AdditionalFile17*Dineretal2021*Qiime2analyses.ipynb

Manuscript: Diner et al. 2021, Temperature and salinity drive pathogenic vibrio distributions and associations with plankton microbiomes

Contents: Code used to analyze 16S, 18S, and HSP60 amplicon sequences using QIIME2 version 2019.1. These commands are largely pulled from the Qiime2 "Moving Pictures" Tutorial (Most recent version: https://docs.qiime2.org/2020.2/tutorials/moving-pictures/). All amplicons were processed using the same basic code, so differences in processing between amplicons (e.g. databases used) are noted with the relevant commands.

some paths and path-specific files are not included to protect author privacy. The metadata file is included in the manuscript as additional file 5. Raw sequences are available via NCBI as noted in the manuscript.

Anaylsis of 16S, 18S, and HSP60 sequences using Qiime2 version 2019.4

Code written is primarily for the 16S analysis, and notes are made for where the 18S or HSP60 analysis differed

Importing paired-end sequences into Qiime2

qiime tools import --type 'SampleData[PairedEndSequencesWithQuality]' \ --input-path Combomanifest*RacheN*ibrio16S*pipeline* \ *--output-path vib16s*paired-end-demux.qza \ --input-format PairedEndFastqManifestPhred33

For HSP60, only used analysis from FWD reads, which were imported as single end sequences with quality data. As sequence quality was poor for many samples, they were trimmed using bbduk (https://jgi.doe.gov/data-and-tools/bbtools/bb-tools-user-guide/bbduk-guide/) prior to importing into

Qiime2. This is an example of the basic command used for the trimming:

bbduk.sh in1=SampleRead1.fastq in2=SampleRead2.fastq out1=forward*trimmed.fastq out2=reverse*trimmed.fastq ktrim=r k=20 hdist=1 minlen=200 ref= adapters.fa

qiime tools import \ --type 'SampleData[SequencesWithQuality]' \ --input-path manifest*bbduktrim*fwreads \ --output-path FWreads*bbduktrim*HSP60_demux.qza \ --input-format SingleEndFastqManifestPhred33

qiime demux summarize \ --i-data vib16s*paired-end-demux.qza* \ *--o-visualization vib16s*paired-end-demux.qzv

Denoising sequences using dada2

Note: trim is based on primer length, and truncation is based on quality of the sequences

16S analysis

qiime dada2 denoise-paired \ --i-demultiplexed-seqs vib16s*paired-end-demux.qza* \ *--p-trim-left-f 19* \ *--p-trim-left-r 20* \ *--p-trunc-len-f 260* \ *--p-trunc-len-r 190* \ *--p-n-threads 12* \ *--o-table Vibrio16S*table.qza \ --o-representative-sequences Vibrio16S-rep-seqs.qza \ --o-denoising-stats Vibrio16S-dada2stats.qza

18S analysis:

qiime dada2 denoise-paired \ --i-demultiplexed-seqs artifacts/vib18s*paired-end-demux.qza* \ *--p-trim-left-f 15* \ *--p-trim-left-r 15* \ *--p-trunc-len-f 139* \ *--p-trunc-len-r 138* \ *--p-n-threads 12* \ *--o-table Vibrio18S*table.qza \ --o-representative-sequences Vibrio18S-rep-seqs.qza \ --o-denoising-stats Vibrio18S-dada2stats-16S.qza

For HSP60 analysis:

qiime dada2 denoise-single \ --i-demultiplexed-seqs FWreads*bbduktrim*HSP60*demux.qza* \ *--p-trim-left 30* \ *--p-trunc-len 0* \ *--o-table FWreads*dada2default*table.qza* \ *--o-representative-sequences FWreads*dada2default*rep-seqs.qza* \ *--o-denoising-stats FWreads*dada2default.qza

Grouping duplicate samples together based on "replicate" metadata category

This step was omitted for HSP60 sequences as there was only a single sample with no replicates

qiime feature-table group \ --i-table Vibrio16S*table.qza* \ *--p-axis sample* \ *--m-metadata-file DEA2020*Vibrio*Metadata.txt* \ *--m-metadata-column "Replicate"* \ *--p-mode sum* \ *--o-grouped-table merged*Vibrio16S_table.qza

Generating visualization artifacts for analyses

qiime metadata tabulate \ --m-input-file Vibrio16S-dada2stats.qza \ --o-visualization Vibrio16S-dada2stats.qzv

qiime feature-table summarize \ --i-table merged*Vibrio16S*table.qza \ --o-visualization merged*Vibrio16S*table.qzv \ --m-sample-metadata-file DEA2020*Vibrio*Metadata.txt

qiime feature-table tabulate-seqs \ --i-data Vibrio16S-rep-seqs.qza \ --o-visualization Vibrio16S-rep-seqs.qzv

Generating rooted phylogenetic tree

qiime alignment mafft \ --i-sequences Vibrio16S-rep-seqs.qza \ --o-alignment aligned-Vibrio16S-rep-seqs.qza

qiime alignment mask \ --i-alignment aligned-Vibrio16S-rep-seqs.qza \ --o-masked-alignment masked-aligned_Vibrio16S-rep-seqs.qza

qiime phylogeny fasttree \ --i-alignment masked-aligned_Vibrio16S-rep-seqs.qza \ --o-tree Vibrio16S-unrooted-tree.qza

qiime phylogeny midpoint-root \ --i-tree Vibrio16S-unrooted-tree.qza \ --o-rooted-tree Vibrio16S-rooted-tree.qza

Training feature classifier and assigning taxonomy

16S: Used the SILVA database (https://www.arb-silva.de/)

16S Plastid sequences (not included in publication): Phytoref database (http://phytoref.sb-roscoff.fr/)

18S: Used PR2 database (https://pr2-database.org/)

HSP60: Used cpn60 database (http://www.cpndb.ca/) and taxonomy file provided by Kelsey Jesser as used in Jesser and Noble 2018.

qiime feature-classifier extract-reads \ --i-sequences silva13*299*qiime2.qza \ --p-f-primer GTGYCAGCMGCCGCGGTAA \ --p-r-primer CCGYCAATTYMTTTRAGTTT \ --o-reads SILVA*for*qiime2*515F*926R_RED.qza

qiime feature-classifier fit-classifier-naive-bayes \ --i-reference-reads SILVA*for*qiime2*515F*926R*RED.qza* \ *--i-reference-taxonomy silva132*99*taxonomy.qza* \ *--o-classifier SILVA*515F*926R*classifier_RED.qza

qiime feature-classifier classify-sklearn \ --i-classifier SILVA*515F*926R*classifier*RED.qza \ --i-reads Vibrio16S-rep-seqs.qza \ --o-classification Vibrio16S-taxonomy.qza

Generating taxonomy plots

qiime taxa barplot \ --i-table artifacts/merged*Vibrio16S*table.qza \ --i-taxonomy artifacts/Vibrio16S-taxonomy.qza \ --m-metadata-file Rachel*Vibrio*merged*Metadata*C2.txt \ --o-visualization artifacts/Vibrio16S-taxa-bar-plots-SILVA.qzv

***Exporting sequences so they can be imported into R for phyloseq analysisdownstream analyses* 4 things you need to import into phyloseq:** 1. OTU table - this is in a Qiime FeatureTable[Frequency] artifact and will need to be converted to a biom file first, then a .tsv/.txt that can import into R) 2. Taxonomy table - this is also an artifact, but can be exported directly to .txt) 3. Tree - can be rooted or unrooted. This is if you already generated them in Qiime2, otherwise you can generate them in R using ape. 4. Metadata file - this is already in a .txt format ready for R, so doesn't require any conversion.

(1) OTU Table

converting qiime otu table into biom

qiime tools export \ --input-path merged*Vibrio16S*table.qza \ --output-path Vibrio16S_OTUtable

convert from biom to tsv/txt (need to install biom for this step)

biom convert -i exports/Vibrio16S*OTUtable/Vibrio16S*OTUtable.biom -o exports/Vibrio16S_OTUtable.txt --to-tsv

Now you need to open the .txt file and change #OTU ID to OTUID

(2) Taxonomy table

Again this creates a directory with the taxonomy.tsv file in it.

qiime tools export \ --input-path Vibrio16S-taxonomy.qza \ --output-path Vibrio16S-taxonomy

(3) Tree, Can import rooted, unrooted, both, or generate a rooted from unrooted in phyloseq

qiime tools export \ --input-path Vibrio16S-rooted-tree.qza \ --output-path exports/exported-tree

qiime tools export \ --input-path Vibrio16S-unrooted-tree.qza \ --output-path exports/unrooted-exported-tree

Additional note: if you filtered out any sequences (chloroplasts, mitochondria, etc) then your

taxonomy and OTU tables are different lengths. QIIME2 doesn’t filter out taxonomy (unless you do it), so you have to merge the two files in R and output a merged file.