

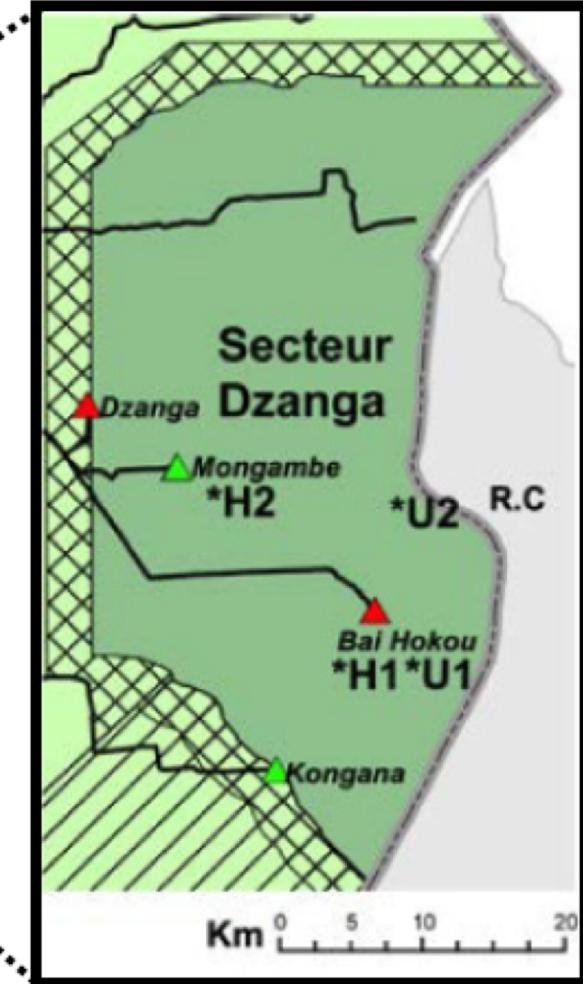
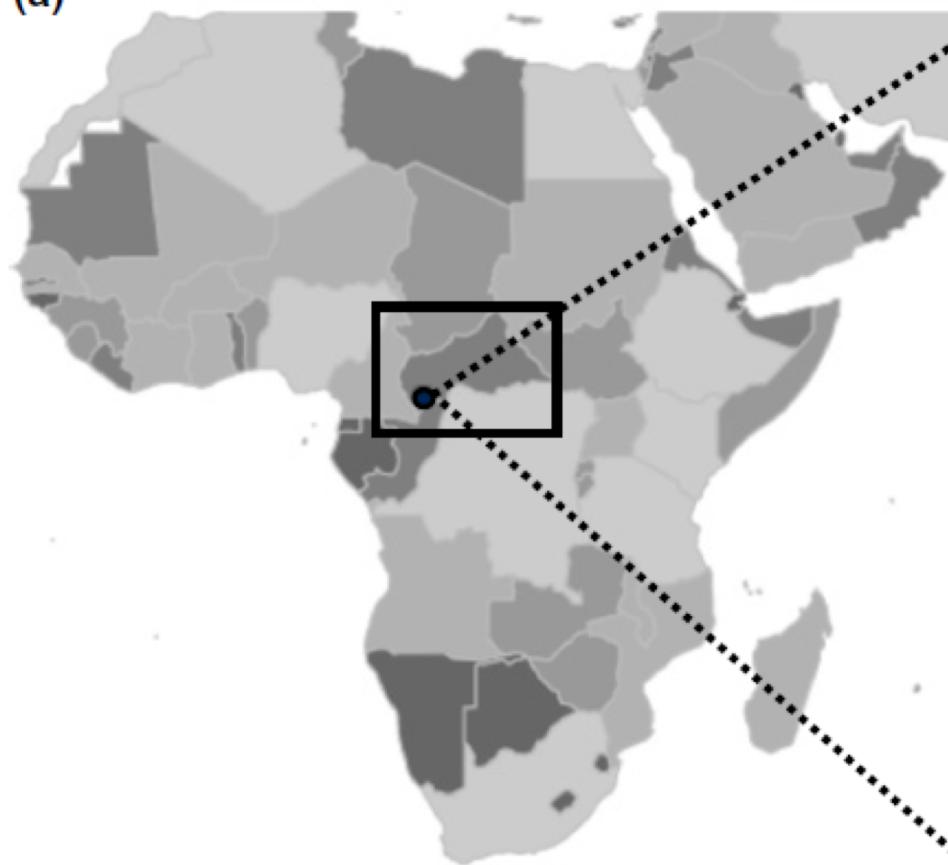
Functional Annotation of microbiomes using Qiime and Picrust

Bryan Featherstone & Dhantha Gunarathna

Gut microbiome composition and metabolomic profiles of wild western lowland gorillas (*Gorilla gorilla gorilla*) reflect host ecology

ANDRES GOMEZ,^{*†} KLARA PETRZELKOVA,^{‡ §¶***} CARL J. YEOMAN,^{† †} KLARA VLCKOVA,[§]
JAKUB MRÁZEK,^{‡ ‡} INGRID KOPPOVA,^{‡ ‡} FRANCK CARBONERO,^{§§} ALEXANDER ULANOV,^{¶¶}
DAVID MODRY,^{¶****} ANGELIQUE TODD,^{†††} MANOLITO TORRALBA,^{‡ ‡ ‡} KAREN E.
NELSON,^{‡ ‡ ‡} H. REX GASKINS,^{*†} BRENDA WILSON,^{*§§§} REBECCA M. STUMPF,^{*¶¶¶} BRYAN A.
WHITE^{*†} and STEVEN R. LEIGH^{*¶¶¶****}

(a)





Based out of Rob Knight's lab in University of California, San Diego.

World's largest crowd-funded citizen science project in existence

6,500 participants with over \$1 million in fundraising

Goals:

Build a large, public data set

Cost effective way for the general public to know their microbial composition

These data would contribute to a greater scientific effort to learn how the microbiome is



Open source bioinformatics pipeline
of microbial data

Can take raw sequence reads
through figures and statistical
analyses

Quality Control

OTU tables

Diversity measures

PICRUSt

Estimates gene families (functionally
annotate) by using categorized
bacteria or archaea of 16S rRNA
sequences

Blast like search against KEGG or
COG databases identifying protein-
coding genes

Installing PICRUSt

Make sure current python version is set to python 2.7

Pip install numpy==1.5

git clone git://github.com/picrust/picrust.git picrust

python setup.py install --user

No need to have root access. By using --user it will install locally

PATH=~/local/bin/:\$PATH

Downloading Data

1. MG-RAST fasta files

- a. <http://metagenomics.anl.gov/mgmain.html?mgpage=search&search=6321> OR Search Project 6321

2. MG-RAST metadata file

- a. Click download search results

3. Download data onto Proteus

MG-RAST Bryan

metagenomics.anl.gov/mgmain.html?mgpage=search&search=6321

MG-RAST search

metagenomics analysis server

6321| search

metadata function organism

Your search returned 60 results. Showing all matches download search results

Created ▾	Study ▾	Metagenome ▾	Seq Type ▾	Biome ▾	Country ▾	Location ▾
2014-09-22	Gut Microbiome composition of Western Lowland Gorillas (G.g.gorilla) and Mountain Gorillas (G.b.beringei)	9_9	metatranscriptome	terrestrial biome	Central African Republic	Central African Republic
2014-09-22	Gut Microbiome composition of Western Lowland Gorillas (G.g.gorilla) and Mountain Gorillas (G.b.beringei)	9_7	metatranscriptome	terrestrial biome	Central African Republic	Central African Republic
2014-09-22	Gut Microbiome composition of Western Lowland Gorillas (G.g.gorilla) and Mountain Gorillas (G.b.beringei)	9_8	metatranscriptome	terrestrial biome	Central African Republic	Central African Republic
2014-09-22	Gut Microbiome	9_6	metatranscriptome	terrestrial	Central African	Central African

Refine Search

Add a search term for a specific metadata field to refine your search. You can use the asterisk (*) symbol as a wildcard.

field PI firstname

term enter searchterm add

Searches [?] Collections [?]

you must be logged in to view stored searches

create new [?]

Store the parameters of your search query.

name enter name

description enter description (optional)

Links 10:25 AM 2/22/2017

Metadata

Downloading Data

1. MG-RAST fasta files

- a. <http://metagenomics.anl.gov/mgmain.html?mgpage=search&search=6321> OR Search Project 6321

2. MG-RAST metadata file

- a. Click download search results

3. Download data onto Proteus

Downloading Data onto Proteus

```
work_dir <- '~/genStats/gorilla'
work_dir <- '~/genStats/gorilla'
dir.create(work_dir, showWarnings=FALSE, recursive=TRUE)

work_dir <- '~/genStats/gorilla'
dir.create(work_dir, showWarnings=FALSE, recursive=TRUE)
work_dir <- '~/genStats/gorilla'
dir.create(work_dir, showWarnings=FALSE, recursive=TRUE)

file <- read.table('~/searchResultsMetadata.txt',
                    sep='\t', header=TRUE, stringsAsFactors=FALSE)
urls <- file[file$sequence_type == 'Amplicon',]$url

out_dir <- file.path(work_dir, 'sequences')
dir.create(out_dir, showWarnings=FALSE, recursive=TRUE)

target <- 'http://api.metagenomics.anl.gov/1/download/%s?file=150.1'

for (url in urls){
  id <- gsub('.*metagenome\\/(.*$', '\\\\1', url)
  file_url <- sprintf(target, id)
  out_path <- paste0(file.path(out_dir, id), '.fna')
  download.file(file_url, destfile=out_path)
```

Data

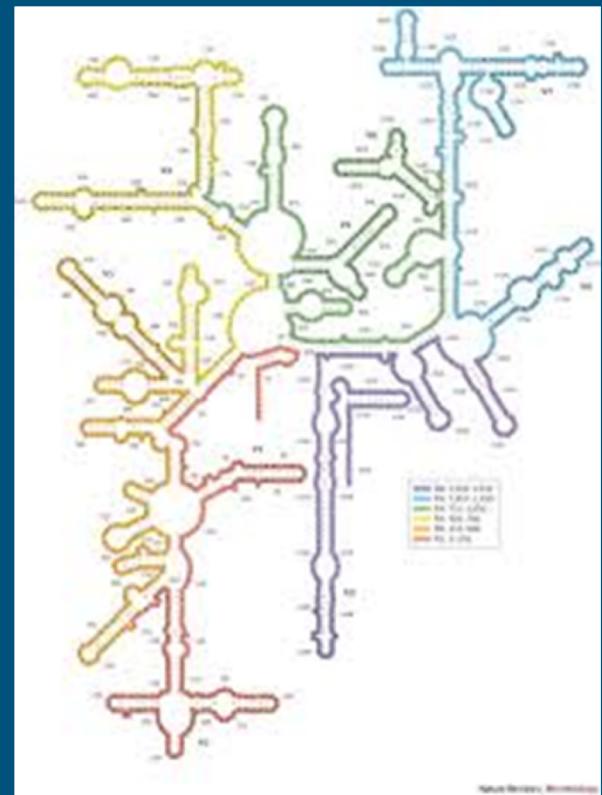
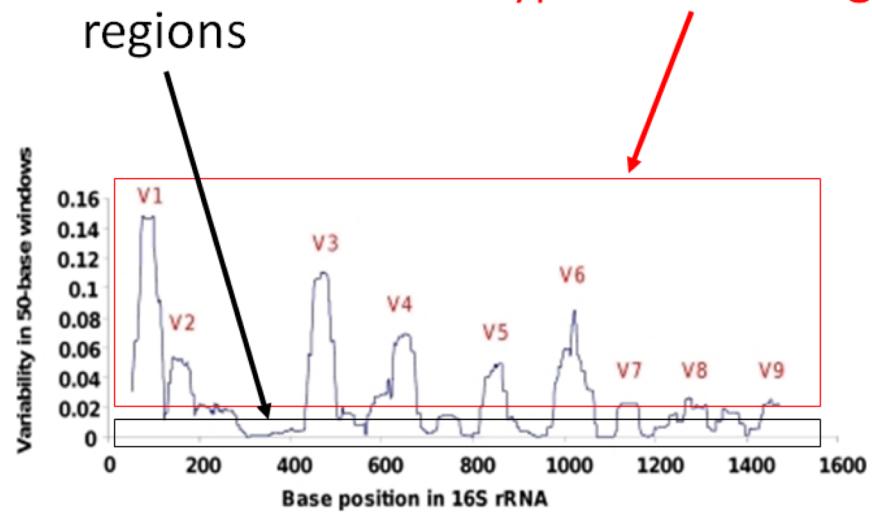
— Fasta files

Reads

```
bsf44@proteus01:~/fastaj> ls
mgm4539180.3.fasta mgm4539181.3.fasta mgm4539182.3.fasta mgm4539183.3.fasta mgm4539184.3.fasta mgm4539185.3.fasta mgm4539186.3.fasta mgm4539187.3.fasta mgm4581310.3.fasta mgm4581339.3.fasta
bsf44@proteus01:~/fastaj> head -n 10 mgm4539180.3.fasta
HCXV15102JH4NE
TTTACCGCCGGCTGCTGGCACGTAGTTAGCCGGTCTTAAAGGTACACTCACTCTCGCTGCTCAATTAAAAGCGTTAACCCGAAGGCCCTCATCCCGACGGCGTCGCTGCATCAGGCTTCGCCATTGGAATATTCCCACGTGCTGCCCTCCGTAGGAGTCTGGCCGTCTCA
TCCCAATCTGGCGGCTGGTCTCAACCCGGCTACCCAATCGCTGGTGGGCCCTGGCCGGCCAACTAGCTAACAGGGCGGGCCCCATCCCTCGCGTCGGCTTCCCTCCGCGGCGATGCCGCTGCGGGAGGGTATCCGTATTACCCACGTTCCGGAGGCCGAGGGCAGGTA
TACCGCCGGCTGCTGGCTATACCCCTGAGCCATAATCACT
HCXV15102JK09Z
TTTACCGCCGGCTGCTGGCACGTAGTTAGCCGGTCTTAAAGGTACACTCACTCTCGCTGCTCAATTAAAAGCGTTAACCCGAAGGCCCTCATCCCGACGGCGTCGCTGCATCAGGCTTCGCCATTGGAATATTCCCACGTGCTGCCCTCCGTAGGAGTCTGGCCGTATCTAGCCAA
TGCCCGGCCCTCTCAGGCCGGTACCCGCTGAAGCCATTACCTCACCAACAGCTGATAGGACGGCACCCATCTCACCGCTAACGCTTCCCTAACAAACATGTGAATAGTTGGAGCATCCGGATTACCCGGTTCAGGAGCTATCCGGTATGAGGCAGGTTAGTCACCGA
TAGCGAACCCGTTGCCACTTCACCATCAAGCAAGCTGATGGATCCGTTGACTTCATGTGTTAACGCTGCCCA
HCXV15102JBDIS
TTTACCGCCGGCTGCTGGCACGTAGTTAGCCGGCTTCTGGTATGGTACCGTCAAACAAAAATCATCCCTATTAGCATTTCTCCCATACAACAGTGCCTTACGACCCGAAGGCCCTCATCACACCGCCGGTGTCCCATCAGGCTTGCCTCCATTGGAAGATTCCCACGTGAGCTCCCGTAGGAGT
GGCCGGTGTCTCAGTCCCAATGTGGCGATCAGTCTCAACTCGGTATGCATCATCGTCTGGTAAGCCTTACCCCAACCTAACACTAATGACCCGGATCCATCTAGGTGACGCCGTAGCGCTTTAACTTGATATCATCGGATACTAAGTTTATTGGTATTAGCATCTGTTCTAAATGTTATCCCAC
CTTGGAGGGCAGGTATTCACGTGTTACTCACCGTTGCCACTCGCTGAAGGGTCAAGCACCTCTCGCTGGCATTGACTTGATGTTAGGCACGCC
HCXV15102JMPHL
TTTACCGCCGGCTGCTGGCACGTAGTTAGCCGGCTTCTTACAGGGTACCGTCACTTCTCGCTCCCTGCACAGAGGTTAACATCGAAAACCTTCTCCCTACCGCCGGTGTGGTCAGGGTGGCCATTGCCCAACTAATCCCACCTGCTGCCCTCCGTACGGAGTCTGGACCGTGTCTCAGTCCAG
GTGGCCGGATCAGCTCGCTCAGCTCGCTGGTGGCCGTTACCTCACCAACTACCTAATGGGACCGAATCTATTCGAGCCGATTCTCTTGAACCTTCAACATGTGTTATTGGTGTCTATGGGTATTAG
HCXV15102JAL8
TTTACCGCCGGCTGCTGGCACGTAGTTAGCCGGCTTCTGGTATGGTACCGTCAAACAAAAATCATCCCTATTAGCATTTCTCCCATACAACAGTGCCTTACGACCCGAAGGCCCTCATCACACCGCCGGTGTCCCATCAGGCTTGTGCTCCATTGGAAGATTCCCACGTGAGCTCCCGTAGGAGT
TGCCGGTGTCTCAGTCCCAATGTGGCGATCAGTCTCAACTCGGTATGCATCATCGTCTGGTAAGCCTTACCCACCAACTAACATGACCCGGATCCATCTAGGTGACGCCGTAGGCCCTTTAACCTGATATCATCGGATACTAAGTTATTGGTATTAGCATCTGTTCTAAATGTTA
bsf44@proteus01:~/fastaj>
```

16S Gene

- Highly conserved regions
- Hypervariable regions

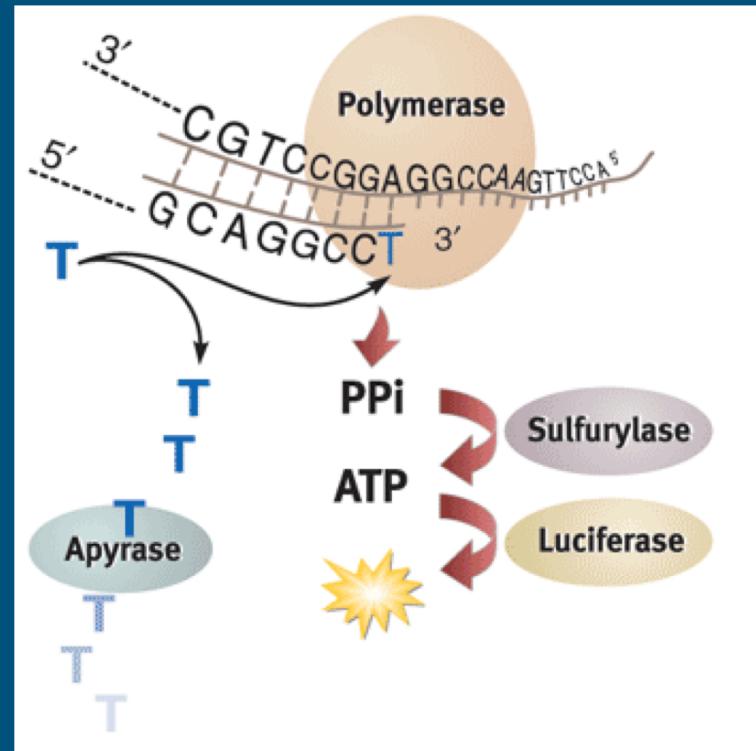


454 Pyrosequencing

Next gen sequencing technology using pyrophosphate release on nucleotide incorporation.

Difficult to read consecutive repeated nucleotides

Denoising to correct for this error



Downloading green genes

Green genes can be downloaded using the following link

http://qiime.org/home_static/dataFiles.html

Green genes is mostly 16S rRNA for browsing, blasting and probing

Picrust requires to use green genes database

All the genes in the green genes database are functionally annotated

Qiime Commands

1. Check user's metadata mapping file for required data

```
validate_mapping_file.py -m Metadata.txt -o  
validate_mapping_file_output
```

2. Add qiime labels. Metadata mapping file with sample IDs and fasta file names

```
add_qiime_labels.py -i fasta/ -m Metadata_corrected.txt -c SampleID -o  
combined_fasta
```

3. Pick closed reference OTUs

```
picked_closed_reference_otus.py -r 97 otus.fasta -t 97 otu_taxonomy.txt
```

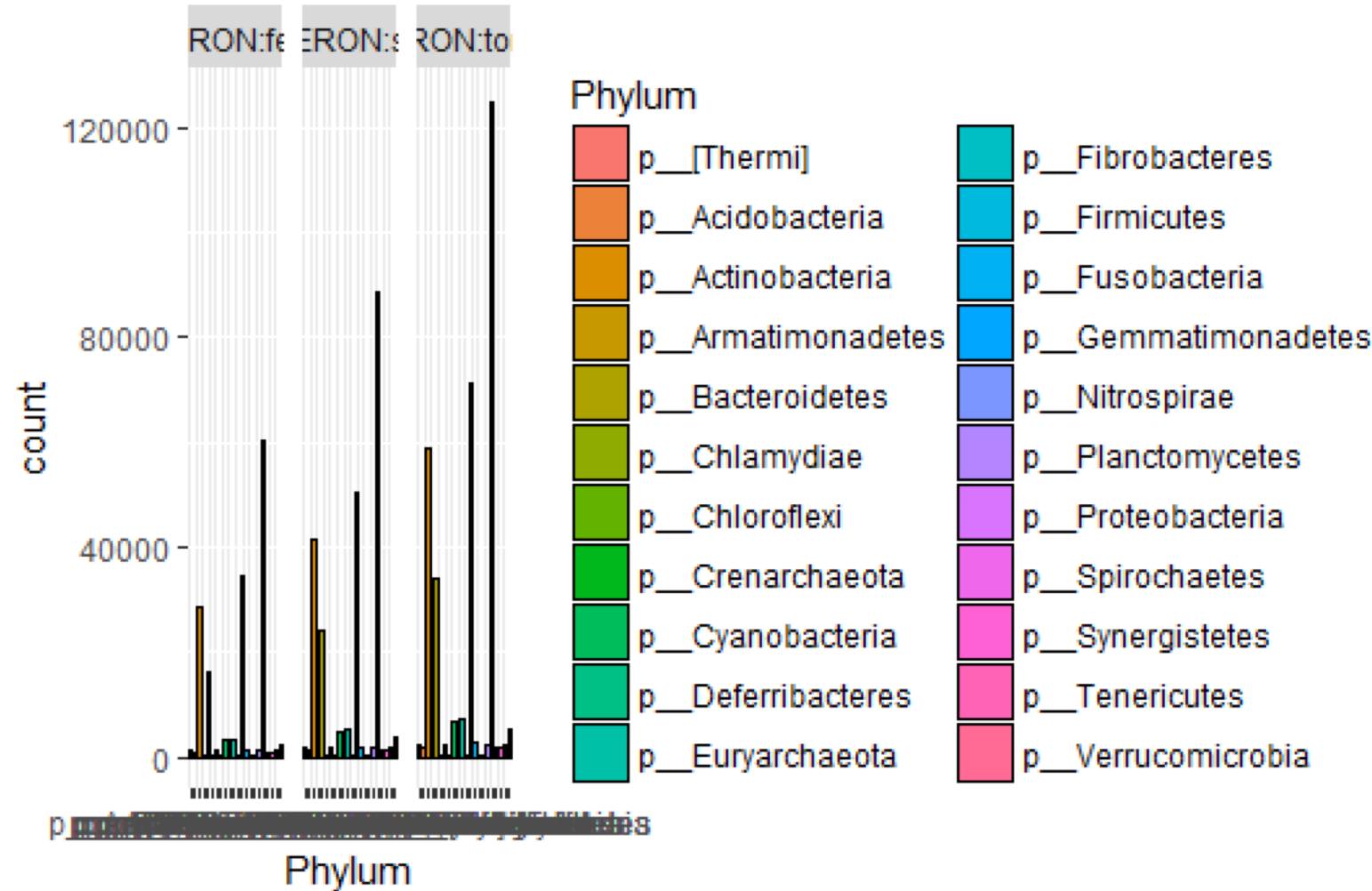
Qiime commands

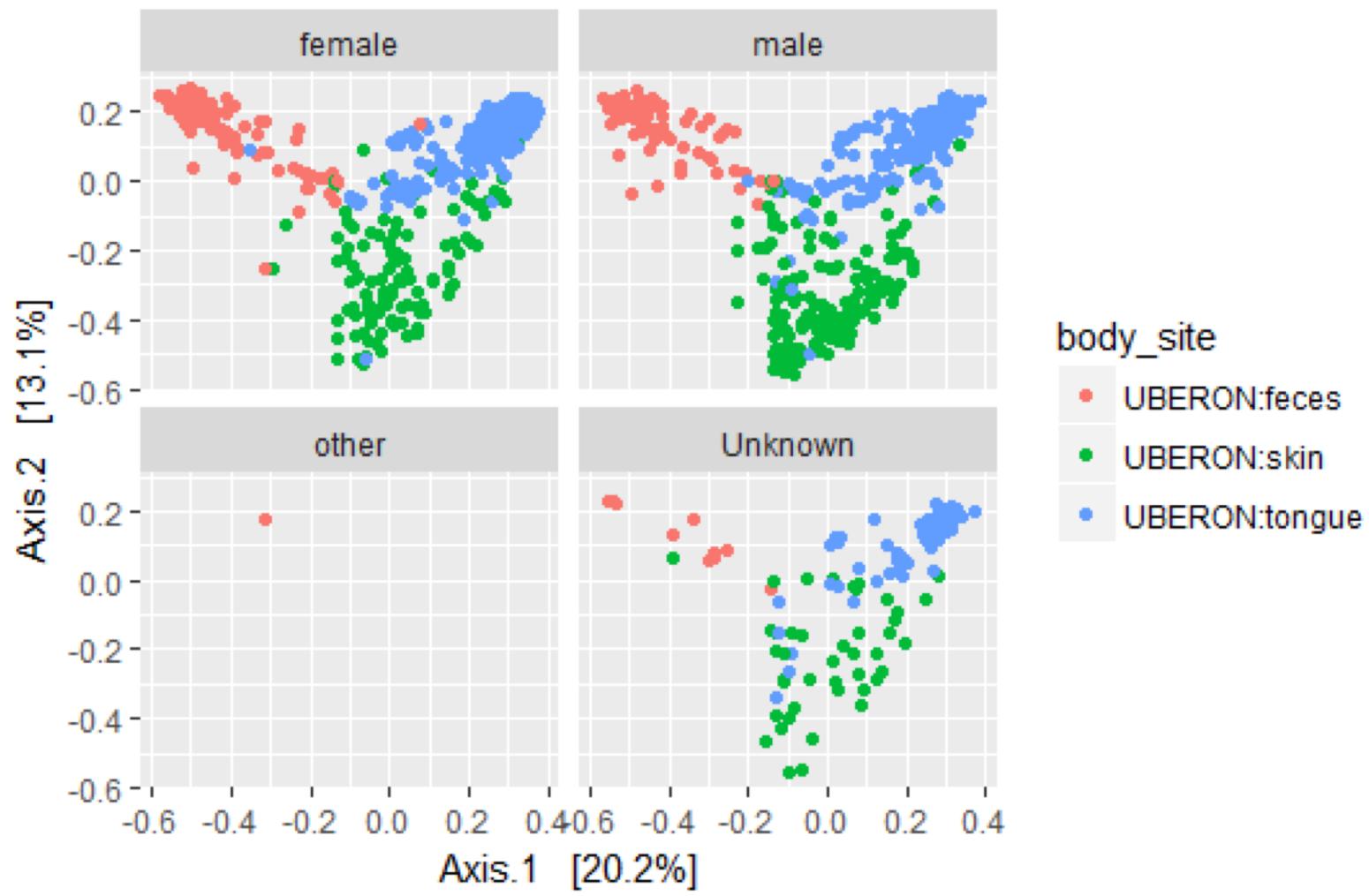
4. Summarize Taxa

```
summarize_taxa_through_plots.py -o taxa_summary -i  
picked_otus_default/otu_table.biom -m Metadata_corrected.txt
```

5. Convert biom (binary file) to json

```
biom convert -t table.txt -o table.from_txt_json.biom --table-type="OTU  
table" --to-json
```





PICRUSt Scripts

1. Normalized OTU table

```
normalized_by_copy_number.py -i picked_otus_default.biom -o  
normalized_otus.biom
```

2. Predict functions for metagenome

```
predict_metagenomes.py -i normalized_otus.biom -o  
metagenome_predictions.biom (-t cog to use COG database)
```

3. Get tab delimited file with NSTI values

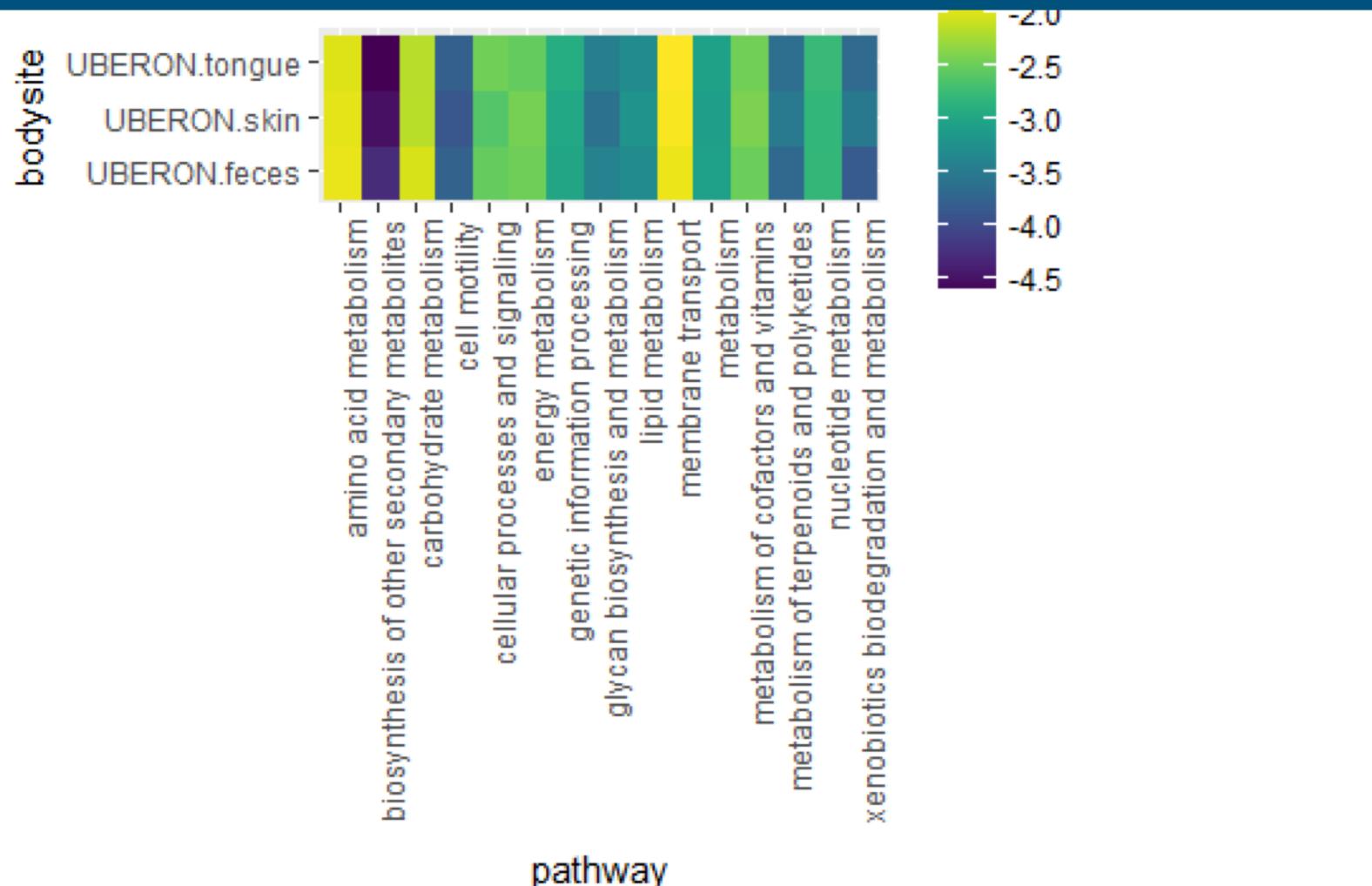
```
perdict metagenomes.py -f -i normalized_otus.biom -o  
metagenome_predictions.tab -a nsti_per_sample.tab
```

KEGG and COG database

KEGG - Database used to map molecular datasets, especially large in genomics, transcriptomics, proteomics, and metabolomics for system functions. Based on speciation events and phylogenetics

Levels of Hierarchy

COG - database consisting of orthologous proteins found across multiple lineages which are likely to correspond to an ancient conserved domain



Other PICRUSt Scripts

The screenshot shows a web browser displaying the PICRUSt 1.1.0 documentation at picrust.github.io/picrust/scripts/. The page title is "PICRUSt Script Index". The main content area lists various PICRUSt scripts with their descriptions:

- [*ancestral_state_reconstruction.py*](#) - Runs ancestral state reconstruction given a tree and trait table.
- [*categorize_by_function.py*](#) - Collapse table data to a specified level in a hierarchy.
- [*compare_biom.py*](#) - Compare the accuracy of biom files (expected and observed) either by observations (default) or by samples.
- [*evaluate_test_datasets.py*](#) - Evaluate the accuracy of character predictions, given directories of expected vs. observed test results.
- [*format_tree_and_trait_table.py*](#) - Formatting script for filtering and reformatting trees and trait tables.
- [*make_test_datasets.py*](#) - Generates test datasets for cross-validation studies of PICRUSt's accuracy.
- [*metagenome_contributions.py*](#) - This script partitions metagenome functional contributions according to function, OTU, and sample, for a given OTU table.
- [*normalize_by_copy_number.py*](#) - Normalize an OTU table by marker gene copy number.
- [*parallel_predict_traits.py*](#) - Runs *predict_traits.py* in parallel.
- [*pool_test_datasets.py*](#) - Pool character predictions within a directory, given directories of expected vs. observed test results.
- [*predict_metagenomes.py*](#) - This script produces the actual metagenome functional predictions for a given OTU table.
- [*predict_traits.py*](#) - Given a tree and a set of known character states (observed traits and reconstructions), output predictions for unobserved character states.
- [*print_picrust_config.py*](#) - Print out the PICRUSt config settings.
- [*run_genome_evaluations.py*](#) - Runs genome evaluations on PICRUSt.
- [*scale_metagenome.py*](#) - This script converts metagenomic relative abundance back to sequence counts, by scaling the relative abundance of each gene in each sample in a biom file by a user-supplied sequencing depth.
- [*start_parallel_jobs.py*](#) - Starts multiple jobs in parallel on multicore or multiprocessor systems.
- [*start_parallel_jobs_sge.py*](#) - Starts multiple jobs in parallel on SGE/qsub based multiprocessor systems.
- [*start_parallel_jobs_torque.py*](#) - Starts multiple jobs in parallel on Torque/qsub based multiprocessor systems.

The right sidebar contains links for "This Page", "Show Source", and "Quick search". The bottom navigation bar includes links for "PICRUSt 1.1.0 documentation" and "index".