]
## sample_data() Sample Data:       [ 22 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 6461 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 6461 reference sequences ]
```

##remove all non bacterial sequences

```
physeq<-rm_nonbac(physeq)
physeq
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 6442 taxa and 22 samples ]
## sample_data() Sample Data:       [ 22 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 6442 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 6442 reference sequences ]
```

##save physeq object as a file

```
##save physeq object as R file
save(physeq, file="RData/physeq.RData")
```

##load physeq

```
load("RData/physeq.RData")
```

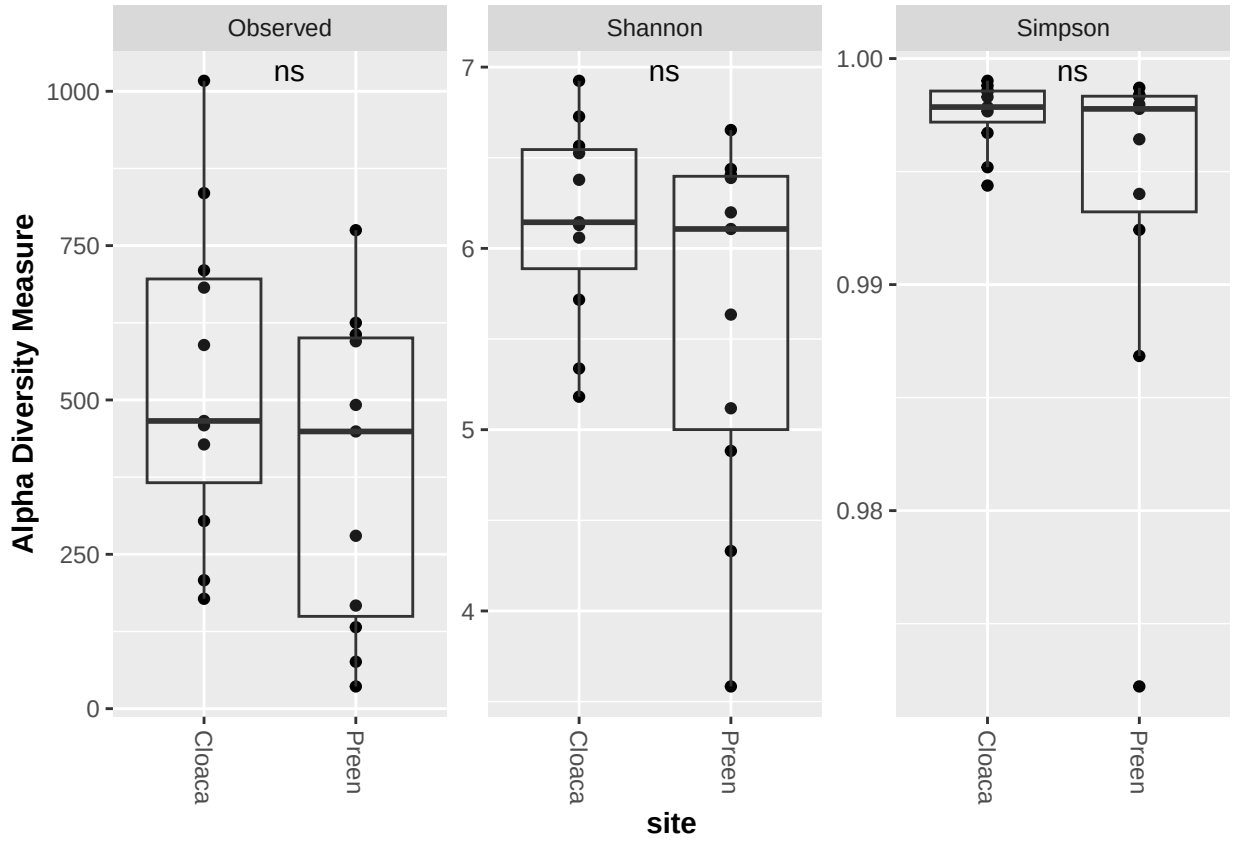
##Alpha Diversity based on site with stats

```
p=plot_richness(physeq,x="site", measures=c("Observed","Simpson", "Shannon"))
BAR <- p + geom_boxplot(data = p$data, aes(x = site, y = value, color = NULL), alpha = 0.1) + theme(axis

bar <- BAR + stat_compare_means(aes(label = ifelse(..p.signif.. < 0.05, ..p.signif.., "")), method = "w

barsite<-bar+ stat_compare_means(aes(label = ..p.signif..), method = "wilcox.test",  label.x = 1.5)

barsite
```



##export tiff with 300dpi

```
ggsave(
  filename="figures/Figure01AlphaDiv.tiff",
  plot = barsite,
  width = 200,
  height = 200,
  units = c("mm"),
  dpi = 300,
)
```

##Export alpha diveristy

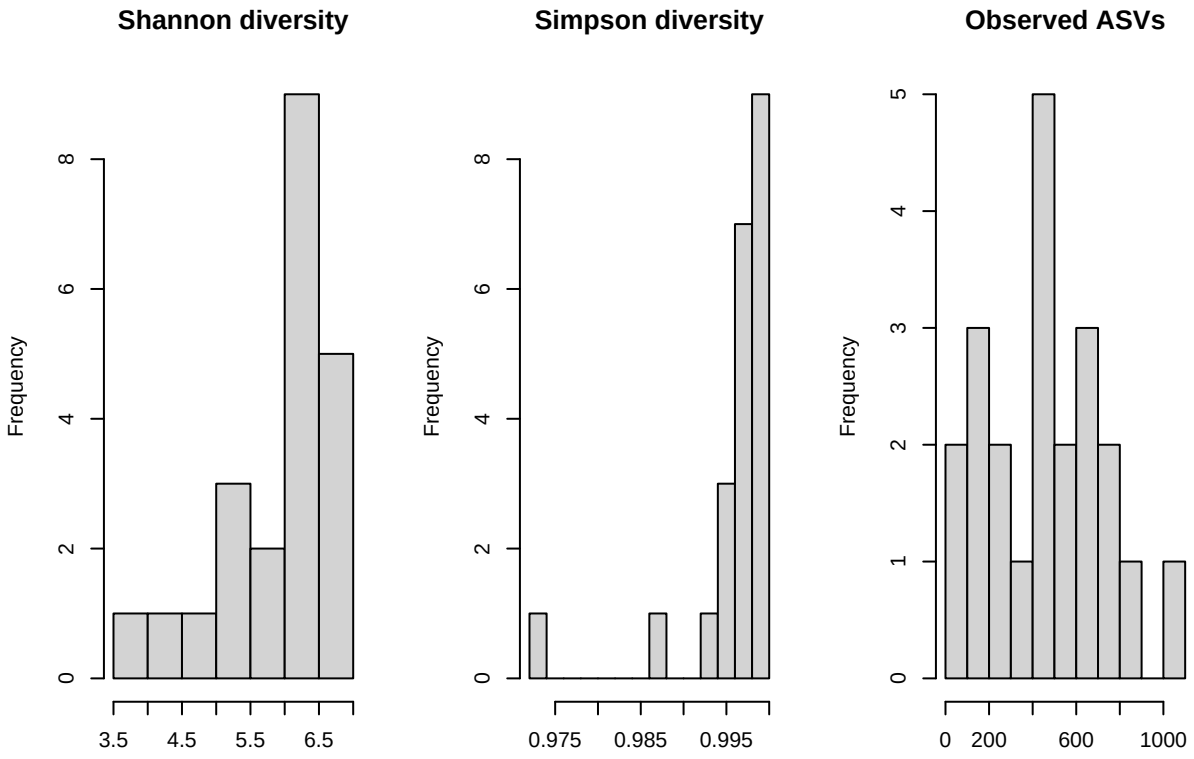
```
alphadiv<-estimate_richness(physeq, measures=c("Observed", "Shannon", "Simpson"))
write.csv(alphadiv, "alphasheets/alpha_div.csv")
```

##add site info to alpha_div file and rename alphadiv import for normality testing

```
meta<-read.csv("alphasheets/alphadiv.csv")
```

##hist

```
par(mfrow = c(1, 3))
hist(meta$Shannon, main="Shannon diversity", xlab="", breaks=10)
hist(meta$Simpson, main="Simpson diversity", xlab="", breaks=10)
hist(meta$Observed, main="Observed ASVs", xlab="", breaks=10)
```



##Test for normality (Shapiro)

```
shapiro.test(meta$Shannon)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  meta$Shannon
## W = 0.88431, p-value = 0.01461
```

```
shapiro.test(meta$Simpson)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  meta$Simpson
## W = 0.55405, p-value = 4.409e-07
```

```
shapiro.test(meta$Observed) #normal
```

```
##
##  Shapiro-Wilk normality test
##
## data:  meta$Observed
## W = 0.97141, p-value = 0.7434
```

##Two factor tests

```
##site
wilcox.test(meta$Simpson ~ meta$site)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  meta$Simpson by meta$site
## W = 79, p-value = 0.2426
## alternative hypothesis: true location shift is not equal to 0
```

```
t.test(meta$Observed ~ meta$site)
```

```
##
##  Welch Two Sample t-test
##
## data:  meta$Observed by meta$site
## t = 1.3524, df = 19.99, p-value = 0.1914
## alternative hypothesis: true difference in means between group Cloaca and group Preen is not equal t
## 95 percent confidence interval:
##  -81.12992 380.22083
## sample estimates:
## mean in group Cloaca  mean in group Preen
##             534.2727             384.7273
```

```
wilcox.test(meta$Shannon ~ meta$site)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  meta$Shannon by meta$site
## W = 79, p-value = 0.2426
## alternative hypothesis: true location shift is not equal to 0
```

##Remove taxa with relative abundance <0.005%

```
minTotRelAbun = .00005
x = taxa_sums(physeq)
keepTaxa = (x / sum(x)) > minTotRelAbun
physeqprune = prune_taxa(keepTaxa, physeq)
physeqprune
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 6376 taxa and 22 samples ]
## sample_data() Sample Data:       [ 22 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 6376 taxa by 7 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 6376 reference sequences ]
```

##save physeq object as a file

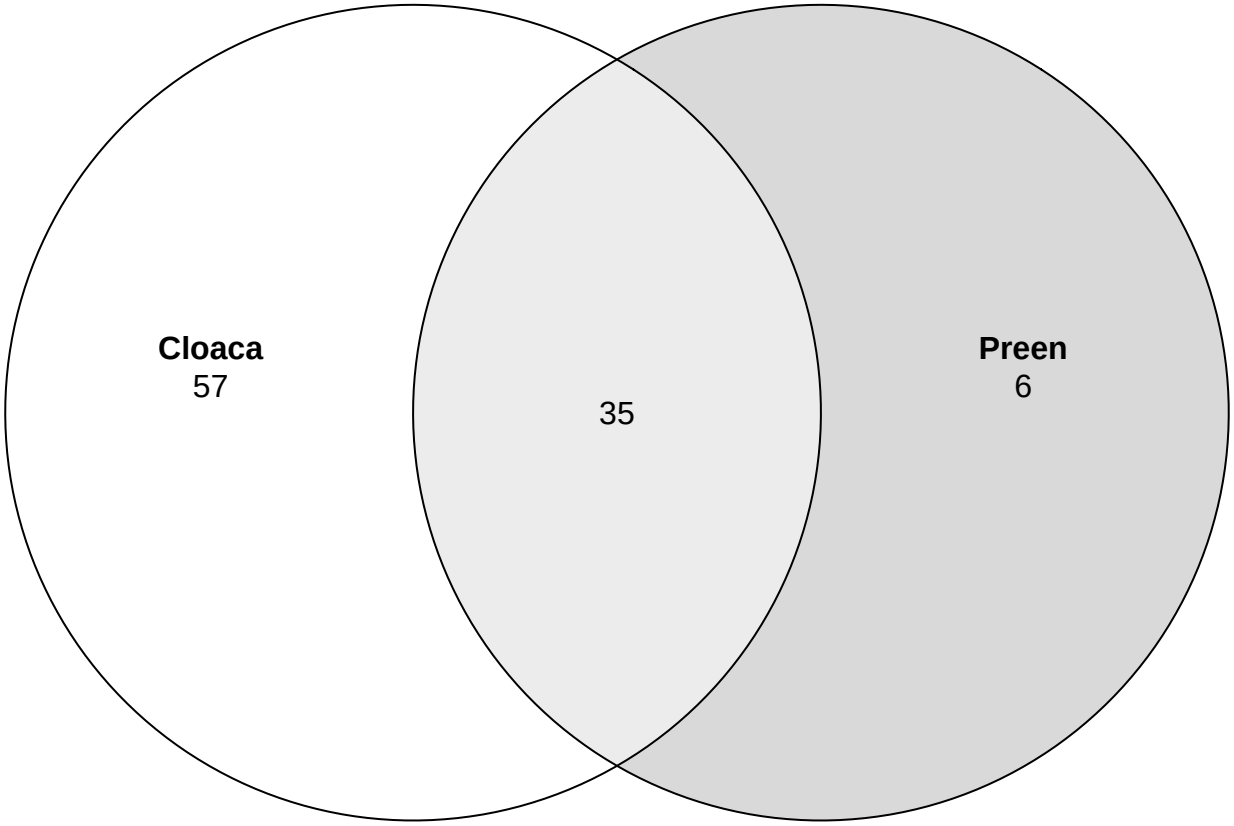
```
##save physeq object as R file
save(physeqprune, file="RData/physeqprune.RData")
```

##load physeq

```
load("RData/physeqprune.RData")
```

##Number of shared ASVs site (found in 50% or more)

```
sitevenn=ps_venn(
  physeqprune,
  "site",
  fraction = .50,
  weight = FALSE,
  relative = TRUE,
  plot = TRUE
)
sitevenn
```



##List of shared ASVs species (found in 50% or more) t=0

```
sitelist=ps_venn(
  physeqprune,
  "site",
  fraction = .5,
  weight = FALSE,
  relative = TRUE,
  plot = FALSE
)
sitelist
```

```
## $Cloaca
##  [1] "ASV41"  "ASV43"  "ASV49"  "ASV52"  "ASV65"  "ASV66"  "ASV67"  "ASV76"
##  [9] "ASV78"  "ASV79"  "ASV82"  "ASV88"  "ASV90"  "ASV93"  "ASV94"  "ASV96"
## [17] "ASV103" "ASV115" "ASV117" "ASV128" "ASV130" "ASV131" "ASV132" "ASV133"
## [25] "ASV134" "ASV136" "ASV145" "ASV146" "ASV148" "ASV150" "ASV151" "ASV152"
```

```
## [33] "ASV157" "ASV158" "ASV159" "ASV162" "ASV163" "ASV170" "ASV172" "ASV175"
## [41] "ASV177" "ASV192" "ASV193" "ASV201" "ASV204" "ASV212" "ASV213" "ASV214"
## [49] "ASV217" "ASV247" "ASV267" "ASV282" "ASV303" "ASV326" "ASV348" "ASV498"
## [57] "ASV506"
##
## $Preen
## [1] "ASV1"   "ASV54"  "ASV63"  "ASV87"  "ASV106" "ASV156"
##
## $Cloaca__Preen
##  [1] "ASV4"   "ASV15"  "ASV20"  "ASV22"  "ASV25"  "ASV26"  "ASV27"  "ASV29"
##  [9] "ASV33"  "ASV38"  "ASV39"  "ASV42"  "ASV45"  "ASV47"  "ASV48"  "ASV53"
## [17] "ASV55"  "ASV56"  "ASV57"  "ASV59"  "ASV64"  "ASV74"  "ASV75"  "ASV77"
## [25] "ASV81"  "ASV84"  "ASV91"  "ASV95"  "ASV102" "ASV112" "ASV116" "ASV124"
## [33] "ASV125" "ASV140" "ASV190"
```

```r
# Load necessary libraries
library(phyloseq)

# Assuming physeqprune is your phyloseq object
# Extract the taxonomy table
tax_table <- as.data.frame(tax_table(physeqprune))

# Function to get genus and species for a list of ASVs
get_genus_species_for_asvs <- function(asv_list, tax_table) {
  # Subset the taxonomy table for the given ASVs
  matched_taxa <- tax_table[rownames(tax_table) %in% asv_list, ]

  # Function to find the most specific identified taxonomic level
  get_first_identified <- function(row) {
    if (!is.na(row["Species"]) && row["Species"] != "" && row["Species"] != "unidentified" && !is.na(ro
      return(paste(row["Genus"], row["Species"], sep = " "))
    } else if (!is.na(row["Genus"]) && row["Genus"] != "" && row["Genus"] != "unidentified") {
      return(paste("Genus:", row["Genus"]))
    } else {
      tax_levels <- c("Family", "Order", "Class", "Phylum", "Kingdom")
      for (col in tax_levels) {
        if (!is.na(row[col]) && row[col] != "" && row[col] != "unidentified") {
          return(paste(col, row[col], sep = ": "))
        }
      }
    }
    return("unidentified")
  }

  # Apply the function to each row
  matched_taxa$First_Identified_Taxa <- apply(matched_taxa, 1, get_first_identified)

  # Return the ASV and First_Identified_Taxa columns
  return(data.frame(ASV = rownames(matched_taxa), First_Identified_Taxa = matched_taxa$First_Identified_
}

# Get genus and species for each group in sitelist
genus_species_cloaca <- get_genus_species_for_asvs(sitelist$Cloaca, tax_table)
genus_species_preen <- get_genus_species_for_asvs(sitelist$Preen, tax_table)
```

```
genus_species_cloaca_preen <- get_genus_species_for_asvs(sitelist$Cloaca__Preen, tax_table)

# Save results to CSV files
write.csv(genus_species_cloaca, "venntaxa/taxa_cloaca.csv", row.names = FALSE)
write.csv(genus_species_preen, "venntaxa/taxa_preen.csv", row.names = FALSE)
write.csv(genus_species_cloaca_preen, "venntaxa/taxa_cloaca_preen.csv", row.names = FALSE)
```

##export tiff with 300dpi

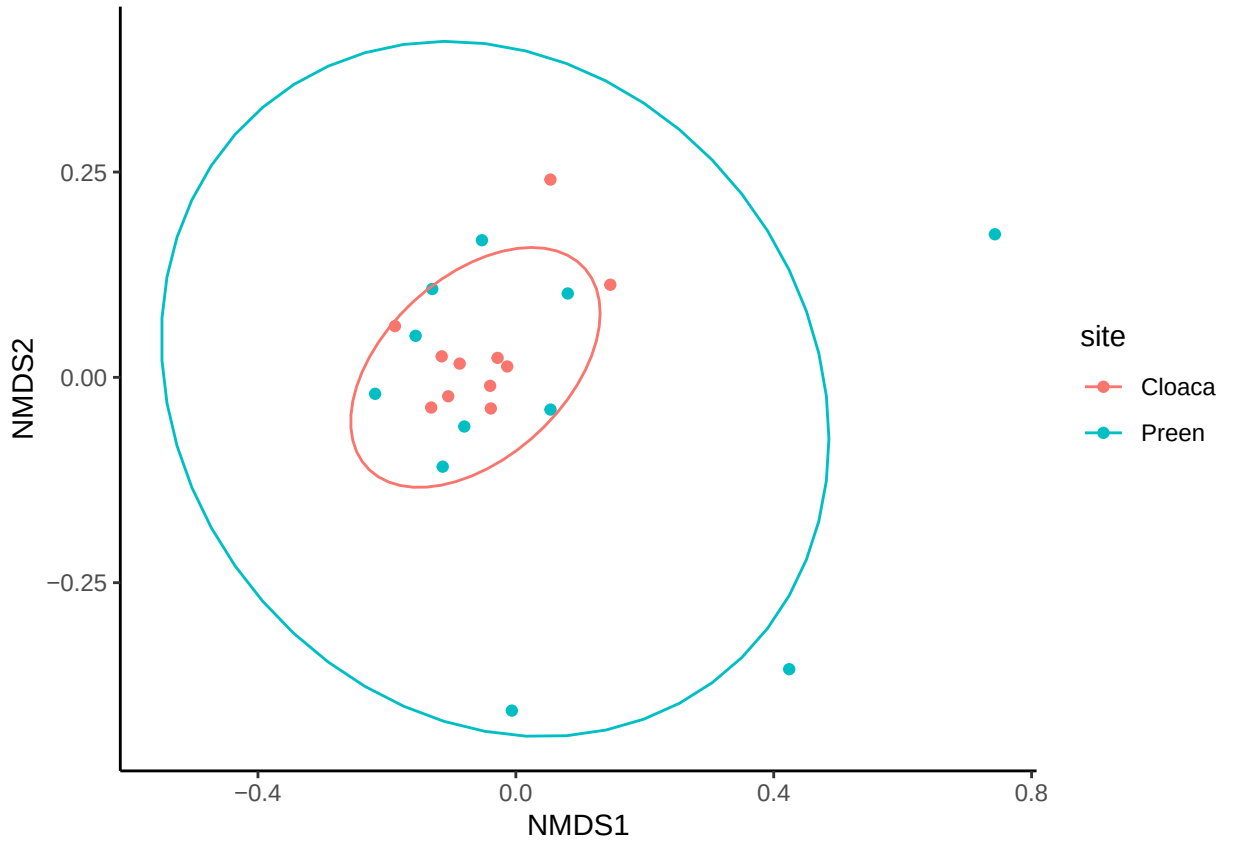
```
ggsave(
  filename="figures/Figure02Venn.tiff",
  plot = sitevenn,
  width = 200,
  height = 200,
  units = c("mm"),
  dpi = 300,
  bg = "white"
)
```

##Bray Curtis Calculation

```
set.seed(777)
dist = phyloseq::distance(physeqprune, method="bray", weighted=TRUE) #calculate Bray-Curtis dissimilari
ordination = ordinate(physeqprune, method="NMDS", distance=dist) #perform ordination on distance matrix
```

##Bray Curtis Site Plot

```
braysite=plot_ordination(physeq, ordination, color="site") +
  theme_classic() +
  theme(strip.background = element_blank()) + stat_ellipse(aes(group=site))
braysite
```



##Bray Curtis Site Stats

```
adonis2(dist ~ sample_data(physeqprune)$site)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = dist ~ sample_data(physeqprune)$site)
##                               Df SumOfSqs      R2      F Pr(>F)
## sample_data(physeqprune)$site  1   0.4271 0.05103 1.0755   0.11
## Residual                      20   7.9412 0.94897
## Total                         21   8.3682 1.00000
```

##Bray Curtis Species ANOSIM

```
anosim <- data.frame(sample_data(physeqprune))
anosim(dist, anosim$site, permutations=9999)
```

```
##
## Call:
## anosim(x = dist, grouping = anosim$site, permutations = 9999)
## Dissimilarity: bray
##
## ANOSIM statistic R: 0.03666
##       Significance: 0.1341
##
```

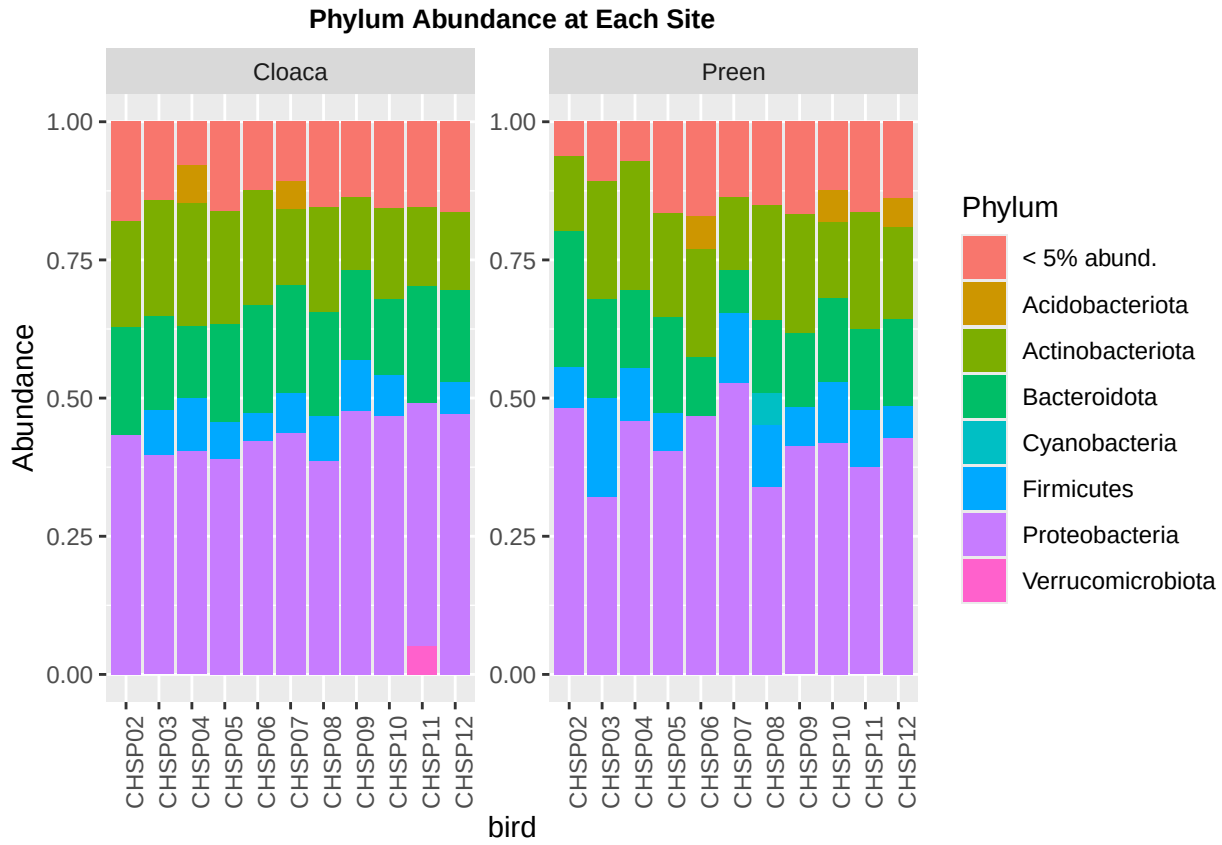
```
## Permutation: free
## Number of permutations: 9999
```

##export tiff with 300dpi

```
ggsave(
  filename="figures/Figure03BetaDiv.tiff",
  plot = braysite,
  width = 250,
  height = 150,
  units = c("mm"),
  dpi = 300,
)
```

##Bar plots of Abundance per individual samples in site (Phylum-Merge <5%)

```
physeq2 = filter_taxa(physeqprune, function(x) mean(x) > 0.05, TRUE)
physeq3 = transform_sample_counts(physeq2, function(x) x / sum(x) )
glom<-psmelt(physeq3)
glom <- tax_glom(physeq3, taxrank = 'Phylum')
data<-psmelt(glom)
data$Phylum <- as.character(data$Phylum)
data$Phylum[data$Abundance < 0.05] <- "< 5% abund."
medians <- ddply(data, ~Phylum, function(x) c(median=median(x$Abundance)))
remainder <- medians[medians$median <= 0.05,]$Phylum
data[data$Phylum %in% remainder,]$Phylum <- "< 5% abund."
data$Phylum[data$Abundance < 0.05] <- "< 5% abund."
spatial_plot <- ggplot(data=data, aes(x=bird, y=Abundance, fill=Phylum)) +
  facet_wrap(~site, scales = "free")
barplotphylum<-spatial_plot + geom_bar(aes(), stat="identity", position="fill") +
  ggtitle("Phylum Abundance at Each Site") +
  theme (axis.text.x = element_text(angle=90),
         plot.title = element_text(size = 10, face = "bold", hjust = .5))
barplotphylum
```



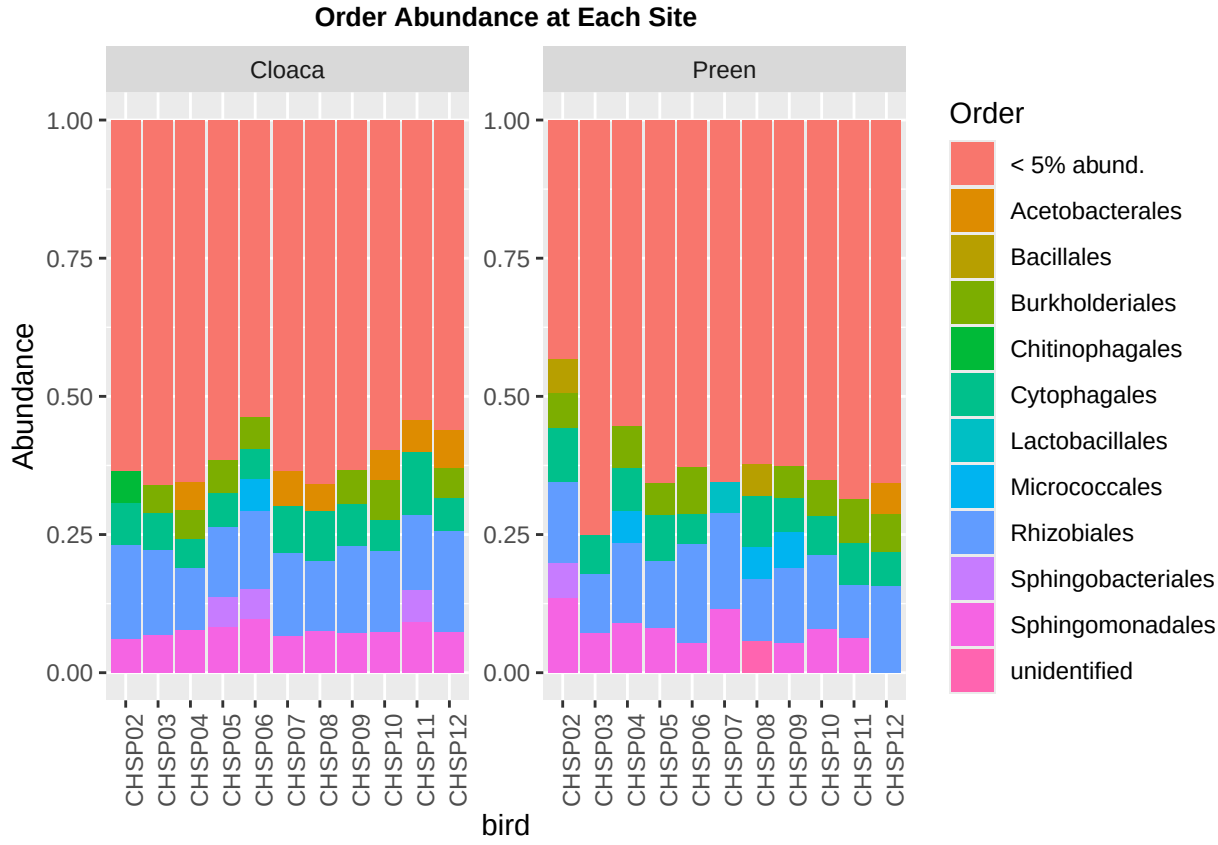
##export tiff with 300dpi

```
ggsave(
  filename="figures/Figure00Barplot.tiff",
  plot = barplotphylum,
  width = 250,
  height = 150,
  units = c("mm"),
  dpi = 300,
)
```

##Bar plots of Abundance per individual samples in species (Order-Merge <5%)

```
physeq2 = filter_taxa(physeqprune, function(x) mean(x) > 0.05, TRUE)
physeq3 = transform_sample_counts(physeq2, function(x) x / sum(x) )
glom<-psmelt(physeq3)
glom <- tax_glom(physeq3, taxrank = 'Order')
data<-psmelt(glom)
data$Order <- as.character(data$Order)
data$Order[data$Abundance < 0.05] <- "< 5% abund."
medians <- ddply(data, ~Order, function(x) c(median=median(x$Abundance)))
remainder <- medians[medians$median <= 0.05,]$Order
data[data$Order %in% remainder,]$Order <- "< 5% abund."
data$Order[data$Abundance < 0.05] <- "< 5% abund."
spatial_plot <- ggplot(data=data, aes(x=bird, y=Abundance, fill=Order)) +
  facet_wrap(~site, scales = "free")
barplotorder<-spatial_plot + geom_bar(aes(), stat="identity", position="fill") +
  ggtitle("Order Abundance at Each Site") +
```

```
  theme (axis.text.x = element_text(angle=90),
         plot.title = element_text(size = 10, face = "bold", hjust = .5))
barplotorder
```



##export tiff with 300dpi

```
ggsave(
  filename="figures/Figure00BarplotOrder.tiff",
  plot = barplotorder,
  width = 250,
  height = 150,
  units = c("mm"),
  dpi = 300,
)
```

#Differential species ID

```
# Extract abundance data (OTU table)
abundance_data <- as.data.frame(otu_table(physeqprune))

# Extract the grouping variable (site) from sample data
site_group <- sample_data(physeqprune)$site

# Ensure the grouping variable is a factor
site_group <- as.factor(site_group)

# Run SIMPER analysis
```

```r
simper_result <- simper(abundance_data, group = site_group, permutations = 100)

# Extract SIMPER results for each pairwise comparison
simper_summary <- summary(simper_result)

# Extract the data frame from the list
simper_df <- simper_summary$Preen_Cloaca

# Convert the data frame to include species names
simper_df <- as.data.frame(simper_df)
simper_df$species <- rownames(simper_df)
rownames(simper_df) <- NULL

# Add comparison information
simper_df$comparison <- "Preen_Cloaca"

# Save SIMPER results to a CSV file
write.csv(simper_df, "differentialexpression/simper_results.csv", row.names = FALSE)

# Extract OTU names and their contributions
otu_contributions <- simper_df %>%
  select(species, average, sd, ratio, ava, avb, cumsum, comparison) %>%
  arrange(species)

# Perform Mann-Whitney U Test on OTU contributions between groups
results <- data.frame()

for (otu in unique(otu_contributions$species)) {
  # Subset the data for the current OTU
  otu_data <- subset(otu_contributions, species == otu)

  # Perform Mann-Whitney U Test
  test_result <- wilcox.test(otu_data$ava, otu_data$avb)

  # Store results
  results <- rbind(results, data.frame(OTU = otu, p.value = test_result$p.value))
}

# Adjust p-values for multiple testing (optional)
results$adj.p.value <- p.adjust(results$p.value, method = "BH")

# Save Mann-Whitney results to a CSV file
write.csv(results, "differentialexpression/mann_whitney_results.csv", row.names = FALSE)

# View significant results
significant_results <- subset(results, adj.p.value < 0.05)

# Save significant results to a CSV file
write.csv(significant_results, "differentialexpression/significant_results.csv", row.names = FALSE)

# Print significant results
print(significant_results)
```

```
## [1] OTU         p.value     adj.p.value
```

```
## <0 rows> (or 0-length row.names)
```