UNL – QIIME2 Tutorial

Authors: JJM.Riethoven, Qidong Jia, Chia-Sin Liew.
BCRF, Center for Biotechnology, University of Nebraska-Lincoln

Acknowledgements: data sets used in this tutorial are courtesy of Dr. Daniel Schachtman and Ms. Morgan McPherson (rhizosphere/root/soil microbiome).

Last update: 22 March 2018.

1 Introduction

The tutorial today will give a high level overview of some of the steps that are involved in the analysis of metagenomics data. We will be using a new incarnation of the QIIME software, namely QIIME2 for the majority of the analysis steps involved in this tutorial. Due to time-constraints we also have prepared input files that can be analyzed in a reasonable amount of time (i.e. these are not complete data sets).

During the tutorial you should be able to copy&paste from this document – if you get an error please let us know. We also have short cut script for most of the steps.

1.1 Requirements checkup

In short, you need:

1. An ssh client (e.g. terminal on Mac, PuTTY on Windows)
2. A file transfer program (preferably Globus)
3. An account with the Holland Computing Center, and DUO authentication set up
4. A Google account (a temporary account is fine)

If any of these are currently not working/fulfilled, please let us know immediately.

1.2 Logging in to the Holland Computing Center

You will either connect via PuTTY (Windows) or an ssh command in a terminal (Mac, Linux).

1. PuTTY. Important settings are hostname "crane.unl.edu" and connection type 'SSH'.
2. SSH command (on Mac): `ssh yourusername@crane.unl.edu`

Let us know if you have trouble logging in.

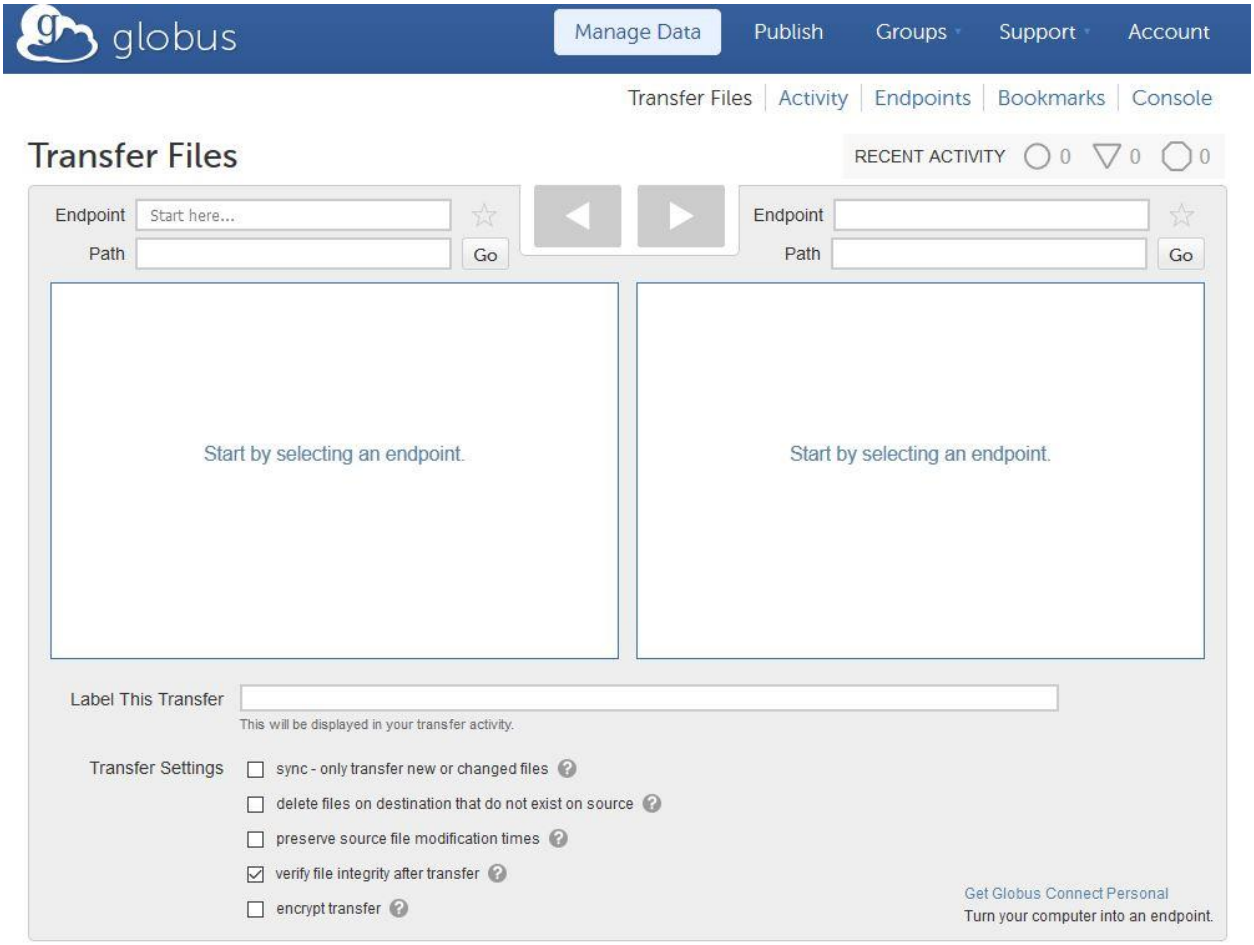
1.3 File Transfer

During the tutorial we will transfer some files from the compute cluster (crane.unl.edu) to our desktop. There are really several ways to do this, but we prefer you to use Globus. Before you continue, make sure your Globus Personal Endpoint is set up on your laptop and running (https://www.globus.org/globus-connect-personal)

First, let's make a simple file on crane.unl.edu to test this. Log on to crane.unl.edu (PuTTY or ssh), then type:

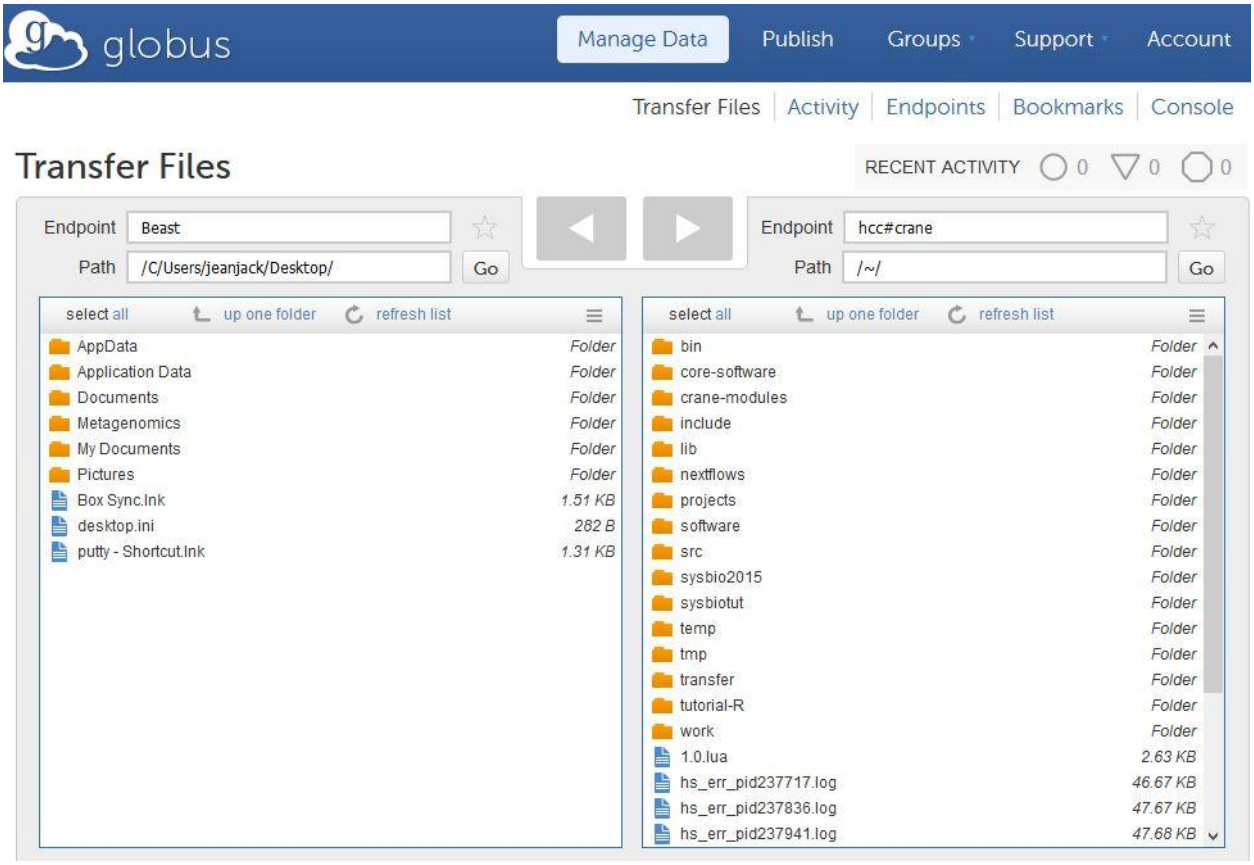
```
echo "this is a simple file" > myfirst.txt
```

In your browser, navigate to https://www.globus.org/app/transfer (you might have to log in). The "Manage Data" tab should appear as shown below. As explained this morning, this is one of the main views that you will use to transfer data within HCC, between your desktop or laptop to HCC, and vice versa.



We are going to use it now to copy the file we just created from crane.unl.edu to your laptop. In the right pane, choose as endpoint hcc#crane, and once chosen it should automatically move to your home directory (designated as /~/). Depending on whether not you have been active previously on crane, you might see a lot of files and directories. The file 'myfirst.txt' should appear as well.

In the left pane, let's connect to your personal Globus endpoint. Use the name that you have set up in the morning. It will also display the content of the folder that it deems your 'home' on your laptop (on Windows this tends to be your 'Documents' folder). For ease of use during this workshop, please navigate to your desktop (Use 'up one folder' and then drill down). On a Windows machine your desktop is (normally) located at C:/Users/<your_username>/Desktop, on a Mac it is located on /Users/<your_username>/Desktop.



Now, on the left pane, find and select (left-click) your 'myfirst.txt' file. One of the big arrows on the top will light up blue, and you can click it to start the copying process from crane to your laptop's desktop. Once it is completed (you get an email and you can see this in the 'Recent activity' panel, look on your desktop and your text file has appeared.

In a couple of minutes, we will come back to Globus to copy the data and scripts required for this workshop to your work directory on crane, so keep a tab open on your browser pointing to Globus.

1.4 Installing Keemei

In one of the early steps of our metagenomics analysis, we need to make a metadata file that contains extra information about all our samples, e.g. genotypes, conditions, treatments, and the like. In QIIME 1 this was called a mapping file. If you have ever worked with Excel or tab-delimited files trying to make an input file for an analysis, you know this can be error-prone. To get some of the issues out of the way a-priori, we will use a Google sheets add-on that will validate our metadata file.

First, log into your Google account. Then go to https://keemei.qiime2.org/, click on the Install menu, and follow the instructions. The big Chrome web store button (works for other browsers as well) is the easiest route, as the second option requires you to have a sheet open in Google Sheets https://docs.google.com/spreadsheets.

1.5 Setting up tutorial environment

During the tutorial we will be doing all the analyses in 'worker nodes' on the HCC cluster. Normally, you will have to write simple job scripts containing your commands and submit these to the job scheduler on the compute cluster. Specifically, for this workshop, we will use interactive commands but we still need to request the resources we need. To do so, type the following exactly as shown:

```
srun --pty --time=6:00:00 --mem=20G  --reservation=qiime2_workshop --
ntasks-per-node=8 --nodes=1 /bin/bash
```

That should move us to a worker node for all further analysis[1].

1.5.1 Working directory

Home directories are read-only for performance reasons on worker nodes, so we need to move to a different location. There is a special location where we have high-performance file access to our data: /work. To access your work directory, you will need to know your primary group on the HCC cluster, and your username. The final location is: /work/yourgroup/yourname. For example, /work/riethoven/jeanjack is the presenter's working directory. Let's move to your own working directory via the change directory command 'cd' (change riethoven and jeanjack into your own group and username):

```
cd     /work/riethoven/jeanjack
```

[1] All steps in this workshop are also available for normal job scheduling on crane.unl.edu: you can submit the 'shortcut' scripts via the sbatch command. Don't do this now (unless instructed) as we will run all analyses interactively, but if you want to rerun this workshop at a later time the above 'srun' command won't work and you will have to submit your scripts. For example:

```
sbatch ./scripts/step1.sh
```

If you type 'pwd' (print working directory) it should show the same location. If not, please ask for help.

You can also use a shortcut via an environment variable: $WORK which points automagically to your work directory.

```
cd    $WORK
```

Once we are in our working directory, let's make a directory that we will use to do the tutorial and move to it.

```
mkdir qiime2-tut

cd qiime2-tut
```

Now, move back to your browser. Either click on this link https://go.unl.edu/qiime2tut , copy/paste, or type it in your browser. It should bring you to Globus again, and the left pane should point to an endpoint called 'QIIME Tutorial'. In the right pane, we want to point to the directory we just made. Use hcc#crane as endpoint, but change the path to /work/yourgroup/yourusername/qiime2-tut (replace yourgroup and yourusername with your own).

Your qiime2-tut folder in Globus should be empty, and the left panel should show you various folders. Select all the folders and files in the left panel and press the arrow to start the data copy to your work qiime2-tut folder. The copy should take a few minutes, maybe a bit longer since we are all copying at the same time.

If you ask for a file listing, you should see something similar like this (actual content might differ slightly):

```
ls -hl

total 472K
-rw-r--r--  1 jeanjack riethoven  129 Oct 13 11:56 CHANGELOG
-rw-r--r--  1 jeanjack riethoven  230 Oct 10 13:18 CONTRIBUTING.md
drwxr-xr-x  4 jeanjack riethoven 4.0K Oct 13 10:54 raw
-rw-r--r--  1 jeanjack riethoven   64 Oct 10 13:18 README.md
drwxr-xr-x  2 jeanjack riethoven 4.0K Oct 13 11:56 scripts
```

If you don't see at least a raw and a scripts directory please let us know.

That concludes setting up for the tutorial. Let's move on to the real work.

2 Raw sequence

In this workshop, we are providing a partial set of data: a rhizosphere/root/soil microbiome, courtesy of Dr. Daniel Schachtman. The data sets have been downscaled to about 5% of their actual size, to make processing it during the workshop a lot faster. The 16S sequencing for these projects were done pair-ended so we will have two files (forward and reverse) for each sample. All initial files are in the FASTQ format, and have been demultiplexed (i.e. they have been divided by sample, and barcodes have been removed).

The raw files are located in the 'raw/soil' directory.

```
ls -alh raw/soil

ls -l /raw/soil | wc -l
```

3 Setting up our data set

Metagenomics (16S/ITS/targeted, even WGS) data normally arrives in our hands from the sequencing center in one of two ways: two or three large files containing forward, reverse (if done), and a barcode file, OR many smaller files. For most metagenomics analysis the yield in bases or reads per sample doesn't have to be large, hence in sequencing samples are often multiplexed by adding a tag (barcode) and sequenced within the same lane ('well'). After sequencing, this has to be split up into the individual samples by their unique barcode. Sequencing centers tend to do this for you, unless they do not want to bother with this process or you have specifically asked not to de-multiplex.

This is important for QIIME2, since the beginning of the analysis deals with either importing data and splitting (de-multiplexing) into samples, OR 'just' importing the already split samples. 'Just' is paraphrased because the first method is strangely enough more straight-forward, although it does take more (computer) time.

During this tutorial we will use the second approach, importing already existing individual samples as FASTQ files, as this is likely the way you will receive data from the sequencing centers. It is also a bit more cumbersome, since we need two files: one metadata file and one manifest file. The manifest file is specific for data imports that do not need to be de-multiplexed, and basically is a file that points each sample to its FASTQ file.

Let us have a look at the metadata file:

```
less soil-metadata.tsv
```

You will see a tab-delimited (column-wise) file with two comment lines (#). Both comment lines are necessary: the first denotes the name of the columns (for some columns these are factors), the second comment line indicates the type of the data in the column: numeric or categorical.

Important to note is that the sample ids (first column) have to be uniquely named.

In our case each sample as a forward and a reverse read, so the number of samples in our soil-metadata.tsv file should be the same as the number of FASTQ files divided by two.

```
wc –l soil-metadata.tsv
```

You'll note there is a discrepancy. What is the cause, how can you make them exact counts?

Let us upload this file to Keemei, and see if there are any errors. There shouldn't be. You can also try file soil-metadata-errors.tsv to find some common errors.

Open your browser on the Globus page, and in one panel move to your qiime2-tut directory in your work directory on crane.unl.edu; on the other panel move to your personal endpoint and find a good spot (a folder under the Desktop is fine). Copy the metadata files there, and upload them to Google Sheets. Select Add-ons->Keemei->Validate QIIME2 to check whether the files are ok.

The second file that QIIME2 needs for sample-based imports is a manifest file. This is a comma-delimited file that will list sample id, absolute path to a forward or reverse FASTQ file, and either 'forward' or 'reverse'. One of the idiosyncrasies of QIIME2 requires the path to be absolute, so we couldn't prepare this file for you in advance. We'll run a little script to generate this file:

```
cd $WORK/qiime2-tut

~jeanjack/core-software/scripts/prep_qiime2_ws_manifest.pl -dir=raw/soil -
metadata=soil-metadata.txt -manifest=soil-manifest.csv

less soil-manifest.csv
```

The main point of this file is to tie samples to their sequence files, whether just one (single-end) or both (paired-end). You should note the second column that shows the full absolute path to your data files.

As a last step in our prep work we need to know how the quality scores of bases in the FASTQ files are encoded. These (tend to be) are scores from 0 to 40, but to compact printing these out we'll print them as their ASCII counterparts. Since many of the lower ASCII values are non-printable (visible), we can add a constant number to them to start them off in the uppercase alphabet (+64) or the punctuation and lowercase alphabet (+33). There are other variants as well. However, this is important for our data import so you should make it a case, especially when using older data, to check this score.

Let's do this now.

```
module load ea-utils
```

```
find raw/soil/ -name "*.fastq" -exec fastq-stats -c 100 \{\} \; | grep
-i phred
```

As you can see everything is Phred 33, which is the default for QIIME2.

4 Importing our data

QIIME2 uses the concept of 'Artifacts' (not: artefacts) as a container for both data, analysis parameters, previous analysis steps, and (current) results. We'll see two main files, qza which is a zipped archive, and qzv which is similar but has visualization files that can be viewed in the terminal (if you're lucky, using X11 or a local Linux/Mac) or uploaded to the QIIME viewer. The latter route is what we will use – a bit cumbersome, but something that anyone can do.

Let's get going and import our FASTQ datafiles into an artifact (don't type anything yet!)

```
module load qiime2

qiime tools import --type 'SampleData[PairedEndSequencesWithQuality]' --
input-path soil-manifest.csv --output-path soil-paired-end-demux.qza --
source-format PairedEndFastqManifestPhred33

qiime tools peek soil-paired-end-demux.qza



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

(these steps are located in scripts/step1.sh)

After we have done an import of all our data to the raw data artifact, as a next step we need to do quality assurance, filtering, and at the same some we create features (the latter used to be called OTUs). Importing takes about 12 minutes, filtering just over an hour – even on this relative small (5% total reads) data set.

```
qiime dada2 denoise-paired   --i-demultiplexed-seqs soil-paired-end-
demux.qza   --o-table soil-table   --o-representative-sequences soil-
rep-seqs   --p-trim-left-f 0   --p-trim-left-r 0   --p-trunc-len-f 240
--p-trunc-len-r 200 --p-n-threads 0 --verbose
```

(located in scripts/step2.sh)

Instead of running interactively while copying/pasting these commands, I want you to run a batch script that will do all this on the cluster. While it is doing this, we have a closer look at quality assurance and artifacts.

```
sbatch scripts/step1_and_2.sh
```

```
squeue –u  yourusername
```

This script will run first step1.sh, and then step2.sh.

5 Quality assurance and filtering

An integral part of any next-generation sequencing analyses is to assure that your data is of high-quality. Many factors can influence the quality, and poor quality and errors will adversely affect the final outcome of our work.

Once simple tool to quickly check for problems is FastQC. It will check the sequence and provide us with statistics on the quality. FastQC has both a graphical user interface and a command-line version. We will be using the command-line version.

First load the environment that is required for FastQC:

```
module load fastqc
mkdir –p qa
```

And then run it on our data files:

```
nohup fastqc  raw/data/*  –o qa/  &
```

(please note that file names and commands are case-sensitive)

After running these commands, you should have several zip and html files in the 'qa' directory (ls –lh qa). To look at the content of the .html files, use Globus to transfer them to your desktop and then double-click the html files from within Windows Explorer or Finder.

16S sequencing is quite different in the resulting files than for example the mainstream DNA or RNA-seq, which tends to be only one or a few species. We have to keep this in mind when we look at the results of the quality control. We also need to reminder that we are only looking at 5% of the number of sequences here. Pick any report to view, and let's look at the 'Per base sequence quality' graph. You will see that the base-by-base quality scores get progressively gets worse towards the end of the sequence. Good (Phred) scores for genome assembly or SNP detection are 30 and above. We will do 'sort of' an assembly when we start joining and clustering the data later in the pipeline. As you can see, there are quite some sequences that trail below this cutoff. That is a problem, and we'll tackle that in the next section.

There are other graphs and sections that do not look good (e.g. the per base sequence content), but normally we look at this when we have millions of reads, not the roughly 700-4000 that we have now per sample. The other section is the overrepresented sequences, but again, keep in mind that we are actually sequencing a very small part of the 16S ribosomal gene, which tends to have very conserved stretches between species and families.

5.1 Filtering

QIIME2 has several mechanisms builtin to filter out spike-ins, chimeric sequences, low quality scores, etc. The three options that are currently available within the QIIME2 pipeline are DADA2, Deblur, and

standard quality trimming. All options behave different, but have a similar approach in that we have to specify the maximum length of a read that will be kept. Choosing this length will depend on the quality of the nucleotides at a position, and is also a function of the length of the 16S/ITS region we sequence, and the length of our reads themselves. In our example we have 2x300bp reads, while we sequenced the V4 region (about 250bp). We are expecting overlap (and this is good: it helps us correct sequencing errors), so we can keep this in mind when we choose where we want to truncate the reads.

Looking at the data in one of the R1 and R2 files, where would you cut off the sequence?

We can also look at the visualization file that QIIME2 created for us: copy soil-paired-end-demux.qzv to your local folder and then drag it into https://view.qiime2.org/

You should see an overview page with data about your (unfiltered) samples, and a tab 'interactive plot' that we would be interested in: it will show you the overall quality boxplot graph over all samples.

Looking at this data, what would be your cutoff for the reads? Is it the same of similar than the one you picked by just looking at one sample in fastqc?

(in the end, for the step2.sh script we picked 240 as cutoff for the forward read, and 200 for the reverse).

Once our second job is finished (check with squeue), we should have a soil-table.qza artifact. We can run the following commands to create a visual table and a list of representative sequences (100% OTUs, or sequence variants). The table will later be used to pick a sampling depth for our diversity analysis, and representative sequences can be used to blast the sequence against the nt database.

```
qiime feature-table summarize  --i-table soil-table.qza  --o-
visualization soil-table.qzv  --m-sample-metadata-file soil-
metadata.tsv

qiime feature-table tabulate-seqs   --i-data soil-rep-seqs.qza   --o-
visualization soil-rep-seqs.qzv
```

(located in step3.sh)

6 Phylogenetic tree

Various diversity metrics (e.g. UniFrac, Faith's) that we are going to calculate downstream depend on a (rooted) phylogenetic tree to estimate evolutionary distances between features (read: genus or species). Since there is no know-it-all and ready-made tree for us that includes all species we likely have in our sample, we are going to generate one ourselves. We will do multiple alignment, mask highly variable regions in the sequences, and then generate the tree with FastTree.

```
qiime alignment mafft   --i-sequences soil-rep-seqs.qza   --o-
alignment soil-aligned-rep-seqs.qza --verbose --p-n-threads -1
```

```
qiime alignment mask   --i-alignment soil-aligned-rep-seqs.qza   --o-
masked-alignment soil-masked-aligned-rep-seqs.qza --verbose

qiime phylogeny fasttree   --i-alignment soil-masked-aligned-rep-
seqs.qza   --o-tree soil-unrooted-tree.qza --verbose --p-n-threads 1

qiime phylogeny midpoint-root   --i-tree soil-unrooted-tree.qza   --o-
rooted-tree soil-rooted-tree.qza
```

(located in step4.sh)

As homework you can unzip/extract the Newick tree file from the soil-rooted-tree artifact and make a proper graph from it. Just be warned it will have many leaves!

7 Diversity analysis

Several of the remaining steps are based off the premise that there is an equal number of reads (~sequencing depth) between samples to do meaningful statistical testing. Of course, this is almost never the case, so we need to pick a depth where we keep the majority of the samples without losing too many sequences. In the output visualization, check the counts/sample data. If we pick the median sampling depth, we'll be throwing away all samples with fewer than that depth, while at the same time the samples with more reads will be subsampled down to the median.

This is a difficult decision, and the answer will vary by experiment. For the tutorial we only have many samples, but only at relative little depth. We can use the soil-table.qzv file to determine what our best sampling depth is. You should already have copied it to your desktop, now move it into the QIIME2 viewer and we will play with sliding the sampling depth up and down.

Pick 'Genotype' in the interactive display to have a good look at the number of samples that are kept as we are checking our sample depth.

For our samples we pick a sampling depth of 340 reads, which leaves us about 40% of all reads in 366 (88%) of our samples. You can tweak this number up to 700 or so to keep more sequences but less sasmples. If you want to run with a higher sampling depth, you'll have to modify the next command.

```
rm -Rf metrics

qiime diversity core-metrics-phylogenetic   --i-phylogeny soil-rooted-
tree.qza   --i-table soil-table.qza   --p-sampling-depth 340   --m-
metadata-file soil-metadata.tsv   --output-dir metrics --verbose --p-
n-jobs 8
```

(located in step5.sh)

This will create various alpha- and beta-diversity metrics, both based on phylogenetic tree and ones that do not need one. Results are stored in the 'metrics' directory; there are both qza and qzv artifact files. Copy the whole directory via Globus to your desktop folder, and let's look at the files that have a

visualization component (drag them into the viewer). Try any *emperor.qzv in the viewer. It will start out with all samples colored the same; use the 'Select Color Category' pull-down to select the category by which you want the samples colored. It is a 3D viewer, so you can click in the graph and drag the picture around the have a better view.

Can you find samples (of which category) that are grouped together? What seems to be, at least for the compositional metric, the largest influencing factor?

For metrics that do not have a visualization component currently, we can use the results as input to other analysis later on, or to make specific PCoA (Emperor) plots.

Let us first see if we can use the diversity metrics to determine if there are significant differences (in the metric) in the composition of the microbial community between groups of samples. We saw, in the graph, quite some separation for some categories, but we would like to attach a p (or q) value to it.

The command for alpha-diversity composition difference analysis is:

```
qiime diversity alpha-group-significance   --i-alpha-diversity
metrics/faith_pd_vector.qza   --m-metadata-file soil-metadata.tsv   --
o-visualization metrics/faith-pd-group-significance.qzv
```

where the input file should point to an alpha-diversity metrics vector (Faith's, Eveness, Shannon, or Observed OTUs); making sure the output points to a similar file of your choosing. You can run step6.sh to calculate all possible alpha-diversity significances.

View one of the significance visualization files in the viewer, and see if you can back up what we find earlier with the statistical tests. (In some metrics we do not find a difference in any category - try to explain why).

We can do a similar analysis for all the beta-diversity metrics, and get statistical information (and a graph) whether whether distances between samples within a group, such as samples from the same genotype (e.g., A), are more similar to each other then they are to samples from the other groups (B, C, D, etc). For our data we have four interesting categories for which we can do this analysis: NitrogenLevel, Type (of sample), Genotype, and GrowthStage. Beta-diversity metrics that are appropriate here are (un)weighted UniFrac, Jaccard, and Bray-Curtis.

Have a look at the scripts/step7.sh shell script: you'll see the command line there, with substitutions for METRIC and CATEGORY. Run the script. It will take all combination of metric and category and do the PERMANOVA test for it. The result will be 16 new visualization artifacts – copy and view (some of) them.

8 Plotting (time) series

The axes in the Emperor (PCoA) plot show the 1st, 2nd, and third principal components of the data that we plot. If we have a numeric variable that can be sorted, we can split our data out over that axis as

well. In that case, we need to add a custom axis to the Emperor over the category that we want to plot. We have a numeric 'GrowthStage', called 'Time', that we want to use.

```
qiime emperor plot   --i-pcoa
metrics/unweighted_unifrac_pcoa_results.qza   --m-metadata-file soil-
metadata.tsv   --p-custom-axes Time   --o-visualization
metrics/unweighted-unifrac-emperor-byTime.qzv

qiime emperor plot   --i-pcoa metrics/bray_curtis_pcoa_results.qza   -
-m-metadata-file soil-metadata.tsv   --p-custom-axes Time   --o-
visualization metrics/bray_curtis-emperor-byTime.qzv
```

(located in step8.sh)

9 Alpha rarefaction

QIIME2 implements an easy way to plot alpha rarefaction plots, which main function is to check whether the full community richness has been observed for certain sampling depths. As long as the graph does not level off, the sampling depth is not (yet) enough. We are not going to run the alpha rarefaction on our samples interactively (since they are subsamples at 5% so we are relatively sure that we won't observe the full richness) during this workshop, but you can run the below command to start it as a batch, and you can come back at the end of today to download and view `alpha-rarefaction.qzv`

```
sbatch ./scripts/step9.sh
```

10 Taxonomic analysis

Up to this point we have looked at high level metrics for our samples, but we haven't dealt with **what** is in our samples: phyla to species. Which species can we find, are they differentially present over groups in categories? To start with this, we need to match the representative sequences we have with known profiles of microbial genomes. We can do that by using creating a Naïve Bayes classifier based upon our sample prep, primers, sequencing method(s) and quality filtering. However, it is involved and most importantly time-consuming so we won't do that during the workshop; instead you can use one of two pre-trained classifiers (Silva, GreenGenes – both for the V4 region, based on 99% OTUs).

```
qiime feature-classifier classify-sklearn   --i-classifier gg-13-8-99-
515-806-nb-classifier.qza   --i-reads soil-rep-seqs.qza   --o-
classification soil-taxonomy.qza --verbose
```

(step10a.sh)

OR

```
qiime feature-classifier classify-sklearn   --i-classifier silva-119-99-515-806-nb-classifier.qza   --i-reads soil-rep-seqs.qza   --o-classification soil-taxonomy.qza --verbose
```

(step10b.sh)

However, please note that the Silva classifier takes more than 30 minutes to run on our reduced dataset. If you are reading this and well-ahead of the presentation, go ahead and run it. If you are following along or are behind, please type:

```
cp pre/soil-taxonomy.qza  .
```

Once we have taxonomic units assigned to each of the representative sequences – hopefully – we can create a simple table visualization artifact (useful if you want to search for a species or lineage), and a more interesting interactive barplot that we will look at in closer detail.

```
qiime metadata tabulate   --m-input-file soil-taxonomy.qza   --o-visualization soil-taxonomy.qzv


qiime taxa barplot   --i-table soil-table.qza   --i-taxonomy soil-taxonomy.qza   --m-metadata-file soil-metadata.tsv   --o-visualization taxa-bar-plots.qzv
```

(step11.sh)

Copy the newly created visualization artifacts and view them. You'll note that there are a couple of samples with no data: these were samples with a low read count and none of the reads was grouped (formed) with a cluster due to low abundance. A representative sequence has to be present in various samples to be counted (a way to remove chimeric sequences). With a full data set it is unlikely you see such empty samples.

11 Differential abundance testing

We saw with the taxa bar plots that they are a good tool to delve into sample composition and taxa abundance, but with so many samples the overview is quickly lost. You can filter out certain categories, e.g. we can only look at bulk soil and see if there is a gradient in taxas if we sort by Nitrogen level of genotype. Another option is to apply ANCOM analysis (Analysis of Composition Of Microbiomes) – we also need to filter out some categories here specifically due to the test assumption that less than 25% of the features change between groups. Since we are aware that there are huge changes between the type of samples we have (bulk soil, rhizosphere, and root) we cannot apply this test over the whole sample

range. Instead, we will only look at the rhizosphere and check whether we see changes in composition between genotypes, nitrogen level, or time (growthstage).

To do this, we will filter (keep) only the rhizosphere by:

```
qiime feature-table filter-samples   --i-table soil-table.qza   --m-
metadata-file soil-metadata.tsv   --p-where "Type='rhizosphere'"   --
o-filtered-table rhizo-table.qza
```

The test also doesn't allow zero values for (abundances of) features, so we add a count of 1 to the abundance of each feature in each sample:

```
qiime composition add-pseudocount   --i-table rhizo-table.qza   --o-
composition-table comp-rhizo-table.qza
```

(both steps can be done via step12.sh)

Once this is done, we have a new table comp-rhizo-table.qza which we can use with the test.

```
qiime composition ancom   --i-table comp-rhizo-table.qza   --m-
metadata-file soil-metadata.tsv   --m-metadata-column Genotype   --o-
visualization ancom-Genotype.qzv
```

(step13.sh for Genotype, Nitrogen Level , and Growthstage)

It takes about 10 min per test, so 30 minutes in total. Depending on time, run the script or:

```
cp pre/ancom*  .
```

and download and view it. You will see that there is category where we find differential abundance in a few genera.

Fin.