**Qiime1 commands to to pick OTUs.**
From EBI website: https://www.ebi.ac.uk/ena/data/view/PRJEB14674, we can find **376** samples, of which **28** are blank samples with scientific name “metagenome”, **348** are with scientific name “human gut metagenome”. We just need to download those 348 samples by clicking “Submitted files (FTP)” column. Some other columns are also download links, but names of files downloaded by clicking “Submitted files (FTP)” are the same as sample IDs in mapping file.

The mapping file provided by authors is very confusing, because it contains more useless sample IDs. There are 384 samples recorded in authors’ mapping file. 36 samples are useless, including 33 blank samples and 3 samples (EP0094201, GMWA.2010, GMWA.2043) not being uploaded to EBI website. In EBI website, we cannot find these 3 samples at all. Then, we can see that the remaining 348 samples are consistent with 348 ones from EBI.

**Use these 348 samples to do analysis.**

During the processes, **21 samples** are excluded from these 348 samples, and then **327 samples are kept.** From these 21 exclusions, 3 samples are deleted automatically by pick OTU process. The output of “picking OTU” step just doesn’t keep these 3 samples. 14 samples are deleted because their reads are less than 5000, which can be seen from the output. From the mapping file, authors indicated that 5 samples should be deleted because 2 of them are “bad metadata” and 3 of them are “duplicate samples”. In addition, a sample is the overlap between 14 samples, whose reads are less than 5000, and 5 samples, who are in exclusion list written by authors. Then, we should exclude 21 samples from these 348 ones. (3+14+5-1=21)

**script1:** validate_mapping_file.py -m     Fasting_Map.txt -o validate_mapping_file_output

This script check whether your mapping file is valid. The mapping file is obtained from authors of the publication.

**validate_mapping_file.py -m** map.txt **-o** vmf-map/

**script2:** split_libraries_fastq.py -i lane1_read1.fastq.gz,lane2_read1.fastq.gz --sample_ids my.sample.1,my.sample.2 -o slout_not_multiplexed_q20/ -q 19 --barcode_type 'not-barcoded'

This script performs demultiplexing of Fastq sequence data where barcodes and sequences are contained in two separate fastq files. We need to input all sample IDs and their file names, respectively.

**split_libraries_fastq.py –i** *type 348 samples' file name here, use comma to separate them* **--sample_id** *type 348 samples' IDs here, use comma to separate them* **-o** slout348/ **-m** map.txt **-q** 19 **--barcode_type** 'not-barcoded' **--phred_offset**=33

**script3:**
pick_closed_reference_otus.py -i $PWD/seqs.fna -r $PWD/refseqs.fna -o $PWD/otus_sortmerna/ -p $PWD/sortmerna_params.txt -t $PWD/taxa.txt
using SortMeRNA 2.0against the August 2013 release of the Greengenes 16S rRNA gene sequence database at 97% similarity to pick out. The parameters are set in "params.txt"

**pick_closed_reference_otus.py -o** 348clotus/ **-i** slout348/seqs.fna **-p** params.txt

Filters samples from an OTU table on the basis of the number of observations in that sample, or on the basis of sample metadata. Mapping file can also be filtered to the resulting set of sample ids

**Script4:** filter_samples_from_otu_table.py -i otu_table.biom -o filtered_otu_table.biom --sample_id_fp ids.txt
After scrip3, we can get the file of OTU table with suffix 'biom'. Scrip4 is to delete all exclusion samples and keep 327 samples in out table.

**filter_otus_by_sample.py -i** 348clotus/sortmerna_picked_otus/seqs_otus.txt **-f** slout348/seqs.fna -s 10567.EC0004201,10567.EC0104201,10567.EP0127201,10567.EP0201201B,10567.FC0024201,10567.FC0333201,10567.FP0016201,10567.FP0016201A,10567.FP0071201,10567.FP0073201,10567.FP0193201,10567.FP0211201,10567.FP0225201,10567.FP0421201,10567.FP0438201,10567.GMWA.1034,10567.GMWA.1036,10567.GMWA.1038,10567.GMWA.1078,10567.GMWA.1090,10567.GMWA.2005 **-o** filtered_otus/

**script5:** make_otu_table.py -i otu_map.txt -o otu_table.biom

This script is to change the result of script4 into ‘BIOM’ file.

**make_otu_table.py -i** filtered_otus/seqs_sfiltered_otus.txt **-o** 348clotus/otu_table_filter.biom

**script6:** filter_otus_from_otu_table.py
Filter OTUs from an OTU table based on their observation counts or identifier

**filter_otus_from_otu_table.py -i** 348clotus/otu_table_filter.biom **-o** 327otus/otn2.biom **-n** 2 **## delete those OTUs whose total reads are 1 and 2. This result includes 4710 OTUs**

**filter_otus_from_otu_table.py -i** 327otus/otn2.biom **-o** 327otus/327no2nof.biom **-e** aligned327/rep_set1_failures.fasta **## delete those OTUs whose total reads are 1 and 2, and delete OTUs that fail to be aligned after picking representative sequence. This result includes 4706 OTUs. ## rep_set1_failures.fasta is obtained from other three steps (see additional steps)**

**command:** biom convert -i input -o output
change out table from “biom” format to txt format with taxonomy.

**command:** biom summarize-table –i
This command to open OUT table (in biom) directly in terminal.

**biom convert -i** otn2.biom **-o** 327no2nof.txt **--to**-tsv --header-key taxonomy **## 4710 OTUs**

**biom convert -i** 327no2nof.biom **-o** 327no2nof.txt **--to**-tsv --header-key taxonomy **## 4706 OTUs**

## Additional steps to get rep_set1_failures.fasta:

```
split_libraries_fastq.py -i lane1_read1.fastq.gz,lane2_read1.fastq.gz --sample_ids
my.sample.1,my.sample.2 -o slout_not_multiplexed_q20/ -q 19 --barcode_type 'not-barcoded'
```

This script performs demultiplexing of Fastq sequence data where barcodes and sequences are contained in two separate fastq files. We need to input all sample IDs and their file names, respectively.

**split_libraries_fastq.py –i** *type 327 samples' file name here, use comma to separate them* **--sample_id** *type 327 samples' IDs here, use comma to separate them* **-o** 327sl/ **-m** map.txt **-q** 19 **--barcode_type** 'not-barcoded' **--phred_offset**=33

```
pick_rep_set.py -i seqs_otus.txt -f seqs.fna -o rep_set1.fna
```

After picking OTUs, we can then pick a representative set of sequences.

**pick_rep_set.py -i** filtered_otus/seqs_sfiltered_otus.txt **-f** 327sl/seqs.fna **-o** pickrepo327/rep_set.fna

```
align_seqs.py -i $PWD/unaligned.fna -t $PWD/core_set_aligned.fasta.imputed -o
$PWD/pynast_aligned_defaults/
```

This script aligns the sequences in a FASTA file to each other or to a template sequence alignment, depending on the method chosen

**align_seqs.py -i** pickrepo327/rep_set.fna **-o** aligned327