QIIME2 Workflow

Monday, April 17, 2023

9:51 AM

**Protocol for QIIME2 v2022.11**

Done in csh; some of the below commands may not work in a bash shell (for ex: setenv is not a bash command. The bash equivalent is export. For example, in bash the command for setting a TMPDIR would be export TMPDIR="/work/WW2DW/Sarah\_Brown\_Microplastics/temp"

Useful Linux commands:

* + ls – lists files in a directory
  + cd – change directory
  + pwd – prints the working directory
  + wget – great for downloading things, especially from a webpage
  + cp – copy a file to a new directory; can use cp \*/directory to copy all files in a directory
  + less – shows the first few lines of a file
  + squeue -u <username> - shows jobs that you’ve submitted to SLURM
  + module avail – lists all programs currently available
  + module add – loads a program
  + module list – lists all programs you currently have loaded
  + zip -r – zips a folder; -r is the recursive command that keeps the file structure the same

Useful SLURM commands:

* + sinfo – views system info
  + sbatch <jobscript> - submits a batch job
  + srun –x11 –pty csh -i – submits an interactive job
  + squeue – shows running jobs
  + scontrol show job <jobid> - show job details
  + scancel <jobid> - cancels a job

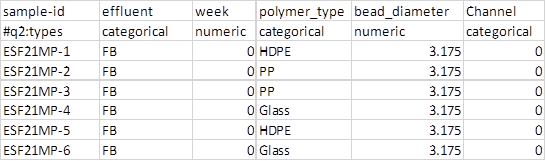
Sbatch options:

|  |  |  |
| --- | --- | --- |
| Job name​ | --job-name=<jobname>​ | Name of job script​ |
| Partition​ | --partition=<partition>​ | compute​ |
| Account​ | --account=<account>​ | Your primary group (minus –hpc)​ |
| Group ID​ | --gid=<group>​ | Your primary group​ |
| Number of tasks​ | --ntasks=<ntasks>​ | 1​ |
| Maximum time​ | --time=<hh:mm:ss>​ | 1:00:00 (1 hour)​ |
| Standard output​ | --output=<outfile>​ | slurm-<jobid>.out​ |
| Standard error​ | --error=<errfile>​ | slurm-<jobid>.out​ |
| Environment vars.​ | --export=NONE​ | Export all environment vars.​ |

Protocol

* + **Create a sample metadata file**

Sample metadata files (which will be used later after samples are denoised and dereplicated in DADA2) should be created in a .tsv or .txt file format. [This page](https://docs.qiime2.org/2022.11/tutorials/metadata/) details the metadata formatting requirements for import to QIIME2. You need an identifier column labeled as sample-id (or any of the other applicable name types listed on the metadata formatting requirements page). Note that whatever you name the samples in the sample-id column should be the same name that you’ll enter in the sample-id column when we create the sample manifest list in step 4a. Here is what the first few lines of my sample metadata sheet looks like:



Additional data can also be added to this file later on. This file should also be checked for errors using [keemei](https://keemei.qiime2.org/).

* + **Checking the files received from the sequencing center**

The fastq files received are in [Casava 1.8 demultiplexed (paired-end) format](https://docs.qiime2.org/2022.11/tutorials/importing/#sequence-data-with-sequence-quality-information-i-e-fastq). There are two fastq.gz files for each sample in the study, each containing the forward or reverse reads for that sample. The file name includes the sample identifier. The forward and reverse read file names for a single sample might look like L2S357\_15\_L001\_R1\_001.fastq.gz and L2S357\_15\_L001\_R2\_001.fastq.gz, respectively. The underscore-separated fields in this file name are:

a. the sample identifier,

b. the barcode sequence or a barcode identifier,

c. the lane number,

d. the direction of the read (i.e. R1 or R2), and

e. the set number.

* + **Copying the files into your scratch space (i.e., your work directory)**

First, change your directory to where the fastq files are located; here I use the following code:

cd /work/WW2DW/microplastics\_test/microplastics\_test\_data

Check for the files:

ls

This should produce a list of all the fastq files in the folder you are currently in.

Copy all files to your scratch space:

cp \* /work/WW2DW/Sarah\_Brown\_Microplastics/microplastics-paired-end-sequences

* + [**Importing the files into QIIME2**](https://docs.qiime2.org/2022.11/tutorials/importing/#sequence-data-with-sequence-quality-information-i-e-fastq)

Now that we’ve transferred the files to our scratch space, we can import them into QIIME2 using the Casava 1.8 file format import method.

Here’s what you will need in order to use this method:

* + one fastq.gz file that contains the forward sequence reads for each sample, and
  + one fastq.gz file that contains the reverse sequence reads for each sample.

All of these files should be together in the same folder. Before we import these files, we can take a look at the file contents to double check that the files contain everything we need using the following code:

less (any file name)

For example:

less ESF21MP-99\_S93\_L001\_R2\_001.fastq.gz

This allows you to scroll through the file contents one page at a time without having to load the whole (very large, very long) file. In this example of the reverse sequence reads, we see the following:

@M04159:12:000000000-JLWR5:1:1101:14336:1903 2:N:0:93

TGACTACCAGGGTTTCTAATCCTGTTTGCTCCCCACGCTTTCGCACCTCAGCGTCAGTATCGAGCCAGTGAGCCGCCTTCGCCACTGGTGTTCCTCCGAATATCTACGAATTTCACCTCTACACTCGGAATTCCACTCACCTCTCTCGTCCTCAAGAACAGGAGTTTCAAATGCAGTTCCAAGGTTGAGCCCTGGGATTTCACCTCTGACTTTCCGGTCCGCCTACGTGCGCTTTACGCCCAGTAATTC

The yellow highlighted portion is the reverse primer.

We can see the forward primer (highlighted in yellow below) as well when we view the forward sequence reads for this sample using the less command:

@M04159:12:000000000-JLWR5:1:1101:14336:1903 1:N:0:93

GTGCCAGCAGCCGCGGTAATACGGAGGGTGCTAGCGTTGTTCGGAATTACTGGGCGTAAAGCGCACGTAGGCGGACCGGTAAGTCAGAGGTGAAATCCCAGGGCTCAACCTTGGAACTGCCTTTGAAACTCCTGTTCTTGAGTTCGAGAGAGGTGAGTGGAATTCCGAGTGTAGAGGTGAAATTCGTAGATATTCGGAGGAACACCAGTGGCGAAGGCGGCTCACTGGCTCGTTACTGACGCTGAGGTGCG

**Tip:** use q to escape lists

The first line of this file contains the quality score information for each read. Unlike some fastq file formats (for example, multiplexed paired end files with sequence quality information, Phred+33 format), in this case the forward and reverse barcodes are not contained in the fastq files. Luckily, since these sequences have already been demultiplexed, we don’t need the barcodes in order to import them into QIIME2. We can now proceed with importing the files using the Casava 1.8 file format.

**Tip:** If you have only one fastq.gz file containing all forward sequences, and one fastq.gz file containing all reverse sequences, your data are not demultiplexed. You will need to use a different file import method and conduct demultiplexing yourself. More information is available [here](https://docs.qiime2.org/2022.11/tutorials/importing/#sequence-data-with-sequence-quality-information-i-e-fastq) and [here](https://forum.qiime2.org/t/how-to-know-if-my-sequence-data-demultiplexed/11988).

First, check to see which version of QIIME 2 is available on Atmos:

Module avail

QIIME2 v2022.11 is the latest version of QIIME2 available on Atmos, and is also the default version (so if you load QIIME2 without specifying the version, this is the version that loads).

Now, load the QIIME2 program:

module add qiime

You can also check to see which programs are loaded to your environment:

Module list

If qiime is loaded, it will be listed as a module after running this command.

Now we’ll use the sbatch command to submit the QIIME2 import job to SLURM, which is a job scheduling system for Linux clusters. Submitting the job on the command line using looks like this:

sbatch qiime tools import \--type 'SampleData[PairedEndSequencesWithQuality]' \--input-path microplastics-paired-end-sequences \--input-format CasavaOneEightSingleLanePerSampleDirFmt \--output-path demux-paired-end.qza

This is what the QIIME2 import code looks like outside of SLURM; this is what you’d be submitting to QIIME2 if you weren’t using SLURM:

qiime tools import \

--type 'SampleData[PairedEndSequencesWithQuality]' \

--input-path microplastics-paired-end-sequences \

--input-format CasavaOneEightSingleLanePerSampleDirFmt \

--output-path demux-paired-end.qza

You should now have an output file called paired-end-demux.qza. If you only get a slurm-somenumber.outfile, check it to see your error message.

**Tips:**

**1.** Run the squeue command to see currently running jobs; use squeue -u <yourusername> to view only your specific jobs.

**2.** If you receive a slurm output file with an error message that says “[No space left on device](https://forum.qiime2.org/t/plugin-error-from-feature-classifier-errno-28-no-space-left-on-device/17557)” try changing your TMPDIR using the following code: setenv TMPDIR /work/group/somefolder

In my case, it was: setenv TMPDIR /work/WW2DW/Sarah\_Brown\_Microplastics/temp

* + **Summarize demultiplexed files**

Now we’ll create a .qzv file from the .qza file that we just generated so that we can create a summary of the results, including the number of sequences acquired from each sample and the distribution of the quality scores for each read.

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

The output here is a .qzv file that can be opened on <https://view.qiime2.org/>

Here is some of the output from the demux-paired-end.qzv file:

Demultiplexed sequence counts summary

forward reads reverse reads

Minimum 30796 30796

Median 56899.0 56899.0

Mean 58750.225166 58750.225166

Maximum 128241 128241

Total 8871284 8871284

We’ll be using this plot to determine where to trim each sequence in the next step.

* + [**Denoise and dereplicate with DADA2**](https://docs.qiime2.org/2022.11/tutorials/atacama-soils/)

In this step, we’ll be merging paired reads, trimming the sequences, and filtering out chimeric sequences. Note that at this point the sequences still have primers attached; we amplified the hypervariable region V4 of the 16s rRNA gene using the 515F (5’-GTGCCAGCAGCCGCGGTAA -3’) and 806R (5’-TGACTACCAGGGTTTCTAAT -3’) primers ([reference](https://earthmicrobiome.org/protocols-and-standards/16s/)). The presence of these primers in the sequences can be seen in the first few lines of each sequence file (as shown above, in step 4). Due to this, we’ll have to trim at least 20 bases from the beginning of the forward and reverse reads of each sequence. Neglecting to trim primers from sequences before taxonomic assignment could impact assignment and/or result in the generation of artificial ASV’s, so primers **must** be removed prior to that point (more info on the importance of primer removal [here](https://forum.qiime2.org/t/different-taxonomic-classification-with-or-without-primers-trimming/10375) and [here](https://forum.qiime2.org/t/forward-and-reverse-primer-removal/23683)).

You can also use deblur to denoise sequences, but I chose to use dada2 because it uses both forward and reverse reads (however, it is much slower than deblur). If you have more than one set of sequences (i.e., separate runs that you plan to merge later), the length at which the sequences from each run are trimmed must be the same, though forward and reverse sequences can have different trimming parameters. I used [this tutorial](https://benjjneb.github.io/dada2/tutorial.html) and [this post](https://forum.qiime2.org/t/finding-primers-in-raw-files-and-quality-control/1621) on the QIIME 2 forum to determine where to trim my sequences. When quality scores begin to drop off significantly (usually going below 20), that is an indication of where they should be trimmed.

**Tip:** If you’re unsure of where to trim your sequences, there is now a program called FIGARO available that can choose trimming parameters for you. You can find a tutorial on FIGARO (including installation instructions) [here](http://john-quensen.com/tutorials/figaro/). If you choose to use FIGARO, I recommend using it in conjunction with your examination of the quality plots to ensure that the recommended trimming parameters are reasonable.

In addition to trimming sequences, dada2 performs chimera checking with the --p-chimera-method command. However, this is not a required command; if you leave it at default (as I did here) it will filter chimeric sequences using “consensus”, where chimeras are detected in samples individually, and chimeras that are found in a significant number of samples are removed.

Additionally, subsequent abundance filtering (such as filtering out singletons) does not need to be conducted ([source](https://forum.qiime2.org/t/filter-features/1493)). According to Callahan et al. (2016), “Note that the abundance p-value is calculated conditional on at least one sequence being observed. As a result, all singleton sequences have an abundance p-value of 1 and are never judged inconsistent with the error model. This means that singletons cannot form their own partitions, and DADA2 will not infer singleton sequences. The effect of this is similar in practice to the UPARSE developer’s recommendation to remove singleton sequences before picking OTUs, and in both cases is driven by the difficulty in robustly differentiating singleton errors from real singleton sequences.” Essentially, it seems that singletons are excluded so that their presence does not affect OTU picking. [This post](https://github.com/benjjneb/dada2/issues/92) states that “singletons cannot appear in the raw output of the dada(...) function, but can appear after merging, if for example the reverse read of one member of a doubleton was mis-assigned due to low quality.” If there are any particular sequences that need to be removed after DADA2 denoising, they can be removed later on using [QIIME 2 filtering methods](https://docs.qiime2.org/2017.4/tutorials/filtering/).

Took a look at the quality plots in my demux-paired-end.qzv file; the quality at the start of both the forward and reverse reads starts off a bit low, but then it increases; we have to trim the first 20 bases anyway, since those are primers. After the first 20 bases quality increases, so I’m not trimming any further from the beginning of the sequences. The quality scores at the end of the forward and reverse sequences decreases a bit but is still good, so I’m just going to trim off the last base (#251) where quality significantly decreases, meaning that I’ll input a trunc-len-f and trunc-len-r of 250.

QIIME2 code:

qiime dada2 denoise-paired \

--i-demultiplexed-seqs demux-paired-end.qza \

--p-trim-left-f 20 \

--p-trim-left-r 20 \

--p-trunc-len-f 250 \

--p-trunc-len-r 250 \

--o-representative-sequences MP-rep-seqs.qza \

--o-denoising-stats dada2-denoising-stats.qza \

--o-table dada2-table

Entered into SLURM as:

sbatch --partition=largemem --ntasks=4 --mem=100g --nodes=2 --time=24:00:00 qiime dada2 denoise-paired \--i-demultiplexed-seqs demux-paired-end.qza \--p-trim-left-f 20 \--p-trim-left-r 20 \--p-trunc-len-f 250 \--p-trunc-len-r 250 \--o-representative-sequences MP-rep-seqs.qza \--o-denoising-stats dada2-denoising-stats.qza \--o-table dada2-table

Also, note that this is the longest step in the analysis, so this may take some time. Once this process is complete, you should have a stats file that tell you how many reads were filtered, denoised, merged, and non-chimeric (i.e., the dada2-denoising-stats.qza file) and artifacts containing the FeatureTable (i.e., the dada2-table file) and corresponding feature sequences available in a FeatureData file (i.e., the MP-rep-seqs-qza file):

dada2-denoising-stats.qza

MP-rep-seqs.qza

dada2-table.qza

Now generate summaries of those as in [this tutorial](https://docs.qiime2.org/2019.4/tutorials/atacama-soils/):

sbatch --partition=largemem --ntasks=4 --mem=100g --nodes=2 --time=24:00:00 qiime feature-table summarize \--i-table dada2-table.qza \--o-visualization dada2-table.qzv \--m-sample-metadata-file MP-metadata.tsv

sbatch --partition=largemem --ntasks=4 --mem=100g --nodes=2 qiime feature-table tabulate-seqs \--i-data MP-rep-seqs.qza \--o-visualization MP-rep-seqs.qzv

sbatch --partition=largemem --ntasks=4 --mem=100g --nodes=2 qiime metadata tabulate \--m-input-file dada2-denoising-stats.qza \--o-visualization dada2-denoising-stats.qzv

* + **Merging denoised data**

If you have more than one sequencing run and you’d like to merge those runs into a single dataset for analysis, this is the point at which you should do that.

Merging example on the forum [here](https://forum.qiime2.org/t/merged-multiple-runs-several-different-experiments-when-to-best-split-them-up/4427) and [here](https://forum.qiime2.org/t/dada2-samples-from-multiple-miseq-sequencing-runs/1311).

* + **Determining taxonomic classification using a Naïve Bayes classifier (q2-feature-classifier)**

Instead of clustering based on OTUs, we’ll be using ASVs, which are higher resolution and may even be able to differentiate organisms at the strain level (see [this post](https://forum.qiime2.org/t/start-analyzing-demultiplex-fastq-files/14465/2)). As stated on the QIIME2 website: “ASVs are a more recent development and provide better resolution in features than traditional OTU-based methods. ASVs can separate features based on differences of a single nucleotide in sequences of 400 bp or more, a resolution not possible even with 99% identity OTU clustering.”

See [this tutorial](https://docs.qiime2.org/2022.11/tutorials/pd-mice/) for more details. The [Moving Pictures Tutorial](https://docs.qiime2.org/2022.11/tutorials/moving-pictures/) also shows how to use the q2-feature-classifier.

Additionally, you will need to select a reference sequence database for use during this step. There are multiple that you can choose from; some of the most common for comparison with bacteria 16S rRNA sequences are SILVA, NCBI, and Greengenes. Some information on the differences between these is available [here](https://forum.qiime2.org/t/green-genes-vs-silva/21248) and [here](https://forum.qiime2.org/t/what-is-the-difference-between-greengenes-and-silva/1560). There is also a QIIME2 plugin called RESCRIPt that allows you to retrieve sequences from various reference databases – a tutorial is available [here](https://forum.qiime2.org/t/processing-filtering-and-evaluating-the-silva-database-and-other-reference-sequence-data-with-rescript/15494).

**Tip:** There are two parts to this section; if a feature classifier trained on sequences with the same primers as your samples doesn’t already exist, you’ll have to create your own feature classifier (start with part **a** below). However, if you already have a feature classifier (you can find some pre-made ones [here](https://docs.qiime2.org/2022.11/data-resources/)) you can jump to part **b**. For this tutorial, I was able to find a pre-existing feature classifier that was trained on sequences that had the same primers as mine (silva-138-99-515-806-nb-classifier.qza) so I jumped straight to part b. However, I’ve included information on how to train your own feature classifier below.

* + [**Train a feature classifier**](https://docs.qiime2.org/2019.4/tutorials/feature-classifier/)[[BS1]](https://usepa-my.sharepoint.com/personal/brown_sarah_epa_gov/Documents/QIIME2/QIIME2%20Atmos%20workflow.docx#_msocom_1)

“Taxonomic classifiers perform best when they are trained based on your specific sample preparation and sequencing parameters, including the primers that were used for amplification and the length of your sequence reads. Therefore, in general you should follow the instructions in [Training feature classifiers](https://docs.qiime2.org/2022.11/tutorials/feature-classifier/) with q2-feature-classifier to train your own taxonomic classifiers.”

QIIME2 compatible reference sequences can be found [here](https://docs.qiime2.org/2022.11/data-resources/). I’m using release 138 of the Silva 16S rRNA database. We need two things in order to train the classifier: reference sequences and their corresponding taxonomic classifications. I uploaded the Silva database to my directory using the wget command:

wget <https://www.arb-silva.de/fileadmin/silva_databases/release_138/Exports/SILVA_138_SSURef_NR99_tax_silva.fasta.gz>

wget /pine/scr/s/b/sbrown21/Sequences-2020/QIIME2-processing-Indian\_Southern\_Oceans/ <https://www.arb-silva.de/fileadmin/silva_databases/qiime/Silva_132_release.zip>

unzip Silva\_132\_release.zip

And I saved the unzipped Silva database in the SILVA\_132\_QIIME\_release folder. To train the classifier, I need to use the files in the rep\_set folder, not in the rep\_set\_aligned folder, according to [this post](https://forum.qiime2.org/t/classifier-training-questions/1162) on the forum. [This post](https://forum.qiime2.org/t/feature-classifier-and-training-percent-alignment/5598/3) also clarifies why using the rep\_set data with a 99% similarity cutoff is the best option: “That cutoff is just specifying the percent similarity used for clustering sequences in that database. Higher will be more specific, i.e., have better taxonomic labels and preserve more unique seq information, and should almost always be used (the only reason to go lower is for memory issues). This has no relation to OTU picking/denoising methods or parameters. Use 99%.”

For the taxonomy file, I had to choose between consensus taxonomy or majority taxonomy; here is the difference between them: “A user of the Silva119 data pointed out that the taxonomy with the SILVA119 release is based only upon the taxonomy string of the representative sequence for the cluster of reads, which could lead to incorrect confidence in taxonomy assignments at the fine level (genus/species). To address this, I have endeavoured to create taxonomy strings that are either consensus (all taxa strings must match for every read that fell into the cluster) or majority (greater than or equal to 90% of the taxonomy strings for a given cluster). If a taxonomy string fails to be consensus or majority, then it becomes ambiguous, moving up the levels of taxonomy until consensus/majority taxonomy strings are met.” There doesn’t seem to be a strict rule on when to use consensus vs. majority taxonomy, and there isn’t a lot of information on the QIIME2 forum about the differences in results from either; I decided to use the majority taxonomy. This should allow for less unassigned sequences, since consensus would require all labels to be the same and would leave no room for mis-labeled reads.

Now I need to import the reference sequences and their taxonomy:

qiime tools import \--type 'FeatureData[Sequence]' \--input-path /pine/scr/s/b/sbrown21/Sequences-2020/QIIME2-processing-Indian\_Southern\_Oceans/SILVA\_132\_QIIME\_release/rep\_set/rep\_set\_16S\_only/99/silva\_132\_99\_16S.fna \--output-path silva\_99\_ref\_seqs.qza

qiime tools import \--type 'FeatureData[Taxonomy]' \--input-format HeaderlessTSVTaxonomyFormat \--input-path/pine/scr/s/b/sbrown21/Sequences-2020/QIIME2-processing-Indian\_Southern\_Oceans/SILVA\_132\_QIIME\_release/taxonomy/16S\_only/99/majority\_taxonomy\_7\_levels.txt \--output-path silva\_99\_ref\_taxonomy.qza

Now we’ll extract reference reads from the unaligned Silva 99% reference sequences that correspond to the specific primers that I used. According to [this post](https://forum.qiime2.org/t/how-can-i-train-classifier-for-paired-end-reads/1512) on the forum, you shouldn’t use the --p-trunc-len parameter for paired-end reads, so I exclude that parameter when extracting reads. Here’s a similar statement from the [feature classifier](https://docs.qiime2.org/2019.4/tutorials/feature-classifier/) tutorial: “For classification of paired-end reads and untrimmed single-end reads, we recommend training a classifier on sequences that have been extracted at the appropriate primer sites, but are not trimmed.”

sbatch -p general -t 4-00:00:00 --mem=6000 --ntasks 1 --wrap=“qiime feature-classifier extract-reads \--i-sequences silva\_99\_ref\_seqs.qza \--p-f-primer AGAGTTTGATCCTGGCTCAG \--p-r-primer GCTGCCTCCCGTAGGAGT \--o-reads extracted-ref-seqs.qza”

And now a Naive Bayes classifier can be trained:

sbatch -p general -t 4-00:00:00 --mem=30000 --ntasks 1 --wrap=“qiime feature-classifier fit-classifier-naive-bayes \--i-reference-reads extracted-ref-seqs.qza \--i-reference-taxonomy silva\_99\_ref\_taxonomy.qza \--o-classifier 8F\_338R\_classifier.qza”

* + **Classify the sequences**

Now that you’ve either created a classifier or downloaded a pre-existing one, we can finally assign taxonomy to our sequences based on the classifier:

sbatch --partition=largemem --ntasks=4 --mem=100g --nodes=2 --time=24:00:00 qiime feature-classifier classify-sklearn \--i-reads MP-rep-seqs.qza \--i-classifier silva-138-99-515-806-nb-classifier.qza \--o-classification MP-taxonomy.qza

Examine the taxonomy associated with the sequences:

sbatch --partition=largemem --ntasks=4 --mem=100g --nodes=2 --time=24:00:00 qiime metadata tabulate \--m-input-file MP-taxonomy.qza \--o-visualization MP-taxonomy.qzv

Now[[SB2]](https://usepa-my.sharepoint.com/personal/brown_sarah_epa_gov/Documents/QIIME2/QIIME2%20Atmos%20workflow.docx#_msocom_2) , finally, we can create a barplot of the taxa in each sample:

sbatch --partition=largemem --ntasks=4 --mem=100g --nodes=2 --time=24:00:00 qiime taxa barplot \--i-table dada2-table.qza \--i-taxonomy MP-taxonomy.qza \--m-metadata-file MP-metadata.tsv \--o-visualization MP\_taxa\_barplot.qzv

* + **Prepare results for import into phyloseq**

At this point, further analyses on our data can be conducted in QIIME 2 (see the ‘Analyze feature table and gain insight’ section in [this](https://docs.qiime2.org/2022.11/tutorials/qiime2-for-experienced-microbiome-researchers/#analyze-feature-table-and-gain-insight) tutorial); however, I’m choosing to export my .qza files from QIIME 2 containing my results for further analysis in R, as R has a wide variety of packages available for advanced data analyses that would be difficult to perform in QIIME 2. I primarily use the [phyloseq](https://joey711.github.io/phyloseq/) package for microbial analyses in R, in conjunction with ggplot2 and other [Tidyverse](https://www.tidyverse.org/) packages.

There are multiple different ways to export your data for use in phyloseq (including, for example, exporting your data into a [BIOM table](https://forum.qiime2.org/t/exporting-and-modifying-biom-tables-e-g-adding-taxonomy-annotations/3630)), but here we’ll be using the [qiime2R](https://github.com/jbisanz/qiime2R) package to easily export our data to R. The benefit of using qiime2R instead of other methods is that it maintains the provenance information associated with every QIIME 2 artifact and doesn’t result in manual editing of the original QIIME 2 artifacts; this way, the settings used to create a specific artifact can be traced back.

* + **Create a phylogenetic tree for export**

First, in addition to the .qza files we created above, we also need to create a phylogenetic tree that can be used when we import all of our data into a phyloseq object in R. The [Moving Pictures Tutorial](https://docs.qiime2.org/2022.11/tutorials/moving-pictures/#taxonomic-analysis) shows how to create both a rooted and unrooted tree in one command; here is the code that I used:

sbatch --partition=largemem --ntasks=4 --mem=100g --nodes=2 --time=24:00:00 qiime phylogeny align-to-tree-mafft-fasttree \--i-sequences MP-rep-seqs.qza \--o-alignment aligned-rep-seqs.qza \--o-masked-alignment masked-aligned-rep-seqs.qza \--o-tree unrooted-tree.qza \--o-rooted-tree rooted-tree.qza

From the output, we’ll be downloading the rooted tree for use in R. Additionally, for more information about tree building in QIIME2, see the [QIIME2 documents on phylogenetic inference](https://docs.qiime2.org/2022.11/tutorials/phylogeny/).

* + **Download qiime artifacts for import into phyloseq**

Here I downloaded my dada2-table.qza file, MP-metadata.tsv file, MP-taxonomy.qza file, and rooted-tree.qza file so that I could use RStudio on my laptop rather than loading R on Atmos. Once qiime2R is installed and loaded in R (instructions in [this](https://github.com/jbisanz/qiime2R) tutorial as well as on the [qiime2 forum](https://forum.qiime2.org/t/tutorial-integrating-qiime2-and-r-for-data-visualization-and-analysis-using-qiime2r/4121)) you can enter the following into R to create a phyloseq object:

physeq<-qza\_to\_phyloseq(

features="inst/artifacts/2020.2\_moving-pictures/table.qza",

tree="inst/artifacts/2020.2\_moving-pictures/rooted-tree.qza",

taxonomy="inst/artifacts/2020.2\_moving-pictures/taxonomy.qza",

metadata = "inst/artifacts/2020.2\_moving-pictures/sample-metadata.tsv"

)

You can now continue your analyses in R.

**References**

Callahan, B. J., McMurdie, P. J., Rosen, M. J., Han, A. W., Johnson, A. J. A., & Holmes, S. P. (2016). DADA2: high-resolution sample inference from Illumina amplicon data. Nature methods, 13(7), 581.

Caporaso, J. G., Lauber, C. L., Walters, W. A., Berg-Lyons, D., Lozupone, C. A., Turnbaugh, P. J., Noah Fierer, N., & Knight, R. (2011). Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample. Proc Natl Acad Sci USA 108, 4516–4522. <http://doi.org/10.1073/pnas.1000080107>