Experiment 19: Larval microbiome by probiotic cocktail treatment and species

Mary English

2/10/2020

```
path <- "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/"
library(dada2)
library(vegan)
library(BiocGenerics)
library(Biostrings)
library(phyloseq)
library(ggplot2)
library(ape)
```

Get forward and reverse reads.

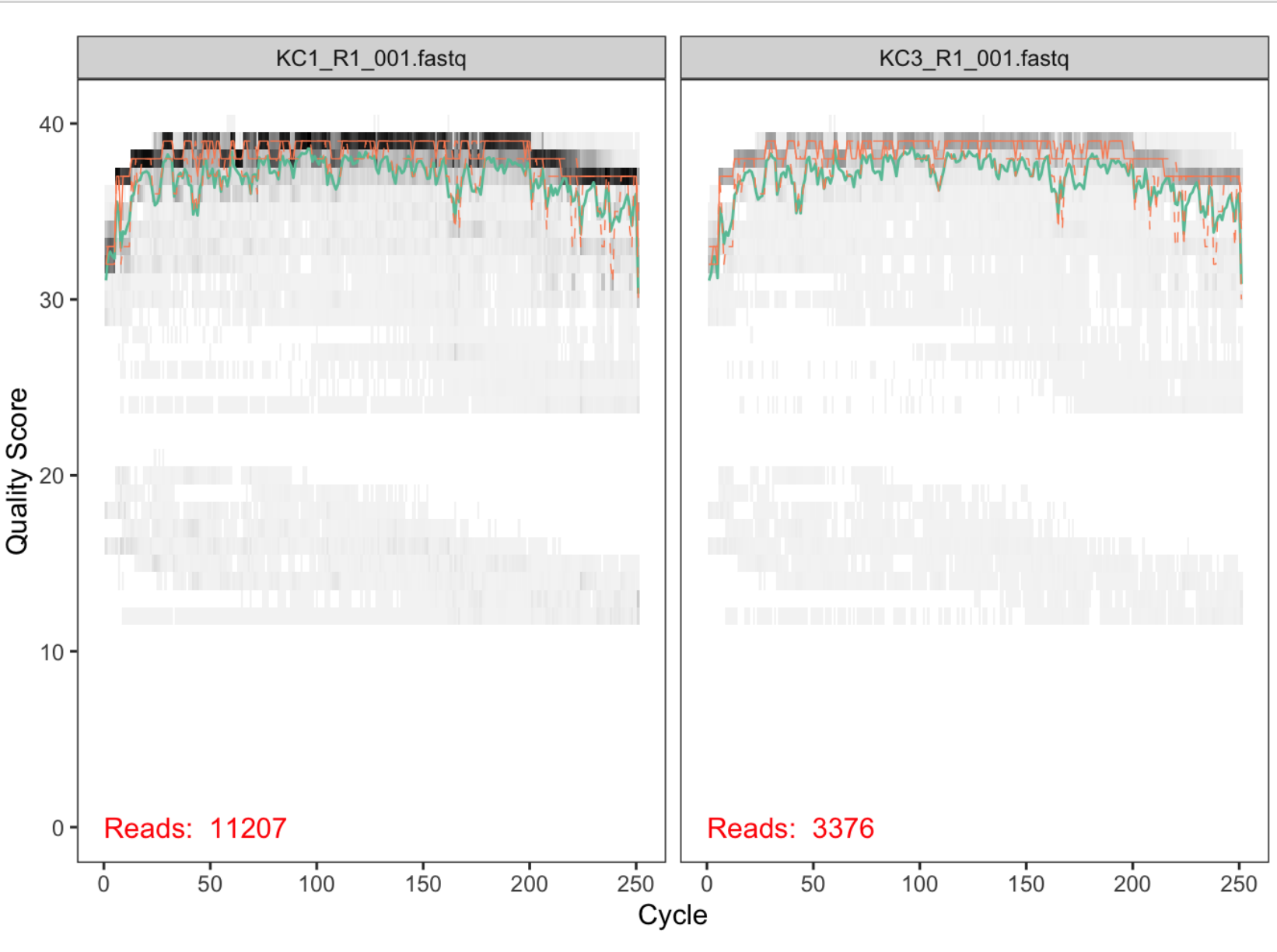
```
fnFs <- sort(list.files(path, pattern="_R1_001.fastq", full.names = TRUE))
fnRs <- sort(list.files(path, pattern="_R2_001.fastq", full.names = TRUE))
```

Extract sample names from the file names and store the sample names in a separate object.

```
sample.names <- sapply(strsplit(basename(fnFs), "_"), `[`, 1)
```

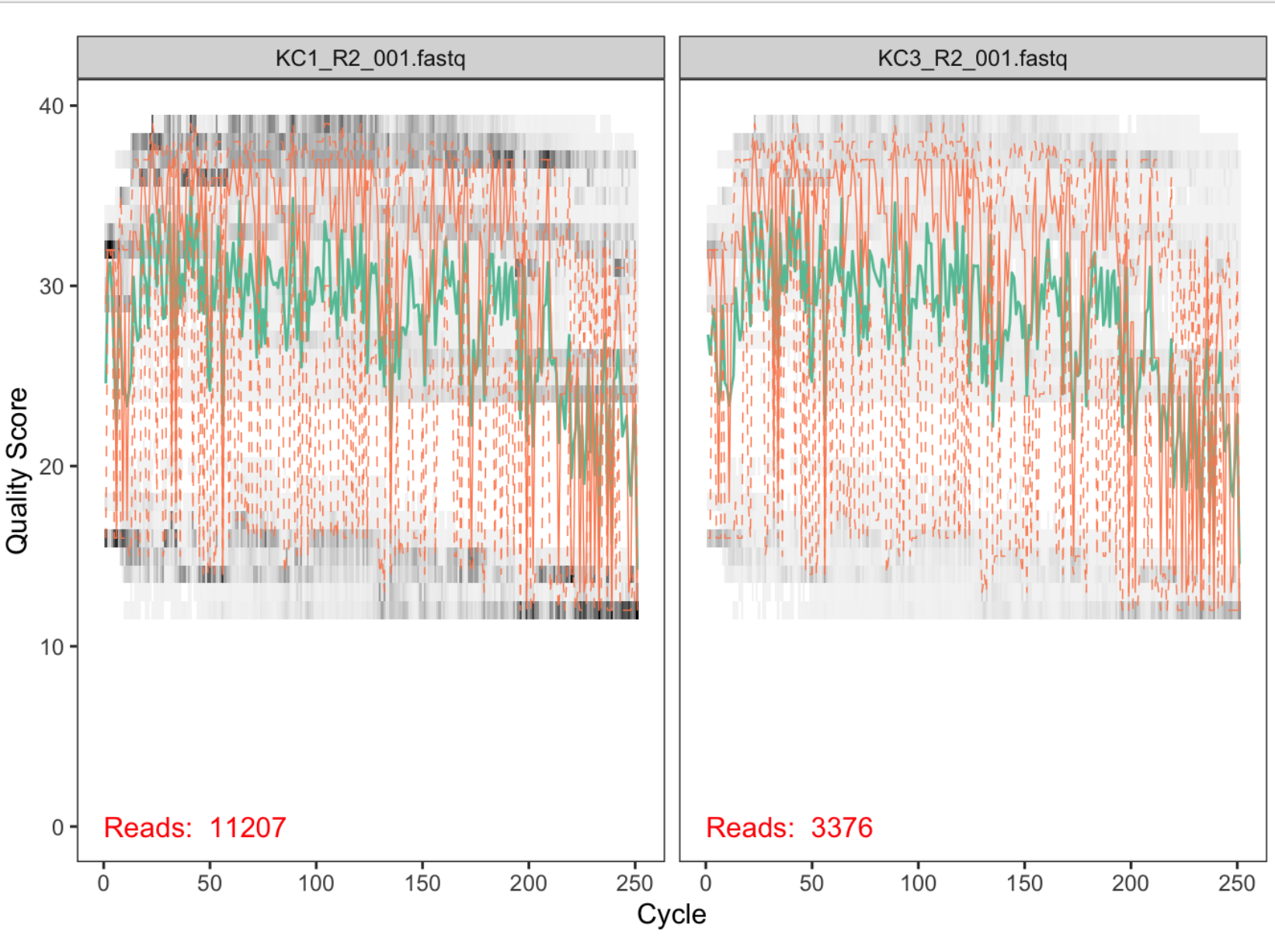
Plot an example of read quality.

```
plotQualityProfile(fnFs[1:2])
```



Forward drops off at about 240.
Reverse drops off at about 140.

```
plotQualityProfile(fnRs[1:2])
```



Distinguish the reads that are quality-controlled (which we'll do next) from original .fastq files

```
filtFs <- file.path(path, "filtered", paste0(sample.names, "_F_filt.fastq.gz"))
filtRs <- file.path(path, "filtered", paste0(sample.names, "_R_filt.fastq.gz"))
names(filtFs) <- sample.names
names(filtRs) <- sample.names
```

Filtering using fwd and rev read quality dropoff measurements, and default parameters for everything else.

```
out <- filterAndTrim(fnFs, filtFs, fnRs, filtRs,
                     truncLen=c(240,140),
                     maxN=0,
                     maxEE=c(2,2),
                     truncQ=2,
                     rm.phix=TRUE,
                     compress=TRUE,
                     multithread=TRUE)
```

Learn errors.

```
errF <- learnErrors(filtFs, multithread=TRUE)
```

```
## 27382560 total bases in 114094 reads from 28 samples will be used for learning the error rates.
```

```
errR <- learnErrors(filtRs, multithread=TRUE)
```

```
## 15973160 total bases in 114094 reads from 28 samples will be used for learning the error rates.
```

Use filtered sequences and learned error rates to make an object for dada2 to work with.

```
dadaFs <- dada(filtFs, err=errF, multithread=TRUE)
dadaRs <- dada(filtRs, err=errR, multithread=TRUE)
```

"Denoise" – inspect the object to see how many true sequence variants there are in all the sequences in a given sample (in this case, sample 1).

```
dadaFs[[1]] # 116 variants
```

```
## dada-class: object describing DADA2 denoising results
## 116 sequence variants were inferred from 4491 input unique sequences.
## Key parameters: OMEGA_A = 1e-40, OMEGA_C = 1e-40, BAND_SIZE = 16
```

Merge forward and reverse reads.

```
mergers <- mergePairs(dadaFs, filtFs, dadaRs, filtRs, verbose=TRUE)
```

Make sequence table.

```
seqtab <- makeSequenceTable(mergers)
dim(seqtab) # 28 x 275. 28 samples; so we lost 1. rip
```

```
## [1]  28 275
```

```
table(nchar(getSequences(seqtab)))
```

```
##
## 251 252 253 254 255 257
##   1   4 262   6   1   1
```

These lengths are all very close in value, so there are no outliers to remove.

Remove bimeras.

```
seqtab.nochim <- removeBimeraDenovo(seqtab, method="consensus", multithread=TRUE, verbose=TRUE)
```

```
## Identified 0 bimeras out of 275 input sequences.
```

```
dim(seqtab.nochim) # no chimeras
```

```
## [1]  28 275
```

Track filtering.

```
getN <- function(x) sum(getUniques(x))
track <- cbind(out, sapply(dadaFs, getN), sapply(dadaRs, getN), sapply(mergers, getN), rowSums(seqtab.nochim))
colnames(track) <- c("input", "filtered", "denoisedF", "denoisedR", "merged", "nonchim")
rownames(track) <- sample.names
write.csv(track, file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_19_filter_track.cs
v")
```

Based on these numbers, I removed samples with <1000 "final" (nonchim) reads. I also removed these from the corresponding metadata table.

```
exp_19_asv <- seqtab.nochim
dim(exp_19_asv) # 28 x 275
```

```
## [1]  28 275
```

```
exp_19_asv_filt <- exp_19_asv[-c(3,4,10,13,14,18,20,21,25,27),]
dim(exp_19_asv_filt) # 18 x 275, 18 samples.
```

```
## [1]  18 275
```

```
write.csv(exp_19_asv_filt, file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_19_asv_f
ilt.csv")
```

Assign taxonomy.

```
exp_19_taxa_filt <- assignTaxonomy(exp_19_asv_filt, "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/e
xp_18/silva_nr_v132_train_set.fa.gz", multithread=TRUE, tryRC = TRUE)
write.csv(exp_19_taxa_filt, file =  "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_19_tax
a_filt.csv")
```

If taxonomy only assigns to kingdom level, try this from https://github.com/benjjneb/dada2/issues/192: tax <- assignTaxonomy(dada2:::rc(getSequences(st)), ref.file))

Remove mitochondria and chloroplasts.

```
reduce_taxa <- apply(exp_19_taxa_filt, 1, function(r) any(r %in% c("Chloroplast", "Mitochondria")))
dim(exp_19_asv_filt) # 18 x 275
```

```
## [1]  18 275
```

```
exp_19_asv_filt_reduc <- exp_19_asv_filt[,!reduce_taxa]
dim(exp_19_asv_filt_reduc) # 18 x 271
```

```
## [1]  18 270
```

```
exp_19_asv_filt_reduc <- t(exp_19_asv_filt_reduc)
dim(exp_19_asv_filt_reduc) # 271 x 18
```

```
## [1] 270  18
```

There are 4 mitochondria and chloroplasts in these reads.
Remove from taxa table.

```
dim(exp_19_taxa_filt) # 275 x 6
```

```
## [1] 275   6
```

```
exp_19_taxa_filt_reduc <- exp_19_taxa_filt[!reduce_taxa,]
dim(exp_19_taxa_filt_reduc) # 271 x 6
```

```
## [1] 270   6
```

Start writing fasta – make 'seqs' BEFORE changing to asvs.

```
seqs <- rownames(exp_19_asv_filt_reduc)
```

Change sequences to "asv001" etc. in taxa and asv tables.

```
library(lessR)
asvs <- to("asv", nrow(exp_19_taxa_filt_reduc))
rownames(exp_19_taxa_filt_reduc) <- asvs
rownames(exp_19_asv_filt_reduc) <- asvs
```

Rewrite asv and taxa tables after the reduction of mitochondria and chloroplast reads.

```
write.csv(exp_19_asv_filt_reduc, file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_19
_asv_filt_reduc.csv")
write.csv(exp_19_taxa_filt_reduc, file =  "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_
19_taxa_filt_reduc.csv")
```

Finish writing fasta of asvs.

```
seqs <- DNAStringSet(seqs)
names(seqs) <- asvs
writeXStringSet(x = seqs, filepath = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_19_fi
lt_reduc_seqs.fa")
```

Following steps are done on command line.
Align and filter seqs.
mothur > align.seqs(fasta=exp_19_filt_reduc_seqs.fa, reference=/nfs0/Mueller_Lab/sequence_databases/16S_db/silva/silva.nr_v132.align, flip=t, processors=48)
mothur > filter.seqs(fasta=exp_19_filt_reduc_seqs.align, processors=20)
Build phylogenetic tree and re-root from midpoint.
FastTreeMP -gtr -nt -log exp_19_filt_reduc_seqs.filter.log exp_19_filt_reduc_seqs.filter.fasta > exp_19_filt_reduc_seqs.filter.tre
/local/cluster/mueller/scripts/Genomics/stat/reroot.pl -midpoint < exp_19_filt_reduc_seqs.filter.tre > exp_19_filt_reduc_seqs.filter.midroot.tre

tree file: exp_19_filt_reduc_seqs.filter.midroot.tre
fasta file: exp_19_filt_reduc_seqs.filter.fasta

Load tree.

```
exp_19_tree <- read.tree(file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_19_filt_re
duc_seqs.filter.midroot.tre")
plot(exp_19_tree)
```

Load fasta file.

```
exp_19_fasta <- readDNAStringSet("~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_19_filt_r
educ_seqs.filter.fasta")
```

Handoff to phyloseq.

Start organizing the different tables needed for phyloseq.

```
meta_19 <- read.table("~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_19_metadata.csv",
                      sep = ",", header = TRUE, row.names = 1)
taxa_19 <- as.matrix(read.table("~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_19_taxa_fi
lt_reduc.csv",
                                sep = ",", header = TRUE, row.names = 1))
asv_19 <- as.matrix(read.table("~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_19/exp_19_asv_filt
_reduc.csv",
                               sep = ",", header = TRUE, row.names = 1))
```

Build phyloseq object.

```
exp_19_phylo <- phyloseq(otu_table(asv_19, taxa_are_rows = TRUE),
                         sample_data(meta_19),
                         tax_table(taxa_19),
                         phy_tree(exp_19_tree),
                         refseq(exp_19_fasta))
```

Prune taxa that <= 0.

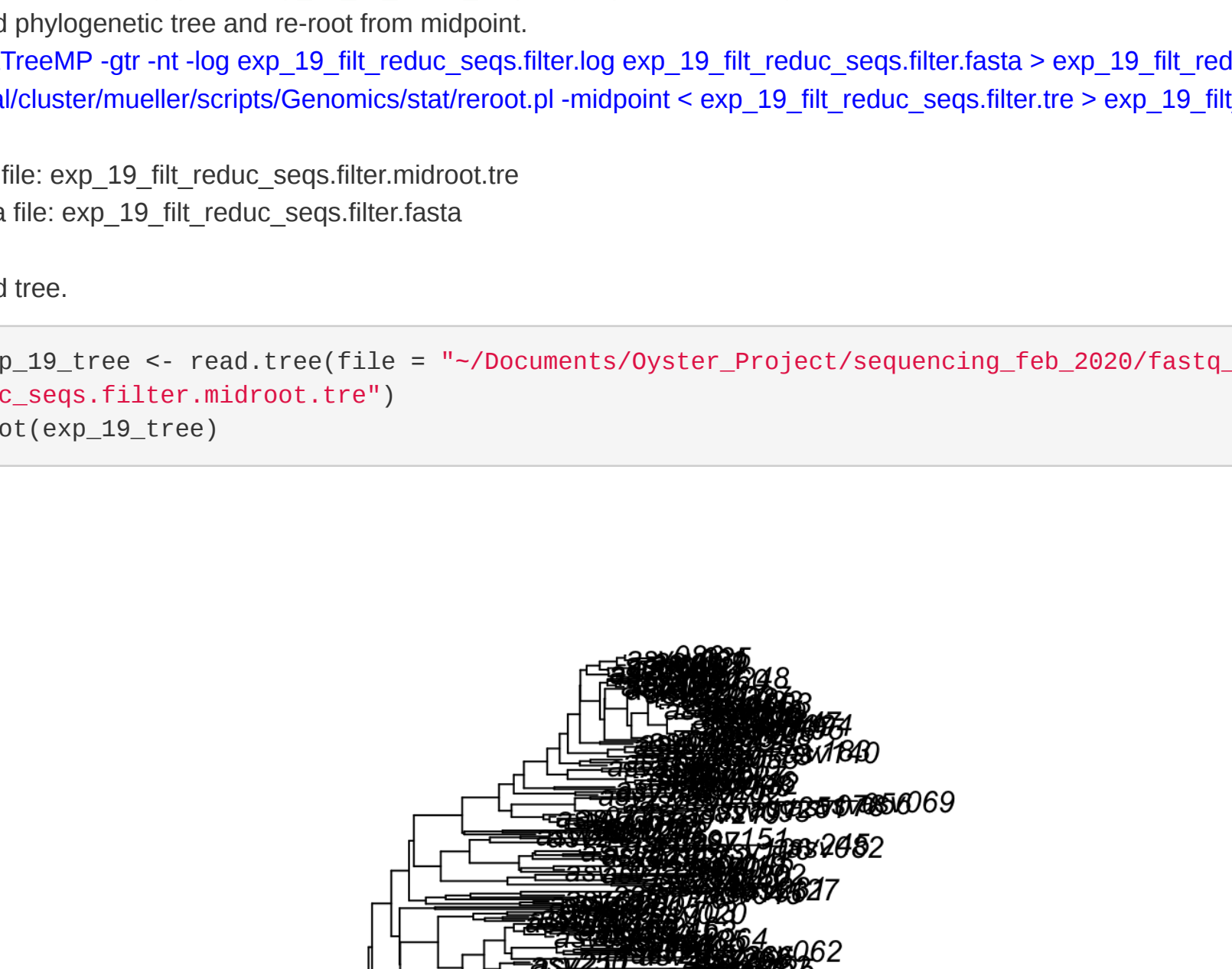
```
exp_19_phylo_sum <- prune_taxa(taxa_sums(exp_19_phylo) > 0, exp_19_phylo)
```

Plot beta diversity - weighted UNIFRAC.

```
exp_19_phylo_sum_wuni <- ordinate(exp_19_phylo_sum, "PCoA", "unifrac", weighted=TRUE)
exp_19_phylo_sum_wuni_plot <- plot_ordination(exp_19_phylo_sum, exp_19_phylo_sum_wuni,
                                              color = "Lineage",
                                              shape = "Treatment",
                                              title = "Exp 19 larval microbiome ordination plot (Weighted UNIFRA
C)")
exp_19_phylo_sum_wuni_plot +
  geom_point(size = 6) +
  scale_colour_manual(values = c("#FDE725FF", "#73D055FF", "#2D708EFF")) +
  theme(panel.background = element_rect(fill = "gray85"),
        panel.border = element_rect(linetype = 1, colour = "black", fill = NA),
        legend.key = element_rect(fill = "gray85"),
        legend.background = element_rect(linetype = 1, colour = "black"))
```

Plot beta diversity - unweighted UNIFRAC.

```
exp_19_phylo_sum_uni <- ordinate(exp_19_phylo_sum, "PCoA", "unifrac", weighted= FALSE)
exp_19_phylo_sum_uni_plot <- plot_ordination(exp_19_phylo_sum, exp_19_phylo_sum_uni,
                                             color = "Lineage",
                                             shape = "Treatment",
                                             title = "Exp 19 larval microbiome ordination plot (Unweighted UNIFRA
C)")
exp_19_phylo_sum_uni_plot +
  geom_point(size = 6) +
  scale_colour_manual(values = c("#FDE725FF", "#73D055FF", "#2D708EFF")) +
  theme(panel.background = element_rect(fill = "gray85"),
        panel.border = element_rect(linetype = 1, colour = "black", fill = NA),
        legend.key = element_rect(fill = "gray85"),
        legend.background = element_rect(linetype = 1, colour = "black"))
```



PERMANOVA
Questions:
> Do probiotic and control cluster separately?
> How much does oyster species affect clustering?

```
exp_19_phylo_sum_wunidist <- UniFrac(exp_19_phylo_sum, weighted = TRUE, normalized=TRUE, parallel=TRUE, fast=TRUE
)
exp_19_phylo_sum_wunidist_treat <- adonis2(exp_19_phylo_sum_wunidist ~ Treatment, data = meta_19)
exp_19_phylo_sum_wunidist_treat
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = exp_19_phylo_sum_wunidist ~ Treatment, data = meta_19)
##           Df SumOfSqs      R2      F Pr(>F)
## Treatment  1  0.01545 0.07607 1.3173  0.161
## Residual  16  0.18765 0.92393
## Total     17  0.20310 1.00000
```

```
exp_19_phylo_sum_wunidist_line <- adonis2(exp_19_phylo_sum_wunidist ~ Lineage, data = meta_19)
exp_19_phylo_sum_wunidist_line
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## adonis2(formula = exp_19_phylo_sum_wunidist ~ Lineage, data = meta_19)
##          Df SumOfSqs      R2      F Pr(>F)
## Lineage   2 0.030401 0.14968 1.3202  0.116
## Residual 15 0.172700 0.85032
## Total    17 0.203100 1.00000
```