# Overview of Pipeline

1. Download fastq files from https://basespace.illumina.com/home/index
2. Filter low quality reads using fastq quality trimmer from FASTX.
3. Join paired end reads using join\_paired\_ends.py from QIIME
4. Filter out the primer/barcode sequence using SeqFilters.jar from https://github.com/rdpstaff/SeqFilters
5. Remove Chimera reads using usearch from <http://www.drive5.com/usearch/>
6. Optional: random subsample reads using in-house JAVA program or QIIME
7. Rename sequence heading using in-house python program if no random subsampling
8. Make mapping file
9. Build OTU table using QIIME
10. Summarize taxonomy using QIIME
11. Alpha analysis using QIIME
12. Beta analysis using QIIME
13. NMDS analysis using R
14. Heatmap generation using R

**Procedure**

A. Download fastq files from https://basespace.illumina.com/home/index

1. Go to <https://basespace.illumina.com/home/index>

2. Login to the website. (ask Jeffrey Wagner ([jrwagner@uga.edu](mailto:jrwagner@uga.edu)) to share the project with your user email address in order to access the sequencing data).

3. Select project under Run section; click Samples to view and download paired end fastq data. (For downloading fastq data from all samples, click download in the overview page and click “Unaligned Data (fastq)”

B. Filter low quality reads using fastq quality trimmer from FASTX

1. Go to <http://hannonlab.cshl.edu/fastx_toolkit/download.html> to download FASTX. Mac-user, might need to use get clone command to download the source code from Source Code GIT Repository

2. Go to the FASTX directory with all the executable files.

3. The usage is: fastq quality trimmer –t minimum quality score –Q33 –i input.fastq -o output.fastq

**Note:** use –h to see the usage. Make sure to have the correct path for the input file if it is not in the same directory as FASTX. Make sure have –Q33 in the usage, which is the data type of the Illumina 1.8 sequencer.

C. Join paired end reads using join\_paired\_ends.py from QIIME

1. Join paired end reads using join\_paired\_ends.py and the command is: join\_paired\_ends.py –f forward\_input.fastq –r reverse\_input.fastq -o output\_file

2. Look for the fastjoin.join.fastq in the output\_file folder, it is the joined fastq file. Rename the file name to be more descriptive.

3. To prepare primer/barcode filtering, convert joined fastq file to fasta file using convert\_fastaqual\_fastq.py from QIIME or fastq\_to\_fasta from FASTX tool (Make sure have –Q33 in the command when fastq\_to\_fasta is used).

(convert\_fastaqual\_fastq.py –c fastq\_to\_fastaqual –f paired\_sfbr\_file –o new\_fastafilename)

creates new\_fastafilename.fna and .qual

D. Filter out the primer/barcode sequence

1. Download SeqFilters.jar from https://github.com/rdpstaff/SeqFilters

2. Run SeqFilters.jar with this command: java -jar /path/to/SeqFilters.jar --forward-primers CAGCMGCCGCGGTAATWC --max-forward 2 --reverse-primers CCGTCAATTCCTTTRAGGTT --max-reverse 1 --seq-file seq\_input.fasta –output\_file\_name

(RDPTools/SeqFilters/dist/SeqFilters.jar)

E. Remove Chimera reads using usearch

1. Download usearch from <http://www.drive5.com/usearch/>

2. Download reference data (either CS Cold or RDP Gold) from <http://www.drive5.com/usearch/manual/uchime_ref.html>

3. Use the following command to run the usearch to remove chimera sequence

path/usearch\_file\_name - -uchime\_ref input\_file.fasta -db reference.fasta –nonchimeras output\_file.fasta -strand plus (usearch\_file\_name will the actual file name downloaded to your computer; if you are in the directory that has the file, use ./ usearch\_file\_name at the beginning of the command)

F. Random subsample reads (optional)

1. Determine the number of subsample sequence and write that number at the command line “int count = “ in the SampleSequence.java file.

2. Go to the directory that has SampleSequence.java and SampleSequence.class

3. Type java SampleSequence input\_file.fasta output\_file.fasta

4. Optional: use count\_seqs.py in QIIME to determine if the number of reads in output\_file.fasta is the same as the number written in the SampleSequence.java

**Note:** If there is a group of samples, the number of subsample sequence will be equivalent to the number of reads of the sample with the lowest number of reads.

G. Rename sequence heading using in-house python program

1. If no random subsample is done, use the in-house python file (Heading Rename Withspace.py) to rename the sequence heading.

2.