XD_QIIME2_processing

Jess Diaz

2023-05-01

Step 1: Import sequencing files

I imported the sequencing files from the BaseSpace backup on one of the lab hard drives into the cluster using the ondemand user interface. I used all samples beginning with the prefix "XEN" from the Kohl1220 run.

Downloaded files were grouped into folders: one folder for each sample, with two fastq.gz files in each folder (F and R). The sample name for these files was located in the folder name, but not in the individual file names. The file names have a plate location and unique sequencing identifier which I can use to find the sample they are associated with if needed.

To use QIIME2, the files needed to be removed from their individual folders and combined into one directory, preferably with more usable file names. The following code was adapted by an example sent by Elizabeth, and is how I conducted this step.

```
# setup: put all sequence folders into one directory, this one is named "main"
# save this as the output directory where the files will ultimately go
output_dir="/ihome/kkohl/jed213/XenopusDevelopment/main"

# loop through all the files within all the folders
# for each one, save a folder name (removes folder name from file name) and a sample name (sample ID ta
# creates a new file path (output_file) within the output directory with the new file name
# moves the file
for file in ./*/*;
  do folder_name=$(echo "$file" | cut -d \/ -f3);
  sample_name=$(echo "$file" | cut -d \/ -f2 | cut -d _ -f1);
  output_file="${output_dir}/${sample_name}_${folder_name}";
  mv -- "$file" "$output_file";
done
```

There are now 608 fastq.gz files in the "main" directory. Empty folders can now be removed, and the process of importing the sequences into QIIME2 can proceed.

Step 2: Generate manifest file

To import the files into QIIME2 I needed to have a manifest file which associates each fastq.gz file with a sample name and whether it is a forward or reverse file. Instructions for this were in Elizabeth's *BaseSpace Protocol* Word Doc.

After making a list of all the sample names that we have sequences for, I created the manifest file to associate each fastq.gz file with a sample name and whether it is a forward or reverse file. Instructions for this were in Elizabeth's *BaseSpace Protocol* Word Doc.

13 of these samples were re-sequenced due to low read counts. Therefore, I have assigned -1 or -2 suffixes to their sample names in the manifest file to differentiate the two. Then later down the pipeline I will remove the sample with the lower read count.

The final file name is called *XD-pe-33-manifest* and saved as a .csv. I then uploaded it to the cluster directory where the sequence files are.

Using QIIME2 in the cluster

The following code is reference for loading qiime2 in the cluster and running an interactive job.

```
# load qiime2
module load qiime2/2022.11

# check it's working
qiime --help

# run interactive job
srun --pty bash

# submit normal job
sbatch [job script]
```

Step 3: Import samples to QIIME2

Instructions for this were in Elizabeth's *BaseSpace Protocol* Word Doc.

```
# first, cd to file directory
# import samples using manifest file
qiime tools import \
  --type 'SampleData[PairedEndSequencesWithQuality]' \
  --input-path XD-pe-33-manifest.csv \
  --output-path paired-end-demux.qza \
  --input-format PairedEndFastqManifestPhred33

# make sure it ran correctly
qiime tools peek paired-end-demux.qza
```

These files have been imported into QIIME2 as Sample Data (paired end with quality info), of the data format SingleLanePerSamplePairedEndFastqDirFmt.

Step 4: Sequence trimming

Instructions for this were in Elizabeth's *BaseSpace Protocol* Word Doc.

Now that the sequences have been uploaded into QIIME2 in a useable format, we can look at the sequence length and quality in order to know how to trim the data.

```
# visualize reads
qiime demux summarize \
  --i-data paired-end-demux.qza \
  --o-visualization paired-end-demux.qzv
```

```
# download .qzv file and view through view.qiime2.org
```

Based on the sequence quality and the recommendations in Elizabeth's protocol, I trimmed using the following parameters. Note, this step had to be done through a bash script because it took too long with an interactive job. Also it should be run outside the working file directory.

```
#!/bin/bash
#SBATCH --partition=htc
#SBATCH --job-name=dada2
#SBATCH --output=outs/dada2.out
#SBATCH --error=errs/dada2.err
#SBATCH --time=0-10:00:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=4
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user=jed213@pitt.edu

module load qiime2/2022.11

cd XenopusDevelopment/main

# trim reads based on sequence length and quality
qiime dada2 denoise-paired \
  --i-demultiplexed-seqs paired-end-demux.qza \
  --o-table table.qza \
  --o-representative-sequences rep-seqs.qza \
  --p-n-threads 0 \
  --verbose \
  --p-trunc-len-f 220 \
  --p-trunc-len-r 220 \
  --p-trim-left-f 19 \
  --p-trim-left-r 20 \
  --o-denoising-stats denoising-stats.qza

# visualize number of reads that passed each filter at each step
qiime metadata tabulate \
  --m-input-file denoising-stats.qza \
  --o-visualization denoising-stats.qzv

# download .qzv file and view through view.qiime2.org
```

Running this gives two important file outputs: table.qza and rep-seqs.qza. It also gives a stats output. At this point we can look at stats from this project, and visualize the number of reads that passed each filter at each step.

Step 5: Set up metadata file

The next steps will require a QIIME2-compatible metadata file. Instructions on proper format are here https://gregcaporaso.github.io/q2book/using/metadata.html. Note that sample names should be in the exact format as they were in the manifest file. Also, any samples not included here will be discarded from

the rest of the analysis (so include only the desired copy of any samples that were sequenced twice, and make sure to include blanks).

To make this file I combined all the metadata files I received from Mauna into one excel sheet: *Xenopus Merged Metadata*, and has an associated .txt file for use in QIIME2 called *XD_qiime_metadata.txt*. There were two samples in the original metadata files (C7 and C20) that were not sequenced so I removed these from the merged metadata. I also added a "days_exposed" category showing how many days the tadpole had been exposed to thyroxine, based on the date thyroxine treatment was started.

Step 6: Remove samples from sequencing redos

This step was done because many of my samples were sequenced twice due to quality control at the sequencing facility. I used the metadata file to exclude any repeated samples.

```
# remove sample duplicates based on what is in the metadata file
qiime feature-table filter-samples \
  --i-table table.qza \
  --m-metadata-file XD_qiime_metadata.txt \
  --o-filtered-table table-filtered.qza
```

More visualization

We can now visualize number of features (reads) per sample. This should match what was in denoising-stats.qzv.

```
# visualize number of features per sample
qiime feature-table summarize \
  --i-table table-filtered.qza \
  --o-visualization table-filtered.qzv \
  --m-sample-metadata-file XD_qiime_metadata.txt

# download .qzv file and view through view.qiime2.org
```

We can also look at what the specific sequences were for each feature. Typically not needed.

```
# visualize sequences for each feature
qiime feature-table tabulate-seqs \
  --i-data rep-seqs.qza \
  --o-visualization rep-seqs.qzv

# download .qzv file and view through view.qiime2.org
```

Step 7: Create phylogenetic trees and classify taxonomy

In order to do other steps, we need a phylogeny for these samples so that we can then assign taxa to the specific features in this dataset. This should also be run through a bash script. Note, this script should be run from outside the main folder, not within it.

```
#!/bin/bash
#SBATCH --partition=htc
#SBATCH --job-name=classify
```

```
#SBATCH --output=outs/classify.out
#SBATCH --error=errs/classify.err
#SBATCH --time=0-15:00:00
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=4
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user=jed213@pitt.edu

module load qiime2/2022.11

# create phylogenetic trees
qiime phylogeny align-to-tree-mafft-fasttree \
  --i-sequences main/rep-seqs.qza \
  --o-alignment main/aligned-rep-seqs.qza \
  --o-masked-alignment main/masked-aligned-rep-seqs.qza \
  --o-tree main/unrooted-tree.qza \
  --o-rooted-tree main/rooted-tree.qza

# classify features to taxa according to the Greengenes database
qiime feature-classifier classify-sklearn \
  --i-classifier training-feature-classifiers/2022.10.backbone.v4.nb.qza \
  --i-reads main/rep-seqs.qza \
  --o-classification main/taxonomy.qza

# visualize each feature associaton with taxa and confidence
qiime metadata tabulate \
  --m-input-file main/taxonomy.qza \
  --o-visualization main/taxonomy.qzv

# download .qzv file and view through view.qiime2.org
```

Step 8: Filter table based on taxonomy

Now that we know the taxonomy associated with each sequence, we can remove any undesired taxa. Specifically, I will remove an reads that are not mapped to a phyla, are mitochondria, are chloroplasts, or are not Bacteria.

```
# filter taxa with no phylum, not bacteria, or are mitochondria or chloroplasts
qiime taxa filter-table \
  --i-table table-filtered.qza \
  --i-taxonomy taxonomy.qza \
  --p-include p__ \
  --p-exclude mitochondria,chloroplast,archaea \
  --o-filtered-table table-filt-bytaxa.qza
```

We can also visualize taxa in each sample (besides being informative, this can help confirm whether this filtering was done correctly).

```
# create taxa visualization
qiime taxa barplot \
  --i-table table-filt-bytaxa.qza \
```

```
  --i-taxonomy taxonomy.qza \
  --m-metadata-file XD_qiime_metadata.txt \
  --o-visualization taxa-bar-plots.qzv

# download .qzv file and view through view.qiime2.org
```

We should also look at read counts again, now that many sequences have been removed through this taxonomic filter.

```
# visualize number of reads per sample after removing these taxa
qiime feature-table summarize \
  --i-table table-filt-bytaxa.qza \
  --o-visualization table-filt-bytaxa.qzv \
  --m-sample-metadata-file XD_qiime_metadata.txt
```

Now the files can be imported into R for analysis. I will need the taxonomy.qza, table-filt-bytaxa.qza, rooted-tree.qza, and XD_qiime_metadata.txt.