Introduction to taxonomic analysis of amplicon and shotgun data using QIIME

Peter Sterk, Oxford e-Research Centre, University of Oxford, UK
European Bioinformatics Institute
Bioplatforms Australia
CSIRO, Australia

Canberra, ACTU July 2014

General information

The following standard icons are used in the hands-on exercises to help you locating:

i Information information.

👣 Instructions to perform.

📄 General information/notes.

! Warning - PLEASE take care and read carefully.

Q Questions.

✸ An optional bonus section for those progressing rapidly.

✸ An optional advanced section for those progressing very rapidly or to be used for future reference.

Resources used

Tools Used

QIIME

http://qiime.org/index.html

Sources of Data

- Sutton et al. (2013). Impact of Long-Term Diesel Contamination on Soil Microbial Community Structure. Appl. Environ. Microbiol. 79(2):619-630.

Tutorial objectives

In this tutorial we will look at the open source software package QIIME (pronounced chime). QIIME stands for Quantitative Insights Into Microbial Ecology. The package contains many tools that enable users to analyse and compare microbial communities. QIIME was originally developed to analyse of Roche 454 amplicon sequencing data. In the latest versions workflows have been added to analyze data from different sequencing platforms, such as Illumina, and different types of data, such as shotgun data. In this course we will use QIIME 1.8, which is the latest version.

We will (re-)introduce you to the Linux operating system to a basic level that is sufficient to run bioinformatics software from preconfigured Linux installations such as the NeCTAR cloud installation we will be using.

After completion of this tutorial, you should be able to perform a taxonomic analysis on a Roche 454 16S rRNA amplicon dataset. In addition you should be able to do 16S taxonomic analysis on shotgun data using the tool rRNASelector in combination with QIIME. Finally you should be able to work out solutions for datasets from other platforms such as Illumina from the information you find on the QIIME web site (http://qiime.org/).

Short introduction to Linux and the NeCTAR Cloud

QIIME runs under Linux or Mac OS X, but not under Windows. For this course a virtual machine was created with Linux as the operating system and all software packages that we need for this course were installed. The image 'lives' in the NeCTAR cloud and with a few pieces of software installed on the course computers, we can access the image and work in a Linux environment.

It is worth mentioning at this stage that if you find QIIME useful for your own project, but you la/ck the expertise to set up your own Linux system, the QIIME developers produce a downloadable disk image that contains the full QIIME package. You will need to install VirtualBox, which is freely available for Linux, Mac and Windows platforms and this will allow you to run QIIME on any of these platforms as a virtual machine. More details can be found at http://qiime.org/install/virtual_box.html.

Another alternative is the Bio-Linux image (http://nebc.nerc.ac.uk/tools/bio-Linux/bio-Linux-7-info). It is a Linux distribution containing many bioinformatics packages, runs live from DVD or memory stick, but can also be installed on an computer, either as the main operating system or under VirtualBox or similar (e.g. VMWare, Parallels). It has the basic packages from QIIME installed, but depending on your requirements you may need to install additional components. It comes with a very good Linux tutorial, so worth investigating if you want to go that route. Finally, you can run QIIME in the Amazon EC2 cloud. For more details, see http://qiime.org/tutorials/working_with_aws.html#working-with-ec2). As a number of you will not have any experience with Linux, this part of the tutorial will teach you the basics of Linux. There are a large number of free Linux tutorials on the web for those who want to learn more at a later stage.

We assume you have successfully connected to the course NeCTAR image and you have the Linux desktop on your screen.

The first steps we will do together now to get you going as quickly as possible.

Most of what we will do will be run from the command line and before we can issue any commands, we will need a terminal window. Click on Applications at the top left of your desktop, then go to Accessories and then click on the first `Terminal` menu item. A terminal window should appear on your desktop.

At the prompt (which ends with $) you can type commands. During this tutorial we will represent the prompt as $ for brevity, do not type a dollar character at the beginning of any of the commands, only type what follows it. To execute a command, press return/enter.

Type the following command to list the files and directories (folders) followed by enter:

```
cd ~/
ls -l
```

You will be presented with a list of the contents of your home directory. Note that Linux does not have a concept of disks like windows (e.g. C:\). Instead it has so called mount points with a directory at its root. /home is where by default the home directories of all users are located. /usr is where a lot of the operating system and programs reside. The directory structure of a Linux system outside the /home area is for this tutorial not important. Also note that where windows uses back slashes to separate directories, Linux uses forward slashes. The desktop is in a folder /home/trainee/Desktop.

To move up to the desktop folder, type

```
cd ~/Desktop
ls -l
```

Note that file and directory names are case-sensitive, ’cd desktop’ does not work. The ’-l’ option after the ls command tells ’ls’ to show a long, more detailed listing of the directory contents showing file permissions, owners and date stamps. On your desktop is a folder called ’Taxonomy, which contains the necessary files for this tutorial.

You can probably work out how to enter this directory now. There are a few more tips for moving around: To go to your home directory, type one of the following (note ’˜’ is short for your home directory):

```
cd
cd ~
```

To go to the tutorial directory, type one of the following:

```
cd /home/trainee/Desktop/Taxonomy
cd ~/Desktop/Taxonomy
t
```

The command ’t’ is an alias that we have set up to make life easier. You can create your own aliases for command that you use frequently using ’alias’, e.g.

```
$ alias d='cd ~/Desktop'
```

From that moment on, you only need to type d followed by ’Enter’ to go to your desktop folder. To move up one directory level (e.g. when you are in Taxonomy and want to go back to Desktop), type:

```
cd ..
```

’..’ is the parent directory, ’.’ Is the current directory. If you’re not sure where you are on the file system, type ’pwd’ to print the current working directory to the terminal window. A few more useful commands: To view text files (or concatenate multiple files into a new file):

```
cat filename1 (filename2 filename 3 ... > newfile
```

To view long text files one screen at a time, use 'less'. Exit 'less' by typing 'q' at the colon.

```
less filename
less -S will truncate long lines

#To copy a file:
cp file newfile
# results in two identical files called file and newfile

#To move or rename a file:

mv file newfile
# renames the file to newfile

mv file ..
# moves the file to the parent directory

#To remove/delete a file:
rm file
# be cautious, in Linux a deleted file is gone forever

#For now this is all we need to know to start the tutorial proper.
```

De novo OTU picking and diversity analysis using 454 data

We will partly re-analyze the data from Sutton et al. (2013). Impact of Long-Term Diesel Contamination on Soil Microbial Community Structure. Appl. Environ. Microbiol. 79(2):619-630. An electronic copy of the paper can be found in your Taxonomy folder. When you have to wait a few minutes for commands to complete, use the time to acquaint yourself with the study. It is a good example of a study that combines the power of next-generation sequencing with environmental observations/measurements.

The analysis we do follows the pipeline described in the QIIME general 454 tutorial (http://qiime.org/tutorials/tutorial.html). Feel free to look at this tutorial for further background information. As our dataset used in the tutorial is a subset of the Sutton data, but more realistic for what you may be doing. In some parts of the analysis and steps we have precomputed analysis on the complete data set for comparison.

```
Go to /home/trainee/Desktop/Taxonomy/sutton/.
```

You will find a file called 'sutton5000.sff', which contains the 5000 random multiplexed reads from this study's 26 samples with quality information and flowgrams. This is a binary file and we will need to extract the sequences and quality scores from it to be able to work with the sequences and quality scores. We will use 'sffinfo' from Roche to do this. This tool is not supplied with QIIME, but it is freely available from Roche on request. We have installed it for you. Alternatively, you can use the much slower tool with the same functionality supplied with QIIME.

Note that in the Taxonomy folder there is a subfolder called 'sutton_full_denoised'. This folder contains the precomputed analysis results from the full and denoised dataset. When you go through this tutorial, it is worth comparing those results with the results you obtain with the reduced dataset.

Prepare files

For this tutorial we have generated a random subset of the original Sutton dataset to complete the necessary steps in a reasonable amount of time.

Generate a fasta and quality score file from sutton.sff:

```
$ sffinfo -s sutton5000.sff > sutton.fna
$ sffinfo -q sutton5000.sff > sutton.qual
```

Note that the >character redirects the output from the screen to a new file. Please inspect the files with ’less’ or ’less -S’ to truncate long lines. You will notice that there is no obvious association of the reads with a particular sample.

That is what we need to do next.

```
$ less -S sutton.fna
```

Or just have a look at the fasta headers and ignore the DNA sequences with the command ’grep’. You need to extract all lines that start with `>` and send the output to ’less’ to be able to view the output one screen at a time. The command to give is:

```
$ grep "^>" sutton.fna | less
```

The following ^ is a so-called regular expression. The ^ character means ”starts with”, so the grep command looks for all lines that start with `>`. The pipe character, `|` is used to pipe or stream the output from the first command (grep) into a second command (less).

Now view the file containing the quality scores:

```
$ less -S sutton.qual
```

The Mapping File

We will use the barcode information to associate reads with one of 26 samples. We will also need to remove the barcodes and primer sequences from the reads as these interfere with the taxonomic analysis.

You will find a file called ’mapping.txt’ in the Taxonomy/sutton/ directory.

Note that this file has to be created for each analysis, as the information is specific for an experiment. The mapping file can also contain information on your experimental design. The format is very strict; columns are separated with a single TAB character; the header names have to be typed exactly as specified in the documentation. A good sample description is useful as it is used in the legends of the figures QIIME generates. We could also specify the reverse primer and remove it from the reads. Unfortunately, the reverse primer sequence was not in the paper, and we will ignore it though we could probably deduce it from longer reads: as a `466 bp` region of the 16S ribosomal RNA gene flanking the V3 and V4 regions was amplified, you’ll have a clue where to look for the reverse primer.

```
$ less -S mapping.txt
```

```

#Next test the mapping file for potential errors:

$ validate_mapping_file.py -m mapping.txt -o mapping_output
```

There shouldn't be any errors. If there are errors, a corrected mapping file will be written to the directory mapping_output.

Assign samples to the reads

Using the mapping file and the Sutton fasta and quality files we are going to now assign samples to the reads.

Type the following command on a single line:

```
$ split_libraries.py -m mapping.txt -f sutton.fna -q sutton.qual -o \
    split_library_output -b 10 -L 500
```

```
Here we specify the following options:
-m for the mapping file
-f for the fasta input file generated from the .sff file
-q for the quality score file generated from the .sff file
-o for the output directory
-b for the length of the barcode (10 in our case)
-L for the maximum length of the read we still accept. Useful if you do \
    not know the reverse primer, but you do know the approximate length \
    of the sequence you have amplified. Reads that are much longer are \
    probably an artifact and it is best to exclude them.
```

When completed please enter the directory ’split_library_output’ and have a look at the log file.

It contains detailed information what was done during the step we’ve just performed. Note that the number of reads assigned to the different samples varies considerably. Knowing where the individual samples were taken may give a clue why this may be! Next have a quick look at the file seqs.fna. What has changed to the header of the reads?

Look at the split libraries results

```
$ cd split_library_output
$ ls -l
$ less split_library_log.txt
$ less -S seqs.fna
```

You will see the reads are now batched to their sample.

To denoise or not to denoise?

The pyrosequencing technology employed by 454 sequencing machines produces characteristic sequencing errors, mostly imprecise signals for longer homopolymers runs. Most of the sequences contain none or only a few errors, but a few sequences contain enough errors to be classified as an additional rare OTU. The goal for the denoising procedure is to reduce the number of erroneous OTUs and thus increasing the accuracy of the whole QIIME pipeline. This is a computationally intensive procedure, which we will skip for this reason. There is a QIIME tutorial that outlines the steps (http://qiime.org/tutorials/denoising_454_data.html) and also includes a warning about new 454 flow patterns introduced in 2012. Note that only amplicon data sets can be denoised with the described procedure. We will not denoise our data today. We did denoise the full dataset in the sutton_full_denoised folder.

Picking Operational Taxonomic Units (OTUs)

We will now use a workflow for de novo OTU picking, taxonomy assignment, phylogenetic tree construction, and OTU table construction QIIME has several workflows to pick OTUs, we will be using the one described in the general overview tutorial (http://qiime.org/tutorials/tutorial.html) It has 7 steps, which are described in some detail in this tutorial.

The described procedure is run with the command from the Taxonomy directory. This step takes about 12mins to run. Please read through the different steps (http://qiime.org/tutorials/tutorial.html) and try to understand the procedure. Remember that an OTU is not the same as a species, but a 'bag/cluster' of highly similar sequences (at least 97% is common for bacteria/archaea), or a single sequence in case of rare OTUs.

```
$ pick_de_novo_otus.py -i split_library_output/seqs.fna -o otus
```

Please do spend some time looking at the output of this pipeline. In particular the file 'seqs_rep_set_tax_assignments.txt' in the 'uclust_assigned_taxonomy' directory. By default QIIME uses the Greengenes 16S reference database to assign taxonomy. It has the following levels: kingdom, phylum, class, order, family, genus, species. It will be immediately clear that most reads cannot be classified up to species level.

As described in step 6 of the QIIME overview tutorial, the pipeline creates a Newick-formatted phylogenetic tree (rep_set.tre) in the otus directory. You can run the program 'figtree' either from the command line or select FigTree from the menu on your desktop (Applications -¿ Other -¿ FigTree) and view the tree by opening the file 'rep_set.tre' in the 'otus' folder (Desktop-¿Taxonomy-¿otus). The tree that is produced is too complex to be of much use. We will look at a different tool, Megan 5, which produces a far more useful tree.

In step 7 of the QIIME overview tutorial a file called otu_table.biom is generated. It is in biom-format, which is increasingly supported by taxonomic software developers. One of the tools that supports the biom format is Megan (http://ab.inf.uni-tuebingen.de/software/megan5/). Megan is a standalone tool for analyzing both taxonomic and functional content of datasets. It is free for academic use, but you will need to request a licence first. We will use Megan version 5 to display a taxonomic tree using the biom output we have just produced.

Note: Sequence errors can give rise to spurious ORFs and we can filter out OTUs that only contain a single sequence (singletons). QIIME allows you to do this quite easily - or you could also remove abundant taxa if you are more interested in rare taxa. To remove singletons, run the following commands:

```
$ cd otus
$ filter_otus_from_otu_table.py -i otu_table.biom
-o otu_table_no_singletons.biom -n 2
```

This removes OTUs with less than 2 sequences. If you use the -k option instead of the -n option, OTUs with more than the specified number of sequences will be removed.

Megan can be opened from the menu under Applications ->Other. From the File menu select Import ->BIOM format. Find your biom file and import it. Megan will generate a tree that is far more informative than the one produced with FigTree. You can change the way Megan displays the data by clicking on the various icons and menu items. Please spend some time exploring your data. The Word Cloud visualization is interesting, too, if you want to find out which samples are similar and which samples stand out.

View OTU statistics

You can generate some statistics, e.g. the number of reads assigned, distribution among samples. Some of the statistics are useful for further downstream analysis, e.g. beta-diversity analysis. Run the following now, again from within the Taxonomy directory, and look at the results. Write down the minimum value under Counts/sample summary. We need it for beta-diversity analysis.

```
$ biom summarize-table -i otus/otu_table.biom âĂŞo \
    otus/otu_table_summary.txt

$ less otus/otu_table_summary.txt
```

Visualize taxonomic composition

We will now group sequences by taxonomic assignment at various levels. The following command produces a number of charts that can be viewed in a browser. The command takes about 5 minutes to complete.

```
$ summarize_taxa_through_plots.py -i otus/otu_table.biom -o \
    wf_taxa_summary -m mapping.txt
```

To view the output, open a web browser from the Applications -¿ Internet menu. You can use Google chrome, Firefox or Chromium. In Google chrome or Chromium, type CTRL-O, or in Firefox use the File menu to select Desktop ->Taxonomy ->wf_taxa_summary ->taxa_summary_plots and open either area_charts.html or bar_chars.html. I prefer the bar charts myself. The top chart visualizes taxonomic composition at phylum level for each of the samples. The next chart goes down to class level and following charts go another level up again. The charts (particularly the ones more at the top) are very useful for discovering how the communities in your samples differ from each other. There is a similar plot in the paper, if you have time, see how our analysis compares with the one described in the paper.

Alpha diversity within samples and rarefaction curves

Alpha diversity is the microbial diversity within a sample. QIIME can calculate a lot of metrics, but for our tutorial, we generate 3 metrics from the alpha rarefaction workflow: chao1 (estimates species richness); observed species metric (the count of unique OTUs); phylogenetic distance. The following workflow generates rarefaction plots to visualize alpha diversity.

Run the following command from within your taxonomy directory, this should take a few minutes:

```
$ alpha_rarefaction.py âĂŞi otus/otu_table.biom âĂŞm mapping.txt âĂŞo \
    wf_arare -t otus/rep_set.tre
```

First we are going to view the rarefaction curves in a web browser by opening /home/trainee/Desktop/Taxonomy/wf_arare/alpha_rarefaction_plots/rarefaction_plots.html. To start select as metric 'chao1' and select as category 'Description'. It is clear that the microbial diversity in some samples is much higher than in other samples. Click around in the legend as this will help you work out which line corresponds with which sample. If you have time you could try to correlate species richness with environmental data from the paper and establish whether our analysis confirms the findings of the authors. Next view the precomputed rarefaction curves which show an increased sequencing depth.

In general the more reads you have, the more OTUs you will observe. If a rarefaction curve start to flatten, it means that you have probably sequenced at sufficient depth, in other words, producing more reads will not significantly add more OTUs. If on the other hand hasn't flattened, you have not sampled enough to capture enough of the microbial diversity and by extrapolating the curve you may be able to estimate how many more reads you will need. Consult the QIIME overview tutorial for further information.

Beta diversity and beta diversity plots

Before we have a quick look at taxonomic analysis of shotgun data, we have a quick look at beta diversity analysis, which is the assessment of differences between microbial communities. As we have already observed, our samples contain different numbers of sequences.

The first step is to remove sample heterogeneity by randomly selecting the same number of reads from every sample. This number corresponds to the 'minimum' number recorded when you looked at the OTU statistics.

Now run the following command

```
$ beta_diversity_through_plots.py -i otus/otu_table.biom -m \
    mapping.txt -o wf_bdiv_even122 -t otus/rep_set.tre -e 112
```

Read through the beta diversity compute section of the QIIME overview tutorial and try to understand this workflow. Tomorrow we will look at visualization of beta diversity analysis results in more detail. Unfortunately we cannot view the PCoA plots that we have just generated using the NeCTAR image as WebGL is not supported. Precomputed plots can be viewed using the browser on your computer, we will make the link available.

Closed reference OTU picking of 16S ribosomal rRNA fragments selected from a shotgun data set

In a closed-reference OTU picking process, reads are clustered against a reference sequence collection and any reads, which do not hit a sequence in the reference sequence collection are excluded from downstream analyses. In QIIME, pick_closed_reference_otus.py is the primary interface for closed-reference OTU picking in QIIME. If the user provides taxonomic assignments for sequences in the reference database, those are assigned to OTUs. We could use this approach to perform taxonomic analysis on shotgun data. We need to perform the following steps:

1. Extract those reads from the data set that contain 16S ribosomal RNA sequence. If there are less than (e.g.) 100 nucleotides of rRNA sequence, the read should be discarded. 2. Remove non-rRNA sequence (flanking regions) from those reads 3. Run closed-reference OUT picking workflow 4. Visualise the results, e.g. in Megan

Extraction of 16S rRNA sequence-containing reads using rRNASelector

We will analyze an Illumina paired-end dataset that has been drastically reduced in size for this tutorial, while preserving the majority of the 16S containing reads. The dataset is from the metagenome described at http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3772140/. There is a pdf in the working directory for this part of the tutorial. This is a paired end dataset, and where read pairs overlapped, they were merged into a single sequence. If read pairs did not overlap, both reads were included in the analysis. QC was performed using the EBI Metagenomics pipeline. We will use a tool called rRNASelector, which is freely available (http://www.ncbi.nlm.nih.gov/pubmed/21887657) to select our 16S rRNA sequence containing reads. The tool invokes hmmsearch and uses trained hidden Markov models to detect reads with 16S rRNA sequence. The tool also trims the reads so that only 16S rRNA sequence is present in the fasta file we will feed into the QIIME workflow.

First, we need to go to our working directory:

```
$ cd ~/Desktop/Taxonomy/A7A/
```

You will find a file called A7A-paired.fasta containing the sequence reads.

Fire up rRNASelector from the command line:

```
$ rRNASelector
```

A graphical interface should appear. Load the sequence file by clicking on 'File Choose' at the top and navigate to the file A7A-paired.fasta. Select the file and click 'Open'. The tool will automatically fill in file names for the result files. Change the Number of CPUs to '2', select Prokaryote 16S (to include both bacterial and archaeal 16S sequences) and specify the location of the hmmsearch file by clicking the second 'File Choose' button. You can type the location manually '/usr/bin/hmmsearch'. Next, click process. The run should take a few minutes to complete.

If all went well, you can close rRNASelector by clicking on Exit. You will have 3 new files in your directory, one containing untrimmed 16S reads, one containing trimmed 16S reads (A7A-paired.prok.16s.trim.fasta; thatâĂŹs the one we want) and a file containing reads that no not contain (sufficient) 16S sequence.

Closed-reference OTU picking workflow and visualization of results in Megan

We are now ready to pick our OTUs. We do that by running the following command (all on one line and no space after gg_otus-12-10):

```
pick_closed_reference_otus.py -i A7A-paired.prok.16s.trim.fasta -o \
    ./cr_uc -r \
    /qiime_software/gg_otus-12_10-release/rep_set/97_otus.fasta -t \
    /qiime_software/gg_otus-12_10-release/taxonomy/97_otu_taxonomy.txt
```

We need to specify the following options:

```
  -i input_file.fasta
  -o output_directory
  -r /path/to/reference_sequences
  -t /path/to/reference_taxonomy
```

The command will take several minutes to run. When finished open Megan as described before, import the otu_table.biom file and explore the results.

Extreme challenge

Many of you will send off samples to be sequenced and quite often providers preprocess the raw data before handing it back to the client. If your reads were demultiplexed and primers were removed, you have a problem as the amplicon workflows from QIIME rely on the presence of primers and barcodes. The first message is that you should insist on getting your data as unprocessed reads. Not all is lost if you end up with a dataset with primers and barcodes stripped, but it requires more work.

This is an exercise in understanding the format QIIME expects and how you can reformat data to allow analysis in QIIME. There is a directory in your Taxonomy folder called BalticSea and it has a number of samples that were sequenced using 454 technology and are demultiplexed, but for the purpose of this exercise, these could have been Illumina files as the format is fastq.

The challenge we give you is to write down how we need to reformat this data (even if you do not know how) to be able to perform a similar analysis we have done with the Red Sea data. This is something you can in a small group if you feel you're not quite up to the challenge.

Hints: Consider using the QIIME function convert_fastaqual_fastq.py What does the mapping you need to run split_libraries.py look like?

Finally

If there is time left you could go back to the polluted railway site study. The aim of this study was to understand interrelationship among microbial community composition, pollution level, and soil geochemical and physical properties. With additional information from the paper, could you come up with some conclusions?

The QIIME 454 overview tutorial at http://qiime.org/tutorials/tutorial.html has a number of additional steps that you may find interesting; so feel free to try some of them out. Note hat we have not installed Cytoscape, so we cannot visualize OTU networks.

We will end this tutorial with a 15-minute summary of what we have done and how well our analysis compares with the one in the paper. Hopefully you will have acquired new skills that allow you to tackle your own taxonomic analyses. There are many more tutorials on the QIIME website that can help you pick the best strategy for your project (http://qiime.org/tutorials/). We picked QIIME for this tutorial as it is widely used and supported, but there are alternatives that might suit your need better (e.g. VAMPS at http://vamps.mbl.edu; mothur at http://www.mothur.org and others).