16S Datasets

We're going to walk through a couple of different analyses of 16S datasets using two different tools, QIIME and Calypso

QIIME

Please note: We do not provide QIIME by default on the images since it requires a significant amount of space. QIIME is quite easy to install and the QIIME developers provide a number of preconfigured QIIME AWS instances that you should be able to start from your account.

To start a QIIME AWS image:

1. Log into your AWS account
2. Set the region to `US Oregon`. Note that is the only region supported by the QIIME developers at the moment
3. Click the `Launch` button to launch a new instance.
4. Under `Community AMIs` type `QIIME` in the search box. This will list the available AMIs. Use the appropriately latest version of the AMI
5. You should be able to use a `t2-micro` instance (one of the free tier instances) for this image.
6. Once the instance has launched you can continue with this tutorial.

Note: The qiime2 images require a username and password. The username is `qiime2` and the password is `qiime2`. If, like me, you have AWS set up to use keys, you may need to tell ssh to temporarily ignore them. For example:

```
ssh -o PubkeyAuthentication=no qiime2@54.187.96.46
```

will login to a machine with the IP address `54.187.96.46` using the username `qiime2` and request a password from you.

Please note there are lots of tutorials available on the QIIME website that walk you through different aspects of QIIME. This tutorial is designed to introduce you to some of the concepts.

Organisms in a drinking water sample

We have summarized data from a drinking water study from the University of Adelaide, Australia. The data comes from this study.

The metadata is also available in tab-separated format and we have a version for Calypso. (See below for the Calypso tutorial).

To import the data into QIIME, you need the sequences.fastq.gz and barcodes.fastq.gz in a directory by themselves. You can download these files from the drinking water directory. For this data, I created a script called split.py which reads all the sequences and splits them based on their barcodes. Notice that in /data/drinking_water/fastq we have selected just 20,000 sequences as a subsample of the data so that you can process the data quickly and efficiently. The entire data set is in /data/drinking_water/fastq_original in case you want to try and run the whole analysis.

Lets walk through running this through QIIME.

0. start qiime and create a directory for the analysis

```
mkdir -p ~/drinking_water/sequences
cd ~/drinking_water/
```

And now we download the sequences from GitHub:

```
curl -Lo sequences/barcodes.fastq.gz https://goo.gl/B58F7M
curl -Lo sequences/sequences.fastq.gz https://goo.gl/tVxpGf
curl -Lo metadata.tsv https://goo.gl/U5zUWQ
```

Note 1: these short Google URLs just point to the GitHub repository of the data, but they are easier to copy and paste! *Note 2:* make sure you include -L on the curl command as curl will need to follow the redirect from Google to GitHub etc!

Now we have a directory called sequences that has the sequence data, and the metadata in our own file, so the directory structure looks like this:

```
drinking_water/
drinking_water/metadata.tsv
drinking_water/sequences/
drinking_water/sequences/barcodes.fastq.gz
drinking_water/sequences/sequences.fastq.gz
```

We are in the directory `drinking_water`, and we can walk through analyzing those sequences:

1. Import the sequences into qiime

```
qiime tools import --type EMPSingleEndSequences --input-path sequences/ --output-path drinki
```

Once that is complete you should get the message:

```
Imported sequences/ as EMPSingleEndDirFmt to drinking_water.qza
```

2. Demultiplex the sequences. This separates the sequences based on the barcodes. As we noted in the description, this data set contains reads from five different sequencing runs, each of which comes from a different sample, either a drinking water site or related site.

```
qiime demux emp-single --i-seqs drinking_water.qza --m-barcodes-file metadata.tsv --m-barcod
```

Upon success you should see:

```
Saved SampleData[SequencesWithQuality] to: demultiplex.qza
```

3. Summarize that data and make a visualization file:

```
qiime demux summarize --i-data demultiplex.qza --o-visualization demultiplex.qzv
```

When this is complete you should see this message:

```
Saved Visualization to: demultiplex.qzv
```

4. Let's visualize that file by uploading it to the QIIME2 website. Start by copying the qzv file onto your computer:

```
scp -o PubkeyAuthentication=no qiime2@54.187.96.46:drinking_water/demultiplex.qzv .
```

Note 1: remember, the password is qiime2 if you used their image *Note 2:* change the IP address to that of your AWS instance

Then you can drag and drop the file to the QIIME2 visualization website.

The sequence count summary should tell you that there are 20,000 sequences per sample:

Sample name	Sequence count
SRR2080436	20,000
SRR2080434	20,000
SRR2080427	20,000
SRR2080425	20,000
SRR2080423	20,000

If you click on "Interactive quality plot" in the top left you will see a plot like this:

You can zoom in and out in the plot and look at the quality of the sequences.

Based on this plot and the information in red (read it!) we will trim to 195 bp using dada2. This removes low quality sequences so we are sure that we are finding the right organisms.

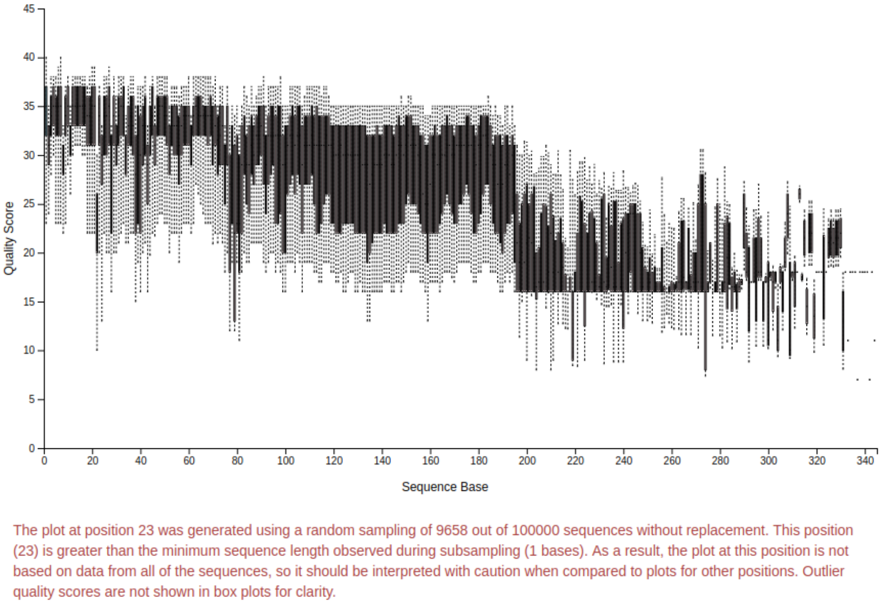
5. denoise that with dada2:

```
qiime dada2 denoise-single --i-demultiplexed-seqs demultiplex.qza --p-trim-left 0 --p-trunc-
```

When this has run, you get a new directory, `dada` with three output files in it. The output reports:

```
Saved FeatureTable[Frequency] to: dada/table.qza
Saved FeatureData[Sequence] to: dada/representative_sequences.qza
Saved SampleData[DADA2Stats] to: dada/denoising_stats.qza
```

These are QIIME format binary files, and so you can't easily read them. If you are curious, the files are zip archives, so you can make a copy of the file into a



The plot at position 23 was generated using a random sampling of 9658 out of 100000 sequences without replacement. This position (23) is greater than the minimum sequence length observed during subsampling (1 bases). As a result, the plot at this position is not based on data from all of the sequences, so it should be interpreted with caution when compared to plots for other positions. Outlier quality scores are not shown in box plots for clarity.

Figure 1:

temporary directory, `unzip` it, and poke around in the files.

6. Summarize the feature table and feature data

```
qiime feature-table summarize --i-table dada/table.qza --o-visualization table.qzv --m-sampl
qiime feature-table tabulate-seqs --i-data dada/representative_sequences.qza --o-visualizati
```

7. Again, you can view those two files using the QIIME2 viewer. You will need to copy the two `.qzv` files to your computer and you can drop them onto the upload link.

The **representative sequences** file contains information about sequences that represents the different groups in your data.

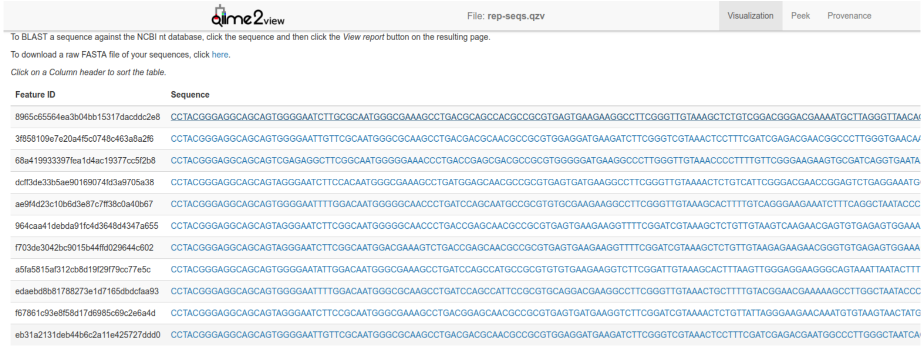
You can click on any of those sequences to BLAST them at the NCBI website, and you can also view the provenance of the sequences:

The **table** file summarizes information about the sequences.

Now that we have looked at the data summaries, we can explore the data in more detail. Lets start with a tree:

8. Generate the phylogenetic tree. We need to start with an alignment of the data:

```
qiime alignment mafft --i-sequences dada/representative_sequences.qza --o-alignment aligned-
```



To BLAST a sequence against the NCBI nt database, click the sequence and then click the *View report* button on the resulting page.

To download a raw FASTA file of your sequences, click here.

Click on a Column header to sort the table.

Feature ID	Sequence
8965c65564ea3b04bb15317dacddc2e8	CCTACGGGAGGCAGCAGTGGGGAATCTTGCGCAATGGGCGAAAGCCTGACGCAGCCACGCCGCGTGAGTGAAGAAGGCCTTCGGGTTGTAAAGCTCTGTCGGACGGGACGAAAATGCTTAGGGTTAACA
3f858109e7e20a4f5c0748c463a8a2f6	CCTACGGGAGGCAGCAGTGGGGAATTGTTCGCAATGGGCGCAAGCCTGACGACGCAACGCCGCGTGGAGGATGAAGATCTTCGGGTCGTAAACTCCTTTCGATCGAGACGAACGGCCCTTGGGTGAACA
68a419933397fea1d4ac19377cc5f2b8	CCTACGGGAGGCAGCAGTCGAGAGGCTTCGGCAATGGGGGAAACCCTGACCGAGCGACGCCGCGTGGGGGATGAAGGCCCTTGGGTTGTAAACCCCTTTTGTTCGGGAAGAAGTGCGATCAGGTGAATA
dcff3de33b5ae90169074fd3a9705a38	CCTACGGGAGGCAGCAGTAGGGAATCTTCCACAATGGGCGAAAGCCTGATGGAGCAACGCCGCGTGAGTGATGAAGGCCTTCGGGTTGTAAAACTCTGTCATTCGGGACGAACCGGAGTCTGAGGAAATG
ae9f4d23c10b6d3e87c7ff38c0a40b67	CCTACGGGAGGCAGCAGTGGGGAATTTTGGACAATGGGGGCAACCCTGATCCAGCAATGCCGCGTGTGCGAAGAAGGCCTTCGGGTTGTAAAGCACTTTTGTCAGGGAAGAAATCTTTCAGGCTAATACCC
964caa41debda91fc4d3648d4347a655	CCTACGGGAGGCAGCAGTAGGGAATCTTCGGCAATGGGGGCAACCCTGACCGAGCAACGCCGCGTGAGTGAAGAAGGTTTTCGGATCGTAAAGCTCTGTTGTAAGTCAAGAACGAGTGTGAGAGTGGAAA
f703de3042bc9015b44ffd029644c602	CCTACGGGAGGCAGCAGTAGGGAATCTTCGGCAATGGACGAAAGTCTGACCGAGCAACGCCGCGTGAGTGAAGAAGGTTTTCGGATCGTAAAGCTCTGTTGTAAGAGAAGAACGGGTGTGAGAGTGGAAA
a5fa5815af312cb8d19f29f79cc77e5c	CCTACGGGAGGCAGCAGTGGGGAATATTGGACAATGGGCGAAAGCCTGATCCAGCCATGCCGCGTGTGTGAAGAAGGTCTTCGGATTGTAAAGCACTTTAAGTTGGGAGGAAGGGCAGTAAATTAATACTTT
edaebd8b81788273e1d7165dbdcfaa93	CCTACGGGAGGCAGCAGTGGGGAATTTTGGACAATGGGCGCAAGCCTGATCCAGCCATTCCGCGTGCAGGACGAAGGCCTTCGGGTTGTAAACTGCTTTTGTACGGAACGAAAAAGCCTTGGCTAATACCC
f67861c93e8f58d17d6985c69c2e6a4d	CCTACGGGAGGCAGCAGTAGGGAATCTTCCGCAATGGGCGAAAGCCTGACGGAGCAACGCCGCGTGAGTGATGAAGGTCTTCGGATCGTAAAACTCTGTTATTAGGGAAGAACAAATGTGTAAGTAACTATG
eb31a2131deb44b6c2a11e425727ddd0	CCTACGGGAGGCAGCAGTGGGGAATTGTTCGCAATGGGCGCAAGCCTGACGACGCAACGCCGCGTGGAGGATGAAGATCTTCGGGTCGTAAACTCCTTTCGATCGAGACGAATGGCCCTTGGGCTAATCA

Figure 2:

However, when we're building trees, we want to ignore some really variable regions, and we can do that with a mask:

9. Mask highly variable positions

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

And then we can make a phylogenetic tree of this data

10. Use fasttree to build a tree

```
qiime phylogeny fasttree --i-alignment masked-aligned-rep-seqs.qza --o-tree unrooted-tree.q
```

11. Add a mid-point root to the tree:

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

We now have a phylogenetic tree of our data that we can explore. QIIME does not (currently) provide a mechanism to visualize this tree, but if you explore the files as I describe above, you can visualize a tree like this:

In addition to making some trees, we can also use the **representative reads** to generate a classification of our data. To do this, we use one of the QIIME machine learning models that they have already trained to classify data sets. In this example, we use the Greengenes 13_8 99% OTUs full-length sequences to test our data.

12. Generate the taxonomy based on this tree using a machine learning classifier

First, we need to download the file of trained classifiers

```
curl -Lo gg-13-8-99-nb-classifier.qza https://goo.gl/ZDg8eH
```

and now we can run the classifier and use that to create a table of the output

```
qiime feature-classifier classify-sklearn --i-classifier gg-13-8-99-nb-classifier.qza --i-re
qiime metadata tabulate --m-input-file taxonomy.qza --o-visualization taxonomy.qzv
```

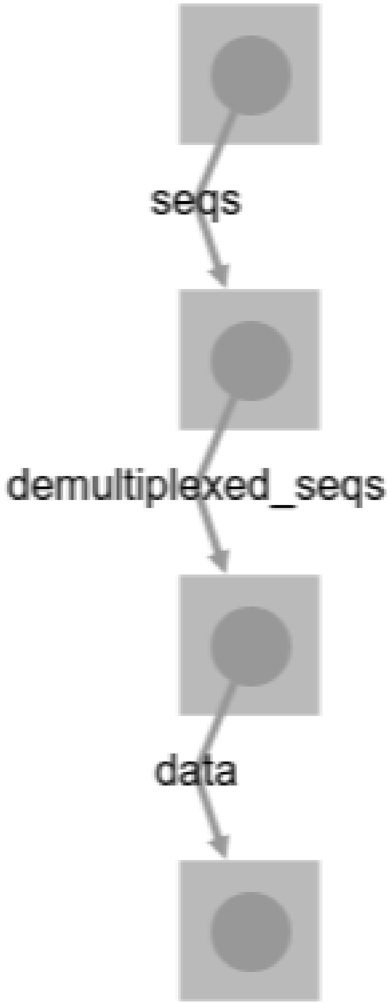


Figure 3:

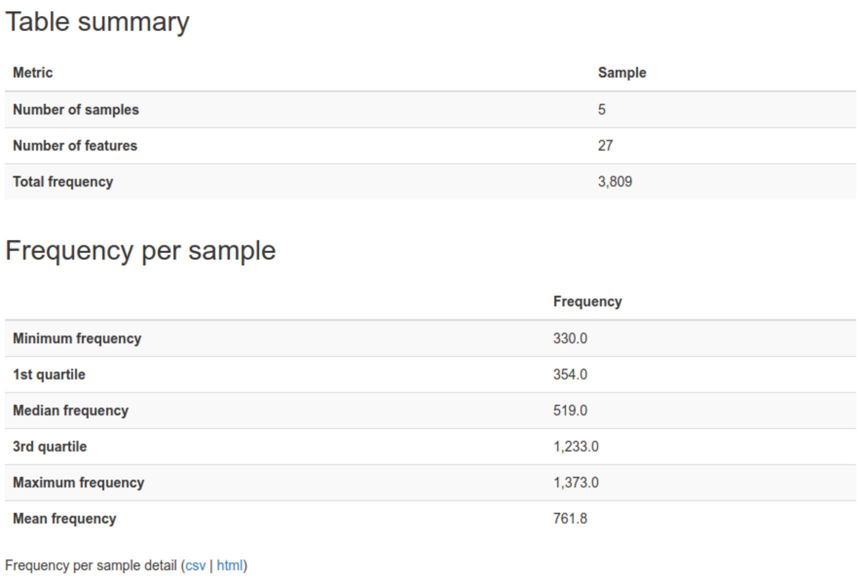


Table summary

Metric	Sample
Number of samples	5
Number of features	27
Total frequency	3,809

Frequency per sample

	Frequency
Minimum frequency	330.0
1st quartile	354.0
Median frequency	519.0
3rd quartile	1,233.0
Maximum frequency	1,373.0
Mean frequency	761.8

Frequency per sample detail (csv | html)

Figure 4:

13. Finally, we want to export this to tab separated values so we can load the classified data into excel or open office and use it to make graphs for our paper:

```
mkdir exported
qiime tools export --input-path taxonomy.qza --output-path exported
qiime tools export --input-path dada/table.qza --output-path exported
```

The two export commands create different files: The first command outputs the taxonomy as tab separated files into a file called `taxonomy.tsv`. This has three columns, the taxonomy ID, the taxonomy string, and the confidence in that identification.

The taxonomy string looks something like:

```
k__Bacteria; p__Actinobacteria; c__Actinobacteria; o__Actinomycetales; f__Propionibacteriace
```

These are the taxonomic rank kingdom, phyla, class, order, family, genus, and species, abbreviated as k__/p__/c__/o__/f__/g__/s__.

The second export command exports the features in Biological Observation Matrix (BIOM) format. This is useful for importing into other programs.

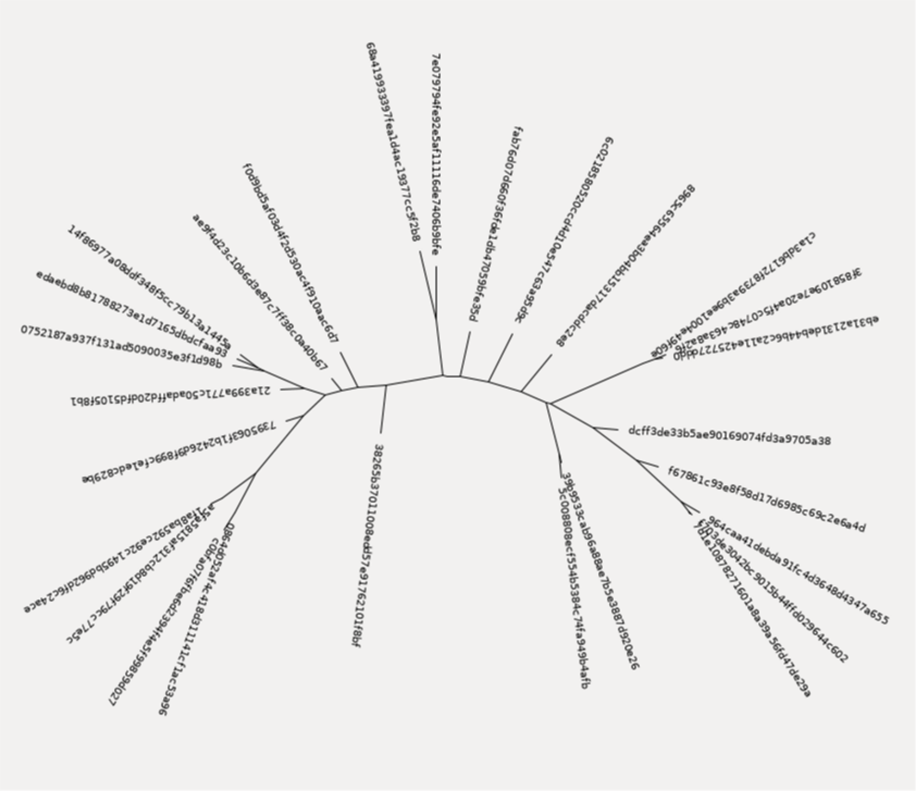

68a41993397fea1d4ac19377cc5f2b8
7e079794fe92e5af11116de7406b9bfe
fab76d07d660f36fde1db47059bfe35d
6c0218580520ccd4d10e547c63a95d9c
8965c6554ea3b04bb15317dacddc2e8
c1a3db6172f8739a3b9ee1004e49f60e
3f858109e7e20a4f5c0748c463a8a2f6
eb31a2131deb44b6c2a11e42572774d0
dcff3de33b5ae90169074fd3a9705a38
f67861c93e8f58d17d6985c69c2e6a4d
964caa41debda91fc4d3648d4347a655
f703de3042bc9015b44ffd029644c602
7b1e10878271601a8a39a56fd47de29a
39b9533cab96a88ae7b5e3887d920e26
5c008808ecf554b5384c74fa949b4afb
38265b37011008edd57e91762101f8bf
0864d052af4c418d311141cf1ac53a96
c0bfa07f6fbe6d2394f4e5f99859d027
a5fa5815af312cb8d19f29f79cc77e5c
11a8ba592ce92c1495bd962df6c24ace
7395063f1b2426d9f899cfe1edc829be
21a399a771c50adaffd20dfd5105f8b1
0752187a937f131ad5090035e3f1d98b
edaebd8b81788273e1d7165dbdcfaa93
14f86977a08ddf348f5cc79b13a1445a
ae9f4d23c10b6d3e87c7ff38c0a40b67
f0d9bd5af03d4f2d530ac4f910aac6d7

Figure 5:

Calypso

Calypso is an easy-to-use online software, allowing non-expert users to mine, interpret and compare taxonomic information from metagenomic or 16S rDNA datasets (Zakrzewski et al., 2017). The Calypso Web Site has lots of options for statistical analysis of metagenomes.

To upload your data to Calypso, you will need three files. These are the same files created above, but we have also provided them for download, too

- taxonomy.tsv
- [feature-table.biom]((../Datasets/drinking_water/Calypso/feature-table.biom)
- metadata

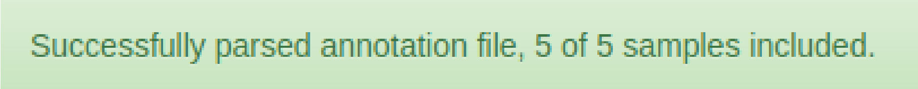
Start at http://cgenome.net/wiki/index.php/Calypso

Click on the Start Using Calypso button:

and then in the right hand pane, we are going to upload our data:

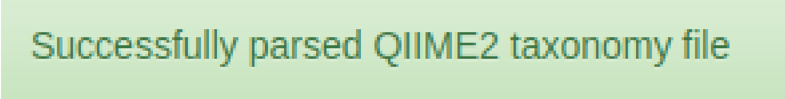
Notes about this form:

1. Leave display as standard 2. Upload the metadata_calypso.tsv metadata file. **NB:** this is a tab-separated V6 format metadata file. You should see a green message:



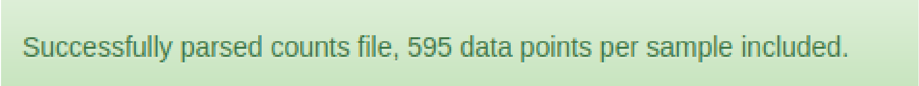
Successfully parsed annotation file, 5 of 5 samples included.

3. Upload the taxonomy.tsv file that you exported from QIIME2. You should see a green message:



Successfully parsed QIIME2 taxonomy file

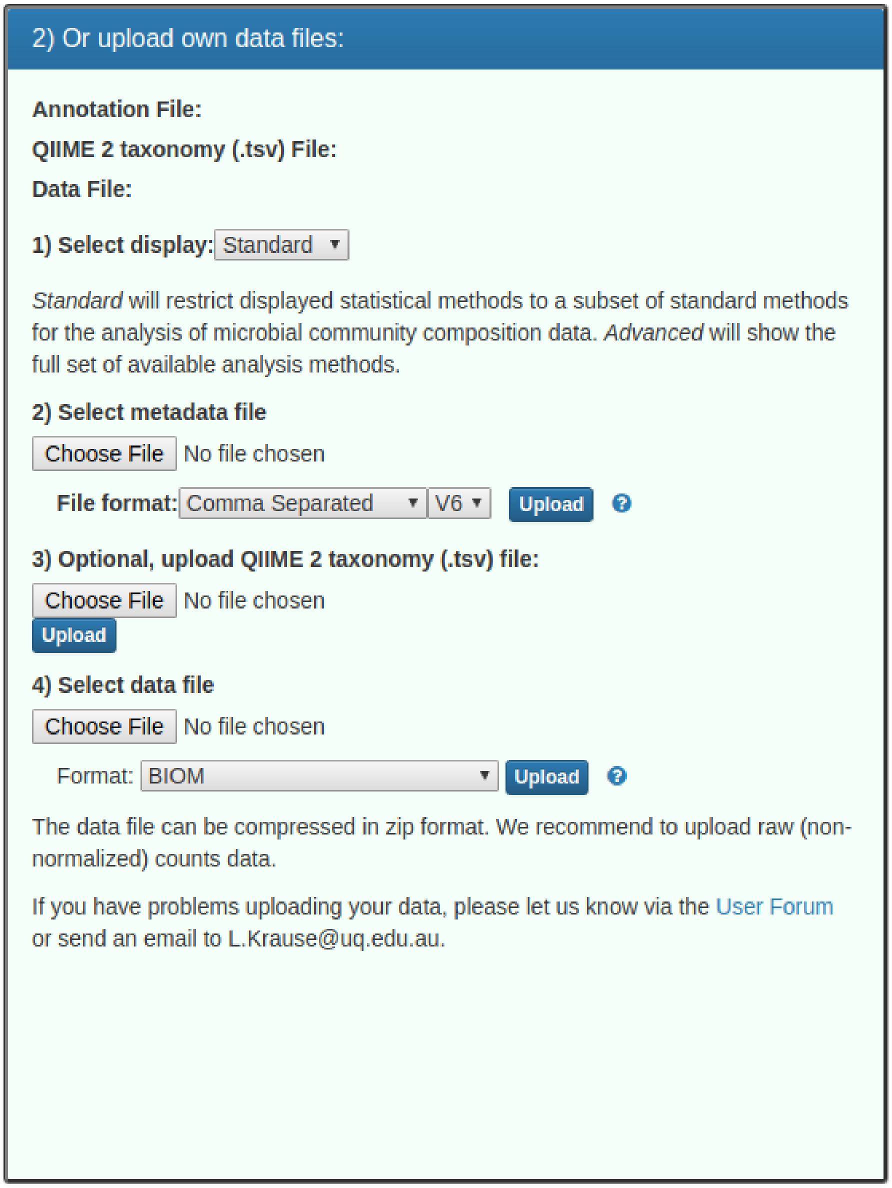
4. Upload the feature-table.biom file as a biom format file. You should see a green message:



Successfully parsed counts file, 595 data points per sample included.

5. Click the next button

On the next page, leave the data filtering box alone:



Annotation File:

QIIME 2 taxonomy (.tsv) File:

Data File:

1) Select display: Standard ▾

Standard will restrict displayed statistical methods to a subset of standard methods for the analysis of microbial community composition data. *Advanced* will show the full set of available analysis methods.

2) Select metadata file

File format: Comma Separated ▾ V6 ▾

3) Optional, upload QIIME 2 taxonomy (.tsv) file:

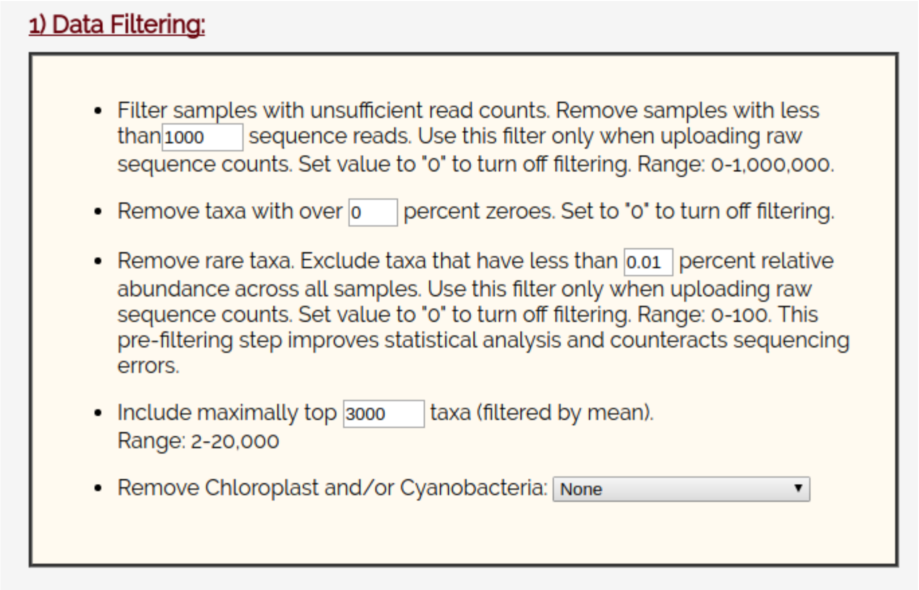
4) Select data file

Format: BIOM ▾

The data file can be compressed in zip format. We recommend to upload raw (non-normalized) counts data.

If you have problems uploading your data, please let us know via the User Forum or send an email to L.Krause@uq.edu.au.

Figure 6:

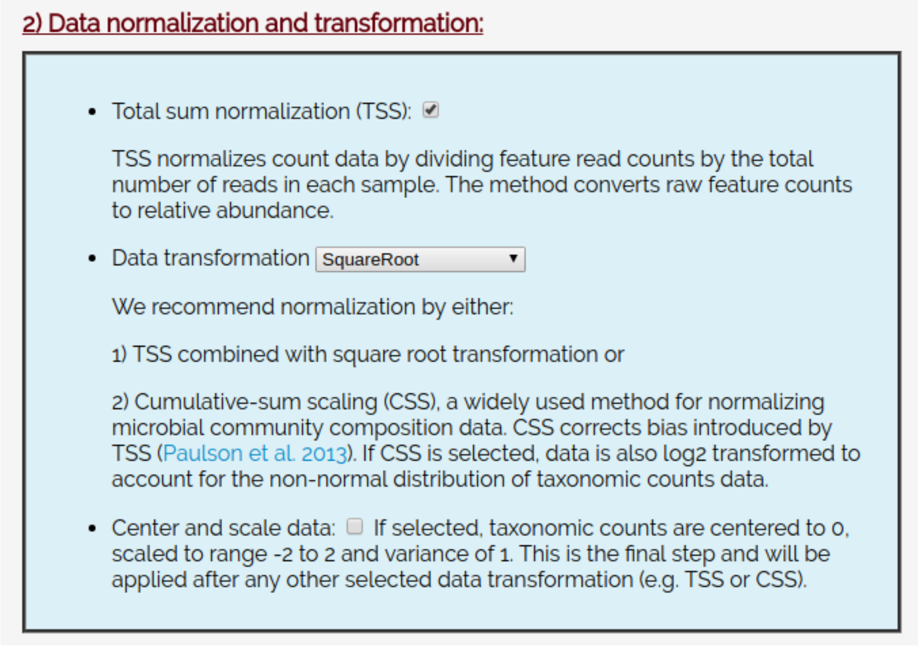


1) Data Filtering:

- Filter samples with unsufficient read counts. Remove samples with less than 1000 sequence reads. Use this filter only when uploading raw sequence counts. Set value to "0" to turn off filtering. Range: 0-1,000,000.
- Remove taxa with over 0 percent zeroes. Set to "0" to turn off filtering.
- Remove rare taxa. Exclude taxa that have less than 0.01 percent relative abundance across all samples. Use this filter only when uploading raw sequence counts. Set value to "0" to turn off filtering. Range: 0-100. This pre-filtering step improves statistical analysis and counteracts sequencing errors.
- Include maximally top 3000 taxa (filtered by mean).
  Range: 2-20,000
- Remove Chloroplast and/or Cyanobacteria: None

Figure 7:

And leave the data normalization set to TSS checked and Square Root.



2) Data normalization and transformation:

- Total sum normalization (TSS): ☑

  TSS normalizes count data by dividing feature read counts by the total number of reads in each sample. The method converts raw feature counts to relative abundance.
- Data transformation SquareRoot

  We recommend normalization by either:

  1) TSS combined with square root transformation or

  2) Cumulative-sum scaling (CSS), a widely used method for normalizing microbial community composition data. CSS corrects bias introduced by TSS (Paulson et al. 2013). If CSS is selected, data is also log2 transformed to account for the non-normal distribution of taxonomic counts data.
- Center and scale data: ☐ If selected, taxonomic counts are centered to 0, scaled to range -2 to 2 and variance of 1. This is the final step and will be applied after any other selected data transformation (e.g. TSS or CSS).

Click Filter and Normalize:

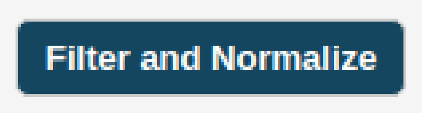


Figure 8:

After a few seconds, you should see:

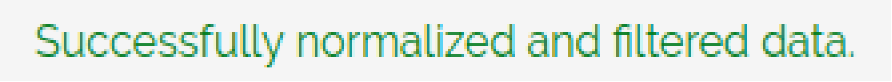
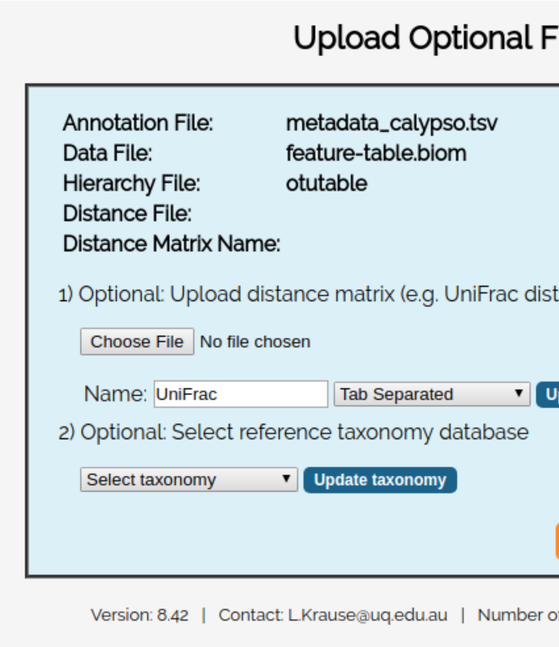

Successfully normalized and filtered data.

Figure 9:


Upload Optional F

Annotation File: metadata_calypso.tsv
Data File: feature-table.biom
Hierarchy File: otutable
Distance File:
Distance Matrix Name:

1) Optional: Upload distance matrix (e.g. UniFrac dist

Name: UniFrac
Tab Separated

2) Optional: Select reference taxonomy database

Version: 8.42 | Contact: L.Krause@uq.edu.au | Number o

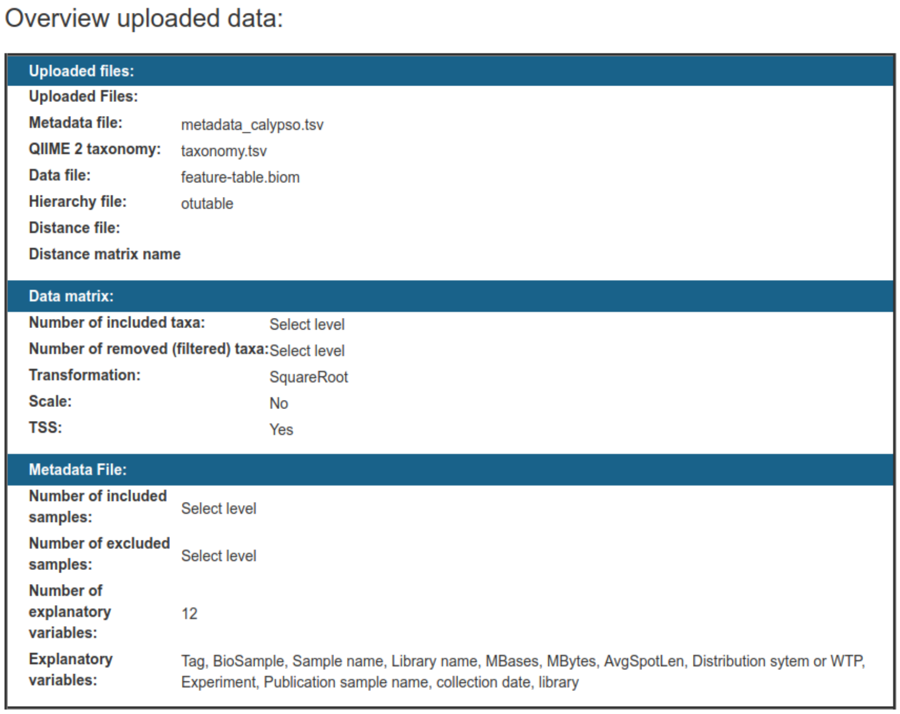
You'll be provided with the ability to upload any optional files.

For this example, we don't have any optional files, so you can just click skip.

Now you will be provided with a summary of your data:

And begin to explore these statistical tools:

You should explore each of them and see what they have to offer!



Overview uploaded data:

Uploaded files:

Uploaded Files:	
Metadata file:	metadata_calypso.tsv
QIIME 2 taxonomy:	taxonomy.tsv
Data file:	feature-table.biom
Hierarchy file:	otutable
Distance file:	
Distance matrix name	

Data matrix:

Number of included taxa:	Select level
Number of removed (filtered) taxa:	Select level
Transformation:	SquareRoot
Scale:	No
TSS:	Yes

Metadata File:

Number of included samples:	Select level
Number of excluded samples:	Select level
Number of explanatory variables:	12
Explanatory variables:	Tag, BioSample, Sample name, Library name, MBases, MBytes, AvgSpotLen, Distribution sytem or WTP, Experiment, Publication sample name, collection date, library

Figure 10:

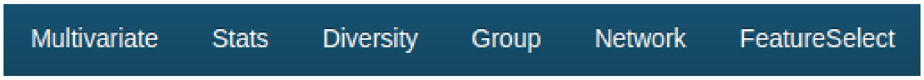
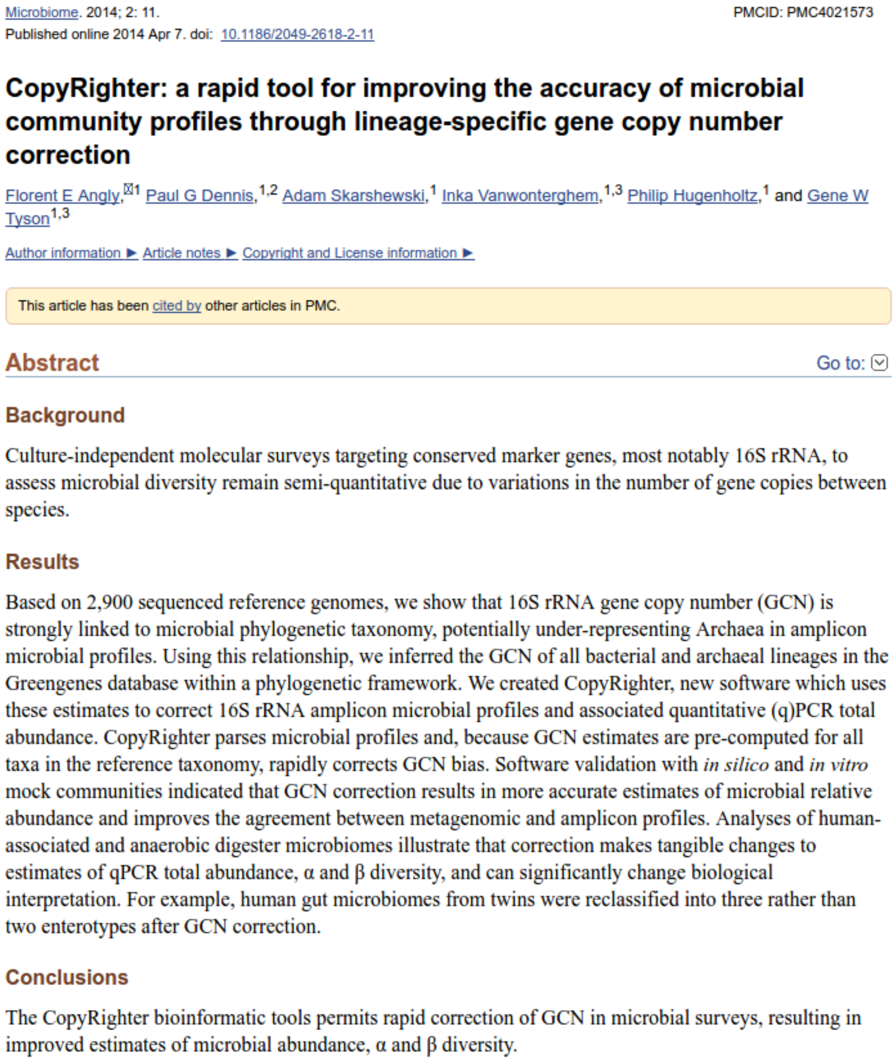


Figure 11:

Adjusting 16S abundance for copy number

Note that different organisms have different numbers of 16S genes, that skews 16S surveys. For example, E. coli has seven copies of the 16S gene, and Vibrio could have up to 13 copies of the 16S gene (https://www.ncbi.nlm.nih.gov/pubmed/19341395). We strongly recommend you read the CopyRighter paper (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4021573/) and consider how the abundance of 16S genes in different organisms affects your conclusions.



Microbiome. 2014; 2: 11.
Published online 2014 Apr 7. doi: 10.1186/2049-2618-2-11
PMCID: PMC4021573

CopyRighter: a rapid tool for improving the accuracy of microbial community profiles through lineage-specific gene copy number correction

Florent E Angly,✉[1] Paul G Dennis,[1,2] Adam Skarshewski,[1] Inka Vanwonterghem,[1,3] Philip Hugenholtz,[1] and Gene W Tyson[1,3]

Abstract

Background

Culture-independent molecular surveys targeting conserved marker genes, most notably 16S rRNA, to assess microbial diversity remain semi-quantitative due to variations in the number of gene copies between species.

Results

Based on 2,900 sequenced reference genomes, we show that 16S rRNA gene copy number (GCN) is strongly linked to microbial phylogenetic taxonomy, potentially under-representing Archaea in amplicon microbial profiles. Using this relationship, we inferred the GCN of all bacterial and archaeal lineages in the Greengenes database within a phylogenetic framework. We created CopyRighter, new software which uses these estimates to correct 16S rRNA amplicon microbial profiles and associated quantitative (q)PCR total abundance. CopyRighter parses microbial profiles and, because GCN estimates are pre-computed for all taxa in the reference taxonomy, rapidly corrects GCN bias. Software validation with *in silico* and *in vitro* mock communities indicated that GCN correction results in more accurate estimates of microbial relative abundance and improves the agreement between metagenomic and amplicon profiles. Analyses of human-associated and anaerobic digester microbiomes illustrate that correction makes tangible changes to estimates of qPCR total abundance, α and β diversity, and can significantly change biological interpretation. For example, human gut microbiomes from twins were reclassified into three rather than two enterotypes after GCN correction.

Conclusions

The CopyRighter bioinformatic tools permits rapid correction of GCN in microbial surveys, resulting in improved estimates of microbial abundance, α and β diversity.

Figure 12: copyrighter