Tutorial pipeline for processing high-throughput *hgcAB* amplicon sequencing data

This tutorial is for processing and classifying mercury methylation genes (*hgcAB*) from a mock community dataset from Gionfriddo et al. 2020. The full-set of paired-end fastq files from the published study can be downloaded from the NCBI SRA database under BioProject: PRJNA608965. A small subset of the mock community dataset is used for this tutorial, 'fastq-sequencing-files.zip'. The tutorial dataset is only a small subset of the mock community data from the paper. Therefore, there are some differences in data outputs from this tutorial with the results from the full dataset.

The tutorial dataset contains *hgcAB* amplicon sequencing of two mock communities. Each community is a mix of three cultured Hg-methylator isolates from the *Deltaproteobacteria*, *Methanomicrobia* (*Euryarchaeota*), and *Firmicutes*. The dataset also includes a salt marsh sediment sample (ID: 1064), and the marsh sample spiked with the two mock communities (1064 + mock community 1, 1064 + mock community 2). The *hgcAB* genes were amplified with primers ORNL-HgcAB-uni-F, ORNL-HgcAB-uni-32R (Gionfriddo et al. 2020) and sequenced on Illumina Miseq 2x300 bp. The *hgcAB* amplicon is ~980 nt bp long, and therefore with short-read sequencing employed in this study, the forward and reverse reads do not overlap. Only the forward reads are used for downstream analyses. Please see the paper for a more detailed explanation of the methods. The trimmed (201 nt bp) forward read *hgcA* sequences are classified using the ORNL compiled Hg-methylator database reference package for short sequences, 'ORNL_HgcA_201.refpkg'.

Before you start:

This tutorial is intended to be run on a Unix/Linux environment. I recommend running Parts 1-4 through a Bioconda environment, which supports 64-bit Linux or Mac OS. The tutorial syntax is written for the versions of programs listed below. If different versions of the programs are used, then some of the command and script syntax may need to be updated. For Parts 1-4 of this tutorial you will need to be able to execute the following programs from the command line:

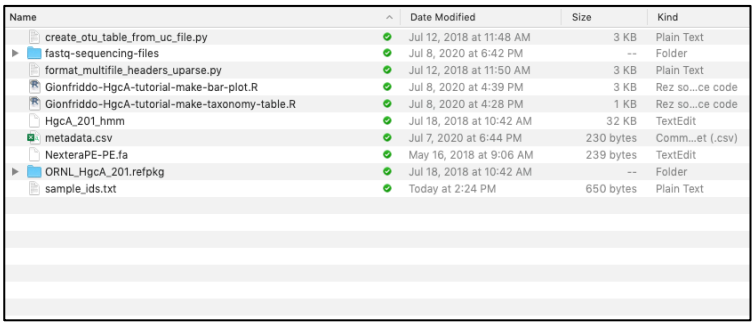
- Python (version 2.7)
- Trimmomatic (version 0.39)
- Vsearch (version 2.7.0)
- Cd-hit (version 4.6.8)
- Emboss (version 6.5.7)
- Hmmer (version 3.2)
- pplacer (version v1.1.alpha17)

Part 5 of the tutorial utilizes two R scripts to process the taxonomic classifications from the pplacer outputs, and producing a bar plot of relative abundances. You will need to install R (version 3.5.1 or later), and the following R libraries and dependencies:

- BoSSA
- phyloseq
- RColorBrewer
- ggplot2

Part 1: editing raw sequence files, trimming, and quality filtering

- Install python script 'format_multifile_headers_uparse.py' & Trimmomatic version 0.39
- Make python script executable:
  chmod +x format_multifile_headers_uparse.py
- The tab-delimited text file 'sample_ids.txt' has sample IDs and the corresponding fastq files on each line:
  SampleID forward_reads.fastq reverse_reads.fastq
- Copy Illumina adapter file 'NexteraPE-PE.fa' from Trimmomatic software folder to working directory with fastq files
- The working directory should look like:



1.1: Reformat read file names with given sample IDs (sample_ids.txt) – will combine all fastq files into single forward (demultiplexed_seqs_1.fq) and reverse (demultiplexed_seqs_2.fq) fastq files into new folder (fastq_with_sampleIDs)

./format_multifile_headers_uparse.py -i sample_ids.txt -o fastq_with_sampleIDs

1.2 Make directory 'trimmomatic-outputs'

mkdir trimmomatic-outputs

1.3 Trim and quality filter reformatted read files – specify that these are paired-end fastq files with 'PE', phred33 refers to format of base quality scores, specify name of log file (samples_trimlog.txt), input fastq files (demultiplexed_seqs_1.fq, demultiplexed_seqs_2.fq), output file names (forward_paired, forward_unpaired, reverse_paired, reverse_unpaired), Illumina adapter and quality filtering parameters (use NexteraPE-PE.fa adapter file), minimum length of reads kept is 36 bp

trimmomatic PE -phred33 -trimlog all_samples_trimlog.txt
fastq_with_sampleIDs/demultiplexed_seqs_1.fq fastq_with_sampleIDs/demultiplexed_seqs_2.fq
trimmomatic-outputs/trimmed_forward_paired trimmomatic-outputs/trimmed_forward_unpaired
trimmomatic-outputs/trimmed_reverse_paired trimmomatic-outputs/trimmed_reverse_unpaired
ILLUMINACLIP:NexteraPE-PE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15
MINLEN:36

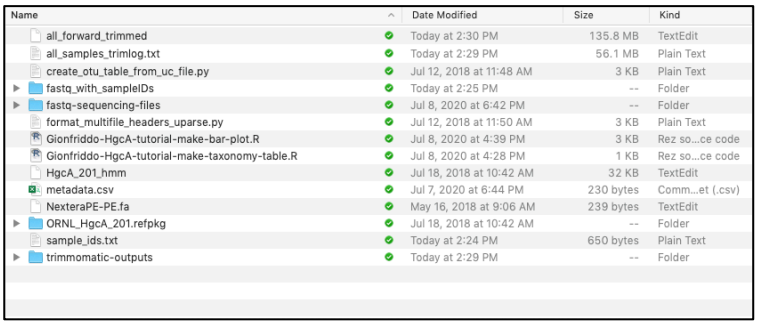
1.4 Since the read sequencing length (2x300bp) was insufficient to fully cover the target sequence length (~1 kbp), and there is no overlap between forward and reverse reads, only forward reads are used for further analysis. Here we combine the forward paired and

unpaired reads from the Trimmomatic outputs into one file (all_forward_trimmed) in our working directory

cat trimmomatic-outputs/trimmed_forward_paired trimmomatic-outputs/trimmed_forward_unpaired > all_forward_trimmed

Part 2: Truncate, dereplicate, sort by size, and remove unique read sequences prior to clustering reads into OTUs

- Make sure that the output of 1.4, 'all_forward_trimmed' is in your working directory
- Install Vsearch (version 2.7.0), Cd-hit (4.6.8), python script create_otu_table_from_uc_file.py
- Make python script executable:
  chmod +x create_otu_table_from_uc_file.py
- The working directory at the beginning of Part 2 should look like:



2.1 Make new output directory: 'vsearch-outputs'

mkdir vsearch-outputs

2.2 Truncate reads to same length (201 bp), add suffix '_201' to each read identifier, and get rid of reads that are too short, output is fasta file 'all_forward_trimmed_201.fa'

vsearch --fastx_filter all_forward_trimmed --fastq_trunclen 201 --label_suffix _201 --fastaout vsearch-outputs/all_forward_trimmed_201.fa

2.3 Dereplicate reads and pull out unique sequences. Relabel remaining sequences 'Uniq'

vsearch --derep_fulllength vsearch-outputs/all_forward_trimmed_201.fa --strand plus --output vsearch-outputs/all_forward_201_uniques.fa --sizeout --relabel Uniq --fasta_width 0

2.4 Sort by size (i.e. the number of replicates for each sequence) – remove singleton sequences by setting minimum to 2. Identifying unique sequences and removing singleton and replicate sequences helps limit the computational power needed for clustering step.

vsearch --derep_fulllength vsearch-outputs/all_forward_201_uniques.fa --minuniquesize 2 --sizein --sizeout --output vsearch-outputs/all_forward_201_uniques_seqs_sorted.fa --fasta_width 0

2.5 Precluster at 98% prior to Chimera detection – output will be a hits table in uclust format (all.preclustered.uc) and fasta file of centroid representative sequences (all.preclustered.fasta)

vsearch --cluster_size vsearch-outputs/all_forward_201_uniques_seqs_sorted.fa --id 0.98 --strand plus --sizein --sizeout --fasta_width 0 --uc vsearch-outputs/all.preclustered.uc --centroids vsearch-outputs/all.preclustered.fasta

2.6 Search 98% centroid sequences (all.preclustered.fasta) for chimeras using denovo chimera detection – non-chimeric sequences are then saved to fasta file (all.denovo.nonchimeras.fasta)

vsearch --uchime_denovo vsearch-outputs/all.preclustered.fasta --sizein --sizeout --fasta_width 0 --nonchimeras vsearch-outputs/all.denovo.nonchimeras.fasta

2.7 Use cd-hit-est to cluster unique non-chimeric, non-singleton sequences at 90% identity – output will be a fasta file of representative sequences (all.denovo.nonchimeras_cluster90)

cd-hit-est -i vsearch-outputs/all.denovo.nonchimeras.fasta -o vsearch-outputs/all.denovo.nonchimeras_cluster90 -c 0.90 -n 4

2.8 Map original trimmed reads (all_forward_trimmed_201.fa) to representative sequence database (all.denovo.nonchimeras_cluster90) at 90% identity to create hits table in uclust format (all_forward_nonchimeras_cluster90_table.uc). Specify that sequences are in the forward direction with 'strand plus'. Save OTU table to main working directory.

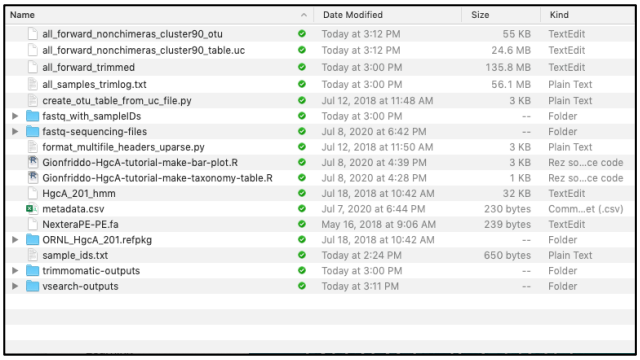
vsearch --usearch_global vsearch-outputs/all_forward_trimmed_201.fa --db vsearch-outputs/all.denovo.nonchimeras_cluster90 --id 0.9 --uc all_forward_nonchimeras_cluster90_table.uc --strand plus

2.9 Convert OTU uclust table to tsv format that can be opened in Excel

./create_otu_table_from_uc_file.py -i all_forward_nonchimeras_cluster90_table.uc -o all_forward_nonchimeras_cluster90_otu

Part 3: Quality filtering centroid representative sequences prior to classification

- Install Emboss (6.5.7), Hmmer (version 3.2)
- Make sure reference package (ORNL_HgcA_201.refpkg) is in working directory
- Make sure hmm reference alignment (HgcA_201_hmm) from reference package (ORNL_HgcA_201.refpkg/ HgcA_201_hmm) is in working directory
- The working directory at the beginning of Part 3 should look like:



3.1 Make new output directory for hmm-filtering steps: 'hmm-outputs'

mkdir hmm-outputs

3.2 Translate centroid representative sequences from cd-hit-est output (step 2.6, all.denovo.nonchimeras_cluster90) from nucleotide to amino acid sequences using 'transeq' from Emboss package

transeq vsearch-outputs/all.denovo.nonchimeras_cluster90 hmm-outputs/all.denovo.nonchimeras_cluster90_aa

3.3 To filter out centroid sequences that are likely not HgcA or do not begin with the cap-helix region (i.e. the reading frame does not begin on the first base), remove sequences with stop codons by deleting sequences that contain '*'

awk '/^>/ {printf("%s%s\t",(N>0?"\n":""),$0);N++;next;} {printf("%s",$0);} END {printf("\n");}' hmm-outputs/all.denovo.nonchimeras_cluster90_aa | awk -F '\t' '!($2 ~ /*/)' | tr "\t" "\n" > hmm-outputs/all.denovo.nonchimeras_cluster90_aa_filtered.fa

3.4 Search filtered centroid sequences (all.denovo.nonchimeras_cluster90_aa_filtered.fa) for HgcA sequences using reference HgcA hmm-profile (HgcA_201_hmm) using hmmsearch. Filter out reads that do not align with reference HgcA sequences using an inclusion E-value cutoff of 1E-7. Output will be a table of reads and E values (all.denovo.nonchimeras_cluster90_aa_filtered_hmm_table), text file showing alignment of all centroid sequences with reference HgcA and E-values (all.denovo.nonchimeras_cluster90_aa_filtered_hmm_output) and alignment of centroid sequences that passed inclusion threshold with reference sequences (all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment)

hmmsearch --tblout hmm-outputs/all.denovo.nonchimeras_cluster90_aa_filtered_hmm_table -o hmm-outputs/all.denovo.nonchimeras_cluster90_aa_filtered_hmm_output --incE 1E-7 -A hmm-outputs/all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment HgcA_201_hmm hmm-outputs/all.denovo.nonchimeras_cluster90_aa_filtered.fa

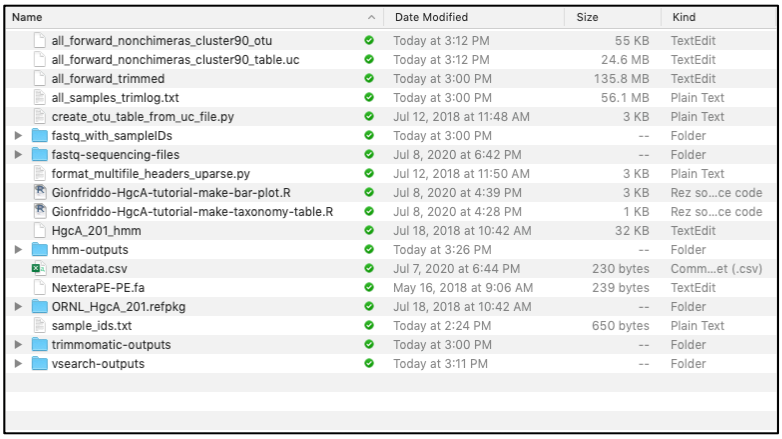
3.5 Align filtered centroid sequences from step 3.2 (all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment) to stockhold formatted alignment of reference sequences in reference package (ORNL_HgcA_201.refpkg/aa_201bp_ref_alignment_stockholm.stockholm) using hmm model (HgcA_201_hmm) produing a Stockholm formatted alignment of filtered centroid sequences and reference

sequences (all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.sto) that can be used for classification

hmmalign -o hmm-outputs/all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.sto --mapali ORNL_HgcA_201.refpkg/aa_201bp_ref_alignment_stockholm.stockholm HgcA_201_hmm hmm-outputs/all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment

Part 4: Classify centroid HgcA sequences based on phylogenetic placement

- Install pplacer (v1.1.alpha17-6-g5cecf99)
- Make sure the ORNL HgcA reference package is in the into working directory (ORNL_HgcA_201.refpkg)
- The working directory at the beginning of Part 4 should look like:



Name	Date Modified	Size	Kind
all_forward_nonchimeras_cluster90_otu	Today at 3:12 PM	55 KB	TextEdit
all_forward_nonchimeras_cluster90_table.uc	Today at 3:12 PM	24.6 MB	TextEdit
all_forward_trimmed	Today at 3:00 PM	135.8 MB	TextEdit
all_samples_trimlog.txt	Today at 3:00 PM	56.1 MB	Plain Text
create_otu_table_from_uc_file.py	Jul 12, 2018 at 11:48 AM	3 KB	Plain Text
fastq_with_sampleIDs	Today at 3:00 PM	--	Folder
fastq-sequencing-files	Jul 8, 2020 at 6:42 PM	--	Folder
format_multifile_headers_uparse.py	Jul 12, 2018 at 11:50 AM	3 KB	Plain Text
Gionfriddo-HgcA-tutorial-make-bar-plot.R	Jul 8, 2020 at 4:39 PM	3 KB	Rez so...ce code
Gionfriddo-HgcA-tutorial-make-taxonomy-table.R	Jul 8, 2020 at 4:28 PM	1 KB	Rez so...ce code
HgcA_201_hmm	Jul 18, 2018 at 10:42 AM	32 KB	TextEdit
hmm-outputs	Today at 3:26 PM	--	Folder
metadata.csv	Jul 7, 2020 at 6:44 PM	230 bytes	Comm...et (.csv)
NexteraPE-PE.fa	May 16, 2018 at 9:06 AM	239 bytes	TextEdit
ORNL_HgcA_201.refpkg	Jul 18, 2018 at 10:42 AM	--	Folder
sample_ids.txt	Today at 2:24 PM	650 bytes	Plain Text
trimmomatic-outputs	Today at 3:00 PM	--	Folder
vsearch-outputs	Today at 3:11 PM	--	Folder

4.1 Place aligned centroid sequences (all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.sto) onto maximum likelihood reference HgcA tree in reference package (HgcA_201.refpkg). Specify to calculate posterior probabilities based on alignment of query sequence with reference sequence (-p), specify that the maximum number of placements to keep is one (--keep-at-most 1), set the maximum branch length to 1 (--max-pend 1) (this ensures that sequences that are highly dissimilar to HgcA are not placed on the tree – this is the last filtering step for identifying non-HgcA sequences). Output will be a jplace file containing reads placed on max-like tree (all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.jplace)

pplacer -p --keep-at-most 1 --max-pend 1 -c ORNL_HgcA_201.refpkg/ hmm-outputs/all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.sto

4.2 Make a sqlite database for classifications (all_forward_nonchimeric_90_201_classify)

rppr prep_db --sqlite all_forward_nonchimeric_90_201_classify -c ORNL_HgcA_201.refpkg/

4.3 Assign taxonomy to reads based on where the query sequences have been placed on reference tree (all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.jplace), uses posterior probability (--pp) and lowest common ancestor of branch to classify, and uses a 90% confidence cut-off for identification as default. Writes classifications to the sqlite database made in step 4.2 (all_forward_nonchimeric_90_201_classify).

```
guppy classify -c ORNL_HgcA_201.refpkg/ --pp --sqlite
all_forward_nonchimeric_90_201_classify
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.jplace
```

4.4 To write guppy classifications to csv file:

```
guppy to_csv --point-mass --pp -o all.denovo.nonchimeras_cluster90_classifications.csv
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.jplace
```

4.5 To make a visualization showing placements on reference tree:

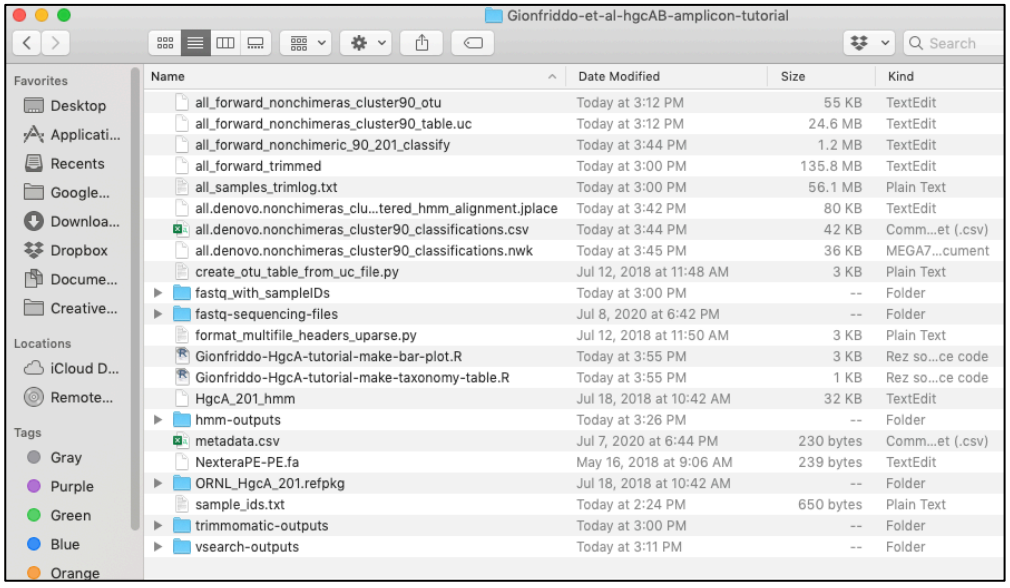
```
guppy tog --pp -o all.denovo.nonchimeras_cluster90_classifications.nwk
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.jplace
```

4.6 The saved terminal output from Parts 1-4 can be found at the end of this document

Part 5: Visualizing the classifications and relative abundance of OTUs using R

The OTU table and pplacer objects are inputed into R in order to process the taxonomy classifications, calculate relative abundances of OTUs for each sample, and create a visualization of the data. To run the R scripts included in this tutorial (shown below, Gionfriddo-HgcA-tutorial-make-taxonomy-table.R and Gionfriddo-HgcA-tutorial-make-bar-plot.R), you will need to install R (version 3.5.1 or later), and the R libraries BoSSA, phyloseq, ggplot2, RColorBrewer.

Before running the scripts, double check that the working directory contains the pplacer outputs, the OTU table output (all_forward_nonchimeras_cluster90_otu), the metadata file for the mock community dataset (metadata.csv), and the reference package (ORNL_HgcA_201.refpkg). The working directory should look like this:



5.1 The R script 'Gionfriddo-HgcA-tutorial-make-taxonomy-table.R' assigns taxonomy to OTUs using the phylogenetic placements and classifications assigned using pplacer. Make sure the path to the working directory in the R script is indeed the path to the tutorial folder on your computer. Currently it is set to '~/Desktop/Gionfriddo-et-al-hgcAB-amplicon-tutorial'

Gionfriddo-HgcA-tutorial-make-taxonomy-table.R

```
#set working directory to folder with pplacer outputs
setwd("~/Desktop/Gionfriddo-et-al-hgcAB-amplicon-tutorial/")
#load programs BoSSA, phyloseq, ggplot2, and RColorBrewer
library(BoSSA)
library(phyloseq)
library(ggplot2)
library(RColorBrewer)
#set path to ORNL HgcA reference package that has tax IDs for taxonomy assigned to hgcA OTUs by pplacer
refpkg_path <- file.path("ORNL_HgcA_201.refpkg/")
taxo <- refpkg(refpkg_path,type="taxonomy")
#input classification database (sqlite formatted file, pplacer object: "all_forward_nonchimeric_90_201_classify") and jplace file with phylogenetic
 placements of hgcA OTUs ("all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.jplace")
sqlite_file <- file.path("all_forward_nonchimeric_90_201_classify")
jplace_file <- file.path("all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.jplace")
pplace_object <- read_sqlite(sqlite_file,jplace_file)
taxo_table <- pplace_to_taxonomy(pplace_object,taxo,tax_name=TRUE,rank=c("phylum","class","order","family","genus","species"))
write.csv(taxo_table,"taxo_table.csv")
```

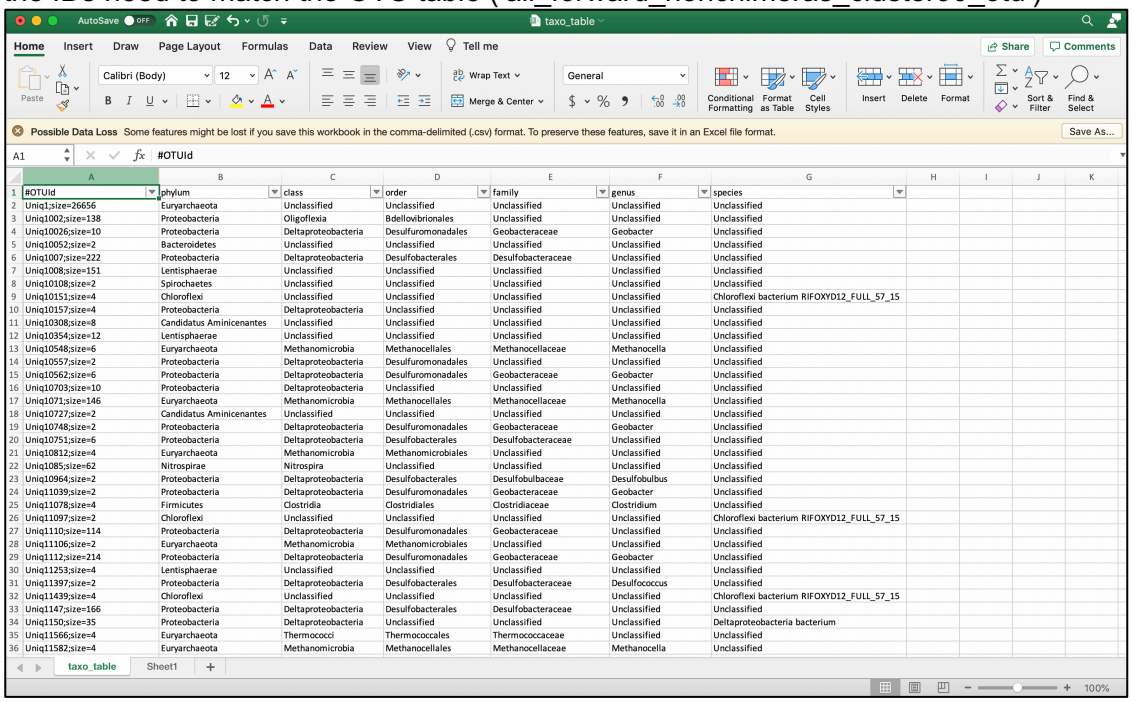
5.2 The output from the R script will be a table, 'taxo_table.csv', with OTU Ids, and taxonomy information. Label the first column '#OTUId'

taxo_table

Possible Data Loss Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

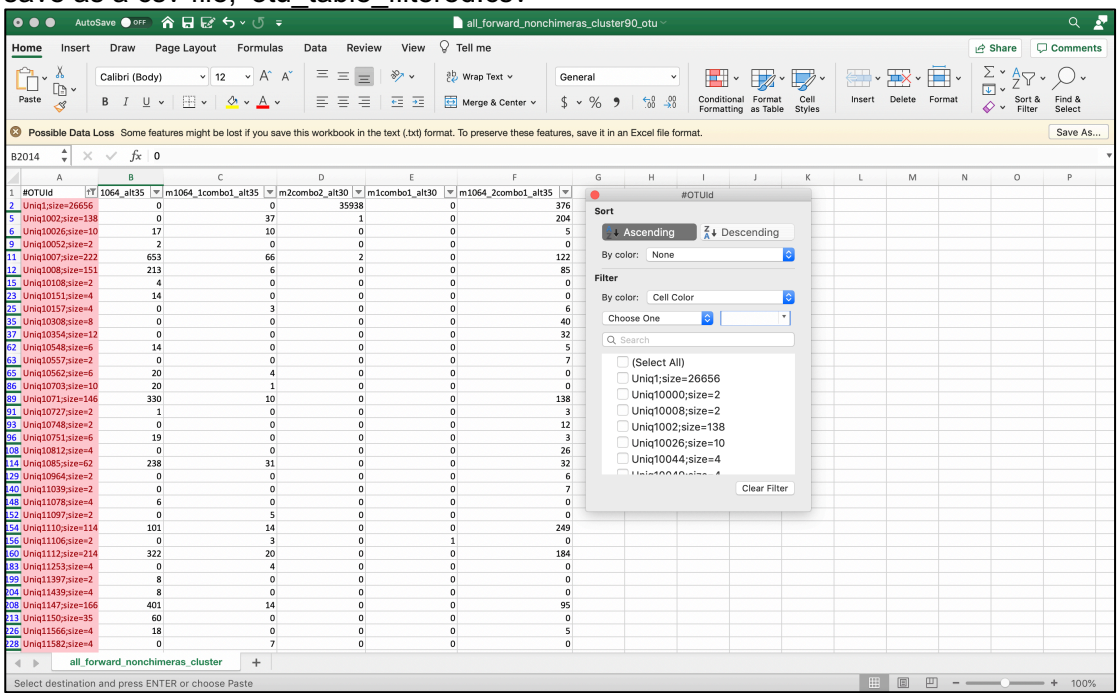
#OTUId	phylum	class	order	family	genus	species
Uniq1;size=26656_1/1-67	Euryarchaeota	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq1002;size=138_1/1-67	Proteobacteria	Oligoflexia	Bdellovibrionales	Unclassified	Unclassified	Unclassified
Uniq10026;size=10_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Geobacter	Unclassified
Uniq10052;size=2_1/1-67	Bacteroidetes	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq1007;size=222_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfobacterales	Desulfobacteraceae	Unclassified	Unclassified
Uniq1008;size=151_1/1-67	Lentisphaerae	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10108;size=2_1/1-67	Spirochaetes	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10151;size=4_1/1-67	Chloroflexi	Unclassified	Unclassified	Unclassified	Unclassified	Chloroflexi bacterium RIFOXYD12_FULL_57_15
Uniq10157;size=4_1/1-67	Proteobacteria	Deltaproteobacteria	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10308;size=8_1/1-67	Candidatus Aminicenantes	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10354;size=12_1/1-62	Lentisphaerae	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10548;size=6_1/1-67	Euryarchaeota	Methanomicrobia	Methanocellales	Methanocellaceae	Methanocella	Unclassified
Uniq10557;size=2_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Unclassified	Unclassified	Unclassified
Uniq10562;size=6_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Geobacter	Unclassified
Uniq10703;size=10_1/1-67	Proteobacteria	Deltaproteobacteria	Unclassified	Unclassified	Unclassified	Unclassified
Uniq1071;size=146_1/1-67	Euryarchaeota	Methanomicrobia	Methanocellales	Methanocellaceae	Methanocella	Unclassified
Uniq10727;size=2_1/1-67	Candidatus Aminicenantes	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10748;size=2_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Geobacter	Unclassified
Uniq10751;size=6_1/7-50	Proteobacteria	Deltaproteobacteria	Desulfobacterales	Desulfobacteraceae	Unclassified	Unclassified
Uniq10812;size=4_1/1-67	Euryarchaeota	Methanomicrobia	Methanomicrobiales	Unclassified	Unclassified	Unclassified
Uniq1085;size=62_1/1-67	Nitrospirae	Nitrospira	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10964;size=2_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfobacterales	Desulfobulbaceae	Desulfobulbus	Unclassified
Uniq11039;size=2_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Geobacter	Unclassified
Uniq11078;size=4_1/1-67	Firmicutes	Clostridia	Clostridiales	Clostridiaceae	Clostridium	Unclassified
Uniq1097;size=2_1/1-67	Chloroflexi	Unclassified	Unclassified	Unclassified	Unclassified	Chloroflexi bacterium RIFOXYD12_FULL_57_15
Uniq1110;size=114_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Unclassified	Unclassified
Uniq11106;size=2_1/11-67	Euryarchaeota	Methanomicrobia	Methanomicrobiales	Unclassified	Unclassified	Unclassified
Uniq1112;size=214_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Geobacter	Unclassified
Uniq11253;size=4_1/1-67	Lentisphaerae	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq11397;size=2_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfobacterales	Desulfobacteraceae	Desulfococcus	Unclassified
Uniq11439;size=4_1/1-67	Chloroflexi	Unclassified	Unclassified	Unclassified	Unclassified	Chloroflexi bacterium RIFOXYD12_FULL_57_15
Uniq1147;size=166_1/1-67	Proteobacteria	Deltaproteobacteria	Desulfobacterales	Desulfobacteraceae	Unclassified	Unclassified
Uniq1150;size=35_1/1-67	Proteobacteria	Deltaproteobacteria	Unclassified	Unclassified	Unclassified	Deltaproteobacteria bacterium
Uniq11566;size=4_1/1-67	Euryarchaeota	Thermococci	Thermococcales	Thermococcaceae	Unclassified	Unclassified
Uniq11582;size=4_1/1-67	Euryarchaeota	Methanomicrobia	Methanocellales	Methanocellaceae	Methanocella	Unclassified

5.3 Edit the OTU IDs to remove the '_1/1-67', '_1/1-62', etc. at the end of the IDs. The format of the IDs need to match the OTU table ('all_forward_nonchimeras_cluster90_otu')



#OTUId	phylum	class	order	family	genus	species
Uniq1;size=26656	Euryarchaeota	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq1002;size=138	Proteobacteria	Oligoflexia	Bdellovibrionales	Unclassified	Unclassified	Unclassified
Uniq10026;size=10	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Geobacter	Unclassified
Uniq10052;size=2	Bacteroidetes	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq1007;size=222	Proteobacteria	Deltaproteobacteria	Desulfobacterales	Desulfobacteraceae	Unclassified	Unclassified
Uniq1008;size=151	Lentisphaerae	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10108;size=2	Spirochaetes	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10151;size=4	Chloroflexi	Unclassified	Unclassified	Unclassified	Unclassified	Chloroflexi bacterium RIFOXYD12_FULL_57_15
Uniq10157;size=4	Proteobacteria	Deltaproteobacteria	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10308;size=8	Candidatus Aminicenantes	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10354;size=12	Lentisphaerae	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10548;size=6	Euryarchaeota	Methanomicrobia	Methanocellales	Methanocellaceae	Methanocella	Unclassified
Uniq10557;size=2	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Unclassified	Unclassified	Unclassified
Uniq10562;size=6	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Geobacter	Unclassified
Uniq10703;size=10	Proteobacteria	Deltaproteobacteria	Unclassified	Unclassified	Unclassified	Unclassified
Uniq1071;size=146	Euryarchaeota	Methanomicrobia	Methanocellales	Methanocellaceae	Methanocella	Unclassified
Uniq10727;size=2	Candidatus Aminicenantes	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10748;size=2	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Geobacter	Unclassified
Uniq10751;size=6	Proteobacteria	Deltaproteobacteria	Desulfobacterales	Desulfobacteraceae	Unclassified	Unclassified
Uniq10812;size=4	Euryarchaeota	Methanomicrobia	Methanomicrobiales	Unclassified	Unclassified	Unclassified
Uniq1085;size=62	Nitrospirae	Nitrospira	Unclassified	Unclassified	Unclassified	Unclassified
Uniq10964;size=2	Proteobacteria	Deltaproteobacteria	Desulfobacterales	Desulfobulbaceae	Desulfobulbus	Unclassified
Uniq11039;size=2	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Geobacter	Unclassified
Uniq11078;size=4	Firmicutes	Clostridia	Clostridiales	Clostridiaceae	Clostridium	Unclassified
Uniq1097;size=2	Chloroflexi	Unclassified	Unclassified	Unclassified	Unclassified	Chloroflexi bacterium RIFOXYD12_FULL_57_15
Uniq1110;size=114	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Unclassified	Unclassified
Uniq11106;size=2	Euryarchaeota	Methanomicrobia	Methanomicrobiales	Unclassified	Unclassified	Unclassified
Uniq1112;size=214	Proteobacteria	Deltaproteobacteria	Desulfuromonadales	Geobacteraceae	Geobacter	Unclassified
Uniq11253;size=4	Lentisphaerae	Unclassified	Unclassified	Unclassified	Unclassified	Unclassified
Uniq11397;size=2	Proteobacteria	Deltaproteobacteria	Desulfobacterales	Desulfobacteraceae	Desulfococcus	Unclassified
Uniq11439;size=4	Chloroflexi	Unclassified	Unclassified	Unclassified	Unclassified	Chloroflexi bacterium RIFOXYD12_FULL_57_15
Uniq1147;size=166	Proteobacteria	Deltaproteobacteria	Desulfobacterales	Desulfobacteraceae	Unclassified	Unclassified
Uniq1150;size=35	Proteobacteria	Deltaproteobacteria	Unclassified	Unclassified	Unclassified	Deltaproteobacteria bacterium
Uniq11566;size=4	Euryarchaeota	Thermococci	Thermococcales	Thermococcaceae	Unclassified	Unclassified
Uniq11582;size=4	Euryarchaeota	Methanomicrobia	Methanocellales	Methanocellaceae	Methanocella	Unclassified

5.4 Using the OTU IDs from the taxonomy table, filter the OTU table 'all_forward_nonchimeras_cluster90_otu' for only those that passed classification guidelines and were identified as HgcA. Copy the OTU IDs from 'taxo_table.csv', and paste below the OTU IDs in the OTU table, then use 'highlight duplicate cells' conditional formatting in Excel. Then filter to view only highlighted cells, copy the filtered OTU table to new Excel spreadsheet, and save as a csv-file, 'otu_table_filtered.csv'



#OTUId	1064_alt35	m1064_1combo1_alt35	m2combo2_alt30	m1combo1_alt30	m1064_2combo1_alt35
Uniq1;size=26656	0	0	35938	0	376
Uniq1002;size=138	0	37	1	0	204
Uniq10026;size=10	17	10	0	0	5
Uniq10052;size=2	2	0	0	0	0
Uniq1007;size=222	653	66	2	0	122
Uniq1008;size=151	213	6	0	0	85
Uniq10108;size=2	4	0	0	0	0
Uniq10151;size=4	14	0	0	0	0
Uniq10157;size=4	0	3	0	0	6
Uniq10308;size=8	0	0	0	0	40
Uniq10354;size=12	0	0	0	0	32
Uniq10548;size=6	14	0	0	0	5
Uniq10557;size=2	0	0	0	0	7
Uniq10562;size=6	20	4	0	0	0
Uniq10703;size=10	20	1	0	0	0
Uniq1071;size=146	330	10	0	0	138
Uniq10727;size=2	1	0	0	0	3
Uniq10748;size=2	0	0	0	0	12
Uniq10751;size=6	19	0	0	0	3
Uniq10812;size=4	0	0	0	0	26
Uniq1085;size=62	238	31	0	0	32
Uniq10964;size=2	0	0	0	0	6
Uniq11039;size=2	0	0	0	0	7
Uniq11078;size=4	6	0	0	0	0
Uniq11097;size=2	0	5	0	0	0
Uniq1110;size=114	101	14	0	0	249
Uniq11106;size=2	0	3	0	1	0
Uniq1112;size=214	322	20	0	0	184
Uniq11253;size=4	0	4	0	0	0
Uniq11397;size=2	8	0	0	0	0
Uniq11439;size=4	8	0	0	0	0
Uniq1147;size=166	401	14	0	0	95
Uniq1150;size=35	60	0	0	0	0
Uniq11566;size=4	18	0	0	0	5
Uniq11582;size=4	0	7	0	0	0

5.5 The R script 'Gionfriddo-HgcA-tutorial-make-bar-plot.R' inputs the OTU table 'otu_table_filtered.csv' and taxonomy table 'taxo_table.csv', transforms the OTU counts as relative abundances for each sample, and creates a bar plot of the relative abundance of each OTU. The bar plot colors correspond to phylum and class of each OTU.

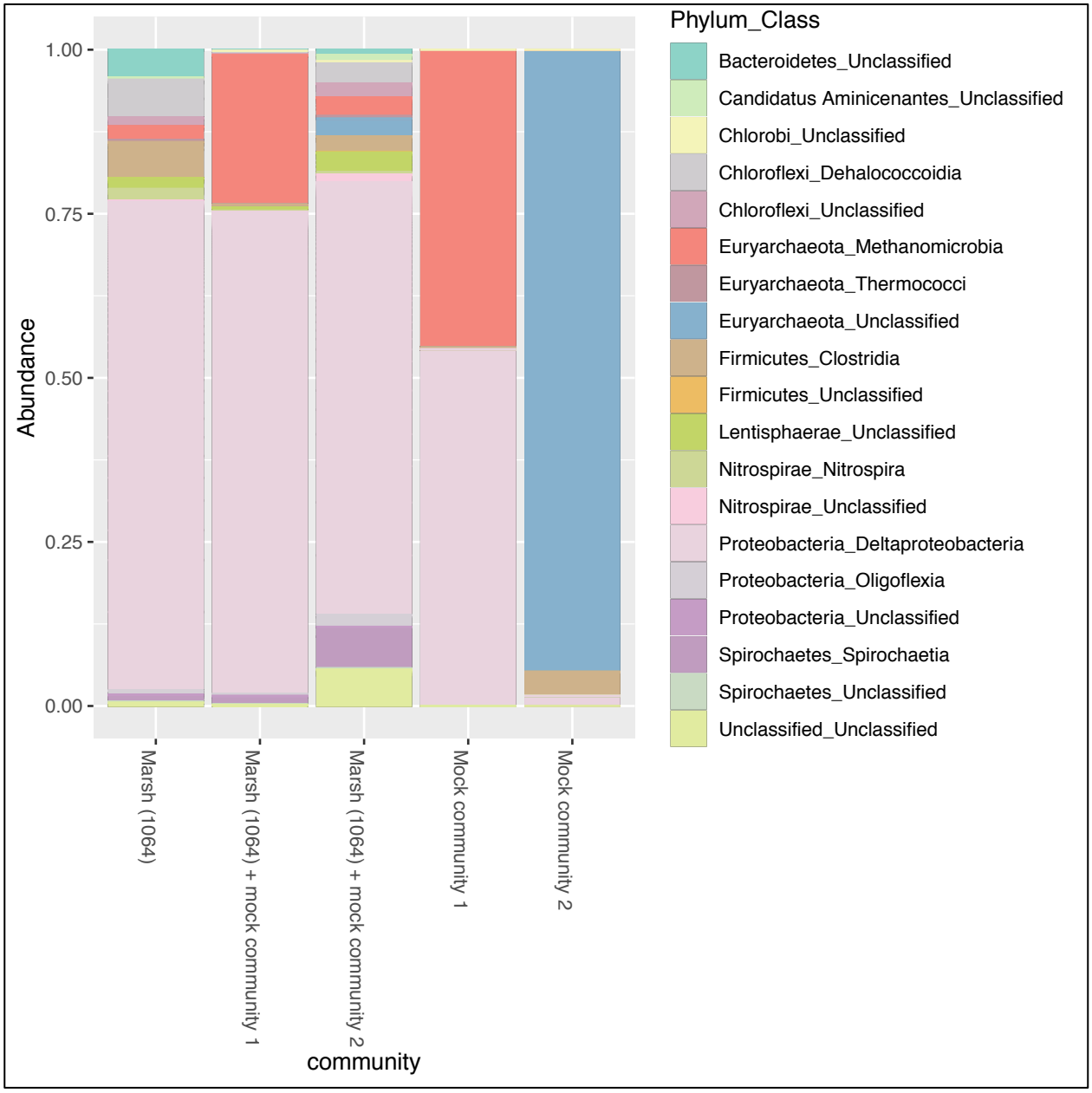
Gionfriddo-HgcA-tutorial-make-bar-plot.R

<functions> Help search

```
#set working directory to folder with pplacer outputs
setwd("~/Desktop/Gionfriddo-et-al-hgcAB-amplicon-tutorial/")
#load programs BoSSA, phyloseq, ggplot2, and RColorBrewer
library(BoSSA)
library(phyloseq)
library(ggplot2)
library(RColorBrewer)

#edit OTU Ids in "taxo_table.csv" to match OTU Ids in OTU table ("all_forward_nonchimeras_cluster90_otu") - manually remove "_1/1-67" part of ID that was added during the hmm alignment steps. Save new taxonomy table: "taxo_table_edited.csv"

#Then filter the OTU table for only OTUs that passed quality filtering steps and were classified as hgcA. In Excel I copy OTU Ids from taxonomy table into #OTUId column, then use "highlight duplicate cells", and filtering to show only OTUIds that have been classified. Then copy to new document, and save as "otu_table_filtered.csv".

#order OTU Ids in both "otu_table_filtered.csv" and "taxo_table_edited.csv" so that the order matches in both

#import the OTU table and taxonomy table into R then convert to matrix
otu_table <- read.csv("otu_table_filtered.csv",header=TRUE,row.names=1, sep=",")
tax_table <- read.csv("taxo_table.csv",header=TRUE,row.names=1, sep=",")
otu_matrix <- as.matrix(otu_table)
tax_matrix <- as.matrix(tax_table)

#import table with metadata, including mock community names
metadata <- read.csv("metadata.csv",header=TRUE,row.names=1, sep=",")


#make phyloseq object with OTU table and taxonomy (i.e. TAX) table
OTU = otu_table(otu_matrix, taxa_are_rows=TRUE)
TAX = tax_table(tax_matrix)
phyloseq_obj = phyloseq(OTU,TAX)

#add sample metadata and create new phyloseq object
community = metadata[1:5,1]
sampledata = sample_data(data.frame(community, row.names=sample_names(phyloseq_obj)))
phyloseq_obj_meta = merge_phyloseq(phyloseq_obj, sampledata)

#transform OTU counts by calculating relative abundance
phyloseq_object_meta_RA = transform_sample_counts(phyloseq_obj_meta, function(x)x/sum(x))

#combine phyla and class taxonomy into one object: Phylum_Class
myranks = c("phylum", "class")
phyloseq_object_meta_RA_tax = prune_taxa(tail(names(sort(taxa_sums(phyloseq_object_meta_RA))), 299), phyloseq_object_meta_RA)
tax_table(phyloseq_object_meta_RA_tax) <- cbind(tax_table(phyloseq_object_meta_RA_tax), Strain=taxa_names(phyloseq_object_meta_RA_tax))
mylabels = apply(tax_table(phyloseq_object_meta_RA_tax)[, myranks], 1, paste, sep="", collapse="_")
tax_table(phyloseq_object_meta_RA_tax) <- cbind(tax_table(phyloseq_object_meta_RA_tax), Phylum_Class=mylabels)
phyloseq_object_meta_RA_tax_merge = tax_glom(phyloseq_object_meta_RA_tax, "Phylum_Class")

#plot community composition as bar chart, set color palette for 20 different variables (colorCount = 20) with RColorBrewer and getPalette function. Save plot as pdf
getPalette = colorRampPalette(brewer.pal(12,"Set3"))
colourCount=20
pdf("mock_community_bar_plot_RA.pdf")
print(plot_bar(phyloseq_object_meta_RA_tax_merge,"community",fill="Phylum_Class") + geom_bar(aes(color=Phylum_Class,fill=Phylum_Class), stat="identity", position="stack") + scale_fill_manual(values = getPalette(colourCount)) + scale_color_manual(values = getPalette(colourCount)))
dev.off()

#end of script
```

setwd(dir)

5.5 The output of the R script is a bar plot (mock_community_bar_plot_RA.pdf) of the relative abundance of HgcA OTUs for the Marsh community (1064), two mock communities, and marsh sample spiked with the two marsh communities. The bars are colored by the phylum and class assigned to each OTU.



Saved Terminal Output from Parts 1-4 of this tutorial:

```
Last login: Mon Jul 13 17:09:47 on ttys000
mac109979:~ cg0$ /Users/cg0/.anaconda/navigator/a.tool ; exit;
(bio) bash-3.2$ cd Desktop/Gionfriddo-et-al-hgcAB-amplicon-tutorial/
(bio) bash-3.2$ ls
Gionfriddo-HgcA-tutorial-make-bar-plot.R
Gionfriddo-HgcA-tutorial-make-taxonomy-table.R
HgcA_201_hmm
NexteraPE-PE.fa
ORNL_HgcA_201.refpkg
create_otu_table_from_uc_file.py
fastq-sequencing-files
format_multifile_headers_uparse.py
metadata.csv
mock_community_bar_plot_RA.pdf
(bio) bash-3.2$ ./format_multifile_headers_uparse.py -i sample_ids.txt
-o fastq_with_sampleIDs
Reformatting m1combo1_alt30...
Reformatting m2combo2_alt30...
Reformatting m1064_1combo1_alt35...
Reformatting m1064_2combo1_alt35...
Reformatting 1064_alt35...
(bio) bash-3.2$ mkdir trimmomatic-outputs
(bio) bash-3.2$ trimmomatic PE -phred33 -trimlog
all_samples_trimlog.txt fastq_with_sampleIDs/demultiplexed_seqs_1.fq
fastq_with_sampleIDs/demultiplexed_seqs_2.fq trimmomatic-outputs/
trimmed_forward_paired trimmomatic-outputs/trimmed_forward_unpaired
trimmomatic-outputs/trimmed_reverse_paired trimmomatic-outputs/
trimmed_reverse_unpaired ILLUMINACLIP:NexteraPE-PE.fa:2:30:10 LEADING:
3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
TrimmomaticPE: Started with arguments:
 -phred33 -trimlog all_samples_trimlog.txt fastq_with_sampleIDs/
demultiplexed_seqs_1.fq fastq_with_sampleIDs/demultiplexed_seqs_2.fq
trimmomatic-outputs/trimmed_forward_paired trimmomatic-outputs/
trimmed_forward_unpaired trimmomatic-outputs/trimmed_reverse_paired
trimmomatic-outputs/trimmed_reverse_unpaired ILLUMINACLIP:NexteraPE-
PE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36
Multiple cores found: Using 4 threads
Using PrefixPair: 'AGATGTGTATAAGAGACAG' and 'AGATGTGTATAAGAGACAG'
Using Long Clipping Sequence: 'GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAG'
Using Long Clipping Sequence: 'TCGTCGGCAGCGTCAGATGTGTATAAGAGACAG'
Using Long Clipping Sequence: 'CTGTCTCTTATACACATCTCCGAGCCCACGAGAC'
Using Long Clipping Sequence: 'CTGTCTCTTATACACATCTGACGCTGCCGACGA'
ILLUMINACLIP: Using 1 prefix pairs, 4 forward/reverse sequences, 0
forward only sequences, 0 reverse only sequences
Input Read Pairs: 252173 Both Surviving: 201887 (80.06%) Forward Only
Surviving: 39934 (15.84%) Reverse Only Surviving: 3246 (1.29%)
Dropped: 7106 (2.82%)
TrimmomaticPE: Completed successfully
(bio) bash-3.2$ cat trimmomatic-outputs/trimmed_forward_paired
trimmomatic-outputs/trimmed_forward_unpaired > all_forward_trimmed
```

```
(bio) bash-3.2$ mkdir vsearch-outputs
(bio) bash-3.2$ vsearch --fastx_filter all_forward_trimmed --
fastq_trunclen 201 --label_suffix _201 --fastaout vsearch-outputs/
all_forward_trimmed_201.fa
vsearch v2.7.0_macos_x86_64, 16.0GB RAM, 8 cores
https://github.com/torognes/vsearch

Reading input file 100%
174211 sequences kept (of which 173618 truncated), 67610 sequences
discarded.
(bio) bash-3.2$ vsearch --derep_fulllength vsearch-outputs/
all_forward_trimmed_201.fa --strand plus --output vsearch-outputs/
all_forward_201_uniques.fa --sizeout --relabel Uniq --fasta_width 0
vsearch v2.7.0_macos_x86_64, 16.0GB RAM, 8 cores
https://github.com/torognes/vsearch

Reading file vsearch-outputs/all_forward_trimmed_201.fa 100%
35016411 nt in 174211 seqs, min 201, max 201, avg 201
Dereplicating 100%
Sorting 100%
73806 unique sequences, avg cluster 2.4, median 1, max 1481
Writing output file 100%
(bio) bash-3.2$ vsearch --derep_fulllength vsearch-outputs/
all_forward_201_uniques.fa --minuniquesize 2 --sizein --sizeout --
output vsearch-outputs/all_forward_201_uniques_seqs_sorted.fa --
fasta_width 0
vsearch v2.7.0_macos_x86_64, 16.0GB RAM, 8 cores
https://github.com/torognes/vsearch

Reading file vsearch-outputs/all_forward_201_uniques.fa 100%
14835006 nt in 73806 seqs, min 201, max 201, avg 201
Dereplicating 100%
Sorting 100%
73806 unique sequences, avg cluster 2.4, median 1, max 1481
Writing output file 100%
19943 uniques written, 53863 clusters discarded (73.0%)
(bio) bash-3.2$ vsearch --cluster_size vsearch-outputs/
all_forward_201_uniques_seqs_sorted.fa --id 0.98 --strand plus --
sizein --sizeout --fasta_width 0 --uc vsearch-outputs/
all.preclustered.uc --centroids vsearch-outputs/all.preclustered.fasta
vsearch v2.7.0_macos_x86_64, 16.0GB RAM, 8 cores
https://github.com/torognes/vsearch

Reading file vsearch-outputs/all_forward_201_uniques_seqs_sorted.fa
100%
4008543 nt in 19943 seqs, min 201, max 201, avg 201
Masking 100%
Sorting by abundance 100%
Counting k-mers 100%
Clustering 100%
```

```
Max number of word counting entries: 98348874

comparing sequences from          0  to       3005
...
     3005  finished       2023  clusters

Apprixmated maximum memory consumption: 15M
writing new database
writing clustering information
program completed !

Total CPU time 0.70
(bio) bash-3.2$ vsearch --usearch_global vsearch-outputs/
all_forward_trimmed_201.fa --db vsearch-outputs/
all.denovo.nonchimeras_cluster90 --id 0.9 --uc
all_forward_nonchimeras_cluster90_table.uc --strand plus
vsearch v2.7.0_macos_x86_64, 16.0GB RAM, 8 cores
https://github.com/torognes/vsearch

Reading file vsearch-outputs/all.denovo.nonchimeras_cluster90 100%
406623 nt in 2023 seqs, min 201, max 201, avg 201
Masking 100%
Counting k-mers 100%
Creating k-mer index 100%
Searching 100%
Matching query sequences: 171680 of 174211 (98.55%)
(bio) bash-3.2$ ./create_otu_table_from_uc_file.py -i
all_forward_nonchimeras_cluster90_table.uc -o
all_forward_nonchimeras_cluster90_otu
100.0%
Writing table...
(bio) bash-3.2$ mkdir hmm-outputs
(bio) bash-3.2$ transeq vsearch-outputs/
all.denovo.nonchimeras_cluster90 hmm-outputs/
all.denovo.nonchimeras_cluster90_aa
Translate nucleic acid sequences
(bio) bash-3.2$ awk '/^>/ {printf("%s%s\t",(N>0?"\n":""),$0);N+
+;next;} {printf("%s",$0);} END {printf("\n");}' hmm-outputs/
all.denovo.nonchimeras_cluster90_aa | awk -F '\t'  '!($2 ~ /\*/)' | tr
"\t" "\n" > hmm-outputs/
all.denovo.nonchimeras_cluster90_aa_filtered.fa
(bio) bash-3.2$ hmmsearch --tblout hmm-outputs/
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_table -o hmm-outputs/
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_output --incE 1E-7 -A
hmm-outputs/all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment
HgcA_201_hmm hmm-outputs/
all.denovo.nonchimeras_cluster90_aa_filtered.fa
(bio) bash-3.2$ hmmalign -o hmm-outputs/
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.sto --
mapali ORNL_HgcA_201.refpkg/aa_201bp_ref_alignment_stockholm.stockholm
```

```
HgcA_201_hmm hmm-outputs/
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment
(bio) bash-3.2$ pplacer -p --keep-at-most 1 --max-pend 1 -c
ORNL_HgcA_201.refpkg/ hmm-outputs/
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.sto
Running pplacer v1.1.alpha17-6-g5cecf99 analysis on hmm-outputs/
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.sto...
Found reference sequences in given alignment file. Using those for
reference alignment.
Pre-masking sequences... sequence length cut from 118 to 67.
Determining figs... figs disabled.
Allocating memory for internal nodes... done.
Caching likelihood information on reference tree... done.
Pulling exponents... done.
Preparing the edges for baseball... done.
working on Uniq4422;size=14_1/1-67 (299/299)...
(bio) bash-3.2$ rppr prep_db --sqlite
all_forward_nonchimeric_90_201_classify -c ORNL_HgcA_201.refpkg/
(bio) bash-3.2$ guppy classify -c ORNL_HgcA_201.refpkg/ --pp --sqlite
all_forward_nonchimeric_90_201_classify
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.jplace
(bio) bash-3.2$ guppy to_csv --point-mass --pp -o
all.denovo.nonchimeras_cluster90_classifications.csv
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.jplace
(bio) bash-3.2$ guppy tog --pp -o
all.denovo.nonchimeras_cluster90_classifications.nwk
all.denovo.nonchimeras_cluster90_aa_filtered_hmm_alignment.jplace
(bio) bash-3.2$
```