# Data analysis working group (DAWG) code for Session 1 on January 25th, 2019

# This dataset is a pyrosequenced ITS dataset that is in the form of a .sff file.

# We want to analyze these ITS sequences with DADA2 in R, which requires demultiplexed fastq files.

# We will need to convert the .sff into .fastq which requires several steps in QIIME1 which you may need to download if it is not already on your computer (<http://qiime.org/install/install.html>).

# After install, activate qiime 1

source activate qiime1

# Set working directory to where the .sff file is stored

cd /Users/admin/Downloads/Berts\_ITS\_Data/

ls #list the files in the directory

1. Convert .sff into .qual and .fna file

process\_sff.py -i ~/Downloads/Berts\_ITS\_Data/

# this makes two files, one that is a .qual file, and another .fna file

# .qual file contains quality information (Phred scores) for each base and is in fasta format

1. combine .qual and .fna into .fastq file

convert\_fastaqual\_fastq.py -f /Users/admin/Downloads/IPQ6P9001.fna -q /Users/admin/Downloads/IPQ6P9001.qual -o /Users/admin//Downloads/fastq\_file/

#This incorporates the sequence file (.fna), the Phred scores (.qual) to create a fastq file

1. Demultiplex .fastq file using program from [www.mrdnafreesoftware.com](http://www.mrdnafreesoftware.com)

# Use “Binning Program”

# Put ITS\_Barcodes.txt in “meta”

# put Fastq file in “fastq”,

# Click “Remove Barcodes”

# Click “Remove Linker Primer Sequence”

# Copy and Paste the Forward Linker Primer

# Copy and Paste the Reverse Primer Sequence

# Click “Bin Fastq” under “Bin Type”

# make sure that “454” is chosen under “Instrument

# “Run”

# this step may take a few minutes

# Now that we have 30 fastq files, we can begin our analysis in R

# DAWG R Code for Spring 2019 #

# source("http://bioconductor.org/biocLite.R")

# install.packages("BiocManager")

# BiocManager::install("ShortRead")

# BiocManager::install("devtools")

# BiocManager::install("dada2", version = "3.8")

# library(dada2)

#packageVersion("dada2")

##### 1. Set working directory ####

setwd("~/Downloads/Berts\_ITS\_Data/demultiplexed\_fastq")

#Tell R to pull files from the path

path<- setwd("~/Downloads/Berts\_ITS\_Data/demultiplexed\_fastq")

list.files(path)

#### 2. Sort the forward reads #####

# Forward fastq filenames have format: SampleName.fastq

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

#You should see in your global env, fnFs chr[1:30] "and the path directory"

# Extract sample names

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

#### 3. Examine the quality profiles ####

## forward reads ##

quartz() # only use if on Mac

plotQualityProfile(fnFs[1:6])

#### 4. Assign Filtering samples ####

# Assign the filenames for the filtered fastq.gz files

# Place filtered files in filtered/ subdirectory

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

# the commands included in this filter and trim are dependent on your specific data-

# especially the truncLen- because you want to trim your seqs based on the quality profiles you generated- quality scores of 30 + are good

# so you should choose the trunclen to reflect the quality scores of your data

out.2<- filterAndTrim(fnFs, filtFs, truncLen=400,

minLen = 200, maxN=0, maxEE=5, truncQ = 2,

multithread=TRUE, compress=TRUE)

# set multithread to FALSE if using windows

# this step takes awhile to perform

# setting trunQ=2 will remove sequences that have at least one read with a high probability of erroneous base assignment (>63%)

# minLen will remove sequences that are less than the specific value

# maxN = 0 will remove all sequences that have ambiguous base cales

# truncLen = 400 will cut off each sequence at base pair 400 and needs to be changed to account for every sequencing run

# look at quality profiles to determine what minLen and truncLen should be set to

# maxEE = 5 sets the maximum number of "expected errors" allowed in a read to be 5- in Illumina set to 2,2 as default but we are working with longer reads

out.2

head(out.2)

#### 5. Learn the Error Rates ####

# learn error rates

# error models are run specific

errf<- learnErrors(filtFs.1, multithread=TRUE) #set multithread=FALSE if on windows

plotErrors(errf, nominalQ = TRUE)

# shows error rates of transitions from one nucleotide to another

# black line shows the estimated error rates after convergence of the machine-learning algorithm

# red line shows the error rates expected under the nominal definition of the Q-score

# want the samples (dots) to track well with the black line

#### 6. Dereplicate the sequences ####

derepFs <- derepFastq(filtFs.1, verbose=TRUE)

# If there are 10 sequences that are the same, derepFs will contain 1 of the sequences

# speeds up downstream processing

names(derepFs) <- sample.names.1

##### 7. Call ESVs ####

# incorporates consensus quality profiles & abundances of unique sequences

# determines if the sequence is more likely to be of biological origin or spurious

dadaFs<-dada(derepFs, err=errf, HOMOPOLYMER\_GAP\_PENALTY=-1, BAND\_SIZE=32, USE\_QUALS=TRUE, multithread=TRUE)

# set multithread=FALSE if usin windows

# homopolymer gap penalty of -1, band size = 32, and setting USE\_QUALS=TRUE is suggested by the dada2 team

# The cost of gaps in homopolymer regions (>=3 repeated bases). Default is NULL, which causes homopolymer gaps to be treated as normal gaps.

# The default value of BAND\_SIZE is 16. If DADA is applied to sequencing technologies with high rates of indels, such as 454 sequencing, the BAND\_SIZE parameter should be increased.

# USE\_QUALS: If TRUE, the dada(...) error model takes into account the consensus quality score of the dereplicated unique sequences. If FALSE, quality scores are ignored

dadaFs[[1]]

# output shows X ESVs inferred from XXX input seqs from the first sample

# OMEGA-A is default parameter sets the level of “statistical evidence” (think p-value) required for inferences of a new ESV

# OMEGA-C The threshold at which unique sequences inferred to contain errors are corrected in the final output. The probability that each unique sequence is generated at its observed abundance from the center of its final partition is evaluated, and compared to OMEGA\_C. If that probability is >= OMEGA\_C, it is "corrected", i.e. replaced by the partition center sequence

# BandSize was set by the user in the filterAndTrim step

seqtab<-makeSequenceTable(dadaFs)

dim(seqtab)

#30 samples with 696 ESVs

#### 8. Remove Chimeric Sequences ####

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

#147 seqs were removed because they were chimeras

dim(seqtab.nochim)

# 30 samples with 549 ESVs

sum(seqtab.nochim)/sum(seqtab)

# ~95% of the sequences in the dataset are real biological sequences

getN <- function(x) sum(getUniques(x))

track <- cbind(out.2, sapply(dadaFs, getN), rowSums(seqtab.nochim))

colnames(track) <- c("input", "filtered", "denoisedF", "nonchim")

rownames(track) <- sample.names

head(track)

# Check this to see how many of your samples were dropped and if there are any steps where a majority of your samples were dropped

##### 9. Assign Taxonomy with Unite ####

taxa <- assignTaxonomy(seqtab.nochim,

"/Users/admin/Downloads/Unite/sh\_general\_release\_dynamic\_01.12.2017.fasta", multithread=TRUE)

taxa #check to make sure taxonomy looks appropriate