Microbiome Workflow Code

Jonas Frankel-Bricker

5/1/2019

Contents

Packages Required

```r
# R version v 3.5.1
library(dada2)
# v 1.8
library(phangorn)
# v 2.4
library(phyloseq)
# v 1.24.2
library(genefilter)
# v 1.62.0
library(tidyverse)
# v 1.2.1
library(lme4)
# v 1.1-19
library(car)
# v 3.0-2
library(cvequality)
# v 0.1.3
library(sjstats)
# v 0.17.2
library(plyr)
# v 1.8.4
library(cowplot)
# v 0.0.3
library(devtools)
# v 2.0.1
source_url("https://raw.githubusercontent.com/gauravsk/ranacapa/master/R/ggrare.R")
library(vegan)
# v 2.5-3
library(BiodiversityR)
# 2.10-1
library(Rmisc)
# v 1.5
library(reshape)
# v 3.5.1
```

DADA2 Pipeline

```r
###Set path for raw sequencing read location
raw_path <- file.path("Data/Raw_Adjusted/")
list.files(raw_path)

###Set path where filtered reads will be depositied
filtpath_R1_16S <- file.path("Data/Filtered_16S/R1/")
filtpath_R2_16S <- file.path("Data/Filtered_16S/R2/")

###Assign 16S forward (R1) fastq.gz files
```

```
fnFs_16S <- sort(list.files(raw_path, pattern = "16S_R1.fastq.gz", full.names = TRUE))
fnFs_16S

###Assign 16S reverse (R2) fastq.gz files
fnRs_16S <- sort(list.files(raw_path, pattern = "16S_R2.fastq.gz", full.names = TRUE))
fnRs_16S

###Plot Quality Profile of Forward (R1) 16S Reads (R1_16S_QualityProfile)
plotQualityProfile(fnFs_16S[c(1, 75, 150, 225)])

###Plot Quality Profile of Reverse (R2) 16S Reads (R2_16S_QualityProfile)
plotQualityProfile(fnRs_16S[c(1, 75, 150, 225)])

###Set path filter and trim step
filtFs_16S <- file.path(filtpath_R1_16S, basename(fnFs_16S))
filtRs_16S <- file.path(filtpath_R2_16S, basename(fnRs_16S))

---------------------------------------------------------------------------------------------

###Use FLASH (v 1.2.11) to estimate amplicon insert sizes from unmerged reads

###Example code for first sample
#FLASH Code
./flash A1i16S_R1.fastq.gz A1i16S_R2.fastq.gz --max-overlap=175 --output-directory=A1i16S_FLASH --compre

#Import data into R
A1i16S <- read.csv("Tables/A1i16S_FLASH.csv")

sum(A1i16S$frequency) #=24246 "Total" amplicons

#Point plot
A1i16S_point <- ggplot(A1i16S, aes(amplicon_length, frequency)) + geom_point()
A1i16S_point

---------------------------------------------------------------------------------------------

###Trim and filter raw reads
outfiltered_16S_28 <- filterAndTrim(fnFs_16S, filtFs_16S_28, fnRs_16S, filtRs_16S_28, truncLen = c(278,
summary(outfiltered_16S_28)

###Samples to remove from outfiltered_16S_28 (< 100 filtered reads)
NegCtrl116S
ASW16S
BL116S
FT2116S
NegCtrl216S
BL316S
FT3216S
BL216S


###Subset samples with > 100 filtered reads
keep_16S_28 <- outfiltered_16S_28[, "reads.out"] > 100
```

```
###Create  data frame keeping only samples with > 100 filtered reads
outfiltered_16S_28_new <- outfiltered_16S_28[keep_16S_28 ,]

###Importing filtered reads
filtered_R1_16S_28_2 <- file.path(filtpath_R1_16S_28, basename(filtered_R1_16S_28))
filtered_R2_16S_28_2 <- file.path(filtpath_R2_16S_28, basename(filtered_R2_16S_28))

###Only keep filtered reads for samples with > 100 filtered reads
filtered_R1_16S_28_2 <- filtered_R1_16S_28_2[keep_16S_28]
filtered_R2_16S_28_2 <- filtered_R2_16S_28_2[keep_16S_28]

###Condense sample names
samnames_R1_16S_28_2 <- sapply(strsplit(basename(filtered_R1_16S_28_2), "_"), `[`, 1)
samnames_R2_16S_28_2 <- sapply(strsplit(basename(filtered_R2_16S_28_2), "_"), `[`, 1)

###Rename filtered reads with condensed sample names
names(filtered_R1_16S_28_2) <- samnames_R1_16S_28_2
names(filtered_R2_16S_28_2) <- samnames_R2_16S_28_2

###Learn error rates
set.seed(97500)
errF_16S_28_2 <- learnErrors(filtered_R1_16S_28_2, multithread = TRUE)
errR_16S_28_2 <- learnErrors(filtered_R2_16S_28_2, multithread = TRUE)


###Composite code for dereplication of reads, dada2 pipeline incorporating error rates, and merging PE r

for(sam in samnames_R1_16S_28_2) {
  cat("Processing:", sam, "\n")
  derepF_28 <- derepFastq(filtered_R1_16S_28_2[[sam]])
  ddF_28 <- dada(derepF_28, err = errF_16S_28_2)
  derepR_28 <- derepFastq(filtered_R2_16S_28_2[[sam]])
  ddR_28 <- dada(derepR_28, err = errR_16S_28_2)
  merger_16S_28 <- mergePairs(ddF_28, derepF_28, ddR_28, derepR_28)
  mergers_16S_28_2[[sam]] <- merger_16S_28
}

###Make sequence table with merged reads
seqtab_16S_4 <- makeSequenceTable(mergers_16S_28_2)

###Remove chimeras
seqtab_16S_nochim_4 <- removeBimeraDenovo(seqtab_16S_4, method = "consensus")

saveRDS(seqtab_16S_nochim_4, "Tables/seqtab_16S_nochim_4.rds")

###Track reads through pipeline
getN_16S <- function(x) sum(getUniques(x))
track_16S_4 <- cbind(outfiltered_16S_28_new, rowSums(seqtab_16S_nochim_4))
colnames(track_16S_4) <- c("input_4", "filtered_4", "nochim_4")
rownames(track_16S_4) <- samnames_R1_16S_28_2

write.csv(track_16S_4, "Tables/track_16S_4.csv")
track_16S_4 <- read.csv("Tables/track_16S_4.csv")
```

```
###Assign taxonomy to seqtab_nochim_4 using SILVA v132 Database
taxa_16S_4 <- assignTaxonomy(seqtab_16S_nochim_4, "SILVA/16S/silva_nr_v132_train_set.fa.gz")

saveRDS(taxa_16S_4, "Tables/taxa_16S_4.rds")

###Sequence Alignment
alignment_16S_4 <- AlignSeqs(DNAStringSet(seqs_16S_4), anchor = NA)

###Construct neighbor joining tree
phang.align_16S_4 <- phyDat(as(alignment_16S_4, "matrix"), type = "DNA")
dm_16S_4 <- dist.ml(phang.align_16S_4)
treeNJ_16S_4 <- NJ(dm_16S_4)

###Fit GTR+G+I ML tree
fit_16S_4 = pml(treeNJ_16S_4, data = phang.align_16S_4)
fitGTR_16S_4 <- update(fit_16S_4, k=4, inv=0.2)
fitGTR_16S_4 <- optim.pml(fitGTR_16S_4, model = "GTR", optInv = TRUE, optGamma = TRUE, rearrangement =

###Combine into a Phyloseq Object
pstree_16S <- phyloseq(otu_table(seqtab_16S_nochim_4, taxa_are_rows = FALSE, errorIfNULL = TRUE), sampl

saveRDS(pstree_16S, "Data/pstree_16S.rds")
```

Identification and Removal of Contaminant OTU Sequences

```
library(decontam)

### Import phyloseq object
pstree_16S <- readRDS("Data/pstree_16S.rds")
pstree_16S_kit <- subset_samples(pstree_16S, Extraction_Kit !=
    "4")

### Identify Contaminants with 'prevalence' method and
### threshold set to 0.5
sample_data(pstree_16S_kit)$is.neg <- sample_data(pstree_16S_kit)$Type ==
    "kit"
contam.prev <- isContaminant(pstree_16S_kit, method = "prevalence",
    neg = "is.neg", threshold = 0.5)
table(contam.prev$contaminant)

### Prune contaminant OTUs from dataset
pstree_16S_decom <- prune_taxa(!contam.prev$contaminant, pstree_16S_kit)
pstree_16S_decom

saveRDS(pstree_16S_decom, "*Decom_Data/pstree_16S_decom.rds")
```

Larval Gut Microbiome Comparisons

Initial Processing and Alpha Diversity Measurements

```
### Import phyloseq object
pstree_16S_decom <- readRDS("*Decom_Data/pstree_16S_decom.rds")

### Subset_samples for treatments A and B, experiment == 'exp'
Q1_16S_decom <- subset_samples(pstree_16S_decom, Experiment ==
    "exp")
Q1_16S_decom <- subset_samples(Q1_16S_decom, Treatment == "A" |
    Treatment == "B")

### Rename values if needed
as.character(Q1_16S_decom@sam_data$Treatment)
Q1_16S_decom@sam_data$Treatment <- revalue(Q1_16S_decom@sam_data$Treatment,
    c(A = "Non-Fungal"))
Q1_16S_decom@sam_data$Treatment <- revalue(Q1_16S_decom@sam_data$Treatment,
    c(B = "Fungal"))

saveRDS(Q1_16S_decom, "*Decom_Data/Q1_16S_decom.rds")

### Alpha diversity calculations
set.seed(42387)
meta_Q1_decom <- Q1_16S_decom@sam_data
richness_Q1_decom <- data.frame(estimate_richness(Q1_16S_decom,
    split = TRUE, measures = NULL))
richness_Q1_decom$Sample_ID <- rownames(richness_Q1_decom)

alpha_Q1_decom <- left_join(meta_Q1_decom, richness_Q1_decom)

write.csv(alpha_Q1_decom, "*Decom_Tables//alpha_Q1_decom.csv")

alpha_Q1_decom <- read.csv("*Decom_Tables/alpha_Q1_decom.csv")

alpha_Q1_decom_A <- subset(alpha_Q1_decom, Treatment == "Non-Fungal")
alpha_Q1_decom_B <- subset(alpha_Q1_decom, Treatment == "Fungal")

alpha_Q1_decom_A <- subset(alpha_Q1_decom, Treatment == "Non-Fungal")
alpha_Q1_decom_B <- subset(alpha_Q1_decom, Treatment == "Fungal")

### Linear mixed effects models

# Simpson
Q1_decom_simpson_final <- lmer(data = alpha_Q1_decom, Simpson ~
    Treatment + (1 | Source_Pop), REML = TRUE)

Anova(Q1_decom_simpson_final, test.statistic = "F")

### Coefficient of Variation
cv(alpha_Q1_decom_A$Simpson, alpha_Q1_decom_B$Simpson)

### CV significance Asymptotic test
```

```
asymptotic_test(alpha_Q1_decom$Simpson, alpha_Q1_decom$Treatment)

### M-SLRT Test
mslr_test(nr = 10000, alpha_Q1_decom$Simpson, alpha_Q1_decom$Treatment)


# Shannon
Q1_decom_shannon_final <- lmer(data = alpha_Q1_decom, Shannon ~
    Treatment + (1 | Source_Pop), REML = TRUE)

Anova(Q1_decom_shannon_final, test.statistic = "F")

### Coefficient of Variation
cv(alpha_Q1_decom_A$Shannon, alpha_Q1_decom_B$Shannon)

### CV significance Asymptotic test
asymptotic_test(alpha_Q1_decom$Shannon, alpha_Q1_decom$Treatment)

### M-SLRT Test
mslr_test(nr = 10000, alpha_Q1_decom$Shannon, alpha_Q1_decom$Treatment)
```

Rarefaction, Filtering, and Relative Abundance Barplots

```
### Import phyloseq object
Q1_16S_decom <- readRDS("*Decom_Data/Q1_16S_decom.rds")
set.seed(73250)

### Make rarecurve
Q1_rarecurve_decom <- ggrare(Q1_16S_decom, step = 100, color = "Treatment",
    se = FALSE) + scale_color_manual(values = c("#0000FF", "#FF0000")) +
    geom_vline(xintercept = 5251, linetype = "dashed") + xlab("Sample Read Coverage") +
    ylab("OTU Richness") + theme_bw()
Q1_rarecurve_decom$labels$colour <- "Larva Type"
Q1_rarecurve_decom <- Q1_rarecurve_decom + theme(text = element_text(size = 25),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )

### Create sample sums table
Q1_samsums_16S_decom <- data.frame(sample_sums(Q1_16S_decom))
colnames(Q1_samsums_16S_decom) <- "Read_Counts"
write.csv(Q1_samsums_16S_decom, "*Decom_Tables//Q1_samsums_16S_decom.csv")

### Rarefy to 5251 (3 sample cut)
Q1_16S_decom_rare <- rarefy_even_depth(Q1_16S_decom, sample.size = 5251)

### 3 samples removed:A3V16S, B1VII16S, A2ii16S

### Filter out singletons
singfilterfun_Q1 <- filterfun(kOverA(1, 1))

Q1_16S_decom_filter1 <- filter_taxa(Q1_16S_decom_rare, function(x) singfilterfun_Q1(x),
    TRUE)

### Convert tree to relative abundance
Q1_16S_decom_relative1 <- transform_sample_counts(Q1_16S_decom_filter1,
    function(x) x/sum(x))

### Must appear at greater than 0.1% in at least 1 occurrence
### (removes OTUs with 5 reads or less in each sample across
### all samples)
singfilterfun_Q1_2 <- filterfun(kOverA(1, 0.001))

Q1_16S_decom_relative2 <- filter_taxa(Q1_16S_decom_relative1,
    function(x) singfilterfun_Q1_2(x), TRUE)
Q1_16S_decom_relative2

saveRDS(Q1_16S_decom_relative2, "*Decom_Data/Q1_16S_decom_relative2.rds")

### Jumping to Barplots
Q1_16S_decom_relative2 <- readRDS("*Decom_Data/Q1_16S_decom_relative2.rds")
AvsBrelative2new <- Q1_16S_decom_relative2

as.character(AvsBrelative2new@sam_data$Treatment)
AvsBrelative2new@sam_data$Treatment <- revalue(AvsBrelative2new@sam_data$Treatment,
    c(A = "Non-Fungal"))
AvsBrelative2new@sam_data$Treatment <- revalue(AvsBrelative2new@sam_data$Treatment,
    c(B = "Fungal"))

### Tax_glom to family
pstree_16S_Q1_relative2_family <- tax_glom(AvsBrelative2new,
    taxrank = "Family")

### Identify top 15 families
family_15_Q1 <- names(sort(taxa_sums(pstree_16S_Q1_relative2_family),
    TRUE)[1:15])

### Prune top 15
family_15_Q1_prune <- prune_taxa(family_15_Q1, pstree_16S_Q1_relative2_family)

### Make table
family_15_Q1_table <- cbind(tax_table(family_15_Q1_prune))

### Consolidate table formatting
family_15_Q1_table[family_15_Q1, "Family"] <- as(tax_table(family_15_Q1_prune)[family_15_Q1,
    "Family"], "character")

tax_table(family_15_Q1_prune) <- family_15_Q1_table

### Transform to percentages
family_15_Q1_merge <- merge_samples(family_15_Q1_prune, "Treatment")
family_15_Q1_percent <- transform_sample_counts(family_15_Q1_merge,
    function(x) 100 * x/sum(x))
```

```r
### Plot relative abundances of top 15 families for C
### (family_15_Q3_plot)
AvsB_barplot_final <- plot_bar(family_15_Q1_percent, fill = "Family") +
    xlab("Larva Type") + ylab("Relative Abundance") + scale_fill_manual(values = c("#FFCC00",
    "#FF9900", "#FF6600", "#FF3300", "#CC0033", "#FF0033", "#FF0099",
    "#660066", "#CC00FF", "#9900FF", "#3399CC", "#0099FF", "#00FFFF",
    "#33FFCC", "#66FF00")) + theme_bw()
AvsB_barplot_final
```

Beta Diversity Measures

```r
### Testing Beta diversity and Betadisper
Q1_16S_decom_relative2 <- readRDS("*Decom_Data/Q1_16S_decom_relative2.rds")
set.seed(84980)

### Calculate Bray Distance
df_bray_Q1 <- as(sample_data(Q1_16S_decom_relative2), "data.frame")
d_bray_Q1 <- distance(Q1_16S_decom_relative2, "bray")

### Nested PERMANOVA
nested.npmanova(d_bray_Q1 ~ Treatment + Source_Pop, method = "bray",
    data = df_bray_Q1, permutations = 999)

# Betadisper
Q1_betadisper1 <- betadisper(d_bray_Q1, df_bray_Q1$Treatment)
permutest(Q1_betadisper1, permutations = 999)


### Calculate Unweighted UniFrac Distance
df_unifrac_Q1 <- as(sample_data(Q1_16S_decom_relative2), "data.frame")
d_unifrac_Q1 <- distance(Q1_16S_decom_relative2, "unifrac")

### Nested PERMANOVA
nested.npmanova(d_unifrac_Q1 ~ Treatment + Source_Pop, method = "unifrac",
    data = df_unifrac_Q1, permutations = 999)

# Betadisper
Q1_betadisper2 <- betadisper(d_unifrac_Q1, df_unifrac_Q1$Treatment)
permutest(Q1_betadisper2, permutations = 999)


### Calculate Weighted UniFrac Distance
df_wunifrac_Q1 <- as(sample_data(Q1_16S_decom_relative2), "data.frame")
d_wunifrac_Q1 <- distance(Q1_16S_decom_relative2, "wunifrac")

### Nested PERMANOVA
nested.npmanova(d_wunifrac_Q1 ~ Treatment + Source_Pop, method = "wunifrac",
    data = df_wunifrac_Q1, permutations = 999)

# Betadisper
Q1_betadisper3 <- betadisper(d_wunifrac_Q1, df_wunifrac_Q1$Treatment)
permutest(Q1_betadisper3, permutations = 999)
```

```
### Calculate Jaccard index
df_jaccard_Q1 <- as(sample_data(Q1_16S_decom_relative2), "data.frame")
d_jaccard_Q1 <- distance(Q1_16S_decom_relative2, "jaccard")

### Nested PERMANOVA
nested.npmanova(d_jaccard_Q1 ~ Treatment + Source_Pop, method = "jaccard",
    data = df_jaccard_Q1, permutations = 999)

# Betadisper
Q1_betadisper4 <- betadisper(d_jaccard_Q1, df_jaccard_Q1$Treatment)
permutest(Q1_betadisper4, permutations = 999)
```

Linear mixed models for relative abundances of the top 15 shared families

```
### LMERS on top 15 families shared between groups
Q1_16S_decom_relative2 <- readRDS("*Decom_Data//Q1_16S_decom_relative2.rds")

### Subset for non-fungal larvae
Q1_16S_nonfungal <- subset_samples(Q1_16S_decom_relative2, Treatment ==
    "Non-Fungal")

### Subset for fungal larvae
Q1_16S_fungal <- subset_samples(Q1_16S_decom_relative2, Treatment ==
    "Fungal")


### Relative Abundances of Target Taxa

### Bacillaceae Non-Fungal
Bacillaceae_Q1_nonfungal <- subset_taxa(Q1_16S_nonfungal, Family ==
    "Bacillaceae")
# Relative abundance
sum(sample_sums(Bacillaceae_Q1_nonfungal))/nsamples(Q1_16S_nonfungal)
0.0009523275

# Fungal
Bacillaceae_Q1_fungal <- subset_taxa(Q1_16S_fungal, Family ==
    "Bacillaceae")
# Relative abundance
sum(sample_sums(Bacillaceae_Q1_fungal))/nsamples(Q1_16S_fungal)
0.0954952

# Subset for combined data
Bacillaceae_Q1 <- subset_taxa(Q1_16S_decom_relative2, Family ==
    "Bacillaceae")

# Metadata Sheet
Bacillaceae_Q1_meta <- Bacillaceae_Q1@sam_data

# Abundance variable
```

```
Bacillaceae_Q1_meta$Abundance <- sample_sums(Bacillaceae_Q1)

# Dataframe
Bacillaceae_Q1_meta_data <- as(Bacillaceae_Q1_meta, Class = "data.frame")

### Final linear mixed-effects model
Q1_Bacillaceae_treat_final <- lmer(data = Bacillaceae_Q1_meta_data,
    Abundance ~ Treatment + (1 | Source_Pop), REML = TRUE)
car::Anova(Q1_Bacillaceae_treat_final, test.statistic = "F")

### Repeat workflow for: Bdellovibrionaceae, Beijerinckiaceae,
### Burkholderiaceae, Caulobacteraceae, Chitinophagaceae,
### Enterobacteriaceae, env.OPS_17, Flavobacteriaceae,
### Microbacteriaceae, Pseudomonadaceae, Rhizobiaceae,
### Sphingobacteriaceae, Sphingomonadaceae, Weeksellaceae
```

Transstadial Transmssion (Non-Fungal Mosquitoes)

Initial Processing and Alpha Diversity Measurements

```
### Import phyloseq object
pstree_16S_decom <- readRDS("*Decom_Data/pstree_16S_decom.rds")

### Subset samples for treatments A and C
Q6_1_16S_decom <- subset_samples(pstree_16S_decom, Experiment ==
    "exp" & Sex == "female" & Frozen == "FALSE")
Q6_1_16S_decom <- subset_samples(Q6_1_16S_decom, Treatment ==
    "A" | Treatment == "C")

### Rename values if needed
Q6_1_16S_decom@sam_data$Age <- revalue(Q6_1_16S_decom@sam_data$Age,
    replace = c(larva = "Larvae"))
Q6_1_16S_decom@sam_data$Age <- revalue(Q6_1_16S_decom@sam_data$Age,
    replace = c(adult = "Adults"))

saveRDS(Q6_1_16S_decom, "*Decom_Data//Q6_1_16S_decom.rds")

Q6_1_16S_decom <- readRDS("*Decom_Data//Q6_1_16S_decom.rds")

### Alpha diversity calculations
set.seed(42387)
meta_Q6_1_decom <- Q6_1_16S_decom@sam_data
richness_Q6_1_decom <- data.frame(estimate_richness(Q6_1_16S_decom,
    split = TRUE, measures = NULL))
richness_Q6_1_decom$Sample_ID <- rownames(richness_Q6_1_decom)

alpha_Q6_1_decom <- left_join(meta_Q6_1_decom, richness_Q6_1_decom)

write.csv(alpha_Q6_1_decom, "*Decom_Tables/alpha_Q6_1_decom.csv")

alpha_Q6_1_decom <- read.csv("*Decom_Tables/alpha_Q6_1_decom.csv")
```

```
alpha_Q6_1_decom_A <- subset(alpha_Q6_1_decom, Age == "Larvae")
alpha_Q6_1_decom_C <- subset(alpha_Q6_1_decom, Age == "Adults")


### Linear mixed models

# Simpson
Q6_1_decom_simpson_final <- lmer(data = alpha_Q6_1_decom, Simpson ~
    Treatment + (1 | Source_Pop), REML = TRUE)
Anova(Q6_1_decom_simpson_final, test.statistic = "F")

### Coefficient of Variation
cv(alpha_Q6_1_decom_A$Simpson, alpha_Q6_1_decom_C$Simpson)

### CV significance Asymptotic test
asymptotic_test(alpha_Q6_1_decom$Simpson, alpha_Q6_1_decom$Treatment)

### M-SLRT Test
mslr_test(nr = 10000, alpha_Q6_1_decom$Simpson, alpha_Q6_1_decom$Treatment)


# Shannon
Q6_1_decom_shannon_final <- lmer(data = alpha_Q6_1_decom, Shannon ~
    Treatment + (1 | Source_Pop), REML = TRUE)
Anova(Q6_1_decom_shannon_final, test.statistic = "F")

### Coefficient of Variation
cv(alpha_Q6_1_decom_A$Shannon, alpha_Q6_1_decom_C$Shannon)

### CV significance Asymptotic test
asymptotic_test(alpha_Q6_1_decom$Shannon, alpha_Q6_1_decom$Treatment)

### M-SLRT Test
mslr_test(nr = 10000, alpha_Q6_1_decom$Simpson, alpha_Q6_1_decom$Treatment)
```

Rarefaction, Filtering, and Relative Abundance Barplots

```
### Import phyloseq object
Q6_1_16S_decom <- readRDS("*Decom_Data/Q6_1_16S_decom.rds")
set.seed(73250)

### Make rarecurve
as.character(Q6_1_16S_decom@sam_data$Age)

Q6_1_rarecurve_decom <- ggrare(Q6_1_16S_decom, step = 100, color = "Age",
    se = FALSE) + geom_vline(xintercept = 2560, linetype = "dashed") +
    xlab("Sample Read Coverage") + ylab("OTU Richness") + theme_bw()
Q6_1_rarecurve_decom$labels$colour <- "Non-Fungal Mosquitoes"
Q6_1_rarecurve_decom <- Q6_1_rarecurve_decom + theme(text = element_text(size = 25),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )

### Create sample sums table
Q6_1_samsums_16S_decom <- data.frame(sample_sums(Q6_1_16S_decom))
colnames(Q6_1_samsums_16S_decom) <- "Read_Counts"
write.csv(Q6_1_samsums_16S_decom, "*Decom_Tables//Q6_1_samsums_16S_decom.csv")

### Rarefy to 2560 (10 samples cut)
Q6_1_16S_decom_rare <- rarefy_even_depth(Q6_1_16S_decom, sample.size = 2560)

### 10 samples removed: C4VII16S, C4VI16S, C3VI16S, C4V16S,
### A3V16S, C1V16S, C3IV16S, C1IV16S, C1i16S, C4iii16S

### Filter out singletons
singfilterfun_Q6_1 <- filterfun(kOverA(1, 1))

Q6_1_16S_decom_filter1 <- filter_taxa(Q6_1_16S_decom_rare, function(x) singfilterfun_Q6_1(x),
    TRUE)

### Convert tree to relative abundance
Q6_1_16S_decom_relative1 <- transform_sample_counts(Q6_1_16S_decom_filter1,
    function(x) x/sum(x))

### Must appear at greater than 0.2% in at least 1 occurrence
### (removes OTUs with 5 reads or less in each sample)
singfilterfun_Q6_1_2 <- filterfun(kOverA(1, 0.002))

Q6_1_16S_decom_relative2 <- filter_taxa(Q6_1_16S_decom_relative1,
    function(x) singfilterfun_Q6_1_2(x), TRUE)

saveRDS(Q6_1_16S_decom_relative2, "*Decom_Data/Q6_1_16S_decom_relative2.rds")


### Barplot of the relative abundances of the top 15 shared
### families between groups
Q6_1_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_1_16S_decom_relative2.rds")
AvsCrelative2new <- Q6_1_16S_decom_relative2

### Tax_glom to family
pstree_16S_Q6_1_relative2_family <- tax_glom(AvsCrelative2new,
    taxrank = "Family")

### Identify top 15 families
family_15_Q6_1 <- names(sort(taxa_sums(pstree_16S_Q6_1_relative2_family),
    TRUE)[1:15])

### Prune top 15
family_15_Q6_1_prune <- prune_taxa(family_15_Q6_1, pstree_16S_Q6_1_relative2_family)

### Make table
family_15_Q6_1_table <- cbind(tax_table(family_15_Q6_1_prune))

### Consolidate table formatting
```

```
family_15_Q6_1_table[family_15_Q6_1, "Family"] <- as(tax_table(family_15_Q6_1_prune)[family_15_Q6_1,
    "Family"], "character")

tax_table(family_15_Q6_1_prune) <- family_15_Q6_1_table

### Transform to percentages
family_15_Q6_1_merge <- merge_samples(family_15_Q6_1_prune, "Age")
family_15_Q6_1_percent <- transform_sample_counts(family_15_Q6_1_merge,
    function(x) 100 * x/sum(x))

### Plot relative abundances of top 15 families for C
### (family_15_Q3_plot)
AvsC_barplot_final <- plot_bar(family_15_Q6_1_percent, fill = "Family") +
    xlab("Non-Fungal Mosquitoes") + ylab("Relative Abundance") +
    scale_fill_manual(values = c("#FF6600", "#FF3300", "#FF0033",
        "#FF0066", "#FF0099", "#CC00CC", "#9900FF", "#0000FF",
        "#3399CC", "#0099FF", "#33CCFF", "#00FFFF", "#33FFCC",
        "#00FF99", "#66FF00")) + theme_bw()
AvsC_barplot_final
```

Beta Diversity Measures

```
set.seed(84980)
Q6_1_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_1_16S_decom_relative2.rds")

### Calculate Bray Distance
df_bray_Q6_1 <- as(sample_data(Q6_1_16S_decom_relative2), "data.frame")
d_bray_Q6_1 <- distance(Q6_1_16S_decom_relative2, "bray")

### Nested PERMANOVA
nested.npmanova(d_bray_Q6_1 ~ Treatment + Source_Pop, data = df_bray_Q6_1,
    method = "bray", permutations = 999)

# Betadisper
Q6_1_betadisper1 <- betadisper(d_bray_Q6_1, df_bray_Q6_1$Treatment)
permutest(Q6_1_betadisper1, permutations = 999)


### Calculate Unweighted UniFrac Distance
df_unifrac_Q6_1 <- as(sample_data(Q6_1_16S_decom_relative2),
    "data.frame")
d_unifrac_Q6_1 <- distance(Q6_1_16S_decom_relative2, "unifrac")

### Nested PERMANOVA
nested.npmanova(d_unifrac_Q6_1 ~ Treatment + Source_Pop, data = df_unifrac_Q6_1,
    method = "unifrac", permutations = 999)

# Betadisper
Q6_1_betadisper2 <- betadisper(d_unifrac_Q6_1, df_unifrac_Q6_1$Treatment)
permutest(Q6_1_betadisper2, permutations = 999)
```

```r
### Calculate Weighted UniFrac Distance
df_wunifrac_Q6_1 <- as(sample_data(Q6_1_16S_decom_relative2),
    "data.frame")
d_wunifrac_Q6_1 <- distance(Q6_1_16S_decom_relative2, "wunifrac")

### Nested PERMANOVA
nested.npmanova(d_wunifrac_Q6_1 ~ Treatment + Source_Pop, method = "wunifrac",
    data = df_wunifrac_Q6_1, permutations = 999)

# Betadisper
Q6_1_betadisper3 <- betadisper(d_wunifrac_Q6_1, df_wunifrac_Q6_1$Treatment)
permutest(Q6_1_betadisper3, permutations = 999)


### Calculate Jaccard index
df_jaccard_Q6_1 <- as(sample_data(Q6_1_16S_decom_relative2),
    "data.frame")
d_jaccard_Q6_1 <- distance(Q6_1_16S_decom_relative2, "jaccard")

### Nested PERMANOVA
nested.npmanova(d_jaccard_Q6_1 ~ Treatment + Source_Pop, method = "jaccard",
    data = df_jaccard_Q6_1, permutations = 999)

# Betadisper
Q6_1_betadisper4 <- betadisper(d_jaccard_Q6_1, df_jaccard_Q6_1$Treatment)
permutest(Q6_1_betadisper4, permutations = 999)
```

Linear mixed models for relative abundances of the top 15 shared families

```r
### LMERS on top 15 families shared between groups
Q6_1_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_1_16S_decom_relative2.rds")

### Subset for treatment non-fungal larvae
Q6_1_16S_larvae <- subset_samples(Q6_1_16S_decom_relative2, Treatment ==
    "A")

### Subset for treatment non-fungal adults
Q6_1_16S_adults <- subset_samples(Q6_1_16S_decom_relative2, Treatment ==
    "C")

### Relative Abundances of Target Taxa Beijerinckiaceae Larvae
Beijerinckiaceae_Q6_1_larvae <- subset_taxa(Q6_1_16S_larvae,
    Family == "Beijerinckiaceae")
# Relative abundance
sum(sample_sums(Beijerinckiaceae_Q6_1_larvae))/nsamples(Q6_1_16S_larvae)

# Adults
Beijerinckiaceae_Q6_1_adults <- subset_taxa(Q6_1_16S_adults,
    Family == "Beijerinckiaceae")
# Relative abundance
sum(sample_sums(Beijerinckiaceae_Q6_1_adults))/nsamples(Q6_1_16S_adults)
```

```
# Subset for combined data
Beijerinckiaceae_Q6_1 <- subset_taxa(Q6_1_16S_decom_relative2,
    Family == "Beijerinckiaceae")

# Metadata Sheet
Beijerinckiaceae_Q6_1_meta <- Beijerinckiaceae_Q6_1@sam_data

# Abundance variable
Beijerinckiaceae_Q6_1_meta$Abundance <- sample_sums(Beijerinckiaceae_Q6_1)

# Dataframe
Beijerinckiaceae_Q6_1_meta_data <- as(Beijerinckiaceae_Q6_1_meta,
    Class = "data.frame")

### Final linear mixed-effects model
Q6_1_Beijerinckiaceae_treat_final <- lmer(data = Beijerinckiaceae_Q6_1_meta_data,
    Abundance ~ Treatment + (1 | Source_Pop), REML = TRUE)
car::Anova(Q6_1_Beijerinckiaceae_treat_final, test.statistic = "F")

### Repeat workflow for: Burkholderiaceae, Chitinophagaceae,
### Corynebacteriaceae, Enterobacteriaceae, Family_XI,
### Microbacteriaceae, Moraxellaceae, Pseudomonadaceae,
### Rhizobiaceae, Rhodobacteraceae, Sphingobacteriaceae,
### Sphingomonadaceae, Staphylococcaceae, Weeksellaceae
```

Transstadial Transmission (Fungal Mosquitoes)

Initial Processing and Alpha Diversity Measurements

```
### Import phyloseq object
pstree_16S_decom <- readRDS("*Decom_Data/pstree_16S_decom.rds")

### Subset samples for treatments B and D
Q6_2_16S_decom <- subset_samples(pstree_16S_decom, Experiment ==
    "exp" & Sex == "female" & Frozen == "FALSE")
Q6_2_16S_decom <- subset_samples(Q6_2_16S_decom, Treatment ==
    "B" | Treatment == "D")

### Rename values if needed
Q6_2_16S_decom@sam_data$Age <- revalue(Q6_2_16S_decom@sam_data$Age,
    replace = c(larva = "Larvae"))
Q6_2_16S_decom@sam_data$Age <- revalue(Q6_2_16S_decom@sam_data$Age,
    replace = c(adult = "Adults"))

saveRDS(Q6_2_16S_decom, "*Decom_Data/Q6_2_16S_decom.rds")

Q6_2_16S_decom <- readRDS("*Decom_Data/Q6_2_16S_decom.rds")

### Alpha diversity calculations
set.seed(42387)
meta_Q6_2_decom <- Q6_2_16S_decom@sam_data
```

```
richness_Q6_2_decom <- data.frame(estimate_richness(Q6_2_16S_decom,
    split = TRUE, measures = NULL))
richness_Q6_2_decom$Sample_ID <- rownames(richness_Q6_2_decom)

alpha_Q6_2_decom <- left_join(meta_Q6_2_decom, richness_Q6_2_decom)

write.csv(alpha_Q6_2_decom, "*Decom_Tables/alpha_Q6_2_decom.csv")

alpha_Q6_2_decom <- read.csv("*Decom_Tables/alpha_Q6_2_decom.csv")

alpha_Q6_2_decom_B <- subset(alpha_Q6_2_decom, Age == "Larvae")
alpha_Q6_2_decom_D <- subset(alpha_Q6_2_decom, Age == "Adults")


### Linear mixed models

# Simpson
Q6_2_decom_simpson_final <- lmer(data = alpha_Q6_2_decom, Simpson ~
    Treatment + (1 | Source_Pop), REML = TRUE)
Anova(Q6_2_decom_simpson_final, test.statistic = "F")

### Coefficient of Variation
cv(alpha_Q6_2_decom_B$Simpson, alpha_Q6_2_decom_D$Simpson)

### CV significance Asymptotic test
asymptotic_test(alpha_Q6_2_decom$Simpson, alpha_Q6_2_decom$Treatment)

### M-SLRT Test
mslr_test(nr = 10000, alpha_Q6_2_decom$Simpson, alpha_Q6_2_decom$Treatment)

# Shannon
Q6_2_decom_shannon_final <- lmer(data = alpha_Q6_2_decom, Shannon ~
    Treatment + (1 | Source_Pop), REML = TRUE)
Anova(Q6_2_decom_shannon_final, test.statistic = "F")

### Coefficient of Variation
cv(alpha_Q6_2_decom_B$Shannon, alpha_Q6_2_decom_D$Shannon)

### CV significance Asymptotic test
asymptotic_test(alpha_Q6_2_decom$Shannon, alpha_Q6_2_decom$Treatment)

### M-SLRT Test
mslr_test(nr = 10000, alpha_Q6_2_decom$Shannon, alpha_Q6_2_decom$Treatment)
```

Rarefaction, Filtering, and Relative Abundance Barplots

```
### Import phyloseq object
Q6_2_16S_decom <- readRDS("*Decom_Data/Q6_2_16S_decom.rds")
set.seed(73250)

### Make rarecurve
Q6_2_rarecurve_decom <- ggrare(Q6_2_16S_decom, step = 100, color = "Age",
    se = FALSE) + geom_vline(xintercept = 2229, linetype = "dashed") +
    xlab("Sample Read Coverage") + ylab("OTU Richness") + theme_bw()
Q6_2_rarecurve_decom$labels$colour <- "Fungal Mosquitoes"
Q6_2_rarecurve_decom <- Q6_2_rarecurve_decom + theme(text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )


### Create sample sums table
Q6_2_samsums_16S_decom <- data.frame(sample_sums(Q6_2_16S_decom))
colnames(Q6_2_samsums_16S_decom) <- "Read_Counts"
write.csv(Q6_2_samsums_16S_decom, "*Decom_Tables/Q6_2_samsums_16S_decom.csv")


### Rarefy to 2229 (10 samples cut)
Q6_2_16S_decom_rare <- rarefy_even_depth(Q6_2_16S_decom, sample.size = 2229)

### 10 Samples Removed: D3VI16S, D1i16S, D1ii16S, D4VII16S,
### D4ii16S, D4V16S, D1iii16S, D2VI16S, D4i16S, D2ii16S

### filter out singletons
singfilterfun_Q6_2 <- filterfun(kOverA(1, 1))

Q6_2_16S_decom_filter1 <- filter_taxa(Q6_2_16S_decom_rare, function(x) singfilterfun_Q6_2(x),
    TRUE)

### Convert tree to relative abundance
Q6_2_16S_decom_relative1 <- transform_sample_counts(Q6_2_16S_decom_filter1,
    function(x) x/sum(x))

### Must appear at greater than 0.3% in at least 1 occurrence
### (removes OTUs with 5 reads or less in each sample across
### all samples)
singfilterfun_Q6_2_2 <- filterfun(kOverA(1, 0.003))

Q6_2_16S_decom_relative2 <- filter_taxa(Q6_2_16S_decom_relative1,
    function(x) singfilterfun_Q6_2_2(x), TRUE)

saveRDS(Q6_2_16S_decom_relative2, "*Decom_Data/Q6_2_16S_decom_relative2.rds")

### Barplot of the relative abundances of the top 15 families
### shared across groups
Q6_2_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_2_16S_decom_relative2.rds")
BvsDrelative2new <- Q6_2_16S_decom_relative2

### Tax_glom to family
pstree_16S_Q6_2_relative2_family <- tax_glom(BvsDrelative2new,
    taxrank = "Family")

### Identify top 15 families
family_15_Q6_2 <- names(sort(taxa_sums(pstree_16S_Q6_2_relative2_family),
    TRUE)[1:15])
```

```
### Prune top 15
family_15_Q6_2_prune <- prune_taxa(family_15_Q6_2, pstree_16S_Q6_2_relative2_family)

### Make table
family_15_Q6_2_table <- cbind(tax_table(family_15_Q6_2_prune))

### Consolidate table formatting
family_15_Q6_2_table[family_15_Q6_2, "Family"] <- as(tax_table(family_15_Q6_2_prune)[family_15_Q6_2,
    "Family"], "character")

tax_table(family_15_Q6_2_prune) <- family_15_Q6_2_table


### Transform to percentages
family_15_Q6_2_merge <- merge_samples(family_15_Q6_2_prune, "Age")
family_15_Q6_2_percent <- transform_sample_counts(family_15_Q6_2_merge,
    function(x) 100 * x/sum(x))


### Plot relative abundances of top 15 families for C
### (family_15_Q3_plot)
BvsD_barplot_final <- plot_bar(family_15_Q6_2_percent, fill = "Family") +
    xlab("Fungal Mosquitoes") + ylab("Relative Abundance") +
    scale_fill_manual(values = c("#FFCC00", "#FF3300", "#FF0066",
        "#FF0099", "#CC00CC", "#CC00FF", "#9900FF", "#6600FF",
        "#0000FF", "#3399CC", "#0099FF", "#00FFFF", "#33FFCC",
        "#00FF99", "#66FF00")) + theme_bw()
BvsD_barplot_final
```

Beta Diversity Measures

```
set.seed(84980)
Q6_2_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_2_16S_decom_relative2.rds")

### Calculate Bray Distance
df_bray_Q6_2 <- as(sample_data(Q6_2_16S_decom_relative2), "data.frame")
d_bray_Q6_2 <- distance(Q6_2_16S_decom_relative2, "bray")

### Nested PERMANOVA
nested.npmanova(d_bray_Q6_2 ~ Treatment + Source_Pop, data = df_bray_Q6_2,
    method = "bray", permutations = 999)

# Betadisper
Q6_2_betadisper1 <- betadisper(d_bray_Q6_2, df_bray_Q6_2$Treatment)
permutest(Q6_2_betadisper1, permutations = 999)


### Calculate Unweighted UniFrac Distance
df_unifrac_Q6_2 <- as(sample_data(Q6_2_16S_decom_relative2),
    "data.frame")
d_unifrac_Q6_2 <- distance(Q6_2_16S_decom_relative2, "unifrac")
```

```
### Nested PERMANOVA
nested.npmanova(d_unifrac_Q6_2 ~ Treatment + Source_Pop, data = df_unifrac_Q6_2,
    method = "unifrac", permutations = 999)

# Betadisper
Q6_2_betadisper2 <- betadisper(d_unifrac_Q6_2, df_unifrac_Q6_2$Treatment)
permutest(Q6_2_betadisper2, permutations = 999)


### Calculate Weighted UniFrac Distance
df_wunifrac_Q6_2 <- as(sample_data(Q6_2_16S_decom_relative2),
    "data.frame")
d_wunifrac_Q6_2 <- distance(Q6_2_16S_decom_relative2, "wunifrac")

### Nested PERMANOVA
nested.npmanova(d_wunifrac_Q6_2 ~ Treatment + Source_Pop, method = "wunifrac",
    data = df_wunifrac_Q6_2, permutations = 999)

# Betadisper
Q6_2_betadisper3 <- betadisper(d_wunifrac_Q6_2, df_wunifrac_Q6_2$Treatment)
permutest(Q6_2_betadisper3, permutations = 999)


### Calculate Jaccard index
df_jaccard_Q6_2 <- as(sample_data(Q6_2_16S_decom_relative2),
    "data.frame")
d_jaccard_Q6_2 <- distance(Q6_2_16S_decom_relative2, "jaccard")

### Nested PERMANOVA
nested.npmanova(d_jaccard_Q6_2 ~ Treatment + Source_Pop, method = "jaccard",
    data = df_jaccard_Q6_2, permutations = 999)

# Betadisper
Q6_2_betadisper4 <- betadisper(d_jaccard_Q6_2, df_jaccard_Q6_2$Treatment)
permutest(Q6_2_betadisper4, permutations = 999)
```

Linear mixed models for relative abundances of the top 15 shared families

```
### LMERS on top 15 families shared between groups
Q6_2_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_2_16S_decom_relative2.rds")

### Subset for fungal larvae
Q6_2_16S_larvae <- subset_samples(Q6_2_16S_decom_relative2, Treatment ==
    "B")

### Subset for fungal adults
Q6_2_16S_adults <- subset_samples(Q6_2_16S_decom_relative2, Treatment ==
    "D")


### Relative Abundances of Target Taxa
### Bacillaceae Larvae
Bacillaceae_Q6_2_larvae <- subset_taxa(Q6_2_16S_larvae, Family ==
    "Bacillaceae")
# Relative abundance
sum(sample_sums(Bacillaceae_Q6_2_larvae))/nsamples(Q6_2_16S_larvae)

# Adults
Bacillaceae_Q6_2_adults <- subset_taxa(Q6_2_16S_adults, Family ==
    "Bacillaceae")
# Relative abundance
sum(sample_sums(Bacillaceae_Q6_2_adults))/nsamples(Q6_2_16S_adults)

# Subset for combined data
Bacillaceae_Q6_2 <- subset_taxa(Q6_2_16S_decom_relative2, Family ==
    "Bacillaceae")

# Metadata Sheet
Bacillaceae_Q6_2_meta <- Bacillaceae_Q6_2@sam_data

# Abundance variable
Bacillaceae_Q6_2_meta$Abundance <- sample_sums(Bacillaceae_Q6_2)

# Dataframe
Bacillaceae_Q6_2_meta_data <- as(Bacillaceae_Q6_2_meta, Class = "data.frame")

### Final linear mixed-effects model
Q6_2_Bacillaceae_treat_final <- lmer(data = Bacillaceae_Q6_2_meta_data,
    Abundance ~ Treatment + (1 | Source_Pop), REML = TRUE)
car::Anova(Q6_2_Bacillaceae_treat_final, test.statistic = "F")

### Repeat workflow for: Burkholderiaceae, Corynebacteriaceae,
### Enterobacteriaceae, Family_XI, Flavobacteriaceae,
### Microbacteriaceae, Micrococcaceae, Moraxellaceae,
### Pseudomonadaceae, Rhizobiaceae, Sphingobacteriaceae,
### Sphingomonadaceae, Staphylococcaceae, Weeksellaceae
```

Adult Gut Microbiome Comparisons

```
### Import phyloseq object
pstree_16S_decom <- readRDS("*Decom_Data/pstree_16S_decom.rds")

### Subset samples for treatments C and D
Q3_16S_decom <- subset_samples(pstree_16S_decom, Target == "16S" &
    Experiment == "exp" & Frozen == "FALSE")
Q3_16S_decom <- subset_samples(Q3_16S_decom, Treatment == "C" |
    Treatment == "D")

### Rename values if needed
as.character(Q3_16S_decom@sam_data$Treatment)
Q3_16S_decom@sam_data$Treatment <- revalue(Q3_16S_decom@sam_data$Treatment,
    c(C = "Non-Fungal"))
```

```
Q3_16S_decom@sam_data$Treatment <- revalue(Q3_16S_decom@sam_data$Treatment,
    c(D = "Fungal"))

saveRDS(Q3_16S_decom, "*Decom_Data/Q3_16S_decom.rds")
Q3_16S_decom <- readRDS("*Decom_Data/Q3_16S_decom.rds")

### Alpha diversity calculations
set.seed(42387)
meta_Q3_decom <- Q3_16S_decom@sam_data
richness_Q3_decom <- data.frame(estimate_richness(Q3_16S_decom,
    split = TRUE, measures = NULL))
richness_Q3_decom$Sample_ID <- rownames(richness_Q3_decom)

alpha_Q3_decom <- left_join(meta_Q3_decom, richness_Q3_decom)

write.csv(alpha_Q3_decom, "*Decom_Tables/alpha_Q3_decom.csv")

alpha_Q3_decom <- read.csv("*Decom_Tables//alpha_Q3_decom.csv")

alpha_Q3_decom_C <- subset(alpha_Q3_decom, Treatment == "Non-Fungal")
alpha_Q3_decom_D <- subset(alpha_Q3_decom, Treatment == "Fungal")


### Linear mixed models

# Simpson
Q3_decom_simpson_final <- lmer(data = alpha_Q3_decom, Simpson ~
    Treatment + (1 | Source_Pop), REML = TRUE)
Anova(Q3_decom_simpson_final, test.statistic = "F")

### Coefficient of Variation
cv(alpha_Q3_decom_C$Simpson, alpha_Q3_decom_D$Simpson)

### CV significance Asymptotic test
asymptotic_test(alpha_Q3_decom$Simpson, alpha_Q3_decom$Treatment)

### M-SLRT Test
mslr_test(nr = 10000, alpha_Q3_decom$Simpson, alpha_Q3_decom$Treatment)


# Shannon
Q3_decom_shannon_final <- lmer(data = alpha_Q3_decom, Shannon ~
    Treatment + (1 | Source_Pop), REML = TRUE)
Anova(Q3_decom_shannon_final, test.statistic = "F")

### Coefficient of Variation
cv(alpha_Q3_decom_C$Shannon, alpha_Q3_decom_D$Shannon)

### CV significance Asymptotic test
asymptotic_test(alpha_Q3_decom$Shannon, alpha_Q3_decom$Treatment)

### M-SLRT Test
mslr_test(nr = 10000, alpha_Q3_decom$Shannon, alpha_Q3_decom$Treatment)
```

Rarefaction, Filtering, and Relative Abundance Barplots

Beta Diversity Measures

Linear mixed models for relative abundances of the top 15 shared families

```r
### LMERS on top 15 families shared between groups
Q3_16S_decom_relative2 <- readRDS("*Decom_Data/Q3_16S_decom_relative2.rds")

### Subset for non-fungal adults
Q3_16S_nonfungal <- subset_samples(Q3_16S_decom_relative2, Treatment ==
    "Non-Fungal")

### Subset for fungal adults
Q3_16S_fungal <- subset_samples(Q3_16S_decom_relative2, Treatment ==
    "Fungal")


### Relative Abundances of Target Taxa

### Burkholderiaceae Non-Fungal
Burkholderiaceae_Q3_nonfungal <- subset_taxa(Q3_16S_nonfungal,
    Family == "Burkholderiaceae")
# Relative abundance
sum(sample_sums(Burkholderiaceae_Q3_nonfungal))/nsamples(Q3_16S_nonfungal)

# Fungal
Burkholderiaceae_Q3_fungal <- subset_taxa(Q3_16S_fungal, Family ==
    "Burkholderiaceae")
# Relative abundance
sum(sample_sums(Burkholderiaceae_Q3_fungal))/nsamples(Q3_16S_fungal)

# Subset for combined data
Burkholderiaceae_Q3 <- subset_taxa(Q3_16S_decom_relative2, Family ==
    "Burkholderiaceae")

# Metadata Sheet
Burkholderiaceae_Q3_meta <- Burkholderiaceae_Q3@sam_data

# Abundance variable
Burkholderiaceae_Q3_meta$Abundance <- sample_sums(Burkholderiaceae_Q3)

# Dataframe
Burkholderiaceae_Q3_meta_data <- as(Burkholderiaceae_Q3_meta,
    Class = "data.frame")

### Final linear mixed-effects model
Q3_Burkholderiaceae_treat_final <- lmer(data = Burkholderiaceae_Q3_meta_data,
    Abundance ~ Treatment + (1 | Source_Pop), REML = TRUE)
car::Anova(Q3_Burkholderiaceae_treat_final, test.statistic = "F")

### Repeat workflow for: Corynebacteriaceae,
### Enterobacteriaceae, Family_XI, Geodermatophilaceae,
### Lactobacillaceae, Microbacteriaceae, Micrococcaceae,
### Moraxellaceae, Pseudomonadaceae, Rhizobiaceae,
### Rhodobacteraceae, Sphingomonadaceae, Staphylococcaceae,
### Streptococcaceae
```

Positive Controls (Eggs and Food)

```
### Import phyloseq object
pstree_16S_decom <- readRDS("*Decom_Data/pstree_16S_decom.rds")

### Subset for ctrl_pos
Q8_16S_decom <- subset_samples(pstree_16S_decom, Experiment ==
    "ctrl_pos")
Q8_16S_decom <- subset_samples(Q8_16S_decom, Type != "rear_water")
meta <- Q8_16S_decom@sam_data

saveRDS(Q8_16S_decom, "*Decom_Data/Q8_16S_decom.rds")


### Make rarecurve
Q8_16S_decom <- readRDS("*Decom_Data/Q8_16S_decom.rds")
Q8_16S_decom@sam_data$Type <- revalue(Q8_16S_decom@sam_data$Type,
    replace = c(eggs = "Eggs"))
Q8_16S_decom@sam_data$Type <- revalue(Q8_16S_decom@sam_data$Type,
    replace = c(food = "Food"))

Q8_rarecurve_decom <- ggrare(Q8_16S_decom, step = 100, color = "Type",
    se = FALSE) + geom_vline(xintercept = 7104, linetype = "dashed") +
    theme_bw()
Q8_rarecurve_decom <- Q8_rarecurve_decom + theme(text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )
Q8_rarecurve_decom


### Create sample sums table
Q8_samsums_16S_decom <- data.frame(sample_sums(Q8_16S_decom))
colnames(Q8_samsums_16S_decom) <- "Read_Counts"
write.csv(Q8_samsums_16S_decom, "*Decom_Tables/Q8_samsums_16S_decom.csv")


.  ###Rarefy to 7104 (0 samples cut)
set.seed(73250)
Q8_16S_decom_rare <- rarefy_even_depth(Q8_16S_decom, sample.size = 7104)

### filter out singletons
singfilterfun_Q8 <- filterfun(kOverA(1, 1))

Q8_16S_decom_filter1 <- filter_taxa(Q8_16S_decom_rare, function(x) singfilterfun_Q8(x),
    TRUE)
```

```
Q8_16S_decom_filter1


### Convert tree to relative abundance
Q8_16S_decom_relative1 <- transform_sample_counts(Q8_16S_decom_filter1,
    function(x) x/sum(x))

### Must appear at greater than 0.1% in at least 1 occurrence
### (removes OTUs with less than 7 reads in each sample)
singfilterfun_Q8 <- filterfun(kOverA(1, 0.001))

Q8_16S_decom_relative2 <- filter_taxa(Q8_16S_decom_relative1,
    function(x) singfilterfun_Q8(x), TRUE)
Q8_16S_decom_relative2



saveRDS(Q8_16S_decom_relative2, "*Decom_Data/Q8_16S_decom_relative2.rds")
Q8_16S_decom_relative2 <- readRDS("*Decom_Data/Q8_16S_decom_relative2.rds")
```

SCML Read Count Calibration

```
transstadial_spike_decom <- readRDS("*Decom_Data/pstree_16S_decom.rds")

### Part 1 (Non-Fungal Mosquitoes)
AvC_spike <- subset_samples(transstadial_spike_decom, Experiment ==
    "exp_spike" & Spike_Amount == "a")
AvC_spike <- subset_samples(AvC_spike, Treatment == "A" | Treatment ==
    "C")
metaAvsC <- AvC_spike@sam_data

### S. ruber Read Counts

# Tax glom to genus
AvsC_spike_genus <- tax_glom(AvC_spike, taxrank = "Genus")

# Create dataframe
AvsC_spike_genus_data <- psmelt(AvsC_spike_genus)
AvsC_spike_genus_data$Genus <- as.character(AvsC_spike_genus_data$Genus)

# Aggregate
AvsC_spike_genus_counts <- aggregate(Abundance ~ Sample + Genus,
    AvsC_spike_genus_data, FUN = sum)

# Reorganize Genus columns
AvsC_spike_genus_counts_final <- cast(AvsC_spike_genus_counts,
    Sample ~ Genus)
colnames(AvsC_spike_genus_counts_final)[1] <- "Sample_ID"

# Individual Salinibacter counts/sample
AvsC_spike_salinibacter <- AvsC_spike_genus_counts_final$Salinibacter
```

```
# Conversion Factor Calculation and Total Read Calibration

metaAvsC <- AvC_spike@sam_data

# Create new dataframe
AvsC_samnames <- AvsC_spike_genus_counts_final$Sample_ID
AvsC_countdata <- data_frame(AvsC_samnames, AvsC_spike_salinibacter)
AvsC_countdata$AvsC_samnames <- metaAvsC$Sample_ID
colnames(AvsC_countdata)[1] <- "Sample_ID"

# Michael Wohjahn loop to calculate conversion factor
x_AvsC <- AvsC_spike_salinibacter

AvsC_conv_func <- function(x_a) {
    object_AvsC <<- c()
    for (i in 1:length(x_AvsC)) {
        out_AvsC <- (mean(x_AvsC)/x_AvsC[i])
        object_AvsC <<- rbind(object_AvsC, out_AvsC)
    }
}

AvsC_conv_func(AvsC_spike_salinibacter)
object_AvsC

# Create conversion factor variable
AvsC_countdata$Conv_Factor <- object_AvsC

# Join datasheets
metaAvsC_2 <- left_join(metaAvsC, AvsC_countdata)
metaAvsC_2$Conv_Factor <- as.vector(metaAvsC_2$Conv_Factor)

# Total Population and Calibrated Counts
metaAvsC_2$Total_Counts <- sample_sums(AvC_spike)
metaAvsC_2 <- mutate(metaAvsC_2, Calibrated_Counts = Total_Counts *
    Conv_Factor)

### Final Lmer
AvsC_spike_treat_final <- lmer(data = metaAvsC_2, Calibrated_Counts ~
    Treatment + (1 | Source_Pop), REML = TRUE)
AvsC_spike_treat_final
Anova(AvsC_spike_treat_final, test.statistic = "F")


### Part 2 (Fungal Mosquitoes)

# Phyloseq object
transstadial_spike_decom <- readRDS("*Decom_Data/pstree_16S_decom.rds")
meta <- transstadial_spike_decom@sam_data

BvD_spike <- subset_samples(transstadial_spike_decom, Experiment ==
    "exp_spike" & Spike_Amount == "a")
BvD_spike <- subset_samples(BvD_spike, Treatment == "B" | Treatment ==
    "D")
```

```
metaBvsD <- BvD_spike@sam_data

### S. ruber Read Counts

# Tax glom to genus
BvsD_spike_genus <- tax_glom(BvD_spike, taxrank = "Genus")

# Create dataframe
BvsD_spike_genus_data <- psmelt(BvsD_spike_genus)
BvsD_spike_genus_data$Genus <- as.character(BvsD_spike_genus_data$Genus)

# Aggregate
BvsD_spike_genus_counts <- aggregate(Abundance ~ Sample + Genus,
    BvsD_spike_genus_data, FUN = sum)

# Reorganize Genus columns
BvsD_spike_genus_counts_final <- cast(BvsD_spike_genus_counts,
    Sample ~ Genus)
colnames(BvsD_spike_genus_counts_final)[1] <- "Sample_ID"

# Individual Salinibacter counts/sample
BvsD_spike_salinibacter <- BvsD_spike_genus_counts_final$Salinibacter


# Conversion Factor Calculation and Total Read Calibration

metaBvsD <- BvD_spike@sam_data

# Create new dataframe
BvsD_samnames <- BvsD_spike_genus_counts_final$Sample_ID
BvsD_countdata <- data_frame(BvsD_samnames, BvsD_spike_salinibacter)
BvsD_countdata$BvsD_samnames <- metaBvsD$Sample_ID
colnames(BvsD_countdata)[1] <- "Sample_ID"

# Michael Wohjahn loop to calculate conversion factor
x_BvsD <- BvsD_spike_salinibacter

BvsD_conv_func <- function(x_BvsD) {
    object_BvsD <<- c()
    for (i in 1:length(x_BvsD)) {
        out_BvsD <- (mean(x_BvsD)/x_BvsD[i])
        object_BvsD <<- rbind(object_BvsD, out_BvsD)
    }
}

BvsD_conv_func(BvsD_spike_salinibacter)
object_BvsD

# Create conversion factor variable
BvsD_countdata$Conv_Factor <- object_BvsD

# Join datasheets
metaBvsD_2 <- left_join(metaBvsD, BvsD_countdata)
```

```
metaBvsD_2$Conv_Factor <- as.vector(metaBvsD_2$Conv_Factor)

# Total Population and Calibrated Counts
metaBvsD_2$Total_Counts <- sample_sums(BvD_spike)
metaBvsD_2 <- mutate(metaBvsD_2, Calibrated_Counts = Total_Counts *
    Conv_Factor)

### Final Lmer
BvsD_spike_treat_final <- lmer(data = metaBvsD_2, Calibrated_Counts ~
    Treatment + (1 | Source_Pop), REML = TRUE)
BvsD_spike_treat_final
Anova(BvsD_spike_treat_final, test.statistic = "F")
```

Paper Figures

Figure 2

```
### 2A (Simpson Diversity)
alpha_Q6_decom <- read.csv("*Decom_Tables/alpha_Q6_decom.csv")
Transstadial_alphaplot_simpson_decom <- ggplot(alpha_Q6_decom,
    aes(Age, Simpson, color = Treatment)) + geom_boxplot() +
    scale_y_continuous(limits = c(0, 1.15), breaks = c(0, 0.25,
        0.5, 0.75, 1)) + scale_x_discrete(limits = c("Larvae",
    "Adults")) + scale_color_manual(values = c("#FF0000", "#0000FF")) +
    geom_jitter(position = position_jitterdodge(jitter.width = 0)) +
    geom_segment(aes(x = 1.2, xend = 2.2, y = 1.05, yend = 1.05),
        color = "black") + annotate(geom = "text", x = 1.7, y = 1.09,
    label = "CV ***", color = "black", size = 7) + geom_segment(aes(x = 1.8,
    xend = 2.2, y = 1.1, yend = 1.1), color = "black") + annotate(geom = "text",
    x = 2, y = 1.14, label = "CV ***", color = "black", size = 7) +
    ylab("Simpson Diversity Index") + xlab("Developmental Stage") +
    theme_bw()

Transstadial_alphaplot_simpson_decom + theme(text = element_text(size = 25),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )


### 2B (Shannon Diversity)
Transstadial_alphaplot_shannon_decom <- ggplot(alpha_Q6_decom,
    aes(Age, Shannon, color = Treatment)) + geom_boxplot() +
    ylim(0, 5) + scale_color_manual(values = c("#FF0000", "#0000FF")) +
    geom_jitter(position = position_jitterdodge(jitter.width = 0)) +
    stat_summary(geom = "text", label = c("a", "a*", "ab", "b*"),
        size = 6, fun.y = max, vjust = -1, position = position_jitterdodge(jitter.width = 0)) +
    ylab("Shannon") + geom_segment(aes(x = 1.2, xend = 2.2, y = 4.4,
    yend = 4.4), color = "black") + annotate(geom = "text", x = 1.7,
    y = 4.6, label = "CV ***", color = "black", size = 7) + geom_segment(aes(x = 1.8,
    xend = 2.2, y = 4.6, yend = 4.6), color = "black") + annotate(geom = "text",
    x = 2, y = 4.8, label = "CV ***", color = "black", size = 7) +
    ylab("Shannon Diversity Index") + xlab("Developmental Stage") +
    theme_bw()

Transstadial_alphaplot_shannon_decom + scale_x_discrete(limits = c("Larvae",
    "Adults")) + theme(text = element_text(size = 25), axis.title.y = element_text(margin = margin(r = 
    axis.title.x = element_text(margin = margin(t = 15)), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = )
```

Figure 3

```
### 3A (Larvae Unweighted UniFrac NMDS)
set.seed(84980)

Q1_16S_decom_relative2 <- readRDS("*Decom_Data/Q1_16S_decom_relative2.rds")
Q1_uni_nmds <- ordinate(Q1_16S_decom_relative2, "NMDS", "unifrac")
Q1_uni_nmds

Q1_uniplot_nmds <- plot_ordination(Q1_16S_decom_relative2, Q1_uni_nmds,
    type = "samples", color = "Treatment") + scale_color_manual(values = c("#0000FF",
    "#FF0000")) + theme_bw()

Q1_uniplot_nmds_final <- Q1_uniplot_nmds + theme(text = element_text(size = 25),
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )
Q1_uniplot_nmds_final$labels$colour <- "Larva Type"
Q1_uniplot_nmds_final


### 3B (Larvae Unweighted UniFrac Betadisper)
df_unifrac_Q1 <- as(sample_data(Q1_16S_decom_relative2), "data.frame")
d_unifrac_Q1 <- distance(Q1_16S_decom_relative2, "unifrac")
Q1_betadisper2 <- betadisper(d_unifrac_Q1, df_unifrac_Q1$Treatment)
Q1_betadisper2_data <- data.frame(Q1_betadisper2$group, Q1_betadisper2$distances)
colnames(Q1_betadisper2_data) <- c("Treatment", "Distance")

Q1_betadisper_uni <- ggplot(Q1_betadisper2_data, aes(Treatment,
    Distance, color = Treatment)) + geom_boxplot() + ylab("Distance from Centroid (Unweighted UniFrac)"
    xlab("Treatment") + scale_y_continuous(limits = c(0.1, 0.55),
    breaks = c(0.1, 0.2, 0.3, 0.4, 0.5)) + scale_color_manual(values = c("#0000FF",
    "#FF0000")) + geom_jitter(position = position_jitterdodge(jitter.width = 0)) +
    geom_segment(aes(x = 1, xend = 2, y = 0.5, yend = 0.5), color = "black") +
    annotate(geom = "text", x = 1.5, y = 0.505, label = "*",
        size = 10, color = "black") + theme_bw()

Q1_betadisper_uni_final <- Q1_betadisper_uni + scale_y_continuous(limits = c(0,
    0.55), breaks = c(0, 0.1, 0.2, 0.3, 0.4, 0.5)) + theme(text = element_text(size = 25),
    axis.title.y = element_text(margin = margin(r = 15)), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = ) + guides(color = FALSE)
Q1_betadisper_uni_final
```

Figure 4

```
set.seed(84980)

### Non-Fungal Dataset
Q6_1_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_1_16S_decom_relative2.rds")
Q6_1_16S_decom_relative2@sam_data$Treatment <- revalue(Q6_1_16S_decom_relative2@sam_data$Treatment,
    c(A = "Larvae"))
Q6_1_16S_decom_relative2@sam_data$Treatment <- revalue(Q6_1_16S_decom_relative2@sam_data$Treatment,
    c(C = "Adults"))

### 4A (Non-Fungal Bray NMDS)
Q6_1_bray_nmds <- ordinate(Q6_1_16S_decom_relative2, "NMDS",
    "bray")
Q6_1_bray_nmds

Q6_1_brayplot_nmds <- plot_ordination(Q6_1_16S_decom_relative2,
    Q6_1_bray_nmds, type = "samples", color = "Treatment", shape = "Treatment") +
    scale_color_manual(values = c("#0000FF", "#0000FF")) + theme_bw()
Q6_1_brayplot_nmds

Q6_1_brayplot_nmds_final <- Q6_1_brayplot_nmds + theme(text = element_text(size = 25),
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = ) + guides(color = FALSE)
Q6_1_brayplot_nmds_final$labels$shape <- "Non-Fungal Mosquitoes"
Q6_1_brayplot_nmds_final

### 4B (Non-Fungal Unweighted UniFrac NMDS)
Q6_1_uni_nmds <- ordinate(Q6_1_16S_decom_relative2, "NMDS", "unifrac")
Q6_1_uni_nmds

Q6_1_uniplot_nmds <- plot_ordination(Q6_1_16S_decom_relative2,
    Q6_1_uni_nmds, type = "samples", color = "Treatment", shape = "Treatment") +
    scale_color_manual(values = c("#0000FF", "#0000FF")) + theme_bw()

Q6_1_uniplot_nmds_final <- Q6_1_uniplot_nmds + theme(text = element_text(size = 25),
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = ) + scale_x_continuous(limits = c(-0.5, 0.7),
    breaks = c(-0.4, -0.2, 0, 0.2, 0.4, 0.6)) + scale_y_continuous(limits = c(-0.65,
    0.35), breaks = c(-0.6, -0.4, -0.2, 0, 0.2)) + guides(color = FALSE)
Q6_1_uniplot_nmds_final$labels$shape <- "Non-Fungal Mosquitoes"
Q6_1_uniplot_nmds_final


### Fungal Dataset
Q6_2_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_2_16S_decom_relative2.rds")
Q6_2_16S_decom_relative2@sam_data$Treatment <- revalue(Q6_2_16S_decom_relative2@sam_data$Treatment,
    c(B = "Larvae"))
Q6_2_16S_decom_relative2@sam_data$Treatment <- revalue(Q6_2_16S_decom_relative2@sam_data$Treatment,
    c(D = "Adults"))

### 4C (Fungal Bray NMDS)
Q6_2_bray_nmds <- ordinate(Q6_2_16S_decom_relative2, "NMDS",
    "bray")
Q6_2_bray_nmds

Q6_2_brayplot_nmds <- plot_ordination(Q6_2_16S_decom_relative2,
    Q6_2_bray_nmds, type = "samples", color = "Treatment", shape = "Treatment") +
    scale_color_manual(values = c("#FF0000", "#FF0000")) + theme_bw()

Q6_2_brayplot_nmds_final <- Q6_2_brayplot_nmds + scale_x_continuous(limits = c(-1.5,
    3.25), breaks = c(-1, 0, 1, 2, 3)) + theme(text = element_text(size = 25),
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = ) + guides(color = FALSE) + scale_y_continuous(limits = c(-4,
    2.5), breaks = c(-4, -2, 0, 2))
Q6_2_brayplot_nmds_final$labels$shape <- "Fungal Mosquitoes"
Q6_2_brayplot_nmds_final

### 4D (Fungal Unweighted UniFrac NMDS)
Q6_2_uni_nmds <- ordinate(Q6_2_16S_decom_relative2, "NMDS", "unifrac")
Q6_2_uni_nmds

Q6_2_uniplot_nmds <- plot_ordination(Q6_2_16S_decom_relative2,
    Q6_2_uni_nmds, type = "samples", color = "Treatment", shape = "Treatment") +
    scale_color_manual(values = c("#FF0000", "#FF0000")) + theme_bw()

Q6_2_uniplot_nmds_final <- Q6_2_uniplot_nmds + theme(text = element_text(size = 25),
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = ) + scale_x_continuous(limits = c(-0.5, 0.7),
    breaks = c(-0.4, -0.2, 0, 0.2, 0.4, 0.6)) + scale_y_continuous(limits = c(-0.65,
    0.35), breaks = c(-0.6, -0.4, -0.2, 0, 0.2)) + guides(color = FALSE)
Q6_2_uniplot_nmds_final$labels$shape <- "Fungal Mosquitoes"
Q6_2_uniplot_nmds_final
```

Figure 5

```
### 5A (Non-Fungal Barplot)
Q6_1_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_1_16S_decom_relative2.rds")
AvsCrelative2new <- Q6_1_16S_decom_relative2

### Tax_glom to family
pstree_16S_Q6_1_relative2_family <- tax_glom(AvsCrelative2new,
    taxrank = "Family")

### Identify top 15 families
family_15_Q6_1 <- names(sort(taxa_sums(pstree_16S_Q6_1_relative2_family),
    TRUE)[1:15])

### Prune top 15
family_15_Q6_1_prune <- prune_taxa(family_15_Q6_1, pstree_16S_Q6_1_relative2_family)

sum(sample_sums(family_15_Q6_1_prune))/sum(sample_sums(pstree_16S_Q6_1_relative2_family))
0.9095544

### Make table
```

```
family_15_Q6_1_table <- cbind(tax_table(family_15_Q6_1_prune))

### Consolidate table formatting
family_15_Q6_1_table[family_15_Q6_1, "Family"] <- as(tax_table(family_15_Q6_1_prune)[family_15_Q6_1,
    "Family"], "character")

tax_table(family_15_Q6_1_prune) <- family_15_Q6_1_table

### Transform to percentages
family_15_Q6_1_merge <- merge_samples(family_15_Q6_1_prune, "Age")
family_15_Q6_1_percent <- transform_sample_counts(family_15_Q6_1_merge,
    function(x) 100 * x/sum(x))

### Plot relative abundances of top 15 families for C
### (family_15_Q3_plot)
AvsC_barplot_final <- plot_bar(family_15_Q6_1_percent, fill = "Family") +
    xlab("Non-Fungal Mosquitoes") + ylab("Relative Abundance (%)") +
    scale_fill_manual(values = c("#FF6600", "#FF3300", "#FF0033",
        "#FF0066", "#FF0099", "#CC00CC", "#9900FF", "#6600FF",
        "#3399CC", "#0099FF", "#33CCFF", "#00FFFF", "#33FFCC",
        "#00FF99", "#66FF00")) + theme_bw()
AvsC_barplot_final <- AvsC_barplot_final + theme(text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )
AvsC_barplot_final <- AvsC_barplot_final + scale_x_discrete(limits = c("Larvae",
    "Adults"))
AvsC_barplot_final

### 5B (Fungal Barplot)
Q6_2_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_2_16S_decom_relative2.rds")
BvsDrelative2new <- Q6_2_16S_decom_relative2

### Tax_glom to family
pstree_16S_Q6_2_relative2_family <- tax_glom(BvsDrelative2new,
    taxrank = "Family")

### Identify top 15 families
family_15_Q6_2 <- names(sort(taxa_sums(pstree_16S_Q6_2_relative2_family),
    TRUE)[1:15])

### Prune top 15
family_15_Q6_2_prune <- prune_taxa(family_15_Q6_2, pstree_16S_Q6_2_relative2_family)

sum(sample_sums(family_15_Q6_2_prune))/sum(sample_sums(pstree_16S_Q6_2_relative2_family))
0.811414

### Make table
family_15_Q6_2_table <- cbind(tax_table(family_15_Q6_2_prune))

### Consolidate table formatting
family_15_Q6_2_table[family_15_Q6_2, "Family"] <- as(tax_table(family_15_Q6_2_prune)[family_15_Q6_2,
    "Family"], "character")
```

```
tax_table(family_15_Q6_2_prune) <- family_15_Q6_2_table

### Transform to percentages
family_15_Q6_2_merge <- merge_samples(family_15_Q6_2_prune, "Age")
family_15_Q6_2_percent <- transform_sample_counts(family_15_Q6_2_merge,
    function(x) 100 * x/sum(x))


### Plot relative abundances of top 15 families for C
### (family_15_Q3_plot)
BvsD_barplot_final <- plot_bar(family_15_Q6_2_percent, fill = "Family") +
    xlab("Fungal Mosquitoes") + ylab("Relative Abundance (%)") +
    scale_fill_manual(values = c("#FFCC00", "#FF3300", "#FF0066",
        "#FF0099", "#CC00CC", "#CC00FF", "#9900FF", "#6600FF",
        "#0000FF", "#3399CC", "#0099FF", "#00FFFF", "#33FFCC",
        "#00FF99", "#66FF00")) + theme_bw()
BvsD_barplot_final <- BvsD_barplot_final + theme(text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )
BvsD_barplot_final <- BvsD_barplot_final + scale_x_discrete(limits = c("Larvae",
    "Adults"))
BvsD_barplot_final
```

Figure 6

```
Genera_master_top <- read.csv("*Decom_Tables/Genera_master_top.csv")

Genera_master_top$Genus <- revalue(Genera_master_top$Genus, c(`Allorhizobium-Neorhizobium-Pararhizobium

genera_logfoldchange <- ggplot(Genera_master_top, aes(Genus,
    Log2_Foldchange, fill = Treatment)) + geom_bar(stat = "identity",
    position = position_dodge2(width = 0.5, preserve = "single")) +
    scale_fill_manual(values = c("#FF0000", "#0000FF")) + scale_x_discrete(limits = c("Acidovorax",
    "Pelomonas", "Cupriavidus", "Delftia", "Aquabacterium", "Corynebacterium",
    "Corynebacterium_1", "Finegoldia", "Peptoniphilus", "Anaerococcus",
    "Agromyces", "Pseudoclavibacter", "Frigoribacterium", "Parafrigoribacterium",
    "Curtobacterium", "Enhydrobacter", "Pseudochrobactrum", "Allor-Neor-Parar-Rhizobium",
    "Aureimonas", "Nosocomiicoccus", "Salinicoccus")) + xlab("") +
    ylab("Log2 Fold Change") + theme_bw()
genera_logfoldchange <- genera_logfoldchange + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, vjust = 0.5), text = element_text(size = 22),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    plot.margin = unit(c(0.5, 0.5, 2, 1), "cm")) + guides(fill = FALSE)
```

Figure 7

```
Genera_masterplotdata_top <- read.csv("*Decom_Tables/Genera_masterplotdata_top.csv")
Larvae_generaabundance <- subset(Genera_masterplotdata_top, Developmental_Stage ==
    "Larvae")
```

```
adults_generaabundance <- subset(Genera_masterplotdata_top, Developmental_Stage ==
    "Adults")

### 7A (Larvae genera)
larvae_genera_barplot <- ggplot(Larvae_generaabundance, aes(Genus,
    Abundance, fill = Treatment)) + geom_bar(stat = "identity",
    position = position_dodge2(width = 0.5, preserve = "single")) +
    scale_x_discrete(limits = c("Acidovorax", "Pelomonas", "Cupriavidus",
        "Delftia", "Aquabacterium", "Corynebacterium", "Corynebacterium_1",
        "Finegoldia", "Peptoniphilus", "Anaerococcus", "Agromyces",
        "Pseudoclavibacter", "Frigoribacterium", "Parafrigoribacterium",
        "Curtobacterium", "Enhydrobacter", "Pseudochrobactrum",
        "Allor-Neor-Parar-Rhizobium", "Aureimonas", "Nosocomiicoccus",
        "Salinicoccus")) + scale_fill_manual(values = c("#FF0000",
    "#0000FF")) + xlab("") + ylab("Relative Abundance (%)") +
    guides(fill = FALSE) + theme_bw()
larvae_genera_barplot + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, vjust = 0.5), text = element_text(size = 22),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    plot.margin = unit(c(0.5, 0.5, 2, 1), "cm"))

### 7B (Adult genera)
adults_genera_barplot <- ggplot(adults_generaabundance, aes(Genus,
    Abundance, fill = Treatment)) + geom_bar(stat = "identity",
    position = position_dodge2(width = 0.5, preserve = "single")) +
    scale_x_discrete(limits = c("Acidovorax", "Pelomonas", "Cupriavidus",
        "Delftia", "Aquabacterium", "Corynebacterium", "Corynebacterium_1",
        "Finegoldia", "Peptoniphilus", "Anaerococcus", "Agromyces",
        "Pseudoclavibacter", "Frigoribacterium", "Parafrigoribacterium",
        "Curtobacterium", "Enhydrobacter", "Pseudochrobactrum",
        "Allor-Neor-Parar-Rhizobium", "Aureimonas", "Nosocomiicoccus",
        "Salinicoccus")) + scale_fill_manual(values = c("#FF0000",
    "#0000FF")) + xlab("") + ylab("Relative Abundance (%)") +
    guides(fill = FALSE) + theme_bw()
adults_genera_barplot + theme(axis.text.x = element_text(angle = 90,
    hjust = 1, vjust = 0.5), text = element_text(size = 22),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    plot.margin = unit(c(0.5, 0.5, 2, 1), "cm"))
```

Figure 8

```
### Developmental Stage SCML Read Counts
all_scml <- read.csv("*Decom_Tables/all_scml.csv")

scml_summary <- summarySE(all_scml, measurevar = "SCML.Reads",
    groupvars = c("Age", "Treatment"))

all_scmlplot <- ggplot(scml_summary, aes(Age, SCML.Reads, fill = Treatment)) +
    geom_bar(position = position_dodge(), stat = "identity",
        width = 0.5) + scale_y_continuous(limits = c(0, 13000),
    breaks = c(0, 4000, 8000, 12000)) + geom_errorbar(aes(ymin = SCML.Reads -
    se, ymax = SCML.Reads + se), width = 0.2, position = position_dodge(0.5)) +
    geom_segment(aes(x = 0.85, xend = 1.85, y = 12000, yend = 12000),
        color = "black") + annotate(geom = "text", x = 1.35,
    y = 12050, label = "*", color = "black", size = 10) + geom_segment(aes(x = 1.15,
    xend = 2.15, y = 12500, yend = 12500), color = "black") +
    annotate(geom = "text", x = 1.65, y = 12550, label = "*",
        color = "black", size = 10) + scale_fill_manual(values = c("#FF0000",
    "#0000FF")) + xlab("Developmental Stage") + ylab("SCML Calibrated Counts") +
    theme_bw()

all_scmlplot_final <- all_scmlplot + scale_x_discrete(limits = c("Larvae",
    "Adults")) + scale_y_continuous(limits = c(0, 13000), breaks = c(5000,
    10000)) + theme(text = element_text(size = 25), axis.title.y = element_text(margin = margin(r = 15)
    axis.title.x = element_text(margin = margin(t = 15)), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = )
all_scmlplot_final
```

Supplementary Figures

S1

```
### S1a (Non-Fungal Bray Betadisper)
Q6_1_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_1_16S_decom_relative2.rds")
Q6_1_16S_decom_relative2@sam_data$Treatment <- revalue(Q6_1_16S_decom_relative2@sam_data$Treatment,
    c(A = "Larvae"))
Q6_1_16S_decom_relative2@sam_data$Treatment <- revalue(Q6_1_16S_decom_relative2@sam_data$Treatment,
    c(C = "Adults"))

df_bray_Q6_1 <- as(sample_data(Q6_1_16S_decom_relative2), "data.frame")
d_bray_Q6_1 <- distance(Q6_1_16S_decom_relative2, "bray")
Q6_1_betadisper1 <- betadisper(d_bray_Q6_1, df_bray_Q6_1$Treatment)
Q6_1_betadisper1_data <- data.frame(Q6_1_betadisper1$group, Q6_1_betadisper1$distances)
colnames(Q6_1_betadisper1_data) <- c("Treatment", "Distance")

Q6_1_betadisper_bray <- ggplot(Q6_1_betadisper1_data, aes(Treatment,
    Distance, color = Treatment)) + geom_boxplot() + ylab("Distance from Centroid (Bray-Curtis)") +
    xlab("Developmental Stage") + scale_color_manual(values = c("#0000FF",
    "#0000FF")) + geom_jitter(position = position_jitterdodge(jitter.width = 0)) +
    theme_bw()

Q6_1_betadisper_bray_final <- Q6_1_betadisper_bray + xlab("Developmental Stage (Non-Fungal)") +
    scale_y_continuous(limits = c(0, 0.8), breaks = c(0, 0.1,
        0.2, 0.3, 0.4, 0.5, 0.6, 0.7)) + geom_segment(aes(x = 1,
    xend = 2, y = 0.76, yend = 0.76), color = "black") + annotate(geom = "text",
    x = 1.5, y = 0.765, label = "***", size = 10, color = "black") +
    theme(text = element_text(size = 25), axis.title.y = element_text(margin = margin(r = 15)),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.border = ) + guides(color = FALSE)
Q6_1_betadisper_bray_final


### S1B (Non-Fungal Unweighted UniFrac Betadisper)
```

```r
df_uni_Q6_1 <- as(sample_data(Q6_1_16S_decom_relative2), "data.frame")
d_uni_Q6_1 <- distance(Q6_1_16S_decom_relative2, "unifrac")
Q6_1_betadisper2 <- betadisper(d_uni_Q6_1, df_uni_Q6_1$Treatment)
Q6_1_betadisper2_data <- data.frame(Q6_1_betadisper2$group, Q6_1_betadisper2$distances)
colnames(Q6_1_betadisper2_data) <- c("Treatment", "Distance")

Q6_1_betadisper_uni <- ggplot(Q6_1_betadisper2_data, aes(Treatment,
    Distance, color = Treatment)) + geom_boxplot() + ylab("Distance from Centroid (Unweighted UniFrac)"
    xlab("Developmental Stage") + scale_color_manual(values = c("#0000FF",
    "#0000FF")) + geom_jitter(position = position_jitterdodge(jitter.width = 0)) +
    theme_bw()

Q6_1_betadisper_uni_final <- Q6_1_betadisper_uni + xlab("Developmental Stage (Non-Fungal)") +
    scale_y_continuous(limits = c(0, 0.7), breaks = c(0, 0.1,
        0.2, 0.3, 0.4, 0.5, 0.6)) + geom_segment(aes(x = 1, xend = 2,
    y = 0.65, yend = 0.65), color = "black") + annotate(geom = "text",
    x = 1.5, y = 0.655, label = "***", size = 10, color = "black") +
    theme(text = element_text(size = 25), axis.title.y = element_text(margin = margin(r = 15)),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.border = ) + guides(color = FALSE)
Q6_1_betadisper_uni_final


### S1C (Fungal Bray Betadisper)
Q6_2_16S_decom_relative2 <- readRDS("*Decom_Data/Q6_2_16S_decom_relative2.rds")
Q6_2_16S_decom_relative2@sam_data$Treatment <- revalue(Q6_2_16S_decom_relative2@sam_data$Treatment,
    c(B = "Larvae"))
Q6_2_16S_decom_relative2@sam_data$Treatment <- revalue(Q6_2_16S_decom_relative2@sam_data$Treatment,
    c(D = "Adults"))

df_bray_Q6_2 <- as(sample_data(Q6_2_16S_decom_relative2), "data.frame")
d_bray_Q6_2 <- distance(Q6_2_16S_decom_relative2, "bray")
Q6_2_betadisper1 <- betadisper(d_bray_Q6_2, df_bray_Q6_2$Treatment)
Q6_2_betadisper1_data <- data.frame(Q6_2_betadisper1$group, Q6_2_betadisper1$distances)
colnames(Q6_2_betadisper1_data) <- c("Treatment", "Distance")

Q6_2_betadisper_bray <- ggplot(Q6_2_betadisper1_data, aes(Treatment,
    Distance, color = Treatment)) + geom_boxplot() + ylab("Distance from Centroid (Bray-Curtis)") +
    xlab("Developmental Stage") + scale_color_manual(values = c("#FF0000",
    "#FF0000")) + geom_jitter(position = position_jitterdodge(jitter.width = 0)) +
    theme_bw()

Q6_2_betadisper_bray_final <- Q6_2_betadisper_bray + xlab("Developmental Stage (Fungal)") +
    scale_y_continuous(limits = c(0, 0.8), breaks = c(0, 0.1,
        0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8)) + geom_segment(aes(x = 1,
    xend = 2, y = 0.725, yend = 0.725), color = "black") + annotate(geom = "text",
    x = 1.5, y = 0.73, label = "***", size = 10, color = "black") +
    theme(text = element_text(size = 25), axis.title.y = element_text(margin = margin(r = 15)),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.border = ) + guides(color = FALSE)
Q6_2_betadisper_bray_final
```

```
### S1D (Fungal Unweighted UniFrac betadisper)
df_uni_Q6_2 <- as(sample_data(Q6_2_16S_decom_relative2), "data.frame")
d_uni_Q6_2 <- distance(Q6_2_16S_decom_relative2, "unifrac")
Q6_2_betadisper2 <- betadisper(d_uni_Q6_2, df_uni_Q6_2$Treatment)
Q6_2_betadisper2_data <- data.frame(Q6_2_betadisper2$group, Q6_2_betadisper2$distances)
colnames(Q6_2_betadisper2_data) <- c("Treatment", "Distance")

Q6_2_betadisper_uni <- ggplot(Q6_2_betadisper2_data, aes(Treatment,
    Distance, color = Treatment)) + geom_boxplot() + ylab("Distance from Centroid (Unweighted UniFrac)")
    xlab("Developmental Stage") + scale_color_manual(values = c("#FF0000",
    "#FF0000")) + geom_jitter(position = position_jitterdodge(jitter.width = 0)) +
    theme_bw()

Q6_2_betadisper_uni_final <- Q6_2_betadisper_uni + xlab("Developmental Stage (Fungal)") +
    scale_y_continuous(limits = c(0, 0.8), breaks = c(0, 0.1,
        0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8)) + geom_segment(aes(x = 1,
    xend = 2, y = 0.725, yend = 0.725), color = "black") + annotate(geom = "text",
    x = 1.5, y = 0.73, label = "***", size = 10, color = "black") +
    theme(text = element_text(size = 25), axis.title.y = element_text(margin = margin(r = 15)),
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.border = ) + guides(color = FALSE)
Q6_2_betadisper_uni_final
```

S2

```
##Non-Fungal Mosquitoes

#S2A Weighted UniFrac
Q6_1_wunifrac_nmds <- ordinate(Q6_1_16S_decom_relative2, "NMDS", "wunifrac")
Q6_1_wunifrac_nmds
2D Stress: 0.0759675

Q6_1_wuniplot_nmds <- plot_ordination(Q6_1_16S_decom_relative2, Q6_1_wunifrac_nmds, type = "samples", c
Q6_1_wuniplot_nmds_final <- Q6_1_wuniplot_nmds + geom_point(size = 4) + theme(text = element_text(size
Q6_1_wuniplot_nmds_final$labels$shape <- "Non-Fungal"

#S2B Jaccard
Q6_1_jaccard_nmds <- ordinate(Q6_1_16S_decom_relative2, "NMDS", "jaccard", binary = TRUE)
Q6_1_jaccard_nmds
2D Stress: 0.1629512

Q6_1_jaccardplot_nmds <- plot_ordination(Q6_1_16S_decom_relative2, Q6_1_jaccard_nmds, type = "samples",

Q6_1_jaccardplot_nmds_final <- Q6_1_jaccardplot_nmds + geom_point(size = 4) + theme(text = element_text
Q6_1_jaccardplot_nmds_final$labels$shape <- "Non-Fungal"
Q6_1_jaccardplot_nmds_final

###Fungal mosquitoes

#S2CWeighted UniFrac
Q6_2_wunifrac_nmds <- ordinate(Q6_2_16S_decom_relative2, "NMDS", "wunifrac")
Q6_2_wunifrac_nmds
```

```
2DStress: 0.1346431

Q6_2_wuniplot_nmds <- plot_ordination(Q6_2_16S_decom_relative2, Q6_2_wunifrac_nmds, type = "samples", co

Q6_2_wuniplot_nmds_final <- Q6_2_wuniplot_nmds + geom_point(size = 4) + theme(text = element_text(size =
Q6_2_wuniplot_nmds_final$labels$shape <- "Fungal"
Q6_2_wuniplot_nmds_final

#S2D Jaccard
Q6_2_jaccard_nmds <- ordinate(Q6_2_16S_decom_relative2, "NMDS", "jaccard", binary = TRUE)
Q6_2_jaccard_nmds
2DStress: 0.1053072

Q6_2_jaccardplot_nmds <- plot_ordination(Q6_2_16S_decom_relative2, Q6_2_jaccard_nmds, type = "samples",

Q6_2_jaccardplot_nmds_final <- Q6_2_jaccardplot_nmds + geom_point(size = 4) + theme(text = element_text
Q6_2_jaccardplot_nmds_final$labels$shape <- "Fungal"
Q6_2_jaccardplot_nmds_final
```

S3

```
### S3A (Larvae Barplot)
Q1_16S_decom_relative2 <- readRDS("*Decom_Data/Q1_16S_decom_relative2.rds")
AvsBrelative2new <- Q1_16S_decom_relative2

### CHANGE NAMES HERE Tax_glom to family
pstree_16S_Q1_relative2_family <- tax_glom(AvsBrelative2new,
    taxrank = "Family")

### Identify top 15 families
family_15_Q1 <- names(sort(taxa_sums(pstree_16S_Q1_relative2_family),
    TRUE)[1:15])

### Prune top 15
family_15_Q1_prune <- prune_taxa(family_15_Q1, pstree_16S_Q1_relative2_family)

sum(sample_sums(family_15_Q1_prune))/sum(sample_sums(pstree_16S_Q1_relative2_family))
0.9720756

### Make table
family_15_Q1_table <- cbind(tax_table(family_15_Q1_prune))

### Consolidate table formatting
family_15_Q1_table[family_15_Q1, "Family"] <- as(tax_table(family_15_Q1_prune)[family_15_Q1,
    "Family"], "character")

tax_table(family_15_Q1_prune) <- family_15_Q1_table

### Transform to percentages
family_15_Q1_merge <- merge_samples(family_15_Q1_prune, "Treatment")
family_15_Q1_percent <- transform_sample_counts(family_15_Q1_merge,
    function(x) 100 * x/sum(x))
```

```
### Plot relative abundances of top 15 families for C
### (family_15_Q3_plot)
AvsB_barplot_final <- plot_bar(family_15_Q1_percent, fill = "Family") +
    xlab("Larva Type") + ylab("Relative Abundance (%)") + scale_fill_manual(values = c("#FFCC00",
    "#FF9900", "#FF6600", "#FF3300", "#CC0033", "#FF0033", "#FF0099",
    "#660066", "#CC00FF", "#9900FF", "#3399CC", "#0099FF", "#00FFFF",
    "#33FFCC", "#66FF00")) + theme_bw()
AvsB_barplot_final <- AvsB_barplot_final + theme(text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )
AvsB_barplot_final <- AvsB_barplot_final + scale_x_discrete(limits = c("Non-Fungal",
    "Fungal"))
AvsB_barplot_final

### S3B (Adult Barplot)
Q3_16S_decom_relative2 <- readRDS("*Decom_Data/Q3_16S_decom_relative2.rds")
CvsDrelative2new <- Q3_16S_decom_relative2

### Tax_glom to family
pstree_16S_Q3_relative2_family <- tax_glom(CvsDrelative2new,
    taxrank = "Family")

### Identify top 15 families
family_15_Q3 <- names(sort(taxa_sums(pstree_16S_Q3_relative2_family),
    TRUE)[1:15])

### Prune top 15
family_15_Q3_prune <- prune_taxa(family_15_Q3, pstree_16S_Q3_relative2_family)

sum(sample_sums(family_15_Q3_prune))/sum(sample_sums(pstree_16S_Q3_relative2_family))
0.80544

### Make table
family_15_Q3_table <- cbind(tax_table(family_15_Q3_prune))

### Consolidate table formatting
family_15_Q3_table[family_15_Q3, "Family"] <- as(tax_table(family_15_Q3_prune)[family_15_Q3,
    "Family"], "character")

tax_table(family_15_Q3_prune) <- family_15_Q3_table


### Transform to percentages
family_15_Q3_merge <- merge_samples(family_15_Q3_prune, "Treatment")
family_15_Q3_percent <- transform_sample_counts(family_15_Q3_merge,
    function(x) 100 * x/sum(x))


### Plot relative abundances of top 15 families for C
### (family_15_Q3_plot)
CvsD_barplot_final <- plot_bar(family_15_Q3_percent, fill = "Family") +
    xlab("Adult Type") + ylab("Relative Abundance (%)") + scale_fill_manual(values = c("#FF3300",
    "#FF0066", "#FF0099", "#CC00CC", "#9966FF", "#6633FF", "#9900FF",
    "#6600FF", "#0000FF", "#3399CC", "#0099FF", "#33CCFF", "#33FFCC",
    "#00FF99", "#00FF66")) + theme_bw()
CvsD_barplot_final <- CvsD_barplot_final + theme(text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )
CvsD_barplot_final <- CvsD_barplot_final + scale_x_discrete(limits = c("Non-Fungal",
    "Fungal"))
CvsD_barplot_final


### S3C (Egg Barplot)
Q8_16S_decom_relative2 <- readRDS("*Decom_Data/Q8_16S_decom_relative2.rds")

### Stacked Barplots Eggs

### Tax_glom to family
Q8_16S_decom_relative2_eggs <- subset_samples(Q8_16S_decom_relative2,
    Type == "Eggs")

pstree_16S_Q8_relative2_family <- tax_glom(Q8_16S_decom_relative2_eggs,
    taxrank = "Family")

### Identify top 15 families
family_15_Q8_eggs <- names(sort(taxa_sums(pstree_16S_Q8_relative2_family),
    TRUE)[1:15])

### Prune top 15
family_15_Q8_eggs_prune <- prune_taxa(family_15_Q8_eggs, pstree_16S_Q8_relative2_family)

### Make table
family_15_Q8_eggs_prune_table <- cbind(tax_table(family_15_Q8_eggs_prune))

### Consolidate table formatting
family_15_Q8_eggs_prune_table[family_15_Q8_eggs, "Family"] <- as(tax_table(family_15_Q8_eggs_prune)[fam
    "Family"], "character")

tax_table(family_15_Q8_eggs_prune) <- family_15_Q8_eggs_prune_table

### Transform to percentages
family_15_Q8_eggs_prune_percent <- transform_sample_counts(family_15_Q8_eggs_prune,
    function(x) 100 * x/sum(x))


### Plot relative abundances of top (family_15_Q8_eggs_plot)
family_15_Q8_eggs_plot <- plot_bar(family_15_Q8_eggs_prune_percent,
    fill = "Family") + xlab("Type") + ylab("Relative Abundance (%)") +
    scale_x_discrete(labels = "Eggs") + scale_fill_manual(values = c("#CCCCCC",
    "#999999", "#FF6600", "#FF3300", "#CC0033", "#FF0033", "#CC00FF",
    "#0000FF", "#3399CC", "#0099FF", "#33CCFF", "#00FFFF", "#33FFCC",
    "#660033", "#66FF00")) + theme_bw()
family_15_Q8_eggs_plot <- family_15_Q8_eggs_plot + theme(text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )
family_15_Q8_eggs_plot


### S3D (Food Barplot)
Q8_16S_decom_relative2 <- readRDS("*Decom_Data/Q8_16S_decom_relative2.rds")
Q8_16S_decom_relative2_food <- subset_samples(Q8_16S_decom_relative2,
    Type == "Food")

### Tax_glom to family
pstree_16S_Q8_relative2_family <- tax_glom(Q8_16S_decom_relative2_food,
    taxrank = "Family")

### Identify top 15 families
family_15_Q8_food <- names(sort(taxa_sums(pstree_16S_Q8_relative2_family),
    TRUE)[1:15])

### Prune top 15
family_15_Q8_food_prune <- prune_taxa(family_15_Q8_food, Q8_16S_decom_relative2_food)

### Make table
family_15_Q8_food_prune_table <- cbind(tax_table(family_15_Q8_food_prune))

### Consolidate table formatting
family_15_Q8_food_prune_table[family_15_Q8_food, "Family"] <- as(tax_table(family_15_Q8_food_prune)[fami
    "Family"], "character")

tax_table(family_15_Q8_food_prune) <- family_15_Q8_food_prune_table

### Transform to percentages
family_15_Q8_food_prune_percent <- transform_sample_counts(family_15_Q8_food_prune,
    function(x) 100 * x/sum(x))


### Plot relative abundances of top (family_15_Q8_food_plot)
family_15_Q8_food_plot <- plot_bar(family_15_Q8_food_prune_percent,
    fill = "Family") + xlab("Type") + ylab("Relative Abundance (%)") +
    scale_fill_manual(values = c("#FFCC00", "#FF0066", "#CC00CC",
        "#6633FF", "#666666", "#6600FF", "#3300FF", "#0000FF",
        "#3399FF", "#0066CC", "#FF3366", "#FF6699", "#00FF99",
        "#00FF66", "#00FF33")) + scale_x_discrete(labels = c("1",
    "2", "3")) + xlab("Days After Food Prepared") + theme_bw()
family_15_Q8_food_plot <- family_15_Q8_food_plot + theme(text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )
family_15_Q8_food_plot
```

S4

```
### S4A (Larva Type Rarecurve)
Q1_16S_decom <- readRDS("*Decom_Data/Q1_16S_decom.rds")
set.seed(73250)

### Make rarecurve
Q1_rarecurve_decom <- ggrare(Q1_16S_decom, step = 100, color = "Treatment",
    se = FALSE) + scale_color_manual(values = c("#0000FF", "#FF0000")) +
    geom_vline(xintercept = 5251, linetype = "dashed") + xlab("Sample Read Coverage") +
    ylab("OTU Richness") + theme_bw()
Q1_rarecurve_decom$labels$colour <- "Larva Type"
Q1_rarecurve_decom <- Q1_rarecurve_decom + theme(text = element_text(size = 25),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )


### S4B (Adult Type Rarecurve)
Q3_16S_decom <- readRDS("*Decom_Data/Q3_16S_decom.rds")

### Make rarecurve
Q3_rarecurve_decom <- ggrare(Q3_16S_decom, step = 100, color = "Treatment",
    se = FALSE) + scale_color_manual(values = c("#0000FF", "#FF0000")) +
    geom_vline(xintercept = 1777, linetype = "dashed") + xlab("Sample Read Coverage") +
    ylab("OTU Richness") + theme_bw()
Q3_rarecurve_decom$labels$colour <- "Adult Type"
Q3_rarecurve_decom <- Q3_rarecurve_decom + theme(text = element_text(size = 20),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )


### S4C (Non-Fungal Mosquitoes Rarecurve)
Q6_1_16S_decom <- readRDS("*Decom_Data/Q6_1_16S_decom.rds")

### Make rarecurve
Q6_1_rarecurve_decom <- ggrare(Q6_1_16S_decom, step = 100, color = "Age",
    se = FALSE) + geom_vline(xintercept = 2560, linetype = "dashed") +
    xlab("Sample Read Coverage") + ylab("OTU Richness") + theme_bw()
Q6_1_rarecurve_decom$labels$colour <- "Non-Fungal"
Q6_1_rarecurve_decom <- Q6_1_rarecurve_decom + scale_color_manual(values = c("#0000FF",
    "#9999FF")) + theme(text = element_text(size = 25), axis.title.y = element_text(margin = margin(r =
    axis.title.x = element_text(margin = margin(t = 15)), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = )


### S4D (Fungal Mosquitoes Rarecurve)
Q6_2_16S_decom <- readRDS("*Decom_Data/Q6_2_16S_decom.rds")

### Make rarecurve
Q6_2_rarecurve_decom <- ggrare(Q6_2_16S_decom, step = 100, color = "Age",
    se = FALSE) + geom_vline(xintercept = 2229, linetype = "dashed") +
    xlab("Sample Read Coverage") + ylab("OTU Richness") + theme_bw()
Q6_2_rarecurve_decom$labels$colour <- "Fungal"
Q6_2_rarecurve_decom <- Q6_2_rarecurve_decom + scale_color_manual(values = c("#FF0000",
    "#FF9999")) + theme(text = element_text(size = 25), axis.title.y = element_text(margin = margin(r =
    axis.title.x = element_text(margin = margin(t = 15)), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), panel.border = )


### S4E (Eggs/Food Rarecurve) Make rarecurve
Q8_16S_decom <- readRDS("*Decom_Data/Q8_16S_decom.rds")
Q8_16S_decom@sam_data$Type <- revalue(Q8_16S_decom@sam_data$Type,
    replace = c(eggs = "Eggs"))
Q8_16S_decom@sam_data$Type <- revalue(Q8_16S_decom@sam_data$Type,
    replace = c(food = "Food"))

Q8_rarecurve_decom <- ggrare(Q8_16S_decom, step = 100, color = "Type",
    se = FALSE) + geom_vline(xintercept = 7104, linetype = "dashed") +
    theme_bw()
Q8_rarecurve_decom <- Q8_rarecurve_decom + theme(text = element_text(size = 25),
    axis.title.y = element_text(margin = margin(r = 15)), axis.title.x = element_text(margin = margin(t
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.border = )
Q8_rarecurve_decom
```