Part 1. Amplicon sequencing.

1. Installation of QIIME

via conda:

```
cd /media/daria/DaryaNika/IB_fall2020/project7
conda activate qiime2-2020.2
```

installation via docker:

Download QIIME 2 Image

```
docker pull quay.io/qiime2/core:2021.2
```

confirm that the image was successfully fetched.

```
docker run -t -i -v $(pwd):/data quay.io/qiime2/core:2021.2 qiime
```

run:

```
docker run --rm -v $(pwd):/data --name=qiime -it
quay.io/qiime2/core:2021.2
```

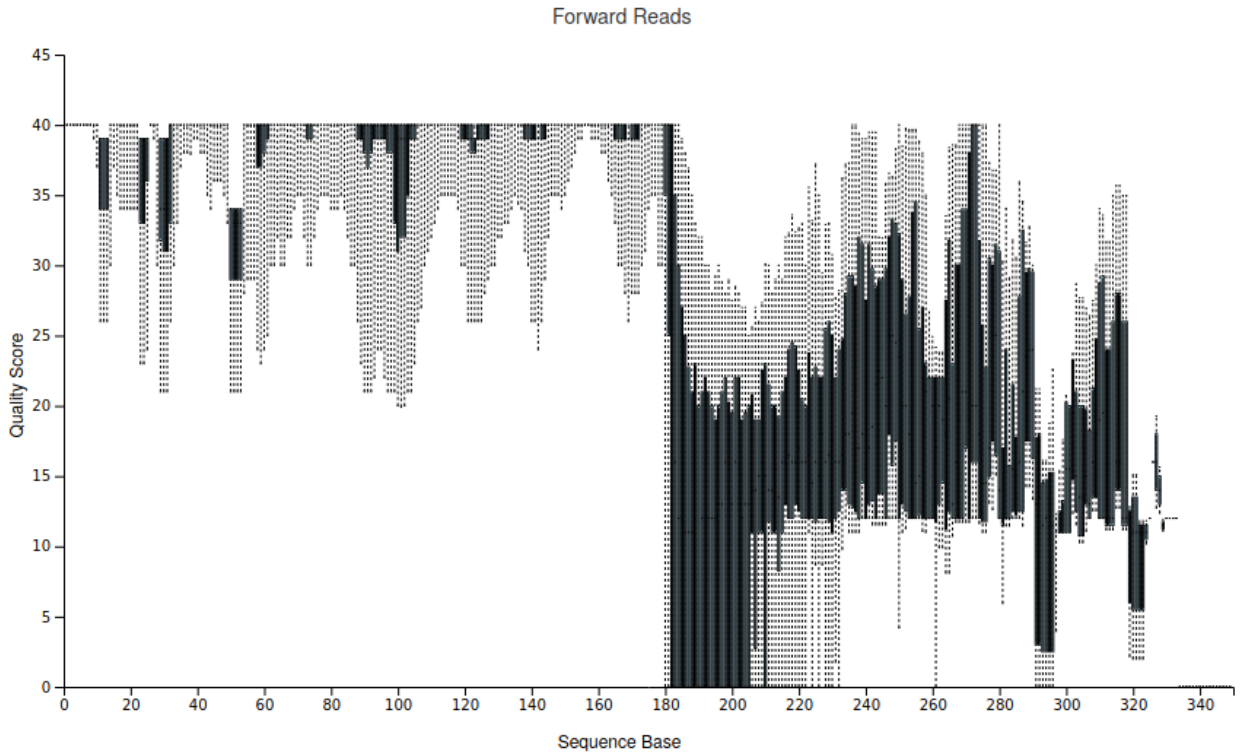
2.Importing data.

```
qiime tools import   --type 'SampleData[SequencesWithQuality]'
--input-path data/manifest.tsv   --output-path sequences.qza
--input-format SingleEndFastqManifestPhred33V2
Imported data/manifest.tsv as SingleEndFastqManifestPhred33V2 to
sequences.qza
```

```
qiime tools validate sequences.qza
Result sequences.qza appears to be valid at level=max.
```

3. Demultiplexing and QC

```
qiime demux summarize   --i-data sequences.qza   --o-visualization
sequences.qzv
Saved Visualization to: sequences.qzv
```



As we can see, the quality falls at 180 bp, so we'll truncate our sequences at 180 bases. A total length of the artificial sequences (barcode+primer) is 32, according to sample-metadata.tsv. Thus we will set further parameters:

```
--p-trim-left 32
--p-trunc-len 150
```

4. Feature table construction (and more QC)

```
qiime dada2 denoise-single   --i-demultiplexed-seqs sequences.qza
--p-trim-left 32 --p-trunc-len 150 --o-representative-sequences
rep-seqs.qza --o-table table.qza --o-denoising-stats stats.qza
Saved FeatureTable[Frequency] to: table.qza
Saved FeatureData[Sequence] to: rep-seqs.qza
Saved SampleData[DADA2Stats] to: stats.qza

qiime metadata tabulate   --m-input-file stats.qza
--o-visualization stats.qzv
Saved Visualization to: stats.qzv
```

sample-id	input	filtered	percentage of input passed filter	denoised	non-chimeric	percentage of input non-chimeric
bone	5788	5589	96.56	5377	5377	92.9
calculu	5362	5183	96.66	5068	4837	90.21

s						

5. FeatureTable and FeatureData summaries

```
qiime feature-table summarize   --i-table table.qza
--o-visualization table.qzv   --m-sample-metadata-file
data/sample-metadata.tsv
Saved Visualization to: table.qzv

qiime feature-table tabulate-seqs  --i-data rep-seqs.qza
--o-visualization rep-seqs.qzv
Saved Visualization to: rep-seqs.qzv
```

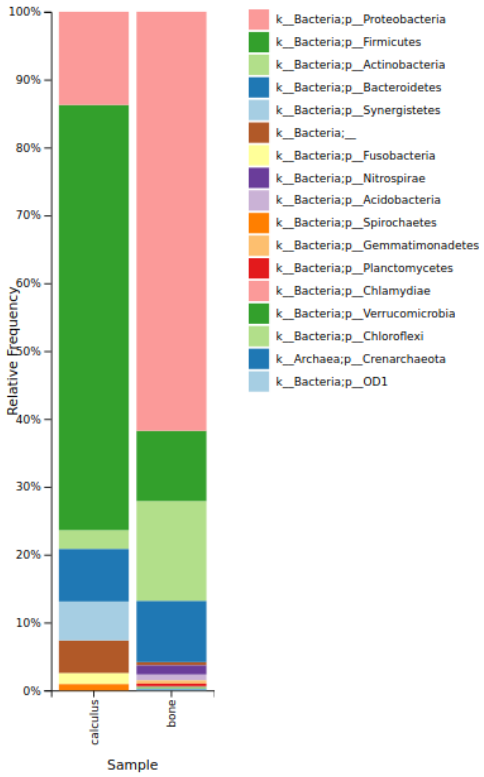
6. Taxonomic analysis

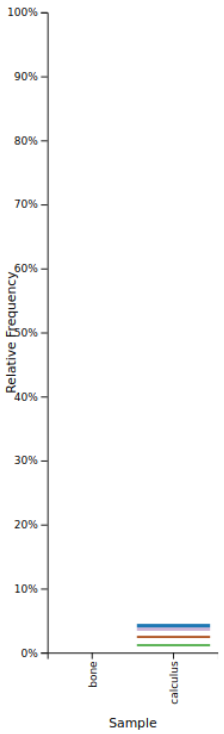
```
wget
https://data.qiime2.org/2020.2/common/gg-13-8-99-nb-classifier.qza

qiime feature-classifier classify-sklearn   --i-classifier
gg-13-8-99-nb-classifier.qza   --i-reads rep-seqs.qza
--o-classification taxonomy.qza
Saved FeatureData[Taxonomy] to: taxonomy.qza

qiime metadata tabulate --m-input-file taxonomy.qza
--o-visualization taxonomy.qzv
Saved Visualization to: taxonomy.qzv

qiime taxa barplot \
  --i-table table.qza \
  --i-taxonomy taxonomy.qza \
  --m-metadata-file data/sample-metadata.tsv \
  --o-visualization taxa-bar-plots.qzv
Saved Visualization to: taxa-bar-plots.qzv
```





Porphyromonas gingivalis 0.290% calculus
Tannerella forsythia 0.517% calculus
Treponema (denticola?) 0.352% calculus
Treponema socranskii 0.559%

Part 2. Shotgun sequencing.

1. Shotgun sequence data profiling

install hclust2

```
python3.7 -m pip install hclust2
```

install metaphlan

```
python3.7 -m pip install metaphlan
```

get profile

```
metaphlan G12_assembly.fna --input_type fasta --nproc 4 >
G12_profile.txt
```

2. Comparison with samples from HMP

Download data from the Human Microbiome Project.
SRS014459-Stool.fasta
SRS014464-Anterior_nares.fasta
SRS014470-Tongue_dorsum.fasta
SRS014472-Buccal_mucosa.fasta
SRS014476-Supragingival_plaque.fasta
SRS014494-Posterior_fornix.fasta

get profiles for data from HMP:

```
for f in *.fasta; do metaphlan $f --input_type fasta --nproc 4 >
${f%.fasta}_profile.txt; done
```

3. Visualization of the metaphlan results with a heat map

merge abundances profile:

```
merge_metaphlan_tables.py *_profile.txt > merged_profile.txt
```

create a species only abundance table, providing the abundance table

```
grep -E "s__|clade" merged_profile.txt | sed 's/^.*s__//g' | cut -f1,3-9 |
sed -e 's/clade_name/body_site/g' > merged_abundance_table_species.txt
```

remove rows containing zeros in all columns (in python):

```
import pandas as pd

merged_abundance = pd.read_csv('merged_abundance_table_species.txt', sep = '\t')
merged_abundance_notallzroes = merged_abundance.loc[(merged_abundance.iloc[:,1:]
!= 0).any(axis=1)]
merged_abundance_notallzroes.to_csv('merged_abundance_table_species_wo_0.txt',
                                    sep='\t',
                                   index=False)
```

generate the species only heatmap by running the following command

(Show the top 25 species using (--ftop 25 argument). Use Bray-Curtis as the distance measure both between samples (s) and between features (f) (microbes), sets the ratio between the width/height of cells to 0.5, uses a log scale for assigning heatmap colors, sets the sample and feature label size to 6, sets the max sample and feature label length to 100, selects the minimum value to display as 0.1, and selects an image resolution of 300):

```
hclust2.py -i merged_abundance_table_species_wo_O.txt -o
abundance_heatmap_species.png --f_dist_f braycurtis --s_dist_f braycurtis
--cell_aspect_ratio 0.5 -l --flabel_size 6 --slabel_size 6
--max_flabel_len 100 --max_slabel_len 100 --minv 0.1 --dpi 300
--image_size 10
```

4. Comparison with ancient Tannerella forsythia genome

download reference:

```
wget
https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/238/215/GCF_000238215.1_ASM23821v1/GCF_000238215.1_ASM23821v1_genomic.fna.gz
gunzip GCF_000238215.1_ASM23821v1_genomic.fna.gz
wget
https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/000/238/215/GCF_000238215.1_ASM23821v1/GCF_000238215.1_ASM23821v1_genomic.gff.gz
gunzip GCF_000238215.1_ASM23821v1_genomic.gff.gz
```

indexing reference genome:

```
bwa index GCF_000238215.1_ASM23821v1_genomic.fna
```

align contigs and sort:

```
bwa mem GCF_000238215.1_ASM23821v1_genomic.fna G12_assembly.fna |
samtools sort -o G12_assembly.bam -
```

get basic statistics:

```
samtools flagstat G12_assembly.bam
905742 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
141 + 0 supplementary
0 + 0 duplicates
16539 + 0 mapped (1.83% : N/A)
0 + 0 paired in sequencing
0 + 0 read1
0 + 0 read2
0 + 0 properly paired (N/A : N/A)
0 + 0 with itself and mate mapped
0 + 0 singletons (N/A : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)
```

index alignment file:

```
samtools index G12_assembly.bam
```

bam to bed

```
bedtools bamtobed -i G12_assembly.bam > G12_assembly.bed
```

intersection of annotation files

```
bedtools intersect -v -a GCF_000238215.1_ASM23821v1_genomic.gff -b
G12_assembly.bed > ref_G12_intersect.gff
```

```
grep -e 'CDS' ref_G12_intersect.gff | cut -f9 | awk -F ';product='
'{print $2}' | cut -d';' -f1 |  sort | uniq | less -S
```

```
grep -e 'CDS' ref_G12_intersect.gff | cut -f9 | awk -F ';locus_tag='
'{print $2}' | cut -d';' -f1 |  wc -l
```

```
191
```

One of the regions of zero coverage contains a gene of conjugative transposon protein Traj, which probably was obtained during the strain evolution.

5. Visualization with pavian

Edit metaphlan output files for pavian:

```
for file in *profile.txt; do cat $file | tail -n +5 | cut -f1,3 |
sed '1 i #SampleID\tMetaphlan2_Analysis' >
${file%.txt}_formatted.txt; done
```