Bird Park Environmental Microbiomes

Dustin Foote, Chris Balakrishnan, Ariane Peralta

Last updated on 29 March, 2023

Project Description: Fill out

Initial Setup

#Import Files ## Environmental Data

Microbial Data

Diversity Metrics - Hypothesis Testing

```
# Rarefy Abundances (min abundance is 8106. We are sampling to 8000)
min(rowSums(otus))
```

```
## [1] 8106
```

```
max(rowSums(otus))
```

```
## [1] 37883
```

```
mean(rowSums(otus))
```

```
## [1] 19232.32
```

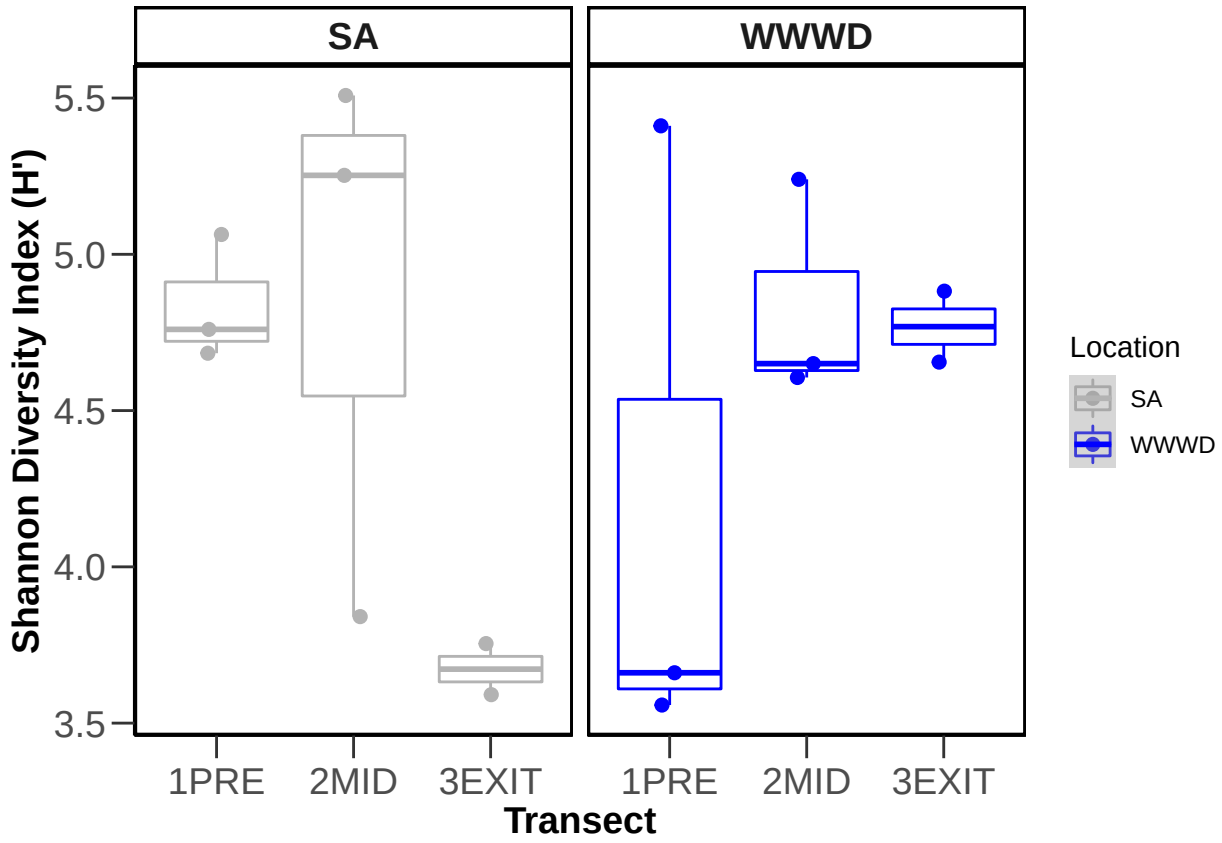
```
SH.r <- rrarefy(otus, 8000)

# Fisher's Alpha
fisher <- fisher.alpha(SH.r)

# Species Richness
richness <- rowSums((SH.r >= 1))

# Shannon Diversity
shannon <- diversity(SH.r, "shannon")

# Simpson's Evenness
```

```r
simp.even <- apply(SH.r, 1, simp_even)

#Pielou's evenness
J <- shannon/log(specnumber(SH.r[,-c(1:1)]))

#combined richness, diversity, evenness
diversity <- cbind(design,richness,shannon,simp.even,J)

diversity$Transect <- as.factor(diversity$Transect)
diversity$Location <- as.factor(diversity$Location)
diversity$Location_ID <- as.factor(diversity$Location_ID)
diversity$Location_ID_order <- as.factor(diversity$Location_ID_order)
str(diversity)
```

```
## 'data.frame':    19 obs. of  10 variables:
##  $ Location         : Factor w/ 3 levels "SA","SOURCE",..: 3 3 3 3 3 3 3 3 1 1 ...
##  $ Transect         : Factor w/ 4 levels "1PRE","2MID",..: 1 1 1 2 2 2 3 3 1 1 ...
##  $ Location_ID      : Factor w/ 7 levels "SA","SA_EXIT",..: 7 7 7 5 5 5 6 6 3 3 ...
##  $ Location_ID_order: Factor w/ 7 levels "1SA_PRE","1WWWD_PRE",..: 2 2 2 4 4 4 6 6 1 1 ...
##  $ Description      : chr  "EMU" "EMU" "EMU" "INFLOW" ...
##  $ Rep              : int  1 2 3 1 2 3 1 2 1 2 ...
##  $ richness         : num  1036 634 576 726 898 ...
##  $ shannon          : num  5.41 3.66 3.56 4.61 5.24 ...
##  $ simp.even        : num  0.069 0.0137 0.0125 0.0541 0.074 ...
##  $ J                : num  0.779 0.568 0.56 0.699 0.771 ...
```

```r
diversity.nosource <- diversity[c(1:16),]

# Graphing Shannon Diversity
p <- ggplot(diversity.nosource, aes(x=Transect, y=shannon, color=Location))+ geom_boxplot() +
      geom_point(aes(color=Location), size=2, position = position_jitterdodge())+ scale_color_manual(na
p1=p+geom_smooth(method="lm")+facet_wrap(~Location)
shannon<-p1 + theme_bw() +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line
          =element_line(colour = "black")) +
    theme(axis.title=element_text(vjust=1,size=14,face="bold"),
          axis.text=element_text(size=14), axis.text.x = element_text(vjust=0.65, hjust=0.5,
          size=14), panel.border = element_rect(colour = "black",size=1)) +
    theme(axis.ticks.length=unit(0.3,"cm")) + labs(x = "Transect", y = "Shannon Diversity Index (H')") +
    theme(strip.text.x = element_text(size=14, face="bold"), strip.text.y =
          element_text(size=14, face="bold"), strip.background = element_rect(colour="black",
          fill="white", size=1))
shannon
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggsave("../figures/SH.bacteria.shannon.png", plot=last_plot(), device=NULL, path=NULL, scale=1, width=7
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
# shannon anova
shannon.lm <- lm(shannon ~ Location*Transect, data = diversity.nosource)
summary(shannon.lm) #NS
```

```
##
## Call:
## lm(formula = shannon ~ Location * Transect, data = diversity.nosource)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02644 -0.19286 -0.07876  0.26728  1.20114
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   4.8356     0.3721  12.997 1.37e-07 ***
## LocationWWWD                 -0.6256     0.5262  -1.189   0.2619
## Transect2MID                  0.0316     0.5262   0.060   0.9533
## Transect3EXIT                -1.1629     0.5883  -1.977   0.0763 .
## LocationWWWD:Transect2MID     0.5904     0.7441   0.793   0.4460
## LocationWWWD:Transect3EXIT    1.7215     0.8320   2.069   0.0654 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6444 on 10 degrees of freedom
## Multiple R-squared:  0.3993, Adjusted R-squared:  0.09895
## F-statistic: 1.329 on 5 and 10 DF,  p-value: 0.3272
```

```
require("emmeans")
```

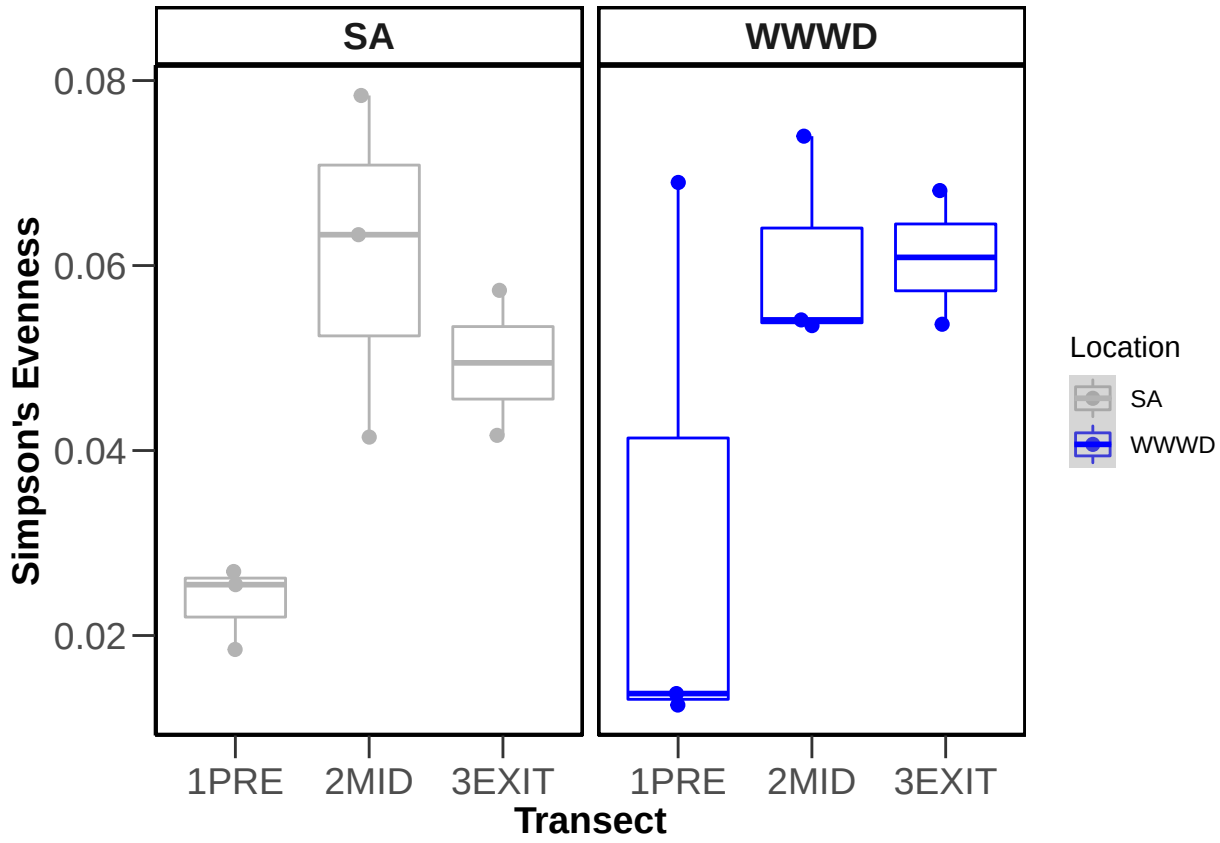
```
## Loading required package: emmeans
```

```
anova(shannon.lm)
```

```
## Analysis of Variance Table
##
## Response: shannon
##                   Df Sum Sq Mean Sq F value Pr(>F)
## Location           1 0.0027 0.00274  0.0066 0.9369
## Transect           2 0.9706 0.48532  1.1686 0.3499
## Location:Transect  2 1.7872 0.89360  2.1517 0.1670
## Residuals         10 4.1530 0.41530
```

```
# Graphing Evenness
p <- ggplot(diversity.nosource, aes(x=Transect, y=simp.even, color=Location))+ geom_boxplot() +
      geom_point(aes(color=Location), size=2, position = position_jitterdodge())+ scale_color_manual(na
p1=p+geom_smooth(method="lm")+facet_wrap(~Location)
even<-p1 + theme_bw() +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line
          =element_line(colour = "black")) +
    theme(axis.title=element_text(vjust=1,size=14,face="bold"),
          axis.text=element_text(size=14), axis.text.x = element_text(vjust=0.65, hjust=0.5,
          size=14), panel.border = element_rect(colour = "black",size=1)) +
    theme(axis.ticks.length=unit(0.3,"cm")) + labs(x = "Transect", y = "Simpson's Evenness") +
    theme(strip.text.x = element_text(size=14, face="bold"), strip.text.y =
          element_text(size=14, face="bold"), strip.background = element_rect(colour="black",
          fill="white", size=1))
even
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggsave("../figures/SH.bacteria.evenness.png", plot=last_plot(), device=NULL, path=NULL, scale=1, width="
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
# evenness anova
even.lm <- lm(simp.even ~ Location*Transect, data = diversity.nosource)
summary(even.lm) #NS
```

```
##
## Call:
## lm(formula = simp.even ~ Location * Transect, data = diversity.nosource)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.019605 -0.007379 -0.001641  0.007379  0.037252
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  0.023636   0.010507   2.250   0.0482 *
## LocationWWWD                 0.008094   0.014859   0.545   0.5979
## Transect2MID                 0.037422   0.014859   2.518   0.0305 *
## Transect3EXIT                0.025844   0.016613   1.556   0.1508
## LocationWWWD:Transect2MID   -0.008615   0.021014  -0.410   0.6905
## LocationWWWD:Transect3EXIT   0.003306   0.023494   0.141   0.8909
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0182 on 10 degrees of freedom
## Multiple R-squared:  0.5393, Adjusted R-squared:  0.309
## F-statistic: 2.342 on 5 and 10 DF,  p-value: 0.1182
```

```
anova(shannon.lm)
```

```
## Analysis of Variance Table
##
## Response: shannon
##                   Df Sum Sq Mean Sq F value Pr(>F)
## Location           1 0.0027 0.00274  0.0066 0.9369
## Transect           2 0.9706 0.48532  1.1686 0.3499
## Location:Transect  2 1.7872 0.89360  2.1517 0.1670
## Residuals         10 4.1530 0.41530
```

Simple Hypothesis Testing - Microbes

```
#PERMANOVA
new.data <-cbind(design,dataREL)
adonis = adonis2(new.data[,-c(1:7)]~Transect*Location, method = "bray", data = new.data, perm=1000)
adonis
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 1000
##
## adonis2(formula = new.data[, -c(1:7)] ~ Transect * Location, data = new.data, permutations = 1000, m
##                   Df SumOfSqs      R2      F   Pr(>F)
## Transect           3   1.2841 0.25434 3.0966 0.001998 **
## Location           1   0.9296 0.18411 6.7249 0.000999 ***
## Transect:Location  2   1.1765 0.23302 4.2556 0.000999 ***
## Residual          12   1.6587 0.32853
## Total             18   5.0488 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Microbial Ordinations

Principal Coordinates Ordination

```
# Principal Coordinates Analysis
dataREL.dist <- vegdist(dataREL, method="bray")

pcoa <- cmdscale(dataREL.dist, k=3, eig=TRUE, add=FALSE)
  # Classical (Metric) Multidimensional Scaling; returns PCoA coordinates
```

```
  # eig=TRUE returns eigenvalues; k = # of dimensions to calculate

explainvar1b <- round(pcoa$eig[1] / sum(pcoa$eig), 3) * 100
explainvar2b <- round(pcoa$eig[2] / sum(pcoa$eig), 3) * 100
sum.eigb <- sum(explainvar1b, explainvar2b)

explainvar1b #30.2
```

```
## [1] 30.2
```

```
explainvar2b #21.6
```

```
## [1] 21.6
```

```
pcoa.groups <- paste(new.data$Location, new.data$Transect, sep = "_")
pcoa.points <- data.frame(pcoa$points, group = pcoa.groups)

# Calculate Centroids (mean and SE)
pcoa.L.centroids <- melt(pcoa.points, id="group", measure.vars = c("X1", "X2"))
pcoa.centroids <- acast(pcoa.L.centroids, variable ~ group, mean)
pcoa.centroids.se <- acast(pcoa.L.centroids, variable ~ group, se)
pcoa.centroids.sd <- acast(pcoa.L.centroids, variable ~ group, sd)

# Combine
pcoa.cent.dataframe <- cbind(t(pcoa.centroids), t(pcoa.centroids.se))
colnames(pcoa.cent.dataframe) <- c("V1", "V2", "V1e", "V2e")
pcoa.cent.trts <- rownames(pcoa.cent.dataframe)
pcoa.cent.dataframe.trts <- as.data.frame(pcoa.cent.dataframe)
dim(pcoa.cent.dataframe.trts)
```

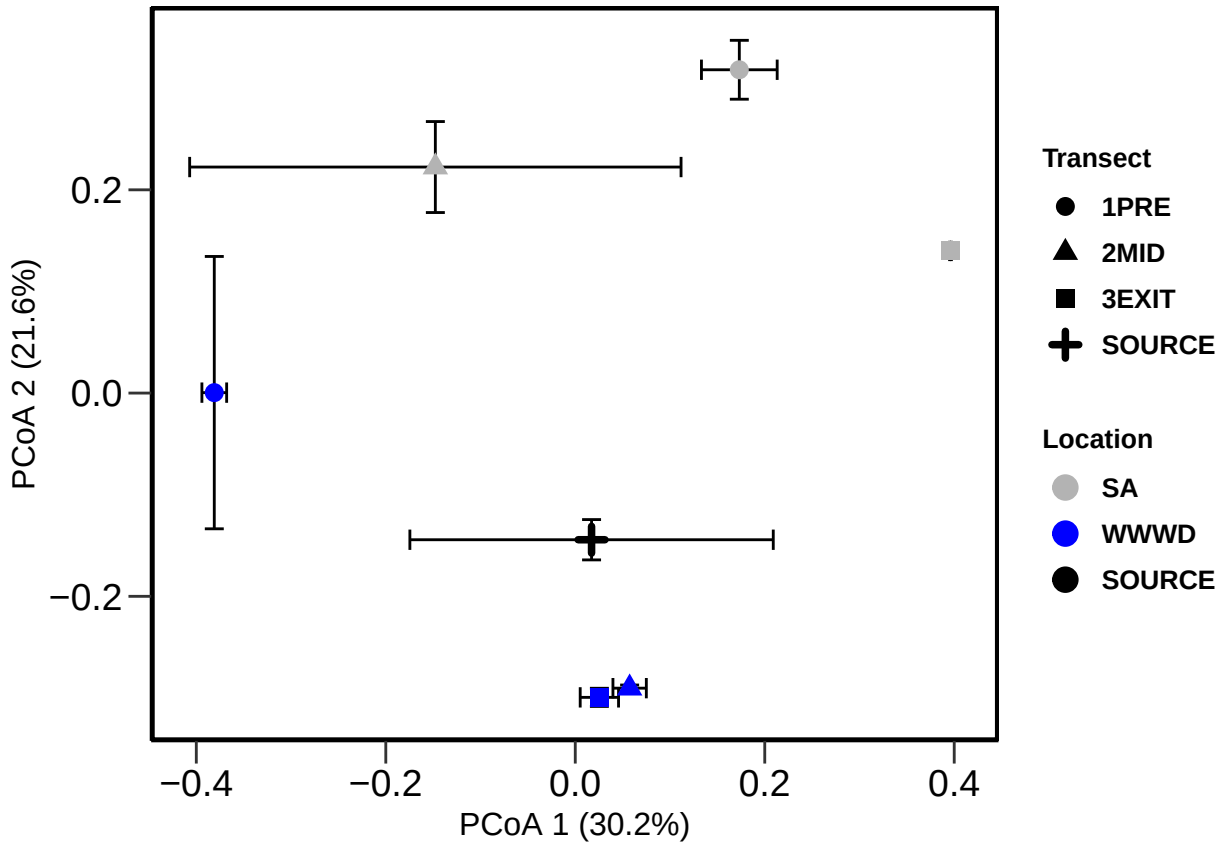
```
## [1] 7 4
```

```
#pcoa.col <- as.factor(sapply(strsplit(pcoa.cent.treats, "_"), `[`, 2)) # Transect
#pcoa.shape <- as.factor(sapply(strsplit(pcoa.cent.treats, "_"), `[`, 1))  # Location

Location <- c("1SA", "1SA", "1SA","3SOURCE","2WWWD", "2WWWD", "2WWWD")
Transect <- c("1PRE", "2MID", "3EXIT","SOURCE","1PRE", "2MID", "3EXIT")

pcoa.cent.dataframe.trts$Location <- as.factor(Location)
pcoa.cent.dataframe.trts$Transect <- as.factor(Transect)
dim(pcoa.cent.dataframe.trts) #28 7
```

```
## [1] 7 6
```

```
# Principal Coordinates Analysis
#Plot
df1a <- as.data.frame(pcoa.cent.dataframe.trts)
SH.bact <- ggplot(df1a, aes(x=V1, y=V2), group = interaction(Location,Transect))+
theme_bw() +
#Set error bars for geom_point
```

```
geom_errorbarh(aes(xmax=V1+V1e, xmin=V1-V1e, height=0.02), colour="black") +
geom_errorbar(aes(ymax=V2+V2e, ymin=V2-V2e, width=0.02), colour="black") +
geom_point(aes(shape = Transect, colour = Location), stroke = 2, size=2) +
#removes gridlines from plot
theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
axis.line = element_line(colour = "black"))+
#Set colors for treatments
scale_colour_manual(labels = c("SA","WWWD","SOURCE"),
                    values = c("gray70","blue", "black")) +
theme(axis.title = element_text(size=12),# face="bold"),
axis.text.x = element_text(size=14, color="black"),  axis.text.y = element_text(size=14, color="black")
panel.border = element_rect(colour = "black", size=1.25)) +
#Set plot title text size
theme(plot.title=element_text(size=12)) +
#Set legend text size
theme(legend.text=element_text(size=10, face="bold"), legend.title = element_text(size=10, face="bold"))
#Sets size of tick marks on axis
theme(axis.ticks.length=unit(0.3,"cm")) +
#Sets labels for plot title, axis titles, and legend headings
xlab("PCoA 1 (30.2%)") + ylab("PCoA 2 (21.6%)")

SH.bact
```




```
ggsave("../figures/SH.bact.ordination.png", plot=last_plot(), device=NULL, path=NULL, scale=1, width=7,
```

```
require(phyloseq)
```

```
## Loading required package: phyloseq
```

```
SH_16s <- import_mothur(mothur_shared_file = "../data/SH.opti_mcc.shared", mothur_constaxonomy_file = "
SH_16s
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 7992 taxa and 19 samples ]
## tax_table()   Taxonomy Table:    [ 7992 taxa by 6 taxonomic ranks ]
```

```
design$Location <- as.factor(design$Location)
design$Transect <- as.factor(design$Transect)
design$Location_ID <- as.factor(design$Location_ID)
str(design)
```

```
## 'data.frame':    19 obs. of  6 variables:
##  $ Location         : Factor w/ 3 levels "SA","SOURCE",..: 3 3 3 3 3 3 3 3 1 1 ...
##  $ Transect         : Factor w/ 4 levels "1PRE","2MID",..: 1 1 1 2 2 2 3 3 1 1 ...
##  $ Location_ID      : Factor w/ 7 levels "SA","SA_EXIT",..: 7 7 7 5 5 5 6 6 3 3 ...
##  $ Location_ID_order: chr  "1WWWD_PRE" "1WWWD_PRE" "1WWWD_PRE" "2WWWD_BIRD" ...
##  $ Description      : chr  "EMU" "EMU" "EMU" "INFLOW" ...
##  $ Rep              : int  1 2 3 1 2 3 1 2 1 2 ...
```

```
sample <- sample_data(design)

sample_data(SH_16s)<- sample

colnames(tax_table(SH_16s))
```

```
## [1] "Rank1" "Rank2" "Rank3" "Rank4" "Rank5" "Rank6"
```

```
colnames(tax_table(SH_16s)) <- c("Kingdom", "Phylum", "Class",
  "Order", "Family", "Genus")

after_remove_low_depth <- prune_samples(sample_sums(SH_16s) >= 6000, SH_16s)
head(sample_sums(after_remove_low_depth))
```

```
##  SH_1 SH_10 SH_11 SH_12 SH_13 SH_14
## 18904 19828 14099 18365 12250 16743
```

```
set.seed(1)
rare <- rarefy_even_depth(after_remove_low_depth, sample.size = 8000,rngseed=TRUE)
```

```
## `set.seed(TRUE)` was used to initialize repeatable random subsampling.

## Please record this for your records so others can reproduce.

## Try `set.seed(TRUE); .Random.seed` for the full vector
```

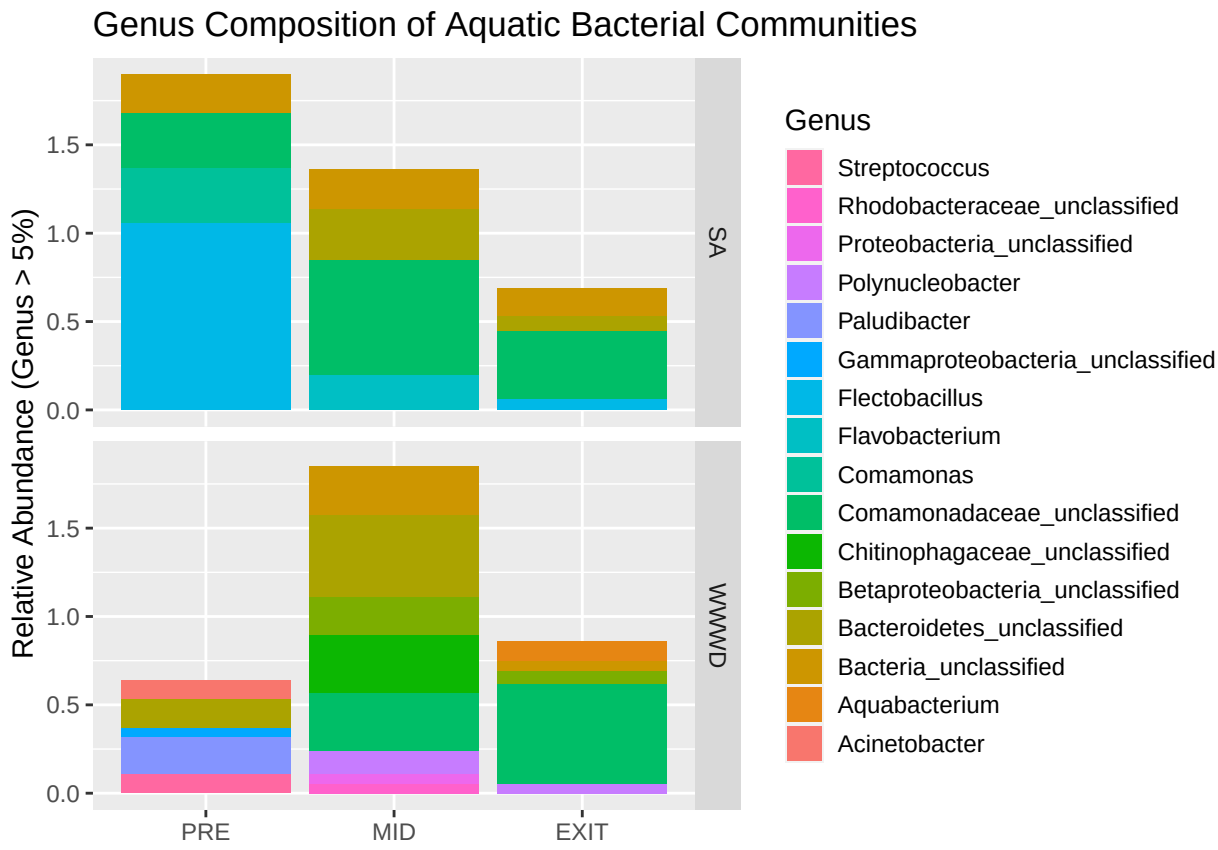
```
## ...

## 26330TUs were removed because they are no longer
## present in any sample after random subsampling

## ...
```

```
#26330TUs were removed because they are no longer present in any sample after random subsampling

head(sample_sums(rare))
```

```
##  SH_1 SH_10 SH_11 SH_12 SH_13 SH_14
##  8000  8000  8000  8000  8000  8000
```

```
#remove the NTC sample. Check to make sure it doesn't have too many sequences before you through it awa
#to_remove <- c("NTC")
#pruned <- prune_samples(!(rownames(sample_data(rare)) %in% to_remove), rare)

#filter out OTUs less than 10
#darte_ed_16s_filter <- filter_taxa(pruned, function(x) sum(x) > 10, TRUE)

 #relative abundance
SH_16s_filter_re <- transform_sample_counts(rare, function(x) x /sum(x))

#Get rid of small taxa
SH_16s_filter2 <- filter_taxa(SH_16s_filter_re, function(x) sum(x) > .001, TRUE)

#Combine OTUs with common taxa
SH_16s_filter_re_g = tax_glom(SH_16s_filter2, "Phylum")
SH_16s_filter_re_g2 = tax_glom(SH_16s_filter2, "Genus")


SH_genus <- SH_16s %>%
  tax_glom(taxrank = "Genus") %>%                      # agglomerate at phylum level
  transform_sample_counts(function(x) {x/sum(x)} ) %>% # Transform to rel. abundance
  psmelt() %>%                                          # Melt to long format
  filter(Abundance > 0.05) %>%                          # Filter out low abundance taxa
  arrange(Genus)                                       # Sort data frame alphabetically by phylum
# Set colors for plotting
genus_colors <- c(
  "salmon", "darkseagreen", "gold", "magenta", "slateblue", "bisque", "darkred", "cadetblue", "darkoran
)


# Plot
a <- list(
  font = list(size = 14),
  xref = "paper",
  yref = "paper",
  yanchor = "bottom",
  xanchor = "center",
  align = "center",
  x = 0.5,
```

```
  y = 1,
  showarrow = FALSE)

SH_genus_v2 <- subset(SH_genus, Location == "SA" | Location == "WWWD")

ggplot(transform(SH_genus_v2, Transect=factor(Transect, levels=c("1PRE","2MID","3EXIT"))), aes(x = Trans
  geom_bar(stat = "identity") +
  #scale_fill_manual(values = genus_colors) +
  scale_x_discrete(labels = c("PRE","MID","EXIT"), drop = TRUE) +
  # Remove x axis title
  theme(axis.title.x = element_blank()) +
  #
  guides(fill = guide_legend(reverse = TRUE, keywidth = 1, keyheight = 1)) +
  ylab("Relative Abundance (Genus > 5%)") +
  ggtitle("Genus Composition of Aquatic Bacterial Communities")
```



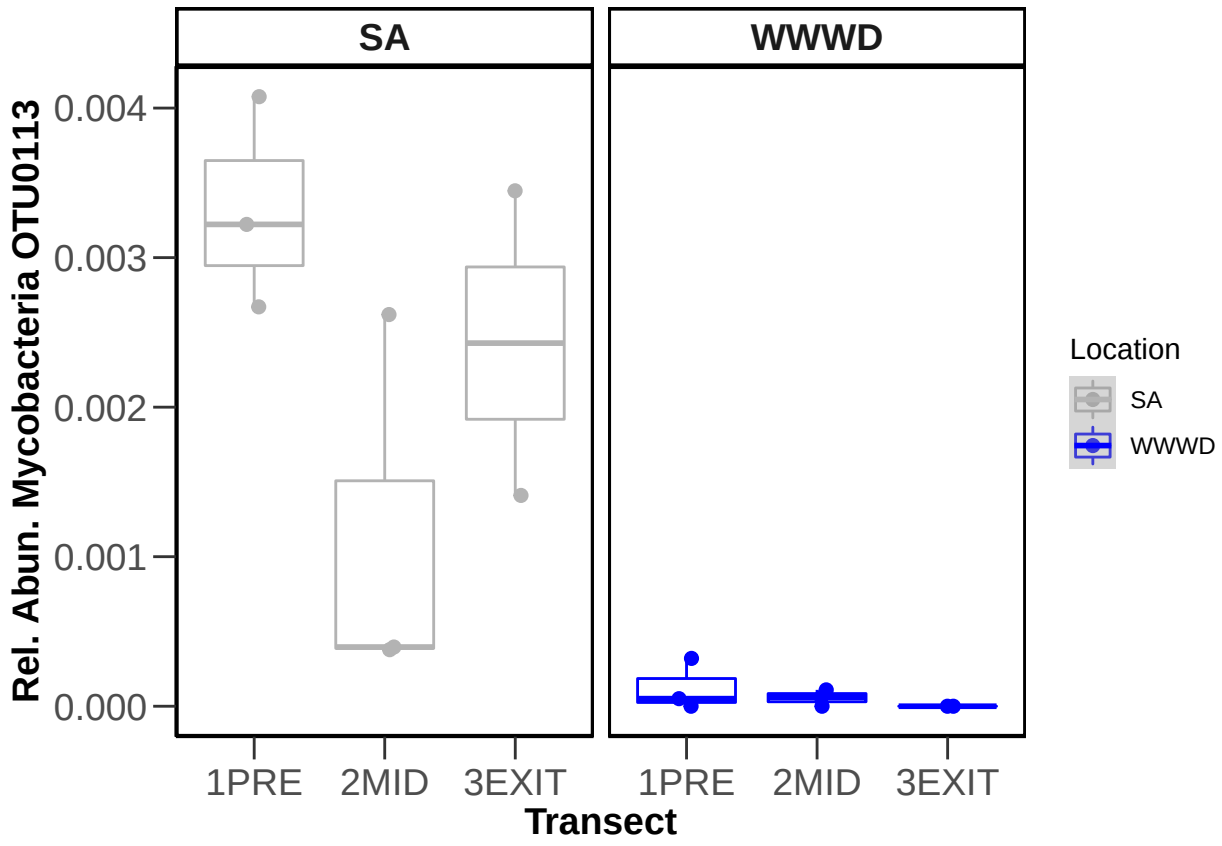

```
ggsave("../figures/genuscomp_updated.png", plot=last_plot(), device=NULL, path=NULL, scale=1, width=7, h
```

OTU graph

```
#NOTE:
#Otu0113 (451 reads)
```

```
#Otu1461 (12 reads)
#Otu2496 (6 reads)
#Otu3114 (4 reads)

#not graphing source
new.data.nosource <- new.data[c(1:16),]
p <- ggplot(new.data.nosource, aes(x=Transect, y=Otu0113, color=Location))+ geom_boxplot() +
      geom_point(aes(color=Location), size=2, position = position_jitterdodge())+ scale_color_manual(nam
p1=p+geom_smooth(method="lm")+facet_wrap(~Location)
Otu0113<-p1 + theme_bw() +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line
          =element_line(colour = "black")) +
    theme(axis.title=element_text(vjust=1,size=14,face="bold"),
          axis.text=element_text(size=14), axis.text.x = element_text(vjust=0.65, hjust=0.5,
          size=14), panel.border = element_rect(colour = "black",size=1)) +
    theme(axis.ticks.length=unit(0.3,"cm")) + labs(x = "Transect", y = "Rel. Abun. Mycobacteria OTU0113
    theme(strip.text.x = element_text(size=14, face="bold"), strip.text.y =
          element_text(size=14, face="bold"), strip.background = element_rect(colour="black",
          fill="white", size=1))
Otu0113
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
ggsave("../figures/OTU0113.png", plot=last_plot(), device=NULL, path=NULL, scale=1, width=7, height=5, c
```

```
## `geom_smooth()` using formula 'y ~ x'
```

Bacterial community indicator species analysis

```
new.data <-cbind(design,dataREL)
library("labdsv")
```

```
## Loading required package: mgcv
```

```
## This is mgcv 1.8-40. For overview type 'help("mgcv-package")'.
```

```
## Registered S3 method overwritten by 'labdsv':
##   method       from
##   summary.dist ade4
```

```
## This is labdsv 2.0-1
## convert existing ordinations with as.dsvord()
```

```
##
## Attaching package: 'labdsv'
```

```
## The following object is masked from 'package:stats':
##
##     density
```

```
group = interaction(new.data$Location_ID)
design.type <- group

dataREL.ind <- dataREL[, colSums(dataREL) > 0.05]
bac.ind <- indval(dataREL.ind, design.type)
levels(design.type)
```

```
## [1] "SA"        "SA_EXIT"   "SA_PRE"    "SOURCE"    "WWWD"      "WWWD_EXIT"
## [7] "WWWD_PRE"
```

```
#"SA"        "SA_EXIT"   "SA_PRE"    "SOURCE"    "WWWD"      "WWWD_EXIT" "WWWD_PRE"
summary(bac.ind)
```

```
##         cluster indicator_value probability
## Otu0029       1          0.5444       0.044
## Otu0022       2          0.7033       0.021
## Otu0046       2          0.6942       0.025
## Otu0032       2          0.6724       0.024
## Otu0007       2          0.6717       0.022
## Otu0031       2          0.6608       0.009
```

```
## Otu0012       2          0.5736       0.019
## Otu0041       2          0.5699       0.023
## Otu0034       2          0.5689       0.024
## Otu0009       2          0.3438       0.031
## Otu0003       2          0.2964       0.038
## Otu0010       3          0.9998       0.003
## Otu0004       3          0.9996       0.002
## Otu0014       3          0.8551       0.003
## Otu0043       3          0.4844       0.008
## Otu0015       4          0.6221       0.012
## Otu0052       5          0.6841       0.004
## Otu0060       5          0.6750       0.003
## Otu0039       5          0.5879       0.031
## Otu0024       5          0.4928       0.030
## Otu0038       5          0.4921       0.036
## Otu0066       5          0.3694       0.037
## Otu0061       6          0.8681       0.001
## Otu0027       6          0.5291       0.037
## Otu0055       6          0.4659       0.040
## Otu0062       6          0.4477       0.029
## Otu0028       6          0.4276       0.032
## Otu0025       7          0.3743       0.019
## Otu0035       7          0.3464       0.016
##
## Sum of probabilities                   =  6.027
##
## Sum of Indicator Values                =  30.99
##
## Sum of Significant Indicator Values    =  17.02
##
## Number of Significant Indicators       =  29
##
## Significant Indicator Distribution
##
##  1  2  3  4  5  6  7
##  1 10  4  1  6  5  2
```

```
inds <- which(bac.ind$pval <= 0.05)
bac.indicators <- as.data.frame(matrix(NA, nrow = length(inds), ncol = 4))
colnames(bac.indicators) <- c("OTU", "Cluster", "IndVal", "Prob")

bac.indicators$OTU <- names(inds)
bac.indicators$Cluster <- bac.ind$maxcls[inds]
bac.indicators$IndVal <- bac.ind$indcls[inds]
bac.indicators$Prob <- bac.ind$pval[inds]

ind.tax <- otu.tax[which(as.character(otu.tax$OTU) %in% bac.indicators$OTU), ]
ind.tax <- ind.tax[match(ind.tax$OTU, bac.indicators$OTU), ]

indicator.bac <- cbind(bac.indicators, ind.tax[, -c(1)])

indicator.bac <- indicator.bac[order(as.numeric(indicator.bac$Cluster)), ]

table(indicator.bac$Cluster)
```

```
##
##  1  2  3  4  5  6  7
##  1 10  4  1  6  5  2
```

```
table(indicator.bac$Phylum)
```

```
##
##        Actinobacteria Bacteria_unclassified         Bacteroidetes
##                     1                     2                     7
##            Firmicutes        Proteobacteria
##                     2                    17
```

```
table(indicator.bac$Cluster)
```

```
##
##  1  2  3  4  5  6  7
##  1 10  4  1  6  5  2
```

```
levels(design.type)
```

```
## [1] "SA"        "SA_EXIT"   "SA_PRE"    "SOURCE"    "WWWD"      "WWWD_EXIT"
## [7] "WWWD_PRE"
```

```
# Export Bacteria Indicator Table
write.table(indicator.bac, "../data/BacterialIndicators_Location.txt",
            sep="\t", row.names = F, quote = F)
```