Bioinformatics Visuals

2025-08-05

```r
# Load required libraries
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3

## Warning: package 'ggplot2' was built under R version 4.3.3

## Warning: package 'tidyr' was built under R version 4.3.3

## Warning: package 'purrr' was built under R version 4.3.3

## Warning: package 'dplyr' was built under R version 4.3.3

## Warning: package 'forcats' was built under R version 4.3.3

## Warning: package 'lubridate' was built under R version 4.3.3

## -- 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 ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## 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
```

```r
library(vegan)
```

```
## Warning: package 'vegan' was built under R version 4.3.3

## Loading required package: permute

## Warning: package 'permute' was built under R version 4.3.3

## Loading required package: lattice
```

```r
library(ggpubr)
```

```
## Warning: package ’ggpubr’ was built under R version 4.3.3
```

```r
library(reshape2)
```

```
## Warning: package ’reshape2’ was built under R version 4.3.3

##
## Attaching package: ’reshape2’
##
## The following object is masked from ’package:tidyr’:
##
##     smiths
```

```r
library(ggvenn)
```

```
## Warning: package ’ggvenn’ was built under R version 4.3.3

## Loading required package: grid
```

Load in the file exports (diversity metrics, taxonomic assignments, unique OTUs)

```r
#Set file path
dir <- file.path(
  "C:", "Users", "Joyalea", "Documents", "UBCO", "Thesis",
  "Final_Files", "RQ4_AMFCommunityResponse", "file_exports_97_blastn_merged"
)

#Load alpha diversity metrics
richness <- read_tsv(file.path(dir, "observed_features_vector.tsv"))
```

```
## New names:
## Rows: 28 Columns: 2
## -- Column specification
## ---------------------------------------------------------- Delimiter: "\t" chr
## (1): ...1 dbl (1): observed_features
## i Use ‘spec()‘ to retrieve the full column specification for this data. i
## Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## * ‘‘ -> ‘...1‘
```

```r
shannon  <- read_tsv(file.path(dir, "shannon_vector_97.tsv"))
```

```
## New names:
## Rows: 28 Columns: 2
## -- Column specification
## ---------------------------------------------------------- Delimiter: "\t" chr
## (1): ...1 dbl (1): shannon_entropy
## i Use ‘spec()‘ to retrieve the full column specification for this data. i
## Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## * ‘‘ -> ‘...1‘
```

```
evenness <- read_tsv(file.path(dir, "evenness_vector.tsv"))
```

```
## New names:
## Rows: 28 Columns: 2
## -- Column specification
## -------------------------------------------------------- Delimiter: "\t" chr
## (1): ...1 dbl (1): pielou_evenness
## i Use ‘spec()‘ to retrieve the full column specification for this data. i
## Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## * ‘‘ -> ‘...1‘
```

```
#Rename ID columns for consistency
colnames(richness)[1] <- "SampleID"
colnames(shannon)[1]  <- "SampleID"
colnames(evenness)[1] <- "SampleID"

#Rename metric columns for clarity
colnames(richness)[2] <- "Richness"
colnames(shannon)[2]  <- "Shannon"
colnames(evenness)[2] <- "Evenness"

#Merge all alpha metrics
alpha_df <- reduce(list(richness, shannon, evenness), inner_join, by = "SampleID")

#Load metadata
metadata <- read_tsv(file.path(
  "C:", "Users", "Joyalea", "Documents", "UBCO", "Thesis",
  "Final_Files", "RQ4_AMFCommunityResponse", "METADATA.tsv"
))
```

```
## Rows: 28 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (4): SampleID, MesocosmID, Treatment, DateHarvested
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
#Merge with metadata
alpha_merged <- inner_join(metadata, alpha_df, by = "SampleID")

#Check column names and that data was correctly uploaded
colnames(alpha_merged)
```

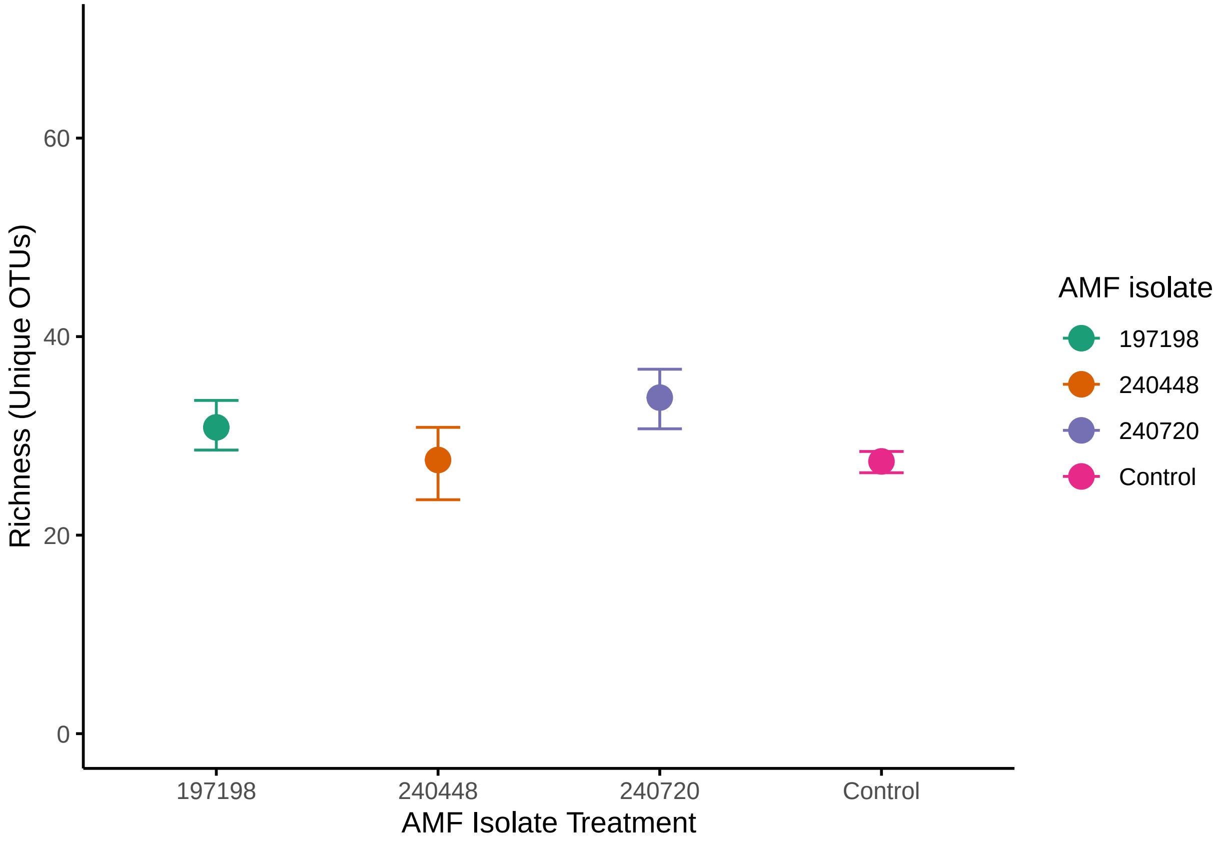
```
## [1] "SampleID"      "MesocosmID"    "Treatment"     "DateHarvested"
## [5] "Richness"      "Shannon"       "Evenness"
```

```
head(alpha_merged)
```

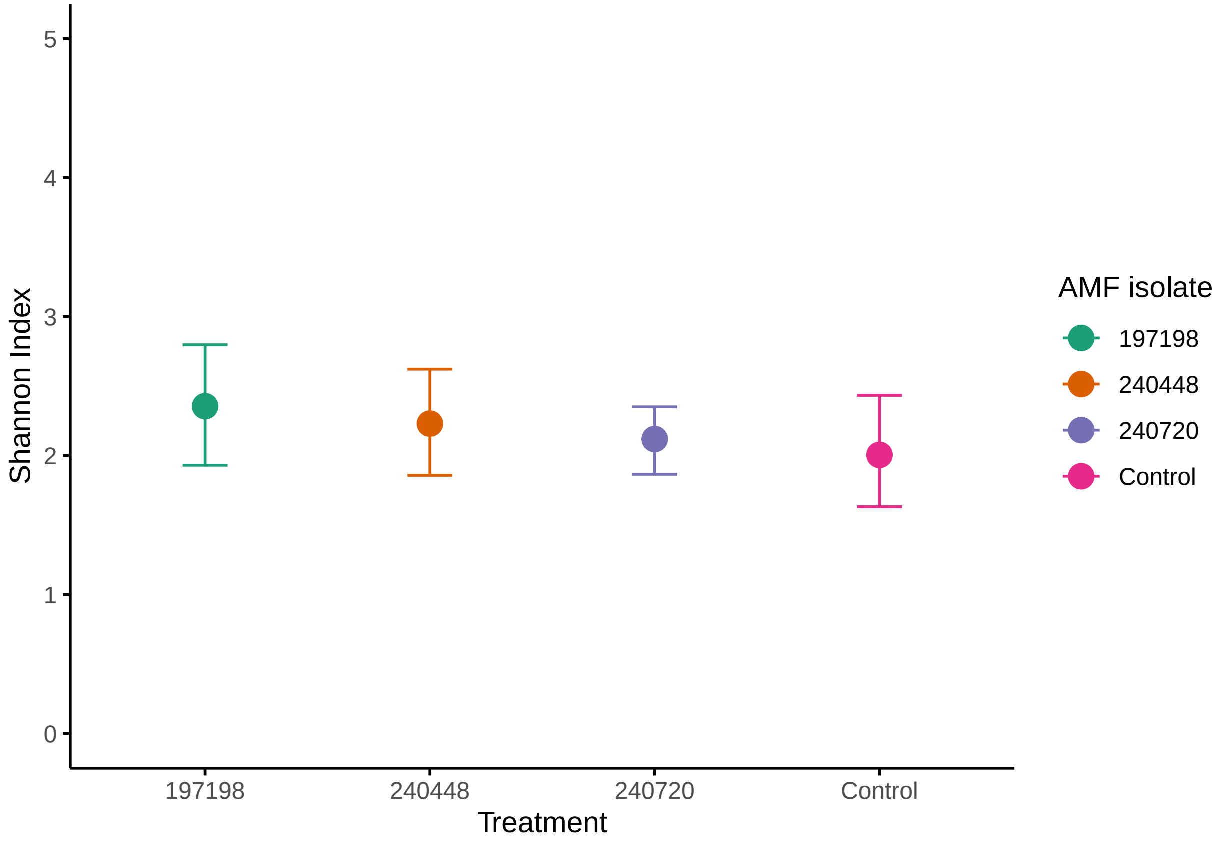
```
## # A tibble: 6 x 7
##   SampleID MesocosmID Treatment DateHarvested Richness Shannon Evenness
##   <chr>    <chr>      <chr>    <chr>            <dbl>   <dbl>    <dbl>
## 1 M1       Mesocosm1  Control  August27            28    1.80    0.375
## 2 M2       Mesocosm2  Control  August27            29    1.78    0.367
## 3 M3       Mesocosm3  Control  August27            26    2.33    0.497
## 4 M4       Mesocosm4  Control  August27            28    1.87    0.390
## 5 M5       Mesocosm5  Control  September4          27    3.03    0.638
## 6 M6       Mesocosm6  Control  September4          25    1.96    0.423
```

Create visuals of the alpha diversity metrics

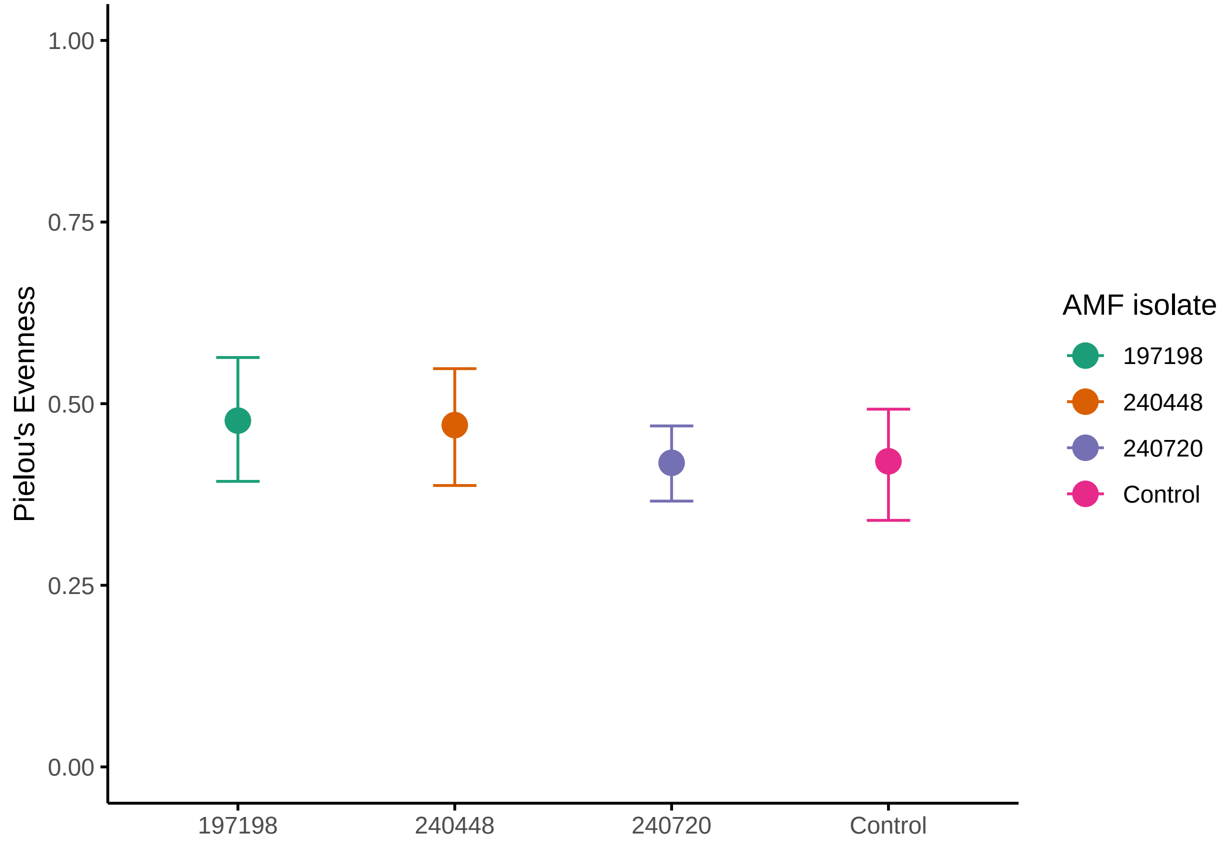
```
#Plot alpha metrics
#RICHNESS
print(mean_richness<-ggplot(alpha_merged, aes(x = Treatment, y = Richness, color = Treatment)) +
  stat_summary(fun = mean, geom = "point", size = 4) +
  coord_cartesian(ylim = c(0,70))+
  scale_colour_brewer(palette = "Dark2") +
  stat_summary(fun.data = mean_cl_boot, geom = "errorbar", width = 0.2) +
  labs(x = "AMF Isolate Treatment", y = "Richness (Unique OTUs)", colour =
         "AMF isolate") +
  theme_minimal()+

  theme(
    panel.grid = element_blank(),                 #Removes all grid lines
    axis.line = element_line(color = "black"),  #Adds black x and y axis lines
    axis.ticks = element_line(color = "black")  #show tick marks
  ))
```



```
#SHANNON
print(mean_shannon<-ggplot(alpha_merged, aes(x = Treatment, y = Shannon, color = Treatment)) +
  stat_summary(fun = mean, geom = "point", size = 4) +
  stat_summary(fun.data = mean_cl_boot, geom = "errorbar", width = 0.2) +
  scale_colour_brewer(palette = "Dark2") +
  coord_cartesian(ylim = c(0,5))+
  labs(y = "Shannon Index", colour =
         "AMF isolate") +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),               #Removes all grid lines
    axis.line = element_line(color = "black"),  #Adds black x and y axis lines
    axis.ticks = element_line(color = "black")  #show tick marks
  ))
```



```
#EVENNESS
print(mean_evenness<- ggplot(alpha_merged, aes(x = Treatment, y = Evenness, color = Treatment)) +
  stat_summary(fun = mean, geom = "point", size = 4) +
  stat_summary(fun.data = mean_cl_boot, geom = "errorbar", width = 0.2) +
  scale_colour_brewer(palette = "Dark2") +
  coord_cartesian(ylim = c(0,1))+
  labs(x = NULL, y = "Pielou's Evenness", colour =
         "AMF isolate") +
  theme_minimal() +
  theme(
    panel.grid = element_blank(),                #Removes all grid lines
    axis.line = element_line(color = "black"),  #Adds black x and y axis lines
    axis.ticks = element_line(color = "black")  #show tick marks
  ))
```



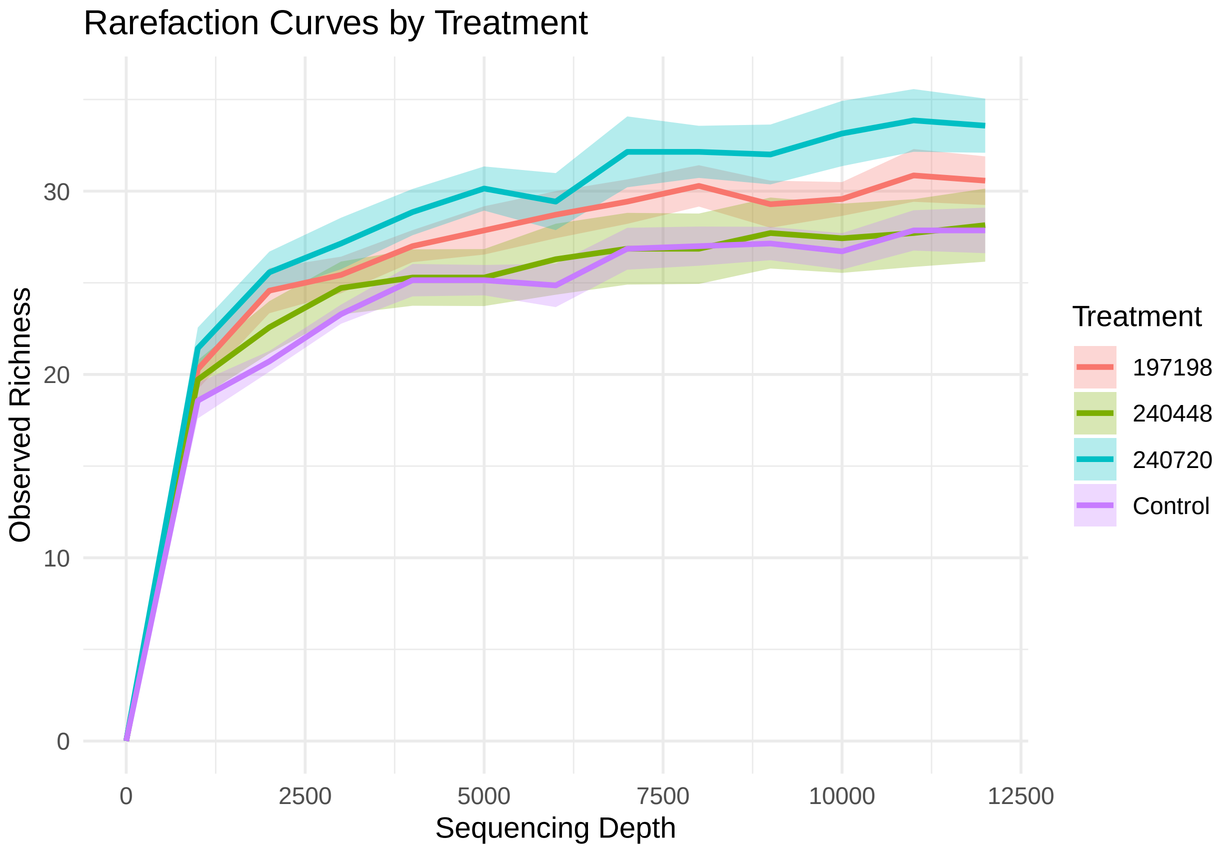
Create rarefication curves

```r
#Set your path
feature_table_path <- file.path(
  "C:", "Users", "Joyalea", "Documents", "UBCO", "Thesis",
  "Final_Files", "RQ4_AMFCommunityResponse", "file_exports_97_blastn_merged",
  "feature_table_97.tsv")

#Load feature table
feature_table <- read_tsv(feature_table_path, skip = 1)
```

```
## Rows: 177 Columns: 29
## -- Column specification -------------------------------------------------------
## Delimiter: "\t"
## chr  (1): #OTU ID
## dbl (28): M1, M10, M11, M12, M13, M14, M15, M16, M17, M18, M19, M2, M20, M21...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

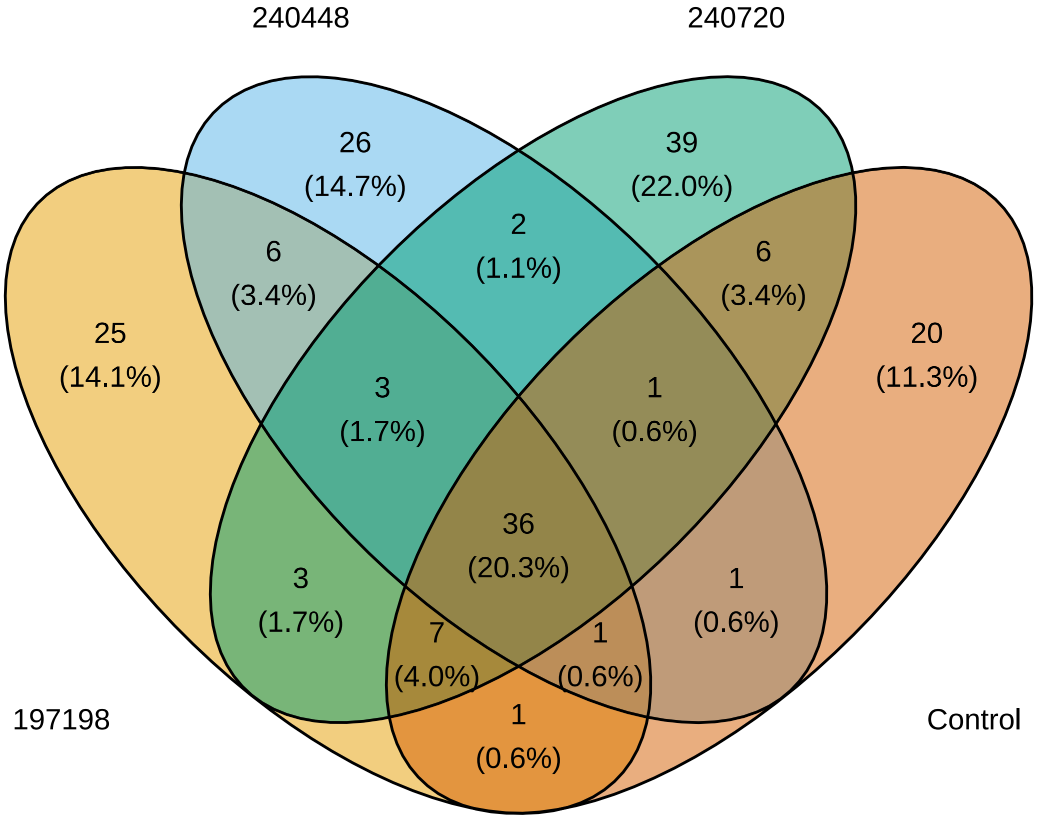
```r
#Convert feature table to a matrix
otu_matrix <- feature_table %>%
  column_to_rownames(var = "#OTU ID") %>%
  t() %>%
  as.data.frame()

#Match metadata to OTU matrix
metadata <- metadata %>% filter(SampleID %in% rownames(otu_matrix))

#Set consistent row order
otu_matrix <- otu_matrix[metadata$SampleID, ]

#Assign treatment colors for plotting
treatment_colors <- RColorBrewer::brewer.pal(n = length(unique(metadata$Treatment)), name = "Set1")
names(treatment_colors) <- unique(metadata$Treatment)
sample_colors <- treatment_colors[metadata$Treatment]

#MAKE THE PLOT LOOKS NICE
#Simulate rarefaction across samples
depths <- seq(0, min(rowSums(otu_matrix)), by = 1000)

rarefied_full <- map_dfr(depths, function(d) {
  rarefied <- rrarefy(otu_matrix, sample = d)
  richness <- rowSums(rarefied > 0)
  tibble(SampleID = names(richness), Richness = richness, Depth = d)
}) %>%
  left_join(metadata, by = "SampleID")

#Summarize curves with with SE
rarefied <- rarefied_full %>%
  group_by(Treatment, Depth) %>%
  summarize(
    MeanRichness = mean(Richness),
    SERichness = sd(Richness) / sqrt(n()),
    .groups = "drop"
  ) %>%
  mutate(Treatment = factor(Treatment))

ggplot(rarefied, aes(x = Depth, y = MeanRichness, color = Treatment)) +
  geom_ribbon(aes(
    ymin = MeanRichness - SERichness,
    ymax = MeanRichness + SERichness,
    fill = Treatment
  ), alpha = 0.3, color = NA) +
  geom_line(size = 1) +
  labs(
    title = "Rarefaction Curves by Treatment",
    x = "Sequencing Depth",
    y = "Observed Richness"
  ) +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Unique OTU visualistion

```
#Convert wide table into long format
otu_table_long <- feature_table %>%
  rename(FeatureID = `#OTU ID`) %>%
  pivot_longer(-FeatureID, names_to = "SampleID", values_to = "Abundance") %>%
  filter(Abundance > 0) %>%  # Keep only present OTUs
  left_join(metadata %>% select(SampleID, Treatment), by = "SampleID")


#Keep distinct OTU-treatment combinations
venn_df <- otu_table_long %>%
  distinct(Treatment, FeatureID)

#Create list of OTUs by treatment
otu_lists <- venn_df %>%
  group_by(Treatment) %>%
  summarize(OTUs = list(unique(FeatureID)), .groups = "drop") %>%
  deframe()

#Plot Venn diagram
ggvenn(otu_lists,
       fill_color = c("#E69F00", "#56B4E9", "#009E73", "#D55E00")[1:length(otu_lists)],
       stroke_size = 0.5,
       set_name_size = 4)
```



## Run SIMPER anlaysis on p/a to investigate what drove the signifcant Jaccard result

```
#Convert OTU table to wide format
otu_wide <- dcast(
  otu_table_long,
  SampleID ~ FeatureID,
  value.var = "Abundance",
  fill = 0
)

otu_wide_mat <- otu_wide %>%
  column_to_rownames("SampleID")

#Align metada
metadata_simper <- metadata %>%
  filter(SampleID %in% rownames(otu_wide_mat)) %>%
  arrange(match(SampleID, rownames(otu_wide_mat)))

#Convert OTU table to presence/absence
otu_pa <- decostand(otu_wide_mat, method = "pa")

#Subset dataset and run simper for DAOM240448 vs DAOM 240720
```

```r
subset_meta_1 <- metadata_simper %>%
  filter(Treatment %in% c("240448", "240720"))

subset_otu_1 <- otu_pa[rownames(otu_pa) %in% subset_meta_1$SampleID, ]

simper_jaccard_1 <- simper(subset_otu_1, subset_meta_1$Treatment, permutations = 999)

simper_df_1 <- as.data.frame(summary(simper_jaccard_1)$`240448_240720`)
simper_df_1$FeatureID <- rownames(simper_df_1)
simper_df_1 <- simper_df_1[order(-simper_df_1$average), ]

top_simper_1 <- head(simper_df_1, 10)
print(top_simper_1)
```

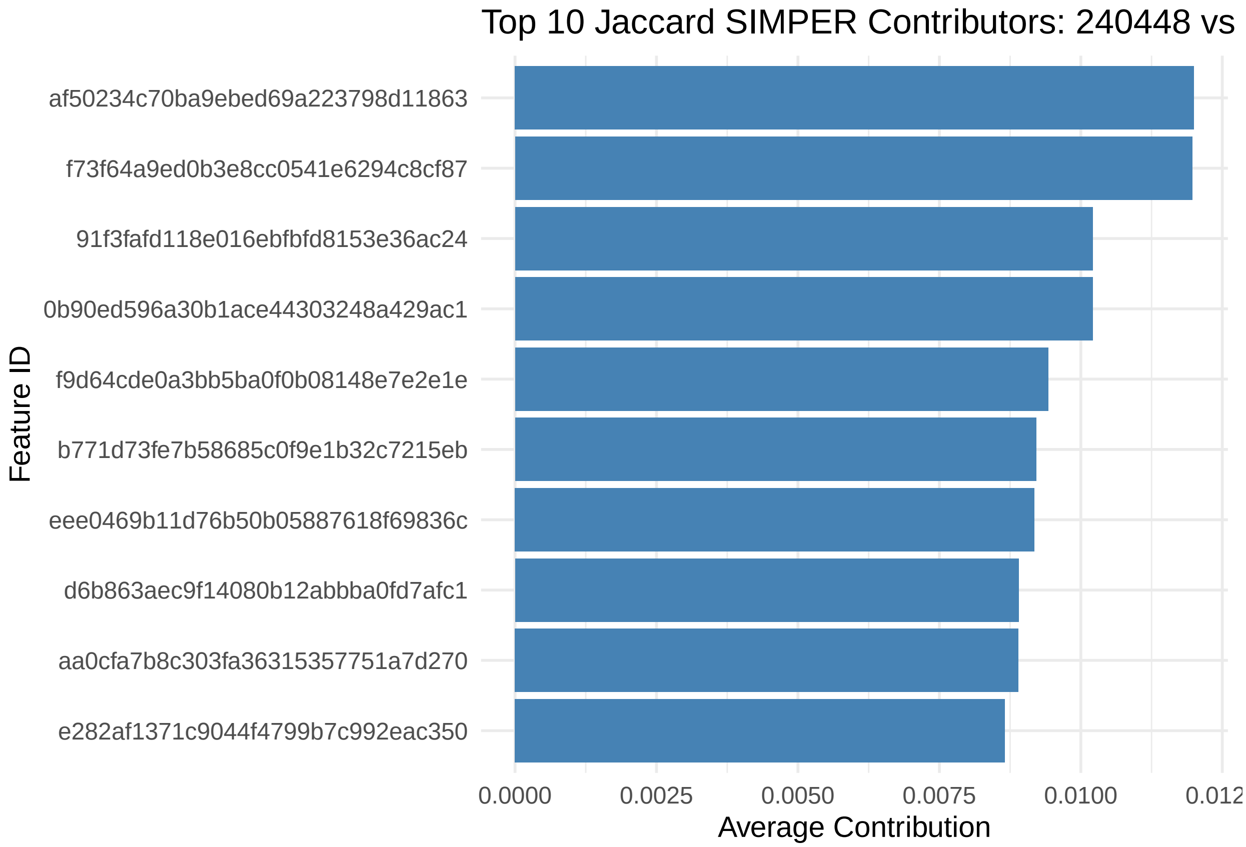
```
##                                      average          sd    ratio       ava
## af50234c70ba9ebed69a223798d11863 0.011998882 0.007107336 1.688239 0.1428571
## f73f64a9ed0b3e8cc0541e6294c8cf87 0.011968258 0.007048819 1.697910 0.1428571
## 0b90ed596a30b1ace44303248a429ac1 0.010210347 0.007640163 1.336404 0.1428571
## 91f3fafd118e016ebfbfd8153e36ac24 0.010210347 0.007640163 1.336404 0.1428571
## f9d64cde0a3bb5ba0f0b08148e7e2e1e 0.009422961 0.008049896 1.170569 0.7142857
## b771d73fe7b58685c0f9e1b32c7215eb 0.009211774 0.008186570 1.125230 0.0000000
## eee0469b11d76b50b05887618f69836c 0.009181298 0.008165458 1.124407 1.0000000
## d6b863aec9f14080b12abbba0fd7afc1 0.008900463 0.007921619 1.123566 0.0000000
## aa0cfa7b8c303fa36315357751a7d270 0.008896403 0.008245560 1.078933 0.4285714
## e282af1371c9044f4799b7c992eac350 0.008660859 0.008365588 1.035296 0.5714286
##                                        avb     cumsum     p
## af50234c70ba9ebed69a223798d11863 0.8571429 0.02565584 0.019
## f73f64a9ed0b3e8cc0541e6294c8cf87 0.8571429 0.05124621 0.016
## 0b90ed596a30b1ace44303248a429ac1 0.7142857 0.07307784 0.036
## 91f3fafd118e016ebfbfd8153e36ac24 0.7142857 0.09490946 0.036
## f9d64cde0a3bb5ba0f0b08148e7e2e1e 0.2857143 0.11505751 0.180
## b771d73fe7b58685c0f9e1b32c7215eb 0.5714286 0.13475400 0.002
## eee0469b11d76b50b05887618f69836c 0.4285714 0.15438532 0.002
## d6b863aec9f14080b12abbba0fd7afc1 0.5714286 0.17341617 0.005
## aa0cfa7b8c303fa36315357751a7d270 0.8571429 0.19243834 0.223
## e282af1371c9044f4799b7c992eac350 0.2857143 0.21095687 0.226
##                                                         FeatureID
## af50234c70ba9ebed69a223798d11863 af50234c70ba9ebed69a223798d11863
## f73f64a9ed0b3e8cc0541e6294c8cf87 f73f64a9ed0b3e8cc0541e6294c8cf87
## 0b90ed596a30b1ace44303248a429ac1 0b90ed596a30b1ace44303248a429ac1
## 91f3fafd118e016ebfbfd8153e36ac24 91f3fafd118e016ebfbfd8153e36ac24
## f9d64cde0a3bb5ba0f0b08148e7e2e1e f9d64cde0a3bb5ba0f0b08148e7e2e1e
## b771d73fe7b58685c0f9e1b32c7215eb b771d73fe7b58685c0f9e1b32c7215eb
## eee0469b11d76b50b05887618f69836c eee0469b11d76b50b05887618f69836c
## d6b863aec9f14080b12abbba0fd7afc1 d6b863aec9f14080b12abbba0fd7afc1
## aa0cfa7b8c303fa36315357751a7d270 aa0cfa7b8c303fa36315357751a7d270
## e282af1371c9044f4799b7c992eac350 e282af1371c9044f4799b7c992eac350
```

```r
#Subset and run SIMPER for DAOM240448 vs the control
subset_meta_2 <- metadata_simper %>%
  filter(Treatment %in% c("240448", "Control"))

subset_otu_2 <- otu_pa[rownames(otu_pa) %in% subset_meta_2$SampleID, ]
```

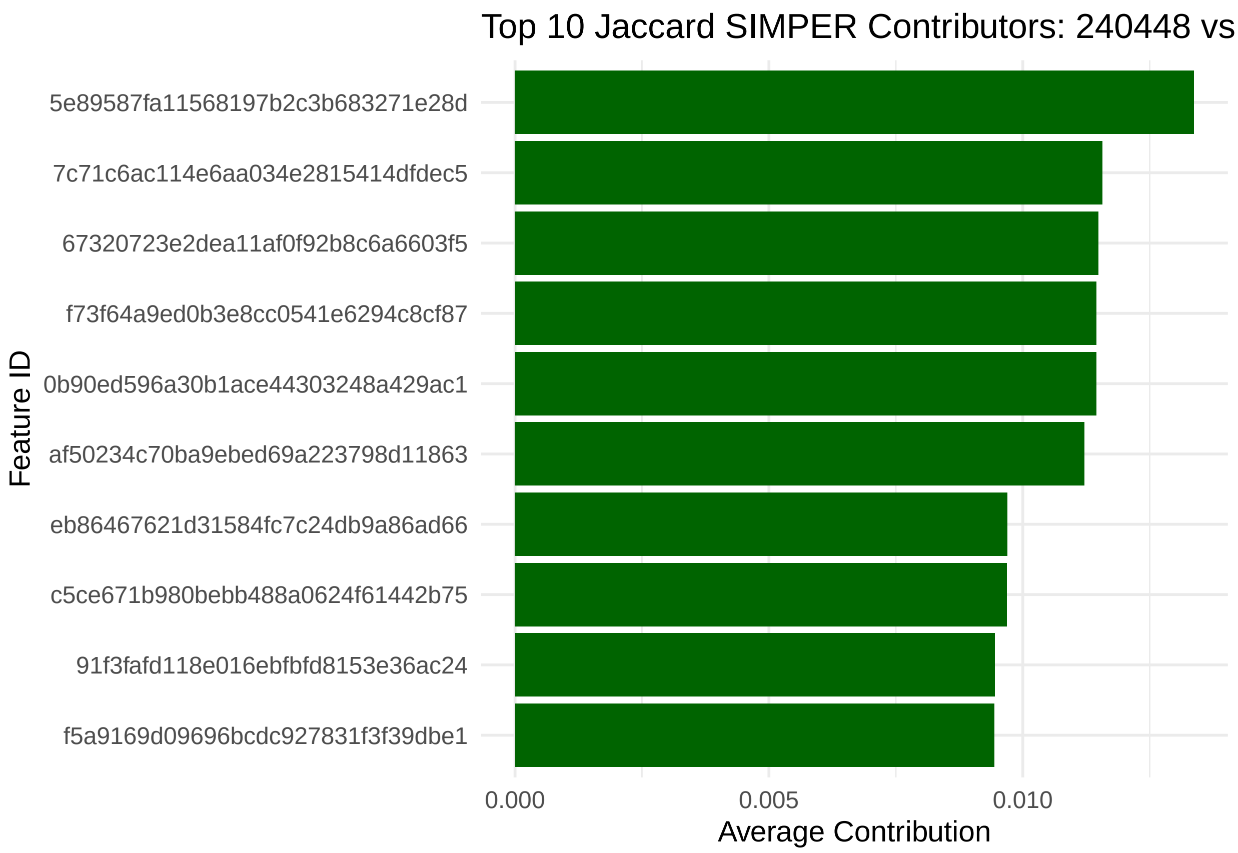
```
simper_jaccard_2 <- simper(subset_otu_2, subset_meta_2$Treatment, permutations = 999)

simper_df_2 <- as.data.frame(summary(simper_jaccard_2)$`Control_240448`)
simper_df_2$FeatureID <- rownames(simper_df_2)
simper_df_2 <- simper_df_2[order(-simper_df_2$average), ]

top_simper_2 <- head(simper_df_2, 10)
print(top_simper_2)
```

```
##                                      average          sd    ratio       ava
## 5e89587fa11568197b2c3b683271e28d 0.013370830 0.007870644 1.698823 0.8571429
## 7c71c6ac114e6aa034e2815414dfdec5 0.011567870 0.008693227 1.330676 0.1428571
## 67320723e2dea11af0f92b8c6a6603f5 0.011488559 0.008615683 1.333447 0.2857143
## 0b90ed596a30b1ace44303248a429ac1 0.011445942 0.008564819 1.336390 0.7142857
## f73f64a9ed0b3e8cc0541e6294c8cf87 0.011445942 0.008564819 1.336390 0.7142857
## af50234c70ba9ebed69a223798d11863 0.011213543 0.008404121 1.334291 0.7142857
## eb86467621d31584fc7c24db9a86ad66 0.009695872 0.008557655 1.133006 0.0000000
## c5ce671b980bebb488a0624f61442b75 0.009688057 0.008967344 1.080371 0.4285714
## 91f3fafd118e016ebfbfd8153e36ac24 0.009444949 0.008696845 1.086020 0.5714286
## f5a9169d09696bcdc927831f3f39dbe1 0.009436835 0.008699967 1.084698 0.1428571
##                                        avb     cumsum     p
## 5e89587fa11568197b2c3b683271e28d 0.1428571 0.02927917 0.007
## 7c71c6ac114e6aa034e2815414dfdec5 0.7142857 0.05461026 0.049
## 67320723e2dea11af0f92b8c6a6603f5 0.8571429 0.07976767 0.043
## 0b90ed596a30b1ace44303248a429ac1 0.1428571 0.10483177 0.044
## f73f64a9ed0b3e8cc0541e6294c8cf87 0.1428571 0.12989586 0.044
## af50234c70ba9ebed69a223798d11863 0.1428571 0.15445105 0.088
## eb86467621d31584fc7c24db9a86ad66 0.5714286 0.17568287 0.058
## c5ce671b980bebb488a0624f61442b75 0.8571429 0.19689758 0.093
## 91f3fafd118e016ebfbfd8153e36ac24 0.1428571 0.21757994 0.155
## f5a9169d09696bcdc927831f3f39dbe1 0.5714286 0.23824453 0.239
##                                                         FeatureID
## 5e89587fa11568197b2c3b683271e28d 5e89587fa11568197b2c3b683271e28d
## 7c71c6ac114e6aa034e2815414dfdec5 7c71c6ac114e6aa034e2815414dfdec5
## 67320723e2dea11af0f92b8c6a6603f5 67320723e2dea11af0f92b8c6a6603f5
## 0b90ed596a30b1ace44303248a429ac1 0b90ed596a30b1ace44303248a429ac1
## f73f64a9ed0b3e8cc0541e6294c8cf87 f73f64a9ed0b3e8cc0541e6294c8cf87
## af50234c70ba9ebed69a223798d11863 af50234c70ba9ebed69a223798d11863
## eb86467621d31584fc7c24db9a86ad66 eb86467621d31584fc7c24db9a86ad66
## c5ce671b980bebb488a0624f61442b75 c5ce671b980bebb488a0624f61442b75
## 91f3fafd118e016ebfbfd8153e36ac24 91f3fafd118e016ebfbfd8153e36ac24
## f5a9169d09696bcdc927831f3f39dbe1 f5a9169d09696bcdc927831f3f39dbe1
```

```
#Plot the top contributors for each contrast
#240448 vs 240720
ggplot(top_simper_1, aes(x = reorder(FeatureID, average), y = average)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(
    title = "Top 10 Jaccard SIMPER Contributors: 240448 vs 240720",
    x = "Feature ID",
    y = "Average Contribution"
  ) +
  theme_minimal()
```



```
#240448 vs Control
ggplot(top_simper_2, aes(x = reorder(FeatureID, average), y = average)) +
  geom_col(fill = "darkgreen") +
  coord_flip() +
  labs(
    title = "Top 10 Jaccard SIMPER Contributors: 240448 vs Control",
    x = "Feature ID",
    y = "Average Contribution"
  ) +
  theme_minimal()
```


Top 10 Jaccard SIMPER Contributors: 240448 vs
Feature ID
5e89587fa11568197b2c3b683271e28d
7c71c6ac114e6aa034e2815414dfdec5
67320723e2dea11af0f92b8c6a6603f5
f73f64a9ed0b3e8cc0541e6294c8cf87
0b90ed596a30b1ace44303248a429ac1
af50234c70ba9ebed69a223798d11863
eb86467621d31584fc7c24db9a86ad66
c5ce671b980bebb488a0624f61442b75
91f3fafd118e016ebfbfd8153e36ac24
f5a9169d09696bcdc927831f3f39dbe1
0.000
0.005
0.010
Average Contribution