# QIIME2 Moving Pictures Tutorial

**BIT 477/577 Class:** **First, submit the analysis.**

This document describes how to do the QIIME2 Moving Pictures Tutorial on the Henry2 cluster.

For doing the tutorial and for running QIIME2 in general, you will do file transfer on a login node, calculations on a compute node, and do visualization elsewhere. In particular:

1) To get the data
   a) Download (using `wget`) all the files while on the login node.
2) To run QIIME2
   a) **For testing:** Try some initial calculations on an interactive compute node.
   b) **For Production:** Cut/paste all instructions into a batch script and submit to LSF.
3) To visualize QIIME2 outputs
   a) **For the tutorial:** Copy the data to a local machine and visualize with QIIME2View.
   b) **For Production:** For larger files in actual production cases, you may want to use the HPC-VCL. This will allow you to do visualizations without moving the output files to a local machine.

Here are some considerations:

- Downloading files (`wget`) can only be done from a login node or HPC-VCL.
- `qiime tools view` uses an internet browser, so it can only be used from the HPC-VCL or [somewhere not on HPC].
- QIIME2 *cannot* be run directly on a login node, but QIIME2 can be run directly on an interactive compute node for the purposes of testing.
- Other than for testing commands, all QIIME2 production runs should be done by submitting a script to the batch scheduler. Required memory should be calculated (estimated). If the required memory is expected to be large, but you really don't know and can't figure out how large, you should use **`-x`**. Also, again, the "wget" command cannot be run from a compute node, and this includes those submitted using bsub.

**Prerequisites:**

- For users with no Linux experience, complete Essential Linux Commands - Fluency Drill.
- For all users, complete Part 1, Part 2, and Part 3 of the Basic HPC Workshop video series.

**Access to QIIME2:**

If you are in the BIT 577 course, you should already have access. For general users, email oit_hpc@help.ncsu.edu and request to be added to the **bioinfo** software group. Test to see if you are in the bioinfo group by typing:

```
groups
```

**Initialize Conda:**

If at this point you have never activated a conda environment, you should do the following commands, then log out and back in again:

```
module load conda
conda init tcsh
```

See the HPC Website for additional documentation for using Conda.

**First do the tutorial on an interactive session on a compute node.**

1. Change directory to `/share`; you will download and generate a significant amount of data.
   **For BIT577f19,**
   ```
   cd /share/bit577f19/$USER/tutorial
   ```
   **For general users,**
   ```
   cd /share/$GROUP/$USER
   ```
2. Begin the tutorial, starting by making the directory.
3. First, do all of the `wget` commands on the login node. You do not need to change shell or activate Conda to download the files with `wget`. **Go through the entire tutorial and `wget` all necessary files before proceeding.**
4. Request an interactive session. This is for 30 minutes. If it is taking too long to reserve the session, request 10 minutes to land in the `debug` queue:
   ```
   bsub -Is -W 30 -n 1 tcsh
   ```
5. Check that you are NOT on a login node by looking at the prompt or typing:
   ```
   echo $HOSTNAME
   ```
6. Check that you are in the same directory you created before doing `wget`.
7. Set the Conda/QIIME2 environment:
   ```
   conda activate /usr/local/usrapps/bioinfo/qiime2/env_2020.2
   ```
8. Cut and paste the rest of the tutorial *as-is*, except that you will not be able to do the `qiime2 tools view`. To visualize the data, you will transfer the outputs to a local computer and use QIIME2View.

**Tar files, copy to local computer, and visualize using QIIME2View:**

Put all of the visualization output into a folder and tar (put in a package for easy file transfer).
**While on Henry2**, do:

```
mkdir my_output
mv *.qzv my_output
tar -cvf my_output.tar my_output
```

Transfer the files to your local computer.
**While in a local terminal**, change the highlighted items as appropriate and do:

```
scp unityid@login.hpc.ncsu.edu:/share/group/unityid/qiime2-moving-pictures-tutorial/my_output.tar .
tar -xvf my_output.tar
```

On your local computer, open QIIME2View and Drag/Drop or Click/Open the output files.

### Redo the tutorial by submitting a job to LSF.

1. Go back to the directory used in the interactive session
   ```
   cd /share/$GROUP/$USER/qiime2-moving-pictures-tutorial
   ```
2. Remove the old outputs.  Keep the downloaded input files.
   ```
   rm -r my_output*
   rm -r *.qza
   ```
3. Create a text file called submit_qiime2_tutorial.csh
4. Copy and paste the example for the basic batch script from the HPC Website into submit_qiime2_tutorial.csh.
5. Add the Conda/QIIME2 environment to the script:
   ```
   conda activate /usr/local/usrapps/bioinfo/qiime2/env_2020.2
   ```
6. Remove 'mycode.exe', and copy and paste all of the QIIME2 commands into submit_qiime2_tutorial.csh.  Do not copy the commands for wget or qiime view.

### Run a batch job

To run a job defined in the LSF submit script submit_qiime2_tutorial.csh, do

```
bsub < submit_qiime2_tutorial.csh
```

The following is the contents of the submit script submit_qiime2_tutorial.csh, which is a small portion of the Moving Pictures tutorial. The run took about 5 minutes and used less than 1.3GB memory (see highlighted output). **To use such a script with bigger data, first test the memory requirements for your data set** to check if you should set the memory usage or request `-x`. For a non-parallel job, set -n 1. For parallel jobs, specify either the proper number of threads, or if the threads depend on the cores available, you may specify `-x.` At this time, QIIME2 does not work under distributed memory parallelism (i.e. MPI), so you must specify the `shared_memory` queue or specify `span[hosts=1].` See Part 4 of the Basic HPC Workshop for more details on submitting parallel jobs.

### submit_qiime2_tutorial.csh

```
#!/bin/tcsh
#BSUB -n 1
#BSUB -W 30
#BSUB -J qiime2_tutorial
#BSUB -o stdout.%J
#BSUB -e stderr.%J
conda activate /usr/local/usrapps/bioinfo/qiime2/env_2020.2
[Copy and paste the steps of the tutorial you want to submit as a batch job]
conda deactivate
```

### LSF output for sample job:

```
[lllowe@login04 qiime2-moving-pictures-tutorial]$ more stdout.595924
Imported emp-single-end-sequences as EMPSingleEndDirFmt to emp-single-end-sequences.qza
Hello, I have finished tools import
Saved SampleData[SequencesWithQuality] to: demux.qza
Saved ErrorCorrectionDetails to: demux-details.qza
I am about to demux
Saved Visualization to: demux.qzv
Saved FeatureTable[Frequency] to: table-dada2.qza
Saved FeatureData[Sequence] to: rep-seqs-dada2.qza
Saved SampleData[DADA2Stats] to: stats-dada2.qza
Saved Visualization to: stats-dada2.qzv

------------------------------------------------------------
Sender: LSF System <lsfadmin@n3n1-7>
Subject: Job 595924: <qiime2_tutorial> in cluster <henry2> Done

Job <qiime2_tutorial> was submitted from host <login04> by user <lllowe> in cluster <henr
y2> at Thu Apr  9 16:20:18 2020
Job was executed on host(s) <n3n1-7>, in queue <cos_serial>, as user <lllowe> in cluster
<henry2> at Thu Apr  9 16:20:24 2020
</home/lllowe> was used as the home directory.
</share/Support/lllowe/qiime2-moving-pictures-tutorial> was used as the working directory
.
Started at Thu Apr  9 16:20:24 2020
Terminated at Thu Apr  9 16:25:23 2020
Results reported at Thu Apr  9 16:25:23 2020

Your job looked like:
```

```
-------------------------------------------------------------
# LSBATCH: User input
#!/bin/tcsh
#BSUB -n 1
#BSUB -W 30
#BSUB -J qiime2_tutorial
#BSUB -o stdout.%J
#BSUB -e stderr.%J
conda activate /usr/local/usrapps/bioinfo/qiime2/env_2020.2

#Starting Tools import
qiime tools import \
  --type EMPSingleEndSequences \
  --input-path emp-single-end-sequences \
  --output-path emp-single-end-sequences.qza

echo "Hello, I have finished tools import"

qiime demux emp-single \
  --i-seqs emp-single-end-sequences.qza \
  --m-barcodes-file sample-metadata.tsv \
  --m-barcodes-column barcode-sequence \
  --o-per-sample-sequences demux.qza \
  --o-error-correction-details demux-details.qza

echo "I am about to demux"

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

qiime dada2 denoise-single \
  --i-demultiplexed-seqs demux.qza \
  --p-trim-left 0 \
  --p-trunc-len 120 \
  --o-representative-sequences rep-seqs-dada2.qza \
  --o-table table-dada2.qza \
  --o-denoising-stats stats-dada2.qza

qiime metadata tabulate \
  --m-input-file stats-dada2.qza \
  --o-visualization stats-dada2.qzv


-------------------------------------------------------------

Successfully completed.

Resource usage summary:

    CPU time :                                    277.87 sec.
    Max Memory :                                  1298.70 MB
    Average Memory :                              690.47 MB
    Total Requested Memory :                      -
    Delta Memory :                                -
    Max Swap :                                    1 MB
    Max Processes :                               5
    Max Threads :                                12
    Run time :                                   313 sec.
    Turnaround time :                            305 sec.

The output (if any) is above this job summary.



PS:

Read file <stderr.595924> for stderr output of this job.
```