***Processing of MiSeq data***

Raw sequence data for all 113 samples in this study (plus an additional 13 cloacal samples and 1 *C. pallida* sample) can be accessed in the Sequence Read Archive under accession number SRP130314. A sequential list of the in-house QIIME.1.8.0 commands used to process the data is given below.

1. Joining paired ends to generate contigs

join\_paired\_ends.py -f <sample\_name\_R1.fastq> -r <sample\_name\_R2.fastq> -m fastq-join -j 50 -p 8 -o <sample\_name>\_joined/

2. Quality trimming, converting from fastq to fasta format

split\_libraries\_fastq.py -i <sample\_name>\_joined/fastqjoin.join.fastq -o <sample\_name>\_trimmed/ -m dummy\_mapping.txt --sample\_id <sample\_name> -q 29 -n 0 --barcode\_type 'not-barcoded' --store\_qual\_scores

3. Concatenating all sequences from all samples

cat \*checked/uchime\_nonchimeras.fna > all\_seqs\_all\_samples.fasta

4. Picking OTUs at 97% similarity (NB: 99% was also run)

pick\_otus.py -i all\_seqs\_all\_samples.fasta -s 0.97 -o uclust\_picked\_otus\_97/

5. Picking representative sequences for each OTU

pick\_rep\_set.py -i uclust\_picked\_otus\_97/all\_seqs\_all\_samples\_otus.txt -f all\_seqs\_all\_samples.fasta -m most\_abundant

6. Assigning taxonomy for each OTU

assign\_taxonomy.py -i all\_seqs\_all\_samples.fasta\_rep\_set.fasta -t SSURef\_NR99\_119\_SILVA\_14\_07\_14\_opt\_SACqiime.tax -r SSURef\_NR99\_119\_SILVA\_14\_07\_14\_opt\_SACqiime.fasta --uclust\_similarity 0.9 --uclust\_max\_accepts 10 --uclust\_min\_consensus\_fraction 0.90 -o uclust\_picked\_otus\_97/uclust\_taxa\_0.9\_10\_0.90/

7. Making OTU table in biom format

make\_otu\_table.py -i uclust\_picked\_otus\_97/all\_seqs\_all\_samples\_otus.txt -t uclust\_picked\_otus\_97/uclust\_taxa\_0.9\_10\_0.90/all\_seqs\_all\_samples.fasta\_rep\_set\_tax\_assignments.txt -o uclust\_picked\_otus\_97/uclust\_taxa\_0.9\_10\_0.90/OTU\_table\_Silva\_119\_all\_seqs.biom

8. Removing singleton OTUs from the dataset

filter\_otus\_from\_otu\_table.py -i uclust\_picked\_otus\_97/uclust\_taxa\_0.9\_10\_0.90/OTU\_table\_Silva\_119\_all\_seqs.biom -n 2 -o uclust\_picked\_otus\_97/uclust\_taxa\_0.9\_10\_0.90/OTU\_table\_singletonfiltered.biom

9. Filtering known PCR contaminant taxa from the dataset by taxonomic name

filter\_taxa\_from\_otu\_table.py -i uclust\_picked\_otus\_97/uclust\_taxa\_0.9\_10\_0.90/OTU\_table\_singletonfiltered.biom -o uclust\_picked\_otus\_97/uclust\_taxa\_0.9\_10\_0.90/OTU\_table\_singleton\_taxafiltered.biom -n Eukaryota

10. Generating Excel-readable OTU table with taxonomic information for use in R (\*employs an in-house perl command written by Dr. Connor Skennerton)

join\_otu\_repset.pl uclust\_picked\_otus\_97/uclust\_taxa\_0.9\_10\_0.90/OTU\_table\_singleton\_taxafiltered.biom all\_seqs\_all\_samples.fasta\_rep\_set.fasta > OTU\_table\_wTaxa\_wSeqs\_97.txt

11. Counting number of remaining sequences per sample to identify the level for rarefaction (found to be 9,878 sequences/sample, equal to the smallest sample, Finch-142)

biom summarize-table -i uclust\_picked\_otus\_97/uclust\_taxa\_0.9\_10\_0.90/OTU\_table\_singleton\_taxafiltered.biom -o OTU\_table\_singleton\_taxafiltered.txt

***Diversity Analyses***

1. Making an OTU Multiple-Sequence Alignment

From the results of Step 5 above (pick\_rep\_set.py), the .fasta sequence file was uploaded to MAFFT Version 7 Online (Katoh and Standley 2013 [[*Molecular Biology and Evolution* **30**:772-780](http://mbe.oxfordjournals.org/content/30/4/772)]) to construct a multiple-sequence alignment selecting the FFT-NS-2 option (fast but rough):

mafft --reorder --memsavetree --retree 2 input

2. Generating a phylogenetic tree

In QIIME.1.8.0

make\_phylogeny.py -i $PWD/aligned.fasta -o $PWD/rep\_phylo.tre

3. Core Diversity Analyses

core\_diversity\_analyses.py -i $PWD/otu\_table.biom -o $PWD/core\_output -m $PWD/map.txt -c Species,Island,Season -t $PWD/rep\_phylo.tre -e 9878 --suppress\_beta\_diversity

***DADA2 Processing***

The same 127 samples were processed using the DADA2 pipeline adapted from Callahan (2016) and the Dada2 Pipeline Tutorial v1.6 (http://benjjneb.github.io/dada2/tutorial.html).

1. Load Files

library(dada2); packageVersion("dada2")

library(ShortRead); packageVersion("ShortRead")

library(ggplot2); packageVersion("ggplot2")

path <- "~/wherefastqfilesare"

fns <- list.files(path)

fns

fastqs <- fns[grepl(".fastq$", fns)]

fastqs <- sort(fastqs) # Sort ensures forward/reverse reads are in same order

fnFs <- fastqs[grepl("\_R1", fastqs)] # Just the forward read files

fnRs <- fastqs[grepl("\_R2", fastqs)] # Just the reverse read files

2. Get sample names, assuming files named as so: SAMPLENAME\_XXX.fastq

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

# Specify the full path to the fnFs and fnRs

fnFs <- file.path(path, fnFs)

fnRs <- file.path(path, fnRs)

plotQualityProfile(fnFs[[1]])

plotQualityProfile(fnFs[[2]])

plotQualityProfile(fnRs[[1]])

plotQualityProfile(fnRs[[2]])

3. Make directory and filenames for the filtered fastqs and filter

filt\_path <- file.path(path, "filtered")

if(!file\_test("-d", filt\_path)) dir.create(filt\_path)

filtFs <- file.path(filt\_path, paste0(sample.names, "\_F\_filt.fastq.gz"))

filtRs <- file.path(filt\_path, paste0(sample.names, "\_R\_filt.fastq.gz"))

# Filter

for(i in seq\_along(fnFs)) {

 fastqPairedFilter(c(fnFs[i], fnRs[i]), c(filtFs[i], filtRs[i]),

 truncLen=c(220,150),

 maxN=0, maxEE=c(1,1), truncQ=2, rm.phix=TRUE,

 compress=TRUE, verbose=TRUE)

}

derepFs <- derepFastq(filtFs, verbose=TRUE)

derepRs <- derepFastq(filtRs, verbose=TRUE)

# Name the derep-class objects by the sample names

names(derepFs) <- sample.names

names(derepRs) <- sample.names

dadaFs.lrn <- dada(derepFs, err=NULL, selfConsist = TRUE, multithread=TRUE)

errF <- dadaFs.lrn[[1]]$err\_out

dadaRs.lrn <- dada(derepRs, err=NULL, selfConsist = TRUE, multithread=TRUE)

errR <- dadaRs.lrn[[1]]$err\_out

plotErrors(dadaFs.lrn[[1]], nominalQ=TRUE)

4. Sample Inference

dadaFs <- dada(derepFs, err=errF, multithread=TRUE)

dadaRs <- dada(derepRs, err=errR, multithread=TRUE)

dadaFs[[1]]

5. Merged Paired Reads

mergers <- mergePairs(dadaFs, derepFs, dadaRs, derepRs, verbose=TRUE)

# Inspect the merger data.frame from the first sample

head(mergers[[1]])

6. Make Sequence Table

seqtab <- makeSequenceTable(mergers)

dim(seqtab)

table(nchar(getSequences(seqtab))) #shows distribution of lengths

# cut to only lengths between 250-256 following inspection

seqtab2 <- seqtab[,nchar(colnames(seqtab)) %in% seq(250,256)]

7. Remove chimeras

seqtab2.nochim <- removeBimeraDenovo(seqtab2, verbose=TRUE)

dim(seqtab2.nochim)

sum(seqtab2.nochim)/sum(seqtab2)

8. Assign taxonomy using Silva Training Set (https://www.arb-silva.de/)

ref\_fasta <- "silva\_nr\_v123\_train\_set.fa.gz"

taxtab <- assignTaxonomy(seqtab2.nochim, refFasta = ref\_fasta)

colnames(taxtab) <- c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus")

write.csv(taxtab, file=“taxtab.csv”)

write.csv(seqtab2.nochim, file=“datable.csv”)

***NMDS, ANOSIM, and SIMPER tests in R***

Generic examples of the three commands are given below, including the options employed for this study. All commands require installation of the ‘vegan’ package in R, as well as two dependencies: ‘lattice’ and ‘permute’. For this study, the following package versions were employed:

‘phyloseq’: v1.25-2

‘vegan’: v2.4-5

 ‘lattice’: v0.20-35

‘permute’: v0.9-4

1. NMDS

NMDS\_analysis=metaMDS(transformed\_data,distance="bray",k=2,trymax=100,engine=c("monoMDS"),autotransform=FALSE)

2. ANOSIM

ANOSIM\_analysis=with(sample\_metadata,anosim(transformed\_data,Season,permutations=999,distance="bray"))

3. Adonis

Adonis\_analysis=adonis(transformed\_data~Season,data=sample\_metadata,permutations=999,method="bray")

4. SIMPER

SIMPER\_analysis=with(sample\_metadata,simper(transformed\_data,Season))

***Data Visualization***

The following additional R packages were used:

‘readxl’: v1.0.0

‘ggplot2’: v2.2.1