Exp18_for_mb668: Larval microbiome by probiotic cocktail treatment

Mary English

6/6/2020

Summary: These samples come from Experiment 18 (10L bucket trial) where larvae were treated with the probiotic cocktail (B11 + DM14 + D16) or not. There were five replicates of each treatment (larvae only, probiotics a single time, probiotics after every water change). DNA was isolated from these 15 samples and the 16S rRNA gene amplified and sequenced on Illumina HiSeq 3000. Sequences were filtered for quality and a total of 1415 samples remained. Beta diversity was plotted and PERMANOVAs performed to look for differences in the community composition between samples depending on the variables a) whether or not probiotics were added at all, and 2) the number of times the probiotic cocktail was added.

```
path <- "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/"
library(dada2)
library(ggplot)
library(RColorBrewer)
library(phangorn)
library(phyloseq)
library(ggplot2)
library(ape)
library(reshape2)
```

Get forward and reverse reads.

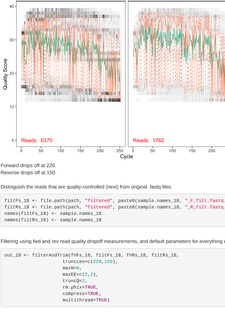
```
fnFs_18 <- sort(list.files(path, pattern="_R1_001.fastq", full.names = TRUE))
fnRs_18 <- sort(list.files(path, pattern="_R2_001.fastq", full.names = TRUE))
```

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

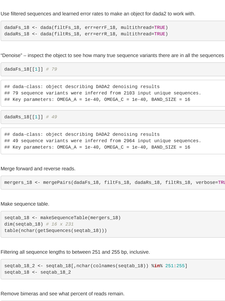
```
sample.names_18 <- sapply(strsplit(basename(fnFs_18), "_"), `[`, 1)
```

Plot an example of read quality.

```
plotQualityProfile(fnFs_18[1:2])
```




```
plotQualityProfile(fnRs_18[1:2])
```



Forward drops off at 220.
Reverse drops off at 150.

Distinguish the reads that are quality-controlled (new) from original .fastq files.

```
filtFs_18 <- file.path(path, "filtered", paste0(sample.names_18, "_F_filt.fastq.gz"))
filtRs_18 <- file.path(path, "filtered", paste0(sample.names_18, "_R_filt.fastq.gz"))
names(filtFs_18) <- sample.names_18
names(filtRs_18) <- sample.names_18
```

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

```
out_18 <- filterAndTrim(fnFs_18, filtFs_18, fnRs_18, filtRs_18,
                        truncLen=c(220,150),
                        maxN=0,
                        maxEE=c(2,2),
                        truncQ=2,
                        rm.phix=TRUE,
                        compress=TRUE,
                        multithread=TRUE)
```

Learn error rates.

```
errF_18 <- learnErrors(filtFs_18, multithread=TRUE)
errR_18 <- learnErrors(filtRs_18, multithread=TRUE)
```

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

```
dadaFs_18 <- dada(filtFs_18, err=errF_18, multithread=TRUE)
dadaRs_18 <- dada(filtRs_18, err=errR_18, 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_18[[1]] # 79

## dada-class: object describing DADA2 denoising results
## 79 sequence variants were inferred from 2103 input unique sequences.
## Key parameters: OMEGA_A = 1e-40, OMEGA_C = 1e-40, BAND_SIZE = 16

dadaRs_18[[1]] # 49

## dada-class: object describing DADA2 denoising results
## 49 sequence variants were inferred from 2964 input unique sequences.
## Key parameters: OMEGA_A = 1e-40, OMEGA_C = 1e-40, BAND_SIZE = 16
```

Merge forward and reverse reads.

```
mergers_18 <- mergePairs(dadaFs_18, filtFs_18, dadaRs_18, filtRs_18, verbose=TRUE)
```

Make sequence table.

```
seqtab_18 <- makeSequenceTable(mergers_18)
dim(seqtab_18) # 15 x 231
table(nchar(getSequences(seqtab_18)))
```

Filtering all sequence lengths to between 251 and 255 bp, inclusive.

```
seqtab_18_2 <- seqtab_18[,nchar(colnames(seqtab_18)) %in% 251:255]
seqtab_18 <- seqtab_18_2
```

Remove bimeras and see what percent of reads remain.

```
seqtab_18.nochim <- removeBimeraDenovo(seqtab_18, method="consensus", multithread=TRUE, verbose=TRUE)

## Identified 2 bimeras out of 227 input sequences.

# 2 bimeras
sum(seqtab_18.nochim)/sum(seqtab_18)

## [1] 0.9984134
```

Track filtering.

```
getN <- function(x) sum(getUniques(x))
track_18 <- cbind(out_18, sapply(dadaFs_18, getN), sapply(dadaRs_18, getN), sapply(mergers_18, getN), rowSums(seqtab_18.nochim))
colnames(track_18) <- c("input", "filtered", "denoisedF", "denoisedR", "merged", "nonchim")
rownames(track_18) <- sample.names_18
write.csv(file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/exp_18_filter_track.csv", x = track_18)
```

Removing one sample – LB – because it had <1000 reads after filtering. This is done later on.

Make asv table.

```
df_18 <- as.data.frame(seqtab_18.nochim)
df_18 <- as.matrix(df_18)
```

Not going to rarefy. Remove sample LB because it has 685 reads; next-lowest is 1043.

```
dim(df_18) # 187 231
df_18_noLB <- df_18[-c(13),] # remove row 12 (LB)
dim(df_18_noLB) # 14 231
df_18_noLB_noneg <- df_18_noLB[-c(15),]# also removing row 15, the negative
dim(df_18_noLB_noneg) # 14 x 225
min(rowSums(df_18_noLB_noneg)) # 1043
rowSums(df_18_noLB_noneg)
exp_18_asv <- df_18_noLB_noneg
write.csv(exp_18_asv, file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/exp_18_asv.csv")
```

Assign taxonomy.

```
exp_18_taxa <- assignTaxonomy(exp_18_asv, "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/silva_nr_v132_train_set.fa.gz",
                              multithread=TRUE,
                              tryRC = TRUE)
```

Create function to remove mitochondria and chloroplasts (below)

```
reduce_taxa <- apply(exp_18_taxa, 1, function(r) any(r %in% c("Chloroplast", "Mitochondria")))
```

Remove from ASV count table and see how many were removed.

```
dim(exp_18_asv) # 14 x 225

## [1] 14 225

write.csv(exp_18_asv, file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/asvs_incl_mitochloro.csv")
exp_18_asv_reduce <- exp_18_asv[,!reduce_taxa]
dim(exp_18_asv_reduce) # 14 x 218

## [1] 14 218
```

225 -> 218, so 7 were mitoChloro.

Get the sequences before renaming them to ASVs.

```
exp_18_asv <- exp_18_asv_reduce
seqs_18 <- colnames(exp_18_asv)
length(seqs_18) # 218

## [1] 218
```

Now transpose (ASVs need to be rows).

```
exp_18_asv <- t(exp_18_asv)
dim(exp_18_asv) # 218 x 14

## [1] 218 14
```

Remove the mitochondria and chloroplasts from taxonomy table. Then, rename rows of both ASV and taxonomy tables.

```
library(janeR)
dim(exp_18_taxa) # 225 x 6.

## [1] 225   6

exp_18_taxa <- exp_18_taxa[!reduce_taxa,]
dim(exp_18_taxa) # 218 x 6

## [1] 218   6

asvs_18 <- t(c("asv", nrow(exp_18_taxa))
rownames(exp_18_asv) <- asvs_18 #bind
write.csv(exp_18_asv, file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/asvs_for_mothur.csv")
rownames(exp_18_taxa) <- asvs_18 #bind
dim(exp_18_taxa) # 218 x 6

## [1] 218   6

write.csv(exp_18_taxa, file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/exp_18_taxa.csv"
)
```

For phyloseq.

```
exp_18_meta <- read.csv("~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/exp_18_nolb_meta.csv",
sep=",", header=T, row.names=1)
```

Make a fasta file for phyloseq object.

```
seqs_18 <- DNAStringSet(seqs_18)
names(seqs_18) <- asvs_18
writeXStringSet(x = seqs_18, filepath = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/exp_18_asv_strings.fa")
```

Align and filter seqs.

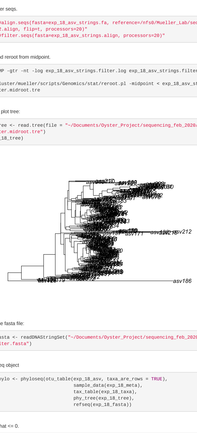
```
mothur "#align.seqs(fasta=exp_18_asv_strings.fa, reference=/nfsdb/Mueller_Lab/sequence_databases/16S_db/silva/silva_nr_v132.align, flip=t, processors=20)"
mothur "#filter.seqs(fasta=exp_18_asv_strings.align, processors=20)"
```

Build tree and reroot from midpoint.

```
FastTreeMP -gtr -nt -log exp_18_asv_strings.filter.log exp_18_asv_strings.filter.fasta > exp_18_asv_strings.filter.tre
/local/cluster/mueller/scripts/Genomics/stat/reroot.pl -midpoint < exp_18_asv_strings.filter.tre > exp_18_asv_strings.filter.midroot.tre
```

Read in and plot tree:

```
exp_18_tree <- read.tree(file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/exp_18_asv_strings.filter.midroot.tre")
plot(exp_18_tree)
```



Get reference fasta file:

```
exp_18_fasta <- readDNAStringSet("~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/exp_18_asv_strings.filter.fasta")
```

build phyloseq object

```
exp_18_phylo <- phyloseq(otu_table(exp_18_asv, taxa_are_rows = TRUE),
                         sample_data(exp_18_meta),
                         tax_table(exp_18_taxa),
                         phy_tree(exp_18_tree),
                         refseq(exp_18_fasta))
```

Prune taxa that <= 0.

```
exp_18_phylo_sum <- prune_taxa(taxa_sums(exp_18_phylo) > 0, exp_18_phylo)
```

Extract sample 1s from phyloseq object into a count file.

```
exp_18_18 <- exp_18_phylo_sum@otu_table[,"18"]
write.csv(exp_18_18, file = "~/Documents/Oyster_Project/sequencing_feb_2020/fastq_files/exp_18/extraction_18.csv"
)
```

Then, use the VLOOKUP function in Excel to match data in taxonomy file to the ASVs listed in the extracted count file.

Done