High Protein Diet Microbiome Analysis

Matthew Snelson

2020-07-09

Contents

Chapter 1

Intro

This is the code for the microbiome analysis associated with this paper - #TODO add link when published. The inspiration for documenting this analysis using bookdown came from Rachel Lappan - I would highly recommend checking out her analysis here.

Chapter 2

Setup and QC

2.1 Install QIIME and acitvate conda environment

This anlaysis was coducted using QIIME 2020.2. First step, download the yml file and then use it to create a conda environment for the install.

```
$ wget https://data.qiime2.org/distro/core/qiime2-2020.2-py36-linux-conda.yml
$ conda env create -n qiime2-2020.2 --file qiime2-2020.2-py36-linux-conda.yml
```

Then activate the conda environment

```
$ conda activate qiime2-2020.2
```

2.2 Quality Check (FastQC)

Run fastqc. Note that these are gzipped fastqc files, but fastqc still works.

```
$ fastqc data-raw/*.fastq.gz
```

Running fastqc generates a fastqc.zip file and fastqc.html per fastq file. Move the fastqc generated files into a `results/fastqc` folder.

```
$ mkdir results/
$ mkdir results/fastqc
$ mv data-raw/*fastqc* results/fastqc/
```

It's also handy to run mutliQC as this allows to look at quality of all samples at once (rather than one at a time with fastQC reports)

First up, install multiQC (if not already installed)

```
$ conda install -c bioconda -c conda-forge multiqc
```

Then run multiqc on the `results/fastqc` folder.

```
$ mkdir results/multiqc
$ multiqc results/fastqc/* -o results/multiqc/
```

Note: FastQC generates fastqc reports for individual files, the benefit of using mutliqc is that we get a report where we can see all the samples at once. Benefit here is if there's one real stinker you'll be able to see it which you might miss if looking at individual fastqc reports. In the plot below the red are all the reverse reads, the quality of the reverse reads tends to always drop off sooner than the forward reads.

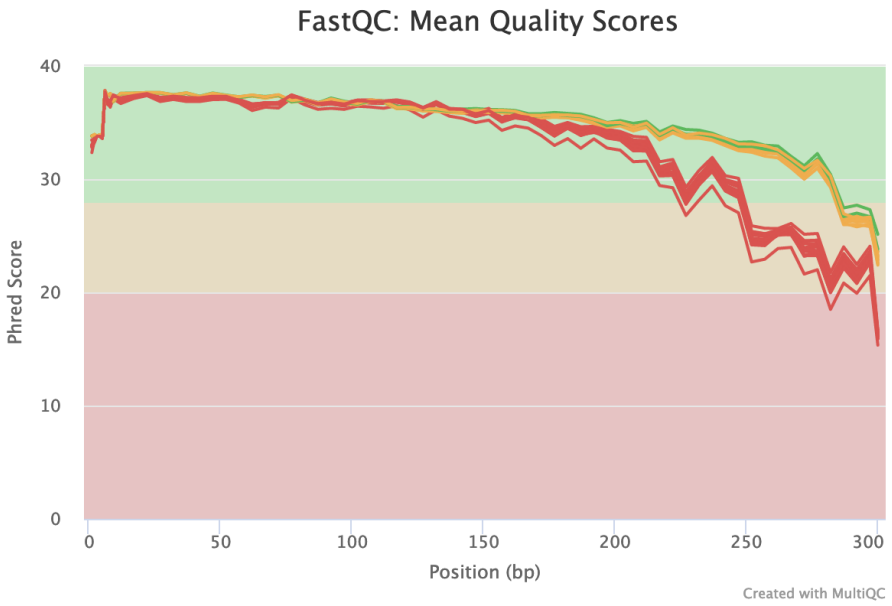

Figure 2.1: multiqc results

Based on the multiQC report: will choose limits of F 260 bp and R : 215 bp

Chapter 3

Pre-processing Sequence Reads

3.1 Import data

First up gotta import reads into a qiime artifact (called here `input-data-casava`)

```
$ qiime tools import \
--type 'SampleData[PairedEndSequencesWithQuality]' \
--input-path data-raw/ \
--input-format CasavaOneEightSingleLanePerSampleDirFmt \
--output-path input-data-casava.qza
```

3.2 Denoise with dada2

This step takes a while. so may want to setup within a tmux session if running on nectar cloud. Note I use a trim-left parameter of 15 on both the forward and reverse reads to deal with the amplicon primers (as seems to work better than using cut-adapt).

```
$ qiime dada2 denoise-paired \
--p-n-threads 8 \
--i-demultiplexed-seqs input-data-casava.qza \
--p-trim-left-f 15 \
--p-trim-left-r 15 \
--p-trunc-len-f 260 \
--p-trunc-len-r 215 \
--output-dir DADA2_denoising_output \
```

```
--verbose \
&> DADA2_denoising.log
```

So now we have a folder, `DADA2_denoisng_output` with following items in there: - denoising_stats.qza - representative_sequences.qza - table.qza

3.3 Check denoising stats

```
$ cd DADA2_denoising_output

$ qiime metadata tabulate \
--m-input-file denoising_stats.qza \
--o-visualization denoising_stats.qzv \
```

3.4 Get FeatureTable [frequency] and FeatureData [sequences]

```
$ qiime feature-table summarize \
  --i-table table.qza \
  --o-visualization table.qzv \
  --m-sample-metadata-file ../metadata.txt

$ qiime feature-table tabulate-seqs \
  --i-data representative_sequences.qza \
  --o-visualization representative_sequences.qzv
```

Note: min freq: 20,718.0 reads

3.5 Train Classifier (for doing taxonomy)

First up, we gotta import a the greengenes database stuff into qiime as qiime artifacts.

```
$ mkdir classifier
$ cd classifier

# download green genes 13_8
$ wget ftp://greengenes.microbio.me/greengenes_release/gg_13_5/gg_13_8_otus.tar.gz
$ gunzip gg_13_8_otus.tar.gz
```

```
$ tar -xvf gg_13_8_otus.tar
$ rm gg_13_8_otus.tar

 # the sequences from greengenes
$ qiime tools import \
--type 'FeatureData[Sequence]' \
--input-path gg_13_8_otus/rep_set/99_otus.fasta \
--output-path 99_otus.qza

# The taxonomy strings from greengenes
$ qiime tools import \
--type 'FeatureData[Taxonomy]' \
--input-format HeaderlessTSVTaxonomyFormat \
--input-path gg_13_8_otus/taxonomy/99_otu_taxonomy.txt \
--output-path 99_otus_16S_taxonomy.qza
```

Extract reads from the relevant region (V3-V5) out of the ref database

```
$ qiime feature-classifier extract-reads \
--i-sequences 99_otus.qza \
--p-f-primer CCTACGGGNGGCWGCAG  \
--p-r-primer GACTACHVGGGTATCTAATCC   \
--p-min-length 300 \
--p-max-length 600 \
--o-reads ref_seqs.qza \
--verbose \
&> 16S_training.log
```

Now, train the classifier on this region. I.e use this `ref_seqs.qza` qiime artifact as the input for feature-classifier.

```
$ qiime feature-classifier fit-classifier-naive-bayes \
--i-reference-reads ref_seqs.qza \
--i-reference-taxonomy 99_otus_16S_taxonomy.qza \
--o-classifier classifier_16S.qza \
--verbose \
&> 16S_classifier.log
```

The output of this is our classifier, which I've called `classifier_16S.qza`, is what will be used in the next step.

3.6 Classify rep seqs

So now we've got the classifier, we can use it to classify our representative seqs (rep-seqs.qza), which I then tabulated and visualised:

```
$ cd ..
$ mkdir taxonomy

$ qiime feature-classifier classify-sklearn \
--i-classifier classifier/classifier_16S.qza \
--i-reads DADA2_denoising_output/representative_sequences.qza \
--o-classification taxonomy/classified_rep_seqs.qza

# Tabulate the features, their taxonomy and the confidence of taxonomy assignment
$ cd taxonomy

$ qiime metadata tabulate \
--m-input-file classified_rep_seqs.qza \
--o-visualization classified_rep_seqs.qzv
```

3.7 Create a phylogenetic tree

Note that this only needs the rep seqs (not the classified rep seqs)

```
$ cd ..

$ qiime phylogeny align-to-tree-mafft-fasttree \
--i-sequences DADA2_denoising_output/representative_sequences.qza \
--output-dir phylogenetic_tree \
--p-n-threads 8 \
--verbose \
&> phylogenetic_tree_generation.log
```

Note: it's the rooted_tree.qza file which will be used in downstream analyses.

Now on to the fun stuff - taxonomy and diversity analyses!

Chapter 4

Taxonomic Analysis

4.1 Taxonomic analysis - taxa barplots

```
$ qiime taxa barplot \
--i-table DADA2_denoising_output/table.qza \
--i-taxonomy taxonomy/classified_rep_seqs.qza \
--m-metadata-file metadata.txt \
--o-visualization taxonomy/taxa_barplots.qzv
```

This provides taxonomic barplots which I can then download as SVG (also allows for download as csv, which act as input for LefSe). Downloaded csvs and have placed in fig/taxa_barplots/ folder.

4.2 Generate heatmap

```
$ mkdir collapsed-tables
$ mkdir heatmaps

$ qiime taxa collapse \
--i-table DADA2_denoising_output/table.qza \
--i-taxonomy taxonomy/classified_rep_seqs.qza \
--p-level 7 \
--o-collapsed-table collapsed-tables/collapsed-table-l7.qza

$ qiime feature-table heatmap \
--i-table collapsed-tables/collapsed-table-l7.qza \
--m-sample-metadata-file metadata.txt \
```

```
--m-sample-metadata-column treatment \
--o-visualization heatmaps/heatmap-l7.qzv

# output heatmap as png/svg
$ qiime tools export \
--input-path heatmap-l7.qzv \
--output-path heatmap-l7
```

Note: As multiple people have noted on the qiime forums, loading the qzv into qiime2view and then trying to export the heatmap figure doesn't seem to work (at least I've never got it to work). So that's why I add in that export command at the end.

Chapter 5

Alpha and Beta Diversity Analyses

NOte: Use sampling depth of 20718 (including all samples)

5.1 Alpha Rarefaction Curves

```
$ mkdir alpha_rarefaction

$ qiime diversity alpha-rarefaction \
--i-table DADA2_denoising_output/table.qza \
--i-phylogeny phylogenetic_tree/rooted_tree.qza \
--p-max-depth 20718 \
--m-metadata-file metadata.txt \
--o-visualization alpha_rarefaction/rarefaction_20718.qzv
```

5.2 Diversity Metrics

```
$ qiime diversity core-metrics-phylogenetic \
  --i-phylogeny phylogenetic_tree/rooted_tree.qza \
  --i-table DADA2_denoising_output/table.qza \
  --p-sampling-depth 20718  \
  --m-metadata-file metadata.txt \
  --output-dir core-metrics-results
```

Then I used this script from Rachel Lappan to iterate the alpha-group-significance plugin.

```
for result in *vector.qza; \
do \
outname=${result/_vector.qza/_group_significance.qzv}; \
qiime diversity alpha-group-significance \
--i-alpha-diversity $result \
--m-metadata-file ../metadata.txt \
--o-visualization $outname; \
done
```

5.3 Beta signficance metrics

```
$ qiime diversity beta-group-significance \
--i-distance-matrix bray_curtis_distance_matrix.qza \
--m-metadata-file ../metadata.txt \
--m-metadata-column treatment \
--o-visualization bray_curtis_distance_matrix.qzv

$ qiime diversity beta-group-significance \
--i-distance-matrix unweighted_unifrac_distance_matrix.qza \
--m-metadata-file ../metadata.txt \
--m-metadata-column treatment \
--o-visualization unweighted_unifrac_distance_matrix.qzv

$ qiime diversity beta-group-significance \
--i-distance-matrix weighted_unifrac_distance_matrix.qza \
--m-metadata-file ../metadata.txt \
--m-metadata-column treatment \
--o-visualization weighted_unifrac_distance_matrix.qzv
```

Chapter 6

Picrust2

Firstly, prepare files for input into picrust2. Note: Will need to be in qiime conda environment for qiime commands to work.

```
$ mkdir picrust2

# export table.qza -> outputs as feature-table.biom
$ qiime tools export  \
--input-path DADA2_denoising_output/table.qza \
--output-path picrust2

# export rep seqs sqz -> outputs as dna-sequences.fasta file.
$ qiime tools export  \
--input-path DADA2_denoising_output/representative_sequences.qza \
--output-path picrust2
```

Install/activate picrust

```
$ conda create -n picrust2 -c bioconda -c conda-forge picrust2=2.3.0_b
$ conda activate picrust2
```

Now run picrust, full pipeline:

```
$ cd picrust2

$ picrust2_pipeline.py -s dna-sequences.fasta -i feature-table.biom -o picrust2_out_pipeline -p 1

# add descriptions (make easier in STAMP)
$ cd picrust2_out_pipeline/

$ add_descriptions.py -i EC_metagenome_out/pred_metagenome_unstrat.tsv.gz -m EC \
                    -o EC_metagenome_out/pred_metagenome_unstrat_descrip.tsv.gz
```

```
$ add_descriptions.py -i pathways_out/path_abun_unstrat.tsv.gz -m METACYC \
                      -o pathways_out/path_abun_unstrat_descrip.tsv.gz
```

These files are now ready for download and visualsation in STAMP.