**Tutorial on Exploring and Analyzing Microbial 16S rRNA Sequencing Data using QIIME2**

**Introduction**

The use of 16S rRNA sequencing data in combined with tools such as QIIME2 provides a greater comprehension of the diversity and composition of microbial communities, as well as their potential function in a variety of ecological and agricultural applications. The 16S rRNA gene is an effective instrument for analyzing the composition and diversity of microbial communities in various environments, including the gut microbiome. QIIME2 is an open-source software application for analyzing and visualizing 16S rRNA sequencing data from microbiome. In this tutorial, I will demonstrate how to use QIIME2 to analyze and visualize 16S rRNA sequencing data from the gut microbiome of bumblebee, *Bombus pascuorum*. This tutorial's data was collected for a journal article about 16S rRNA Gut Bumblebee Targeted Locus (Loci). I downloaded BASH scripts from GitHub (https://github.com/beiko-lab/mimb\_16S) and datasets from the European Bioinformatics Institute Short Read Archive (EBI SRA) (https://www.ebi.ac.uk/ena/browser/view/PRJNA313530). However, due to website and QIIME2 package updates, I encountered difficulties when attempting to execute the published tutorial. Therefore, I present an updated version of the tutorial that incorporates the most recent modifications to the QIIME2 application and the EBI SRA website. This tutorial will demonstrate how to use QIIME2 to examine, analyze, and visualize 16S rRNA sequencing data from the gut microbiome of *Bombus pascuorum*.

**Application**

Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2 Bolyen et al. (2019) - This article introduces QIIME2 and its features, including its ability to analyze large and complex microbiome datasets.

Microbial community analysis in soil Straub et al. (2020) - This article used QIIME2 to analyze microbial community data from soil samples collected in a long-term field experiment. The study used QIIME2 to perform quality control, sequence alignment, taxonomic classification, and diversity analysis of the 16S rRNA sequencing data.

Microbial community analysis in the human oral cavity Kuczynski et al. (2011) - This study used QIIME2 to analyze 16S rRNA sequencing data from oral rinse samples collected from healthy individuals. The study used QIIME2 to perform quality control, sequence alignment, taxonomic classification, and diversity analysis of the data.

**Tutorial**

**1. Downloading Anaconda and activating QIIME2 environment**

# Downloads the QIIME2 conda environment file for Mac OS

wget https://data.qiime2.org/distro/core/qiime2-2023.2-py38-osx-conda.yml

# Creates a new conda environment called 'qiime2-2023.2' based on it

conda env create -n qiime2-2023.2 --file qiime2-2023.2-py38-osx-conda.yml

conda activate qiime2-2023.2

**2. Retrieving sequence data**

The repository contains a BASH script named 'fetchFastq.sh' that automatically downloads the raw FASTQ files and sample metadata directly from EBI SRA. Due to the update on the website, for the script to run properly, the command needs to be change from:

curl -sLo MANIFEST.txt "http://www.ebi.ac.uk/ena/data/warehouse/search?query=%22study\_accession%3D%22${ACCESSION}%22%22&result=read\_run&fields=fastq\_ftp,sample\_alias,sample\_accession&display=report"

to:

curl -sLo MANIFEST.txt "https://www.ebi.ac.uk/ena/portal/api/filereport?accession=${ACCESSION}&result=read\_run&fields=run\_accession,fastq\_ftp,fastq\_md5,fastq\_bytes"

# Change permission of the 'fetchFasta.sh' file to make it executable

chmod +x fetchFasta.sh

# Execute the 'fetchFasta.sh' script

./fetchFasta.sh

**3. Import data (Appendix 1)**QIIME2 uses two main file types for input/output: QIIME artifacts (.qza) and QIIME visualization (.qzv). QIIME artifacts capture the data generated by a particular pipeline step, along with additional information such as software versions, command parameters, and run times. On the other hand, QIIME visualization files contain data intended for visualization and can be opened in a web browser using the qiime tools view command.

By running the fetchFastq.sh script, a directory named data/import/ is created. This directory contains the forward and reverse FASTQ data files, a MANIFEST file, and a metadata.yml file. To import the FASTQ files into the QIIME artifact named reads.qza, the following command should be executed in the same directory where the fetchFastq.sh script is located.

# Importing the FASTQ files into a QIIME

qiime tools import --type 'SampleData[PairedEndSequencesWithQuality]' --input-path data/import --output-path reads

**4. Visualize Sequence Quality**

The quality characteristics of sequences may differ based on experimental factors like the gene being targeted and the sequencing platform being used. These quality features play a role in determining certain sequence processing parameters, such as the truncation parameters for the DADA2 denoising process. To examine these quality features at each base position, the following command will randomly select 10,000 sequences and generate box plots:

# Analyzing quality scores of 10,000 random samples using DADA2 (Figure 1)

qiime demux summarize --p-n 10000 --i-data reads.qza --o-visualization quality\_visualization

# View the plot

tools view quality\_visualization.qzv

**5. Denoising sequences with DADA2**

QIIME2 provides Illumina sequence denoising using DADA2. With the qiime dada2 denoise-paired algorithm, paired-end reads can be combined and denoised. To specify the truncation position for the forward and reverse sequences, use --p-trunc-len-f and --p-trunc-len-r, respectively. These parameters are currently set at 151 and 140. The program can perform parallel computations on four threads with the --p-n-threads 4 option. The --verbose option displays DADA2's progress in the terminal. The denoising process creates two outputs: a table file and a sequence file with a representative sample.  
# The decrease in quality for the forward reads was minimal, whereas the reverse reads displayed a substantial decline in quality, so trim 10 base pairs from the reverse reads  
qiime dada2 denoise-paired --i-demultiplexed-seqs reads.qza --o-table table --o-representative-sequences representative\_sequences --o-denoising-stats denoise\_stats --p-trunc-len-f 150 --p-trunc-len-r 140 --p-trim-left-f 19 --p-trim-left-r 20 --p-n-threads 3

# Create a summary of the output table file showing the sequences/sample spread

qiime feature-table summarize --i-table table.qza --o-visualization table\_summary

**6. Filtering sequence table**

Upon examining the summary table, it was observed that SRR3203007 had a much lower sequencing depth compared to all other sequences. While the other sequences had more than 40000 sequences, SRR3203007 had only 3554. Therefore, it was decided to remove SRR3203007 from the table and exclude it from further analysis. To achieve this, the command to eliminate samples with less than 5000 sequences will be used, which will only remove SRR3203007 from the table.  
# Filter out sequences with few samples

qiime feature-table filter-samples --i-table table.qza --p-min-frequency 5000 --o-filtered-table filtered\_table  
**7. Taxonomic classification**

QIIME2 employs scikit-learn, a Python library for machine learning, to classify sequences. To avoid the need for retraining the classifier between trials and reduce the overall runtime, a reference set can be used to train a naive Bayes classifier, which can be saved as a QIIME2 artifact for future use. However, since the previous classifier artifact generated using an older version of scikit-learn is no longer compatible with the current version, a new classifier (silva-119-99-515-806-nb-classifier) is used instead.  
# Downloading the trained naive Bayes classifier artifact.

wget https://data.qiime2.org/2018.4/common/silva-119-99-515-806-nb-classifier.qza

This classifier artifact is trained on the Silva July, 2014, trimmed to the V4 hyper-variable region with primers 515f/806r, and clustered at 99% sequence identity. This classifier artifact and the scikit-learn Python library are used to instruct QIIME:

# Classify against it with Naive Bayes

qiime feature-classifier classify-sklearn --i-classifier silva-119-99-515-806-nb-classifier.qza --i-reads representative\_sequences.qza --o-classification taxonomy

**8. Visualize taxonomic classifications**

QIIME is capable of generating interactive bar graphs of taxonomic profiles. Using the metadata file produced by the fetchFastq.sh script, the profiles can be sorted by metadata category. The example data metadata file is located at data/METADATA.txt.

# Taxa bar plots (Figure 2)

qiime taxa barplot --i-table filtered\_table.qza --i-taxonomy taxonomy.qza --m-metadata-file data/METADATA.txt --o-visualization taxa-bar-plots

#View the barplot

tools view taxa-barplots.qzv.

**9. Build Phylogeny**

To construct phylogenetic diversity measures such as unweighted and weighted UniFrac, a phylogenetic tree is required. The procedure consists of four steps: multiple sequence alignment, masking, tree building, and tree rooting to generate the.qza artifact that will be used as input to generate phylogenetic-diversity measures. In this experiment, 16S rRNA gene fragments are used to construct a phylogeny.

# Steps for generating a phylogenetic tree

# Aligning denoised sequences with MAFFT   
qiime alignment mafft --i-sequences representative\_sequences.qza --o-alignment aligned\_representative\_sequences

# Masking uniformative positions

qiime alignment mask --i-alignment aligned\_representative\_sequences.qza --o-masked-alignment masked\_aligned\_representative\_sequences

# Build the phylogeny using the FastTree method

qiime phylogeny fasttree --i-alignment masked\_aligned\_representative\_sequences.qza --o-tree unrooted\_tree

# Rooting the tree at midpoint to produce rooted\_tree.qza artifact

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

**Reflection**

During my time learning QIIME2 analysis, I experienced countered several challenges. Identifying and resolving errors in an old script to ensure its proper functioning was the greatest obstacle. This was because the previous tutorial was published in 2018 and directly downloaded FASTQ files from the EBI SRA website. The website and QIIME2 software have since been upgraded, rendering the script obsolete and inoperable. I spent most of the time identifying and correcting the script's bugs so that it would generate identical results to the previous version. One challenge worth mentioning is that this type of analysis cannot be run on Graham because it no longer supports the Anaconda environment. Additionally, the website and QIIME2 may be updated again in the future, requiring users to review the script to ensure that the code continues to function properly.

QIIME2 is a robust open-source bioinformatics software application that enables the analysis and exploration of microbial communities using high-throughput sequencing data. It offers a variety of tools and workflows for processing, analyzing, and visualizing microbiome data, enabling researchers to gain insight into the diversity and composition of microbial communities in various environments, including the gut microbiome. Recent research has shown that alterations in the gut microbiome are associated with a variety of health conditions. The gut microbiome is a complex ecosystem that plays a crucial role in human health and disease. QIIME2 provides an intuitive interface for data visualization, enabling researchers to investigate the relationships between microbial communities and diverse environmental factors. By utilizing these tools, researchers can obtain a deeper understanding of the complex interactions between microorganisms and their host, which could eventually lead to the development of novel therapeutic strategies for a variety of health conditions. Therefore, I believe that it would be beneficial for us, future bioinformaticians, to gain knowledge about this particular type of analysis.  
I found the selected workflow and software to be reasonably user-friendly and well-documented, with online tutorials and instruction available. However, updating the script was difficult due to modifications to the website and the QIIME2 software, which may pose a significant challenge for others learning this tutorial.

**Figures**   
Chart, box and whisker chart

Description automatically generated

**Figure 1**. Quality score box plots from 10,000 reverse reads

Chart, histogram

Description automatically generated

**Figure 2**. Phylum-level taxonomic profiles for the bumblebee gut samples. The samples are classified by type of bee and then by Proteobacteria abundance. Larval samples (La), nesting bees (Nu), foragers from the nest (Fn), foragers from the environment (Fo), and the queen (Qu) are the distinct groups of bees.

**References** (Bolyen et al., 2019; Hall & Beiko, 2018; Kuczynski et al., 2011; Rai et al., 2021; Straub et al., 200

Bolyen, E., Rideout, J. R., Dillon, M. R., Bokulich, N. A., Abnet, C. C., Al-Ghalith, G. A., Alexander, H., Alm, E. J., Arumugam, M., Asnicar, F., Bai, Y., Bisanz, J. E., Bittinger, K., Brejnrod, A., Brislawn, C. J., Brown, C. T., Callahan, B. J., Caraballo-Rodríguez, A. M., Chase, J., … Caporaso, J. G. (2019). Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nature Biotechnology*, *37*(8), 852–857. https://doi.org/10.1038/s41587-019-0209-9

Hall, M., & Beiko, R. G. (2018). *16S rRNA Gene Analysis with QIIME2* (pp. 113–129). https://doi.org/10.1007/978-1-4939-8728-3\_8

Kuczynski, J., Stombaugh, J., Walters, W. A., González, A., Caporaso, J. G., & Knight, R. (2011). Using QIIME to Analyze 16S rRNA Gene Sequences from Microbial Communities. *Current Protocols in Bioinformatics*, *36*(1). https://doi.org/10.1002/0471250953.bi1007s36

Rai, S. N., Qian, C., Pan, J., Rai, J. P., Song, M., Bagaitkar, J., Merchant, M., Cave, M., Egilmez, N. K., & McClain, C. J. (2021). Microbiome data analysis with applications to pre-clinical studies using QIIME2: Statistical considerations. *Genes & Diseases*, *8*(2), 215–223. https://doi.org/10.1016/j.gendis.2019.12.005

Straub, D., Blackwell, N., Langarica-Fuentes, A., Peltzer, A., Nahnsen, S., & Kleindienst, S. (2020). Interpretations of Environmental Microbial Community Studies Are Biased by the Selected 16S rRNA (Gene) Amplicon Sequencing Pipeline. *Frontiers in Microbiology*, *11*. https://doi.org/10.3389/fmicb.2020.550420

**Additional References**

GitHub respiratory for original tutorials: <https://github.com/beiko-lab/mimb_16S>

EBI SRA for dataset: <https://www.ebi.ac.uk/ena/browser/view/PRJNA313530>

Silva classifier file: <https://www.arb-silva.de/documentation/release-119/>

**Appendix**

**1. fetchFasta.sh**

#!/bin/bash

# Set the project accession

ACCESSION=PRJNA313530

# Organize everything in a folder

mkdir -p data

cd data

# Fetch the project file manifest

curl -sLo MANIFEST.txt "https://www.ebi.ac.uk/ena/portal/api/filereport?accession=${ACCESSION}&result=read\_run&fields=run\_accession,fastq\_ftp,fastq\_md5,fastq\_bytes"

# Make the directory that will contain all the files to be imported to QIIME

mkdir -p import

cd import

# Make the required metadata.yml file for QIIME that lists the PHRED offset value

echo "{'phred-offset': 33}" > metadata.yml

# Fetch each of the noted FASTQ files

awk 'BEGIN{FS="\t";}{if (NR>1) {split($2, f, ";"); system("wget " f[1]); system("wget " f[2]);}}' ../MANIFEST.txt

# Print the header of the MANIFEST file that QIIME requires

echo "sample-id,filename,direction" > MANIFEST

# Name the files according to QIIME/Illumina conventions and fill the MANIFEST file needed for `qiime tools import`

for RUN\_ACCESSION in `tail -n +2 ../MANIFEST.txt | cut -f 1`

do

# Rename the first read file to match Illumina file naming convention

mv ${RUN\_ACCESSION}\_1.fastq.gz ${RUN\_ACCESSION}\_S0\_L001\_R1\_001.fastq.gz

# Append a line to MANIFEST file with the updated file name and direction

echo "${RUN\_ACCESSION},${RUN\_ACCESSION}\_S0\_L001\_R1\_001.fastq.gz,forward" >> MANIFEST

# Rename the second read file to match Illumina file naming convention

mv ${RUN\_ACCESSION}\_2.fastq.gz ${RUN\_ACCESSION}\_S0\_L001\_R2\_001.fastq.gz

# Append a line to MANIFEST file with the updated file name and direction

echo "${RUN\_ACCESSION},${RUN\_ACCESSION}\_S0\_L001\_R2\_001.fastq.gz,reverse" >> MANIFEST

done

# Get the sample accession from MANIFEST.txt and assign it to SAMPLE\_ACCESSION variable

SAMPLE\_ACCESSION=tail -n +2 MANIFEST.txt | cut -f 1 | head -n 1

# Use curl command to download the XML data for the specified sample accession

curl -sLo ${SAMPLE\_ACCESSION}.txt "https://www.ebi.ac.uk/ena/data/view/${SAMPLE\_ACCESSION}&display=xml"

# Extract the desired metadata information from the downloaded XML file using awk and format it using sed. The resulting metadata is then written to a file named METADATA.txt with "#" added at the beginning of each line

awk 'BEGIN{ORS="\t"} /<TAG>/{gsub(/.<TAG>/,"");gsub(/</TAG>./,"");print}' ${SAMPLE\_ACCESSION}.txt | sed 's/^/#/' > METADATA.txt

# Extract metadata from all XML files in the list and append to the METADATA.txt file

for SAMPLE\_ACCESSION in `tail -n +2 MANIFEST.txt | cut -f 1`

do

# Use curl command to download the XML data for the specified sample accession

curl -sLo ${SAMPLE\_ACCESSION}.txt "https://www.ebi.ac.uk/ena/data/view/${SAMPLE\_ACCESSION}&display=xml"

# Extract the sample ID, sample alias, and bee type information from the downloaded XML file using awk

SAMPLE\_ID=`awk '/<SAMPLE\_ID>/{gsub(/.\*<SAMPLE\_ID>/,"");gsub(/<\/SAMPLE\_ID>.\*/,"");print}' ${SAMPLE\_ACCESSION}.txt`

SAMPLE\_ALIAS=`awk '/<SAMPLE\_ALIAS>/{gsub(/.\*<SAMPLE\_ALIAS>/,"");gsub(/<\/SAMPLE\_ALIAS>.\*/,"");print}' ${SAMPLE\_ACCESSION}.txt`

BEE\_TYPE=`awk '/<TAG>/{if($0~/bee\_type/)gsub(/.\*<TAG>/,"");gsub(/<\/TAG>.\*/,"");print}' ${SAMPLE\_ACCESSION}.txt`

# Append the extracted metadata to a file named METADATA.txt, separated by tabs

echo -e "${SAMPLE\_ID}\t${SAMPLE\_ALIAS}\t${BEE\_TYPE}" >> METADATA.txt

done