Rarification

s06b_rarefied_table.qza

s06a_alpha_rarefaction.qzv

FASTA QC

samples.txt

FASTQ's

FASTQC files

multiqc_report.html

FASTQ Trim

s03_pe_dmx.qza

s03_pe_dmx_trim.qza

s03_pe_dmx_trim.qzv

Feature Table

s04_table_dada2.qza

s04_table_dada2.qzv

feature table

s04_seqs_dada2.qza

s04_seqs_dada2.qzv

denoising-stats

s04_rep_seqs_dada2.qza

s04_rep_seqs_dada2.qzv

representative-sequences

Taxonomy

s10_taxonomy.qza

s10_taxonomy.qzv

model

gg-13-8-99-515-806-nb-classifier.qza

s10_taxa_bar_plot.qzv

Phylogeny

s05_aligned_rep_seqs.qza

s05_masked_aligned_rep_seqs.qza

aligned-dna-sequences.fasta

s05_unrooted_tree.qza

s05_rooted_tree.qza

tree.nwk

Alpha and Beta diversity

observed_features_vector.qza

observed_features_vector/alpha-diversity.tsv

s08_alpha_faith_pd_per_group.qzv

faith_pd_vector.qza

faith_pd_vector/alpha-diversity.tsv

s08_alpha_evenness_per_group.qzv

evenness_vector.qza

evenness_vector/alpha-diversity.tsv

s08_alpha_shannon_per_group.qzv

shannon_vector.qza

shannon_vector/alpha-diversity.tsv

unweighted_unifrac_distance_matrix.qza

unweighted_unifrac_distance_matrix/distance-matrix.tsv

unweighted_unifrac_emperor.qzv

unweighted_unifrac_pcoa_results.qza

weighted_unifrac_distance_matrix.qza

weighted_unifrac_distance_matrix/distance-matrix.tsv

weighted_unifrac_emperor.qzv

s09_beta_weighted_unifrac_emperor_pcoa.qzv

weighted_unifrac_pcoa_results.qza

jaccard_pcoa_results.qza

jaccard_distance_matrix/distance-matrix.tsv

jaccard_distance_matrix.qza

jaccard_emperor.qzv

s09_beta_bray_curtis_emperor_pcoa.qzv

bray_curtis_pcoa_results.qza

bray_curtis_distance_matrix/distance-matrix.tsv

bray_curtis_distance_matrix.qza

bray_curtis_emperor.qzv

s01_get_data.sh

```
# Name s01_get_data.sh
# Load required modules (this is an example, change it!)
module load FastQC/0.11.9-Java-11
module load MultiQC/1.12-foss-2021b

fastqc --version
multiqc --version

# Base folder (this is an example, change it!)
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics"

# Start message
echo "Started downloading FASTQ files from SRA"
date
echo ""

# Folders
#base_folder="..."
sra_folder="${base_folder}/tools/sratoolkit.3.1.1-ubuntu64/bin" # update x.y.z
data_folder="${base_folder}/data" # may exist, but should not contain the data

# List of SRA IDs
# The next line of code reads the first colimn from the samples.txt file,
# omitting the header line, and saves it to the variable sra_ids.
# It ames that the samples file is in the same folder as the script.
sra_ids=$(awk 'NR > 1 {print $1}' samples.txt)

# Loop over SRA IDs and use fasterq-dump to download the data
for id in $sra_ids
do
  echo "${id}"
  "${sra_folder}/fasterq-dump" $id --split-files --skip-technical --outdir "${data_folder}"
  echo ""

done
# Completion message

echo ""
echo "Done"
date
```

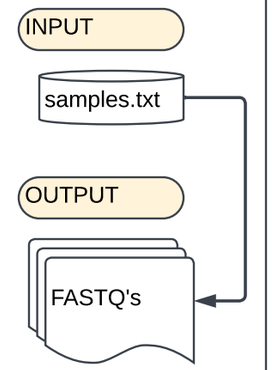
Setting up folders

Samplesheet

Loop through samples and pass each id to fasterq-dump.

--split-files: This option tells fasterq-dump to split paired-end reads into separate files.
--skip-technical: This option skips technical reads, which are often not useful for downstream analysis.

STEP SUMMARY



This scrip will download FASTQ's from the NCBI Sequence Read Archive (SRA) database using the id's from samples.txt as input.

s02_qc

```
# Load required modules (this is an example, change it!)
module load FastQC/0.11.9-Java-11
module load MultiQC/1.12-foss-2021b

fastqc --version
multiqc --version

# Base folder (this is an example, change it!)
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics/"

# Start message
echo "FastQC & MultiQC"
date
echo ""

# Folders
# base_folder="..."
data_folder="${base_folder}/data" # should exist and contain fastq files

# Go to data folder
cd "${data_folder}"

# List of fastq files in data folder
fastq_files=$(ls *.fastq)

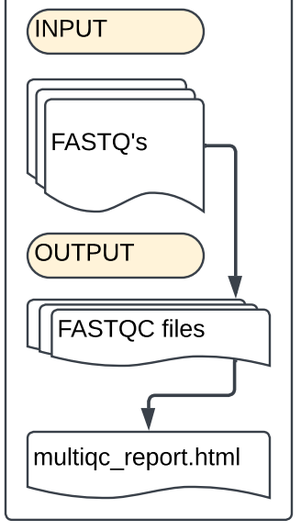
# Run FastQC for all fastq files
fastqc $fastq_files

# Run MultiQC in the current folder
multiqc .
```

make fastc files

make multiqc report

STEP SUMMARY



s03_q2_source_file_prep.sh

```
# name: s03_q2_source_file_prep.sh
# add header to source_files_local.txt
awk 'NR==1 {OFS="	"; print $0}' source_files.txt > source_files_local.txt


# bash one liner to update source_files.txt
awk 'NR > 1 {print $1}'
"/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics/scripts/source_files.txt"
| xargs -I {} sh -c 'echo {} $(find
"/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics/data" -name
{}_1.fastq) $(find "/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics/data"
-name {}_2.fastq)' | tr ' ' '\t' >> source_files_local.txt
```

bash oneliner to manipulate source_files.txt

BEFORE

```
SRR1770766
/mnt/beegfs/home/alexey.larionov/teaching_2024/metagenomics/data/SRR1770766_1.fastq
/mnt/beegfs/home/alexey.larionov/teaching_2024/metagenomics/data/SRR1770766_2.fastq
```

AFTER

```
SRR1770766
/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics/data/SRR1770766_1.fastq
/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics/data/SRR1770766_2.fastq
```

s03_q2_import_and_trim.sh

```
# name : s03_q2_import_and_trim.sh
# Load required modules (this is an example, change it!)
module load QIIME2/2022.8

# Base folder (this is an example, change it!)
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics"

# Start message
echo "QIIME2: Import and Trim"
date
echo ""

# Folders
# base_folder="..."
results_folder="${base_folder}/results"

# make results folder
mkdir -p "${results_folder}"

# source_files.txt filepath
source_filepath="${base_folder}/scripts/source_files_local.txt"

# Importing data to QIIME2. For more details: qiime tools import --help
# Note that file "source_files.txt" should be prepared before you run this script!
qiime tools import \
--type "SampleData[PairedEndSequencesWithQuality]" \
--input-path "${source_filepath}" \
--input-format "PairedEndFastqManifestPhred33V2" \
--output-path "${results_folder}/s03_pe_dmx.qza"

# Trim primers (https://docs.qiime2.org/2022.11/plugins/available/cutadapt/)
# This example shows the case when fragments are longer than reads
# (e.g. ~300bp PCR products sequenced with 150PE Illumina sequencing)
# You should use different approach when reads are longer than PCR fragments
# (e.g. ~300bp PCR productd sequenced with 500PE Illumina sequencing)
qiime cutadapt trim-paired \
--p-front-f ^GTGCCAGCMGCCGCGGTAA \
--p-front-r ^GGACTACHVGGGTWTCTAAT \
--p-match-read-wildcards \
--i-demultiplexed-sequences "${results_folder}/s03_pe_dmx.qza" \
--o-trimmed-sequences "${results_folder}/s03_pe_dmx_trim.qza"

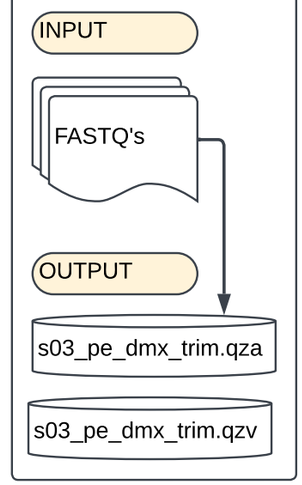
# Make visualisation file (to view at https://view.qiime2.org/)
qiime demux summarize \
--i-data "${results_folder}/s03_pe_dmx_trim.qza" \
--o-visualization "${results_folder}/s03_pe_dmx_trim.qzv"

# Completion message
echo ""
```

This command imports paired-end sequence data into QIIME 2, converting it into a .qza artifact that can be used for further analysis.

This command is used to trim primers from paired-end sequence data using the cutadapt plugin in QIIME 2.

STEP SUMMARY



This script is for processing metagenomic sequencing data using QIIME 2. It imports paired-end sequence data, trims primers, and generates a visualization file for further analysis.

- .qza files: These are data artifacts that contain raw data, intermediate results, or final outputs from various analyses. They encapsulate both the data and metadata, making it easy to track and reproduce analyses12.
- .qzv files: These are visualization artifacts that contain visual representations of the data, such as plots, charts, and summary statistics. They are used to interpret and present the results of analyses

s04_q2_denoise

```
# name: s04_q2_denoise
# Load required modules (this is an example, change it!)
module load QIIME2/2022.8

# Base folder (this is an example, change it!)
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics"

# Start message
echo "QIIME2: Denoise"
date
echo ""

# Folders
# base_folder="..."
results_folder="${base_folder}/results"

# Denoise (default --p-n-reads-learn 1000000)
# In this example we do not aditionally trim data by quality (both trunc-len = 0)
# because the data quality is good from the beginning to the end of the reads.
# Setting the number of threads to 0 requests all cores available on PC.
# This is OK when you use a personal laptop, but should be changed for HPC.
qiime dada2 denoise-paired \
--i-demultiplexed-seqs "${results_folder}/s03_pe_dmx_trim.qza" \
--p-trunc-len-f 0 \
--p-trunc-len-r 0 \
--p-n-threads 12 \
--o-table "${results_folder}/s04_table_dada2.qza" \
--o-denoising-stats "${results_folder}/s04_stats_dada2.qza" \
--o-representative-sequences "${results_folder}/s04_rep_seqs_dada2.qza" \
--verbose

# Summarise feature table
qiime feature-table summarize \
--i-table "${results_folder}/s04_table_dada2.qza" \
--o-visualization "${results_folder}/s04_table_dada2.qzv"

# Visualise statistics
qiime metadata tabulate \
--m-input-file "${results_folder}/s04_stats_dada2.qza" \
--o-visualization "${results_folder}/s04_stats_dada2.qzv"

# Tabulate representative sequences
qiime feature-table tabulate-seqs \
--i-data "${results_folder}/s04_rep_seqs_dada2.qza" \
--o-visualization "${results_folder}/s04_rep_seqs_dada2.qzv"

# Completion message
echo ""
echo "Done"
date
```

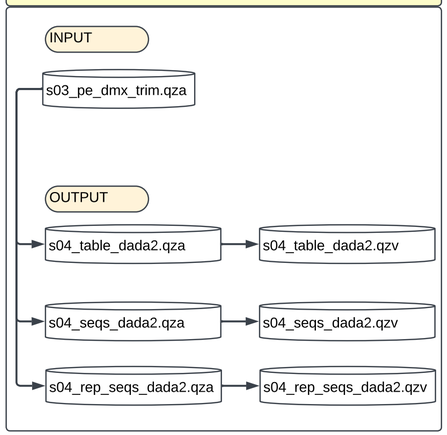
- This command runs the DADA2 algorithm on paired-end sequences to denoise them. This file is in QIIME 2 Artifact format (.qza).

Truncation Lengths: --p-trunc-len-f 0 and --p-trunc-len-r 0.

These parameters set the truncation length for the forward and reverse reads, respectively. A value of 0 means no truncation is applied, which is suitable when the data quality is good throughout the reads.

These commands are used to generate visual summaries and tables for the outputs of the DADA2 denoising process.

STEP SUMMARY



This script uses the DADA2 algorithm within QIIME 2 to clean and refine paired-end sequencing data. The goal is to remove noise and errors from the raw sequence data, resulting in high-quality, accurate sequences that can be used for downstream analysis, such as identifying and quantifying microbial species.

feature table → The s04_table_dada2.qza file is a QIIME 2 Artifact that contains the feature table generated by the DADA2 denoising process.

denoising-stats → Denoising statistics provide detailed information about the performance of the denoising process. These stats typically include:
- Number of input reads: The total number of raw sequences before denoising.
- Number of filtered reads: Sequences that passed initial quality filters.
- Number of denoised reads: Sequences after error correction.
- Number of merged reads: Successfully merged paired-end reads.
- Number of non-chimeric reads: Reads that are not identified as chimeras (artifacts formed by the combination of two or more sequences).

These statistics help you understand how many sequences were retained at each step and assess the overall quality and effectiveness of the denoising process.

representative-sequences → Representative sequences are the unique sequences that remain after denoising and merging paired-end reads. These sequences represent the distinct biological sequences in your dataset. They are used for:
- Taxonomic classification: Identifying the microbial species present in your samples.
- Phylogenetic analysis: Studying the evolutionary relationships between the sequences.
- Feature table creation: Quantifying the abundance of each unique sequence across different samples.

s05_q2_phylogenetic_tree.sh

runs a pipeline that aligns sequences and constructs a phylogenetic tree using MAFFT for alignment and FastTree for tree construction.

file for the aligned sequences.

Masked alignment, removes variable regions that introduce noise into phylogenetic analysis.

unrooted phylogenetic tree.

rooted phylogenetic tree, which is often required for downstream diversity analyses

Export tree as nwk file

Export masked alignments as fasta

```
# name: s05_q2_phylogenetic_tree.sh

module load QIIME2/2022.8

# Folders
# Base folder
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics"
results_folder="${base_folder}/results"


# Perform multiple alignments and build phylogenetic trees
qiime phylogeny align-to-tree-mafft-fasttree \
  --i-sequences "${results_folder}/s04_rep_seqs_dada2.qza" \
  --o-alignment "${results_folder}/s05_aligned_rep_seqs.qza" \
  --o-masked-alignment "${results_folder}/s05_masked_aligned_rep_seqs.qza" \
  --o-tree "${results_folder}/s05_unrooted_tree.qza" \
  --o-rooted-tree "${results_folder}/s05_rooted_tree.qza"

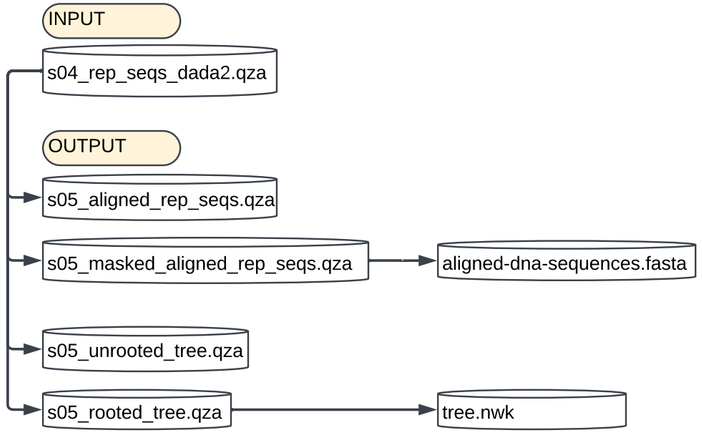

# --- Export tree dta for plotting outside QIIME2 --- #
# Tree files are stored in a separate sub-folder in Results folder.
# They can be used to plot trees in several online tree viewers.
# For instance, tree.nwk file can be viewed using NCBI tree viewer
# https://www.ncbi.nlm.nih.gov/tools/treeviewer/
#
# NCBI tree viewer upload link:
# https://www.ncbi.nlm.nih.gov/projects/treeview/tv.html?appname=ncbi_tviewer&renderer=radial
&openuploaddialog
# Export tree as tree.nwk


qiime tools export \
  --input-path "${results_folder}/s05_rooted_tree.qza" \
  --output-path "${results_folder}/s05_phylogenetic_tree"

# Export masked alignment as aligned-dna-sequences.fasta
# this fasta is not used by NCBI tree viewer, but may be used by other viewers

qiime tools export \
  --input-path "${results_folder}/s05_masked_aligned_rep_seqs.qza" \
  --output-path "${results_folder}/s05_phylogenetic_tree"
```

STEP SUMMARY



This step uses the MAFFT algorithm to align your representative sequences. Multiple sequence alignment arranges the sequences in a way that identifies regions of similarity, which can be indicative of functional, structural, or evolutionary relationships. Masking: After alignment, the sequences are masked to remove highly variable regions that might introduce noise into the phylogenetic analysis. These regions can be problematic because they may not reflect true evolutionary relationships.FastTree: This step uses the FastTree algorithm to construct a phylogenetic tree from the masked aligned sequences. A phylogenetic tree represents the evolutionary relationships between the sequences. The aligned sequences and trees are then exported for visualization in external tools like the NCBI tree viewer.

s06a_q2_rarefaction_plot.sh

```
# name: s06a_q2_rarefaction_plot.sh
# Load required modules
module load QIIME2/2022.8

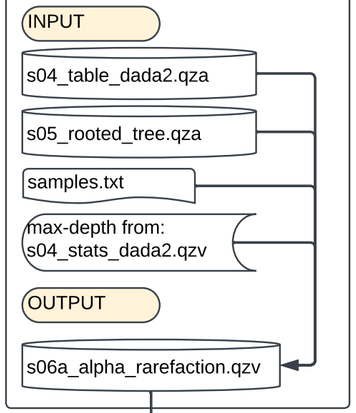

# Base folder
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics"
# base_folder="..."
results_folder="${base_folder}/results"

# Alpha rarefaction
# Max-depth based on max non-chimeric reads in s04_stats_dada2.qzv
# Download csv from qiime2view to get exact numeric rarefaction thresholds

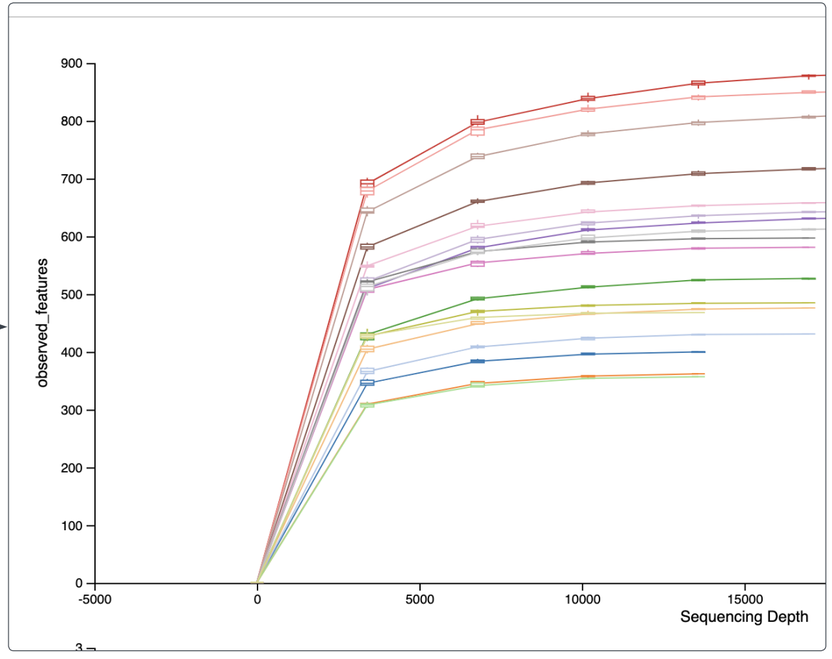
qiime diversity alpha-rarefaction \
--i-table "${results_folder}/s04_table_dada2.qza" \
--i-phylogeny "${results_folder}/s05_rooted_tree.qza" \
--p-max-depth 30559 \ <--- ADDED FROM QIIME2 plot
--m-metadata-file "samples.txt" \
--o-visualization "${results_folder}/s06a_alpha_rarefaction.qzv"
```

Maximum sequencing depth to consider for rarefaction, based on the maximum number of non-chimeric reads in your dataset.

STEP SUMMARY



This part of the pipeline performs alpha rarefaction analysis to assess the diversity within your samples at different sequencing depths



Alpha Rarefaction (sO6a)

- Purpose: Alpha rarefaction is used to assess the diversity within each sample at different sequencing depths. It generates rarefaction curves that show how the number of observed features (e.g., species) increases with sequencing depth.

Rarefaction (sO6b)

- Purpose: Rarefaction (or subsampling) standardizes the sequencing depth across all samples to a specified level. This ensures that all samples are compared at the same depth, which is crucial for accurate diversity and statistical analyses.

This command rarefies (subsamples) your feature table to a specified sequencing depth, ensuring that all samples are compared at the same depth.

s06b_q2_apply_rarefaction.sh

```
# Load required modules
module load QIIME2/2022.8

# Start message
echo "QIIME2: Apply rarefaction"
date
echo ""

# Base folder
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics"

# Folders
results_folder="${base_folder}/results"

# Rarefaction
# Select the sampling-depth as the minimal count of non-chimeric reads (see output of step 4)
qiime feature-table rarefy \
--i-table "${results_folder}/s04_table_dada2.qza" \
--p-sampling-depth 14107 \ \ <--- ADDED FROM s04_seqs_dada2.qza
--o-rarefied-table "${results_folder}/s06b_rarefied_table.qza"
```

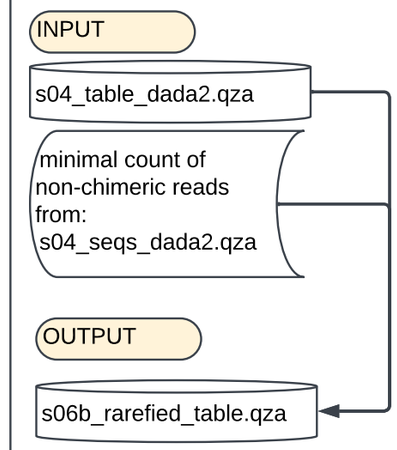
Alpha Rarefaction (sO6a)

- Purpose: Alpha rarefaction is used to assess the diversity within each sample at different sequencing depths. It generates rarefaction curves that show how the number of observed features (e.g., species) increases with sequencing depth.

Rarefaction (sO6b)

- Purpose: Rarefaction (or subsampling) standardizes the sequencing depth across all samples to a specified level. This ensures that all samples are compared at the same depth, which is crucial for accurate diversity and statistical analyses.

STEP SUMMARY



performs rarefaction on your feature table to standardize the sequencing depth across all samples.

Can't really describe it bettwe....
Calculate a whole bunch of diversity metrics

These commands export various alpha and beta diversity metrics from QIIME 2 format to standard file formats that can be used for further analysis outside of QIIME 2.

s07_q2_calculate_diversity_metrics

```
# name: s07_q2_calculate_diversity_metrics

# Load required modules
module load QIIME2/2022.8


# Folders
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics"
results_folder="${base_folder}/results"
diversity_metrics_folder="${results_folder}/s07_diversity_metrics"


# Calculate a whole bunch of diversity metrics
# Select the sampling-depth as the minimal count of non-chimeric reads (see output of step 4)
qiime diversity core-metrics-phylogenetic \
--i-table "${results_folder}/s04_table_dada2.qza" \
--i-phylogeny "${results_folder}/s05_rooted_tree.qza" \
--p-sampling-depth 14107 \ <--- ADDED FROM s04_seqs_dada2.qza
--m-metadata-file "samples.txt" \
--output-dir "${diversity_metrics_folder}"



# Export some results out of QIIME2 format to explore
# (these files can be used for analysis outsede of QIIME2)
# Alpha-diversity metrics
qiime tools export \
  --input-path "${diversity_metrics_folder}/observed_features_vector.qza" \
  --output-path "${diversity_metrics_folder}/observed_features_vector"
qiime tools export \
  --input-path "${diversity_metrics_folder}/faith_pd_vector.qza" \
  --output-path "${diversity_metrics_folder}/faith_pd_vector"
qiime tools export \
  --input-path "${diversity_metrics_folder}/evenness_vector.qza" \
  --output-path "${diversity_metrics_folder}/evenness_vector"
qiime tools export \
  --input-path "${diversity_metrics_folder}/shannon_vector.qza" \
  --output-path "${diversity_metrics_folder}/shannon_vector"
# Beta-diversity metrics
qiime tools export \
  --input-path "${diversity_metrics_folder}/unweighted_unifrac_distance_matrix.qza" \
  --output-path "${diversity_metrics_folder}/unweighted_unifrac_distance_matrix"
qiime tools export \
  --input-path "${diversity_metrics_folder}/weighted_unifrac_distance_matrix.qza" \
  --output-path "${diversity_metrics_folder}/weighted_unifrac_distance_matrix"
qiime tools export \
  --input-path "${diversity_metrics_folder}/jaccard_distance_matrix.qza" \
  --output-path "${diversity_metrics_folder}/jaccard_distance_matrix"
qiime tools export \
  --input-path "${diversity_metrics_folder}/bray_curtis_distance_matrix.qza" \
  --output-path "${diversity_metrics_folder}/bray_curtis_distance_matrix"
```

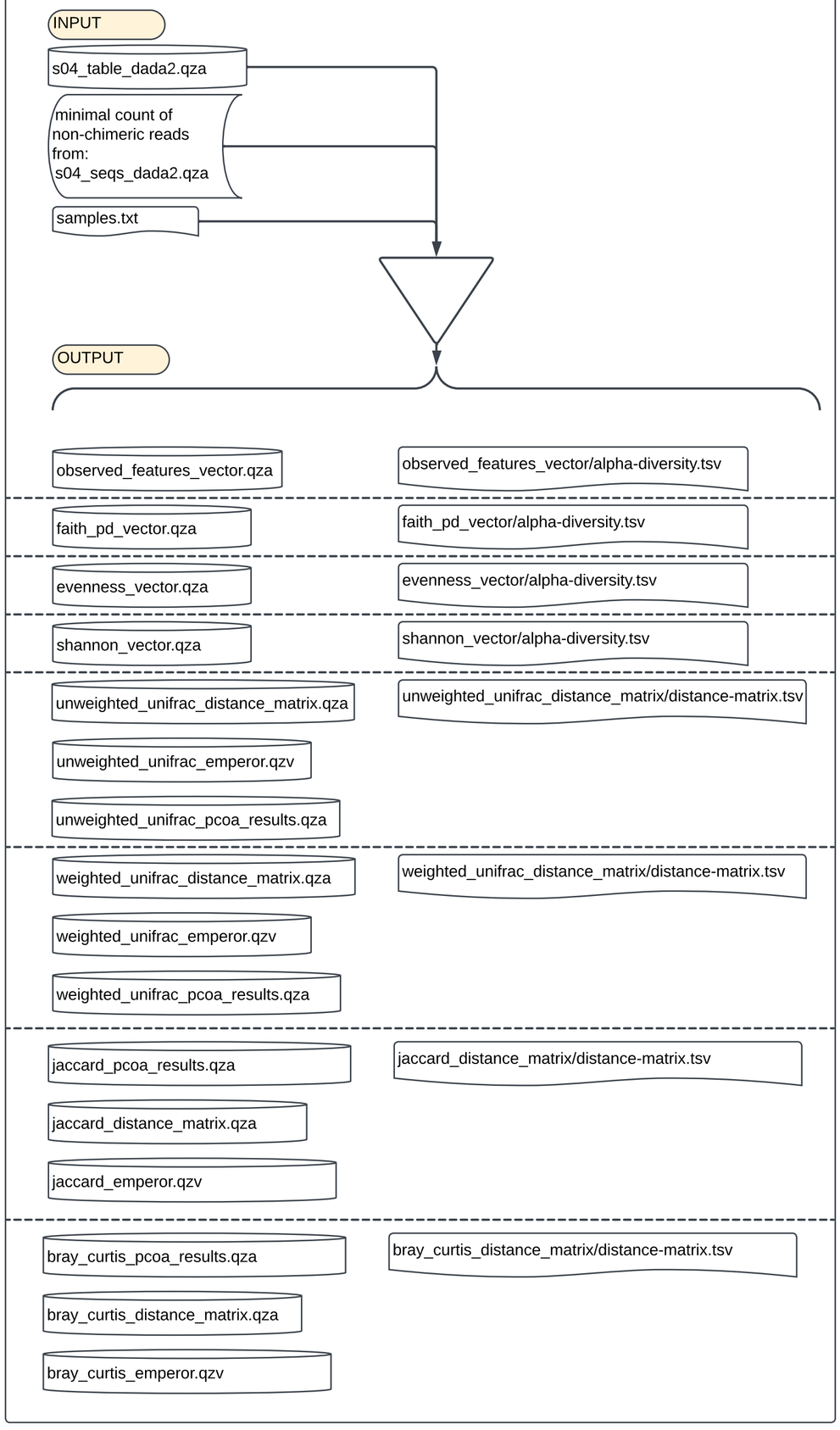
Alpha-Diversity Metrics

1. Observed Features: Exports the number of unique features (e.g., species) observed in each sample.
2. Faith's Phylogenetic Diversity (PD): Exports a measure of phylogenetic diversity that considers the branch lengths of the phylogenetic tree.
3. Evenness: Exports a measure of how evenly the features are distributed within each sample.
4. Shannon Diversity: Exports a measure of both richness and evenness of the features within each sample.

Beta-Diversity Metrics

1. Unweighted UniFrac Distance Matrix: Exports the phylogenetic distance between samples without considering feature abundance.
2. Weighted UniFrac Distance Matrix: Exports the phylogenetic distance between samples while considering feature abundance.
3. Jaccard Distance Matrix: Exports the dissimilarity between samples based on the presence/absence of features.
4. Bray-Curtis Distance Matrix: Exports the dissimilarity between samples based on feature abundance.

STEP SUMMARY



Calculate Alpha and Beta diversity metrics and export from QIIME 2 format to standard file formats that can be used for further analysis outside of QIIME 2.

s08_q2_alpha_diversity_box_plots

```
# Name: s08_q2_alpha_diversity_box_plots
# Load required modules
module load QIIME2/2022.8

# Start message
echo "QIIME2: Alpha-diversity box-plots"
date
echo ""

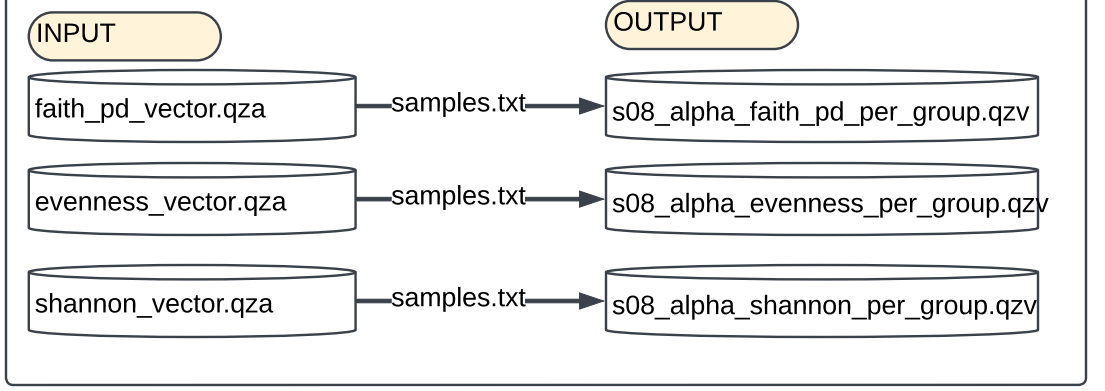
# Folders
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics"
results_folder="${base_folder}/results"
diversity_metrics_folder="${results_folder}/s07_diversity_metrics"

# Visualize relationships between alpha diversity and study metadata
# (uses some files created at the previous step)
qiime diversity alpha-group-significance \
--i-alpha-diversity "${diversity_metrics_folder}/faith_pd_vector.qza" \
--m-metadata-file "samples.txt" \
--o-visualization "${results_folder}/s08_alpha_faith_pd_per_group.qzv"

qiime diversity alpha-group-significance \
--i-alpha-diversity "${diversity_metrics_folder}/evenness_vector.qza" \
--m-metadata-file "samples.txt" \
--o-visualization "${results_folder}/s08_alpha_evenness_per_group.qzv"

qiime diversity alpha-group-significance \
--i-alpha-diversity "${diversity_metrics_folder}/shannon_vector.qza" \
--m-metadata-file "samples.txt" \
--o-visualization "${results_folder}/s08_alpha_shannon_per_group.qzv"
```

STEP SUMMARY



These commands are used to visualize the relationships between alpha diversity metrics and your study metadata.

s09_q2_beta_diversity_pcoa

```
# Name: s09_q2_beta_diversity_pcoa
# Load required modules
module load QIIME2/2022.8

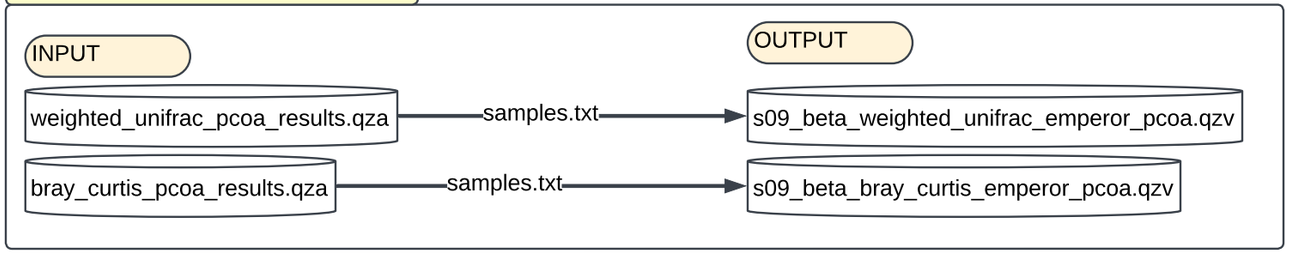
# Load required modules
module load QIIME2/2022.8


# Folders
# Base folder (this is an example, change it!)
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics"
results_folder="${base_folder}/results"
diversity_metrics_folder="${results_folder}/s07_diversity_metrics"


# Use the weighted unifrac distances (custom-axes parameter can be used to
specific any column from your metadata file)
qiime emperor plot \
--i-pcoa "${diversity_metrics_folder}/weighted_unifrac_pcoa_results.qza" \
--m-metadata-file "samples.txt" \
--o-visualization "${results_folder}/s09_beta_weighted_unifrac_emperor_pcoa.qzv"


# Use the bray curtis distances (custom-axes parameter can be used to specific any
column from your metadata file)
qiime emperor plot \
--i-pcoa "${diversity_metrics_folder}/bray_curtis_pcoa_results.qza" \
--m-metadata-file "samples.txt" \
--o-visualization "${results_folder}/s09_beta_bray_curtis_emperor_pcoa.qzv"
```

STEP SUMMARY



This step generates interactive PCoA plots using weighted UniFrac and Bray-Curtis distances to visualize beta diversity. These plots help you explore how microbial communities differ between samples and how they relate to metadata categories.

Classifier: Uses a pre-trained classifier (gg-13-8-99-515-806-nb-classifier.qza) to assign taxonomy.

Make taxonomy barplot

s10_q2_taxonomy_barplot

```
# Name: s10_q2_taxonomy_barplot

# Load required modules
module load QIIME2/2022.8

# Folders
base_folder="/mnt/beegfs/home/s430452/metagenomics_assay/metagenomics"
results_folder="${base_folder}/results"
resources_folder="${base_folder}/resources"

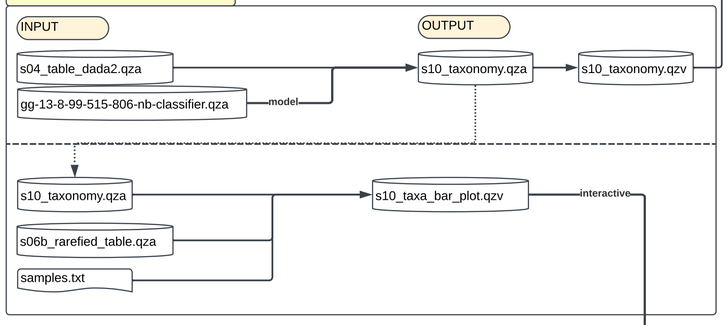

# Assign taxonomy to sequences
qiime feature-classifier classify-sklearn \
--i-classifier "${resources_folder}/gg-13-8-99-515-806-nb-classifier.qza" \
--i-reads "${results_folder}/s04_rep_seqs_dada2.qza" \
--o-classification "${results_folder}/s10_taxonomy.qza"


# Show taxonimies assigned to each ASV (Amplicon Sequence Variant)
qiime metadata tabulate \
--m-input-file "${results_folder}/s10_taxonomy.qza" \
--o-visualization "${results_folder}/s10_taxonomy.qzv"


# Make taxonomy barplot
qiime taxa barplot \
--i-table "${results_folder}/s06b_rarefied_table.qza" \
--i-taxonomy "${results_folder}/s10_taxonomy.qza" \
--m-metadata-file "samples.txt" \
--o-visualization "${results_folder}/s10_taxa_bar_plot.qzv"
```

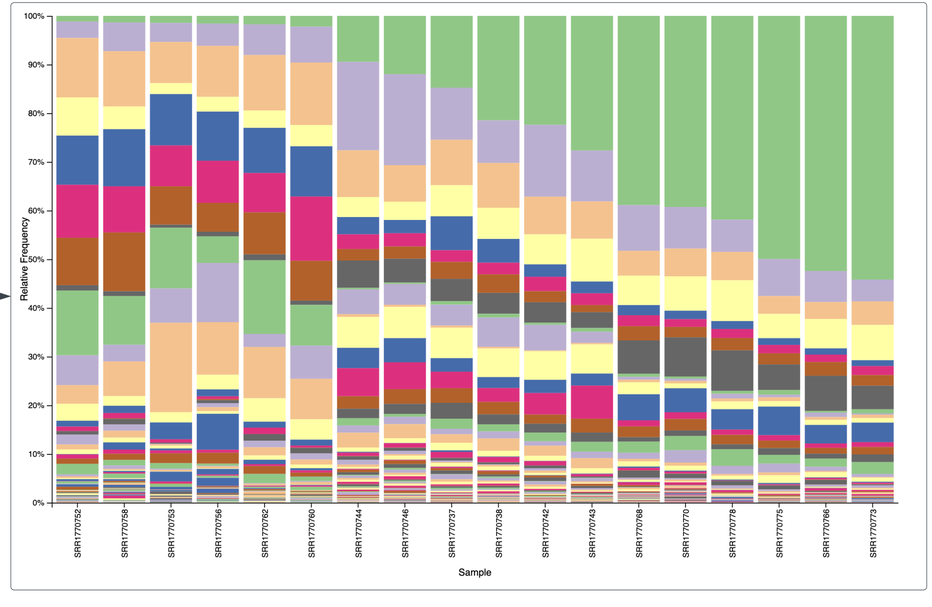
An ASV (Amplicon Sequence Variant) is a unique DNA sequence identified from amplicon sequencing data. Unlike traditional OTUs (Operational Taxonomic Units), which group sequences based on a similarity threshold (e.g., 97%), ASVs represent exact sequences without clustering. This provides higher resolution and more accurate identification of microbial diversity.

STEP SUMMARY

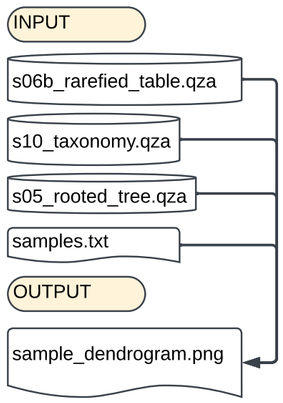


This script assigns taxonomy to your sequences using a pre-trained classifier and then creates a bar plot to visualize the taxonomic composition of your samples. It helps you understand the distribution of different taxa across your samples.

Feature ID #q2:types	Taxon categorical	Confidence categorical
00053205579d520823e3aa493ba8548d	k__Bacteria; p__Acidobacteria; c__Acidobacteria-6; o__iii1-15; f__; g__; s__	0.9629214359028847
00159db925c072323026e4989ee589fd	k__Bacteria; p__Planctomycetes; c__028H05-P-BN-P5; o__; f__; g__; s__	0.9999999450968902
004a0406cfdab0e322c2b49aeb274d92	k__Bacteria; p__Proteobacteria; c__Deltaproteobacteria; o__Myxococcales; f__; g__; s__	0.9998342189875883



STEP SUMMARY



S11_q2_to_R

```
---
title: "S11_q2_to_R"
author: "Matthew Spriggs"
date: "2024-12-13"
output: html_document
---


```{r}
current_dir = dirname(rstudioapi::getActiveDocumentContext()$path)
setwd(current_dir)
setwd('..')
root_dir = getwd()
```

```{r}
data_folder=paste0(root_dir, '/results')
table_qza=file.path(data_folder,"s06b_rarefied_table.qza")
rooted_tree_qza=file.path(data_folder,"s05_rooted_tree.qza")
taxonomy_qza=file.path(data_folder,"s10_taxonomy.qza")

scripts_folder=paste0(root_dir, '/scripts')
metadata_tsv=file.path(scripts_folder,"samples.txt")
```

Import of QIIME2 artifacts & metadata to R
Import to phyloseq data type using qiime2R package. Then data could be further extracted from the phyloseq object, if necessary.

```{r}
# Convert QIIME2 artifacts & metadata to phyloseq
phy <- qza_to_phyloseq(table_qza, rooted_tree_qza, taxonomy_qza, metadata_tsv)

# Extract metadata and distance matrix from phyloseq object
metadata <- data.frame(sample_data(phy))
distance_matrix <- distance(phy, method="bray")
```

Samples' Hierarchical Clustering & Dendrogram

Formatting the dendrogram with general purpose dendextend R package.

```{r}
bray_clust <- hclust(distance_matrix, method="ward.D2")
bray_dend <- as.dendrogram(bray_clust, hang=0.1)

colour_labels <- c('red','green','blue')[ match(as.factor(metadata$group),
                                                c('frue_ch','mtca_au','ukul_za') ) ]
labels_colors(bray_dend) <- colour_labels

plot(bray_dend, main="Samples dendrogram", ylab="Distances")

legend("topright",
       legend=c('frue_ch','mtca_au','ukul_za'),
       col=c('red','green','blue'),
       bty="n",lty=1, cex=0.8)
```

PERMANOVA
Using vegan::adonis: the differences between studied groups are highly significant.

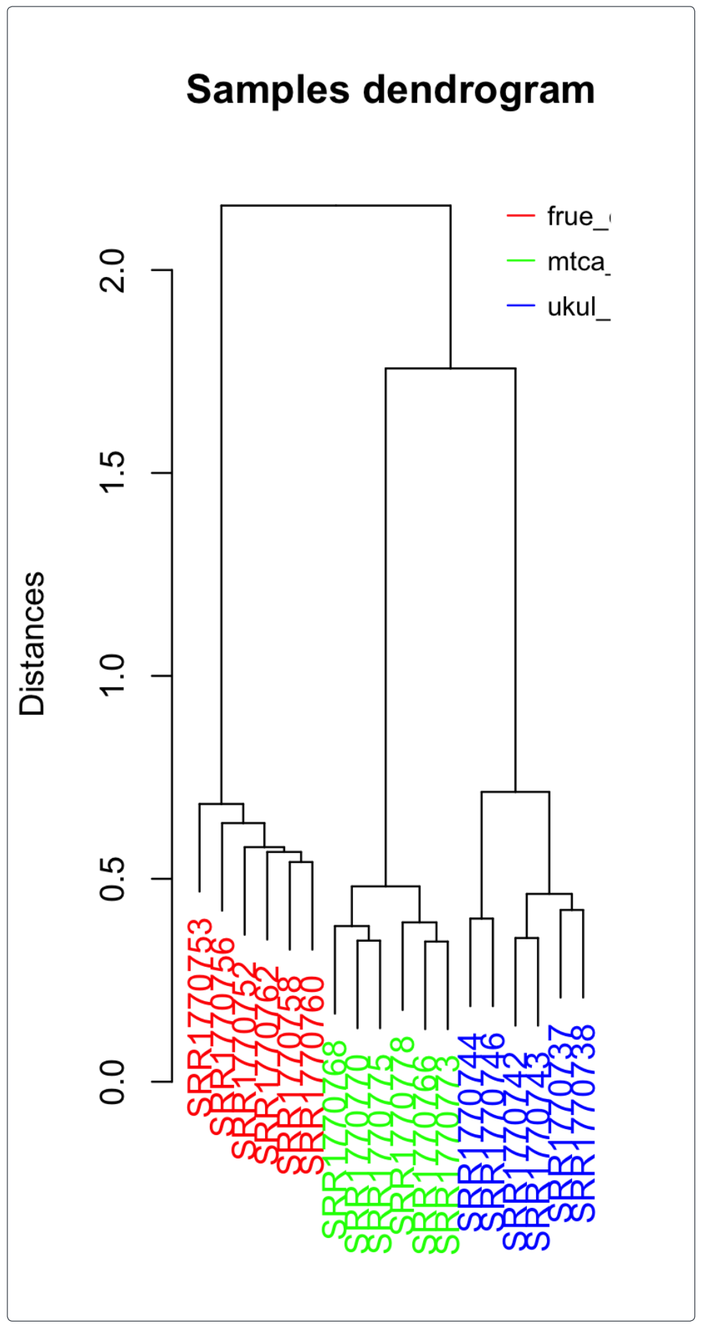
```{r}

# Run PERMANOVA
adonis2(distance_matrix ~ group, data = metadata, permutations=100000)

```

Session info
For reproducible research

```{r}
sessionInfo()
```

This script imports QIIME 2 artifacts and metadata into R, converts them into a phyloseq object, and performs hierarchical clustering and PERMANOVA analysis. To visualize and statistically analyze the relationships between microbial communities in different sample groups.

Text




```
Permutation test for adonis under reduced model
Permutation: free
Number of permutations: 1e+05

adonis2(formula = distance_matrix ~ group, data = metadata, permutations =
1e+05)
         Df SumOfSqs      R2      F
Model     2   3.8749 0.67183 15.354
Residual 15   1.8928 0.32817
Total    17   5.7677 1.00000
         Pr(>F)
Model     1e-05 ***
Residual
Total
---
Signif. codes:
  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05
  ‘.’ 0.1 ‘ ’ 1
```
```