SOP_RRT_QIIME2_Amplicon

September 20, 2025

1 QIIME2 Amplicon Standard Operating Procedure

RRT_SOP_QIIME2Amplicon_v2_20250920

QIIME2 Amplicon is a computational analysis platform for 16S and other microbiome markers. It is most easily run in command line. This guide goes covers

1) Overview
2) Installation
3) Setting a Working Directory
4) Metadata and Manifest preparation
5) QIIME2 main workflow
6) Alterations Required if data is not demultiplexed

Copy and paste the relevant lines into your command line. It is easiest to have an open notepad or word document to make the relevant edits and then copy to command line since it is much harder to make edits in the command line.

2 Install QIIME2 Amplicon

Written for terminal command line in MacOS with Apple Silicon Chip More install options can be found at: sso.services.box.net/sp/ACS.saml2

[ ]:
```
# update conda
conda update conda
```

[ ]:
```
# create a QIIME2 Amplicon environment
CONDA_SUBDIR=osx-64 conda env create \
  --name qiime2-amplicon-2025.7 \
  --file https://raw.githubusercontent.com/qiime2/distributions/refs/heads/dev/
 ↪2025.7/amplicon/released/qiime2-amplicon-macos-latest-conda.yml

conda activate qiime2-amplicon-2025.7
conda config --env --set subdir osx-64
```

[ ]:
```
# test install
conda deactivate
conda activate qiime2-amplicon-2025.7
```

```
qiime info
```

[ ]:
```
# install setuptools to avoid an error come winter 2025 when a package becomes␣
↪outdated
conda install -n qiime2-amplicon-2025.7 "setuptools<81"
```

3 Setting a Working Directory

Open Up Terminal Command Line Having all of your files in one working directory will reduce file path editing. If we do not do this, each file will need to be called as the entire filepath.

[ ]:
```
conda activate qiime2-amplicon-2025.7
```

Edit the working directory to the folder you are working in. WORKDIR="/Users/yourusername/Desktop/FolderWithData"

[ ]:
```
WORKDIR="PATH"
```

4 Manifest and Metadata Creation

Create each of these as a .csv file and then convert to .tsv using script shown at the end of this section. Manifest and metadata files seem similar, but they have different purposes and different structure.

Manifest File Contains sample-id, absolute-filepath, and direction. No other information is required. Column IDs are case and character sensitive.

For the vast majority of samples run with NETL primers, you will only share the R1 filepath which is the forward direction. You will have R2 files returned to you, but you can ignore them.

A three column CSV minimum:

```
A three column CSV minimum:
sample-id  absolute-filepath                          direction
SampleA    /Users/.../SampleA_R1.fastq.gz             forward
SampleB    /Users/.../SampleB_R1.fastq.gz             forward
SampleC    /Users/.../SampleC_R1.fastq.gz             forward
```

Metadata File Must contain #SampleID and Barcode. No other information is required but can be added in other columns if you want. The first two column headers are case and character sensitive.

A two column CSV minimum:

```
#SampleID   Barcode
SampleA     TAAGGTAAGGTG
SampleB     CGCAAATTCGAC
SampleC     ACACGAGCCACA
```

4.0.1 Convert Manifest and Metadata Files to .tsv

```
[ ]: tr ',' '\t' < $WORKDIR/manifest.csv > $WORKDIR/manifest.tsv
```

```
[ ]: tr ',' '\t' < $WORKDIR/metadata.csv > $WORKDIR/metadata.tsv
```

5 QIIME2 Main Workflow

Files Required

(1) *manifest.tsv*

(2) *metadata.tsv*

(3) *silva-138-99-nb-classifier.qza*

Visualizaitons Visuals are denoted by *.qzv* files and can be dragged into the QIIME2 website to look at.

```
[ ]: qiime tools import \
  --type 'SampleData[SequencesWithQuality]' \
  --input-path $WORKDIR/manifest.tsv \
  --output-path $WORKDIR/demux_single.qza \
  --input-format SingleEndFastqManifestPhred33V2
```

```
[ ]: qiime demux summarize \
  --i-data $WORKDIR/demux_single.qza \
  --o-visualization $WORKDIR/demux_single.qzv
```

```
[ ]: qiime dada2 denoise-single \
  --i-demultiplexed-seqs $WORKDIR/demux_single.qza \
  --p-trim-left 0 \
  --p-trunc-len 250 \
  --o-table $WORKDIR/table.qza \
  --o-representative-sequences $WORKDIR/rep-seqs.qza \
  --o-denoising-stats $WORKDIR/denoising-stats.qza
```

```
[ ]: qiime metadata tabulate \
  --m-input-file $WORKDIR/denoising-stats.qza \
  --o-visualization $WORKDIR/denoising-stats.qzv
```

```
[ ]: qiime feature-table summarize \
  --i-table $WORKDIR/table.qza \
  --o-visualization $WORKDIR/table.qzv \
  --m-sample-metadata-file $WORKDIR/metadata.tsv

qiime feature-table tabulate-seqs \
  --i-data $WORKDIR/rep-seqs.qza \
  --o-visualization $WORKDIR/rep-seqs.qzv
```

Using a pretrained classifier is helpful. The file for this classifier is on Rowan's Box, email and ask for it.

Edit the classifier path only.

[ ]:
```
qiime feature-classifier classify-sklearn \
  --i-classifier /Users/.../silva-138-99-nb-classifier.qza \
  --i-reads $WORKDIR/rep-seqs.qza \
  --p-read-orientation auto \
  --p-n-jobs 0 \
  --o-classification $WORKDIR/taxonomy.qza

qiime metadata tabulate \
  --m-input-file $WORKDIR/taxonomy.qza \
  --o-visualization $WORKDIR/taxonomy.qzv
```

[ ]:
```
qiime taxa filter-table \
  --i-table $WORKDIR/table.qza \
  --i-taxonomy $WORKDIR/taxonomy.qza \
  --p-exclude mitochondria,chloroplast \
  --o-filtered-table $WORKDIR/table.noplants.qza
```

Main Visuals of Interest Taxon barplots and CSV files.

[ ]:
```
qiime taxa barplot \
  --i-table $WORKDIR/table.noplants.qza \
  --i-taxonomy $WORKDIR/taxonomy.qza \
  --m-metadata-file $WORKDIR/metadata.tsv \
  --o-visualization $WORKDIR/taxa-barplot.qzv
```

[ ]:
```
set -euo pipefail
TABLE="$WORKDIR/table.noplants.qza"
TAXON="$WORKDIR/taxonomy.qza"
OUTDIR="$WORKDIR/collapsedtaxonomy"
mkdir -p "$OUTDIR"

for L in 1 2 3 4 5 6 7; do
  qiime taxa collapse \
    --i-table "$TABLE" \
    --i-taxonomy "$TAXON" \
    --p-level $L \
    --o-collapsed-table "$OUTDIR/table_L${L}.qza"

  qiime feature-table relative-frequency \
    --i-table "$OUTDIR/table_L${L}.qza" \
    --o-relative-frequency-table "$OUTDIR/table_L${L}_rel.qza"
```

```
  qiime tools export --input-path "$OUTDIR/table_L${L}.qza"     --output-path␣
↪"$OUTDIR/export_L${L}_counts"
  qiime tools export --input-path "$OUTDIR/table_L${L}_rel.qza" --output-path␣
↪"$OUTDIR/export_L${L}_rel"

  biom convert --input-fp "$OUTDIR/export_L${L}_counts/feature-table.biom"␣
↪--output-fp "$OUTDIR/table_L${L}_counts.tsv" --to-tsv --header-key taxonomy
  biom convert --input-fp "$OUTDIR/export_L${L}_rel/feature-table.biom"   ␣
↪--output-fp "$OUTDIR/table_L${L}_rel.tsv"    --to-tsv --header-key taxonomy

  awk -v OFS=',' 'BEGIN{FS="\t"} NR==1{gsub(/^#OTU ID/,"Taxon",$1)} {for(i=1;
↪i<NF;i++) printf "%s,", $i; print $NF}' "$OUTDIR/table_L${L}_counts.tsv" >␣
↪"$OUTDIR/table_L${L}_counts.csv"
  awk -v OFS=',' 'BEGIN{FS="\t"} NR==1{gsub(/^#OTU ID/,"Taxon",$1)} {for(i=1;
↪i<NF;i++) printf "%s,", $i; print $NF}' "$OUTDIR/table_L${L}_rel.tsv"    >␣
↪"$OUTDIR/table_L${L}_rel.csv"

  rm -rf "$OUTDIR/export_L${L}_counts" "$OUTDIR/export_L${L}_rel"
done
```

If you would like to transpose the rows and columns, use this snippet

[ ]:
```
set -euo pipefail
IN="$WORKDIR/collapsedtaxonomy"
OUT="$WORKDIR/transposedtaxonomy"
mkdir -p "$OUT"

for f in "$IN"/table_L{1..7}_{counts,rel}.csv; do
  [[ -f "$f" ]] || continue
  base=$(basename "$f" .csv)
  out="$OUT/${base}_transposed.csv"

  awk -F',' '
    {
      for (i=1; i<=NF; i++) a[NR,i]=$i
      if (NF>p) p=NF
    }
    END {
      for (i=1; i<=p; i++) {
        for (j=1; j<=NR; j++) {
          printf "%s%s", a[j,i], (j==NR ? ORS : OFS)
        }
      }
    }
  ' OFS=',' "$f" > "$out"

  echo "wrote $out"
```

```
done
```

6 Alterations Required if Data is Not Returned from Sequencing Demulitplexed

The following files are required

(1) rename the smaller seq file: *barcodes.fastq.gz*

(2) rename the larger seq file: *sequences.fastq.gz*

(3) metadata file made earlier in this script: *metadata.tsv*

[ ]:
```
qiime tools import \
  --type EMPSingleEndSequences \
  --input-path $WORKDIR/ \
  --output-path $WORKDIR/emp-single.qza

qiime demux emp-single \
  --i-seqs $WORKDIR/emp-single.qza \
  --m-barcodes-file $WORKDIR/metadata.tsv \
  --m-barcodes-column Barcode \
  --o-per-sample-sequences $WORKDIR/demux-single.qza \
  --o-error-correction-details $WORKDIR/demux-emp-details.qza
```

Proceed to *demux summarize* in the main workflow