# PCAMMS

Pipeline for Classifying and Assembling Microbes from Metagenomic Sequencing

\*\*\*THIS PIPELINE IS STILL IN DEVELOPMENT\*\*\*

# Description:

## Brief:

This pipeline is for initial analysis of metagenomic samples sequenced on the Miseq platforms. It was originally designed for detection of pathogens in sterile site infections. The pipeline consists if two scripts, the first (metagenomic\_qc.py) prepares the reads for analysis (including merging reads, removing host and performing basic QC). The second (metagenomic.py) performs classification on the reads using CLARK in light mode, then downloads the representative reference file for each species taxon ID estimated to be significant, and reference assemblies the unpaired reads against his reference.

## In-depth:

\*Metagenomic\_qc.py\*

1) Script will check the input files and directory, if the output directory does not exist it will create it. All files will be written to this directory

\*Note processes are paralleled using Nextflow. This will create a directory called 'work' in your current directory where intermediate files will be kept, this directory will be deleted at the end of the script. If there is already a directory called 'work' in your current directory this will be overwritten, so rename this directory before starting the pipeline.

2) next the script Will copy the raw files from the input directory to output directory and rename based either on the sample number and run name provided (default), or rename based on a given sample list. Option available to not rename the files.

Raw files remain unchanged.

3) Host reads will be removed by mapping against the host reference, default is human. These fastq files replace the raw files in the output directory. Host can be changed see below. Option to not map against host.

4) Paired-end reads will be merged using fastq-join. The paired-end reads will be stored in a dictory within the project folder

5) Basic Quality control will be performed using Fastqc on merged reads, then produce a combined report using Mulitqc.

6) If negatvie controls are given these will be classified using CLARK-L (optional)

7) A directory called qc\_results will be created which will contain:

\* a html report containing the multiqc results

\* a csv file containing information on the host mapping process (input reads, number of reads mapping to host, % reads mapped to host, and the output reads)

\* a csv containing read information (number of parie-end reads, number of merged reads and the % of reads which were sucessfully merged)

\* html(s) of a Krona plot of the results of classifying the negative (if performed)

\*metagenomic.py\*

1) Optional: Map against the negative control(s) and/or a local contamination library to remove any contaminating reads, mapped reads will be removed.

2) The reads will be classified using CLARK in light mode.

3) The significance of classified hits will then be assessed for each species identified in a sample. Using the average read length, size of the genome and the number of reads classified.

(av\_readlength x number of classified hits)/genome size) if this is greater than 0.001 (0.1%) then the hit is considered significant.

This balances not missing hits, but also not performing assembly on very low quality hits.

This cut off can be altered in the metagenomic.py if required (line 423)

4) For those hits considered to be potentially significant the representative genome for the taxon ID will be downloaded from RefSeq and stored in a directory PCAMMS/refs.

5) The paired-end reads will then be assembled against this reference, and the number of mapped reads and genome coverage will be calculated.

6) The taxon abundance across the whole run will be calculated, which includes how many samples each taxon was identified in, how many reads were identified and which samples have the highest number of reads. This is designed to help indicate contamination within the run.

7) a directory called 'results' will be created in the output directory and will contain:

\* runame.csv: csv file containing basic run information for each sample in the run(number of raw reads, number of reads mapping to host, pecentage of reads mappig to host, number of reads not mapping to host , number of reads after paried-end merging, % of reads that were merged , number of reads which were classified, % of reads which were classified)

\* runame\_genome\_cov: csv containing the estimated genome coverage based on calculation described above

\* runame\_referene\_mapping.csv: csv file with the mapping information for each sample and each microbe identified (organism name, organism genome size, bases in the genome covered in the reference assembly, % of genome covered, number of reads mapped, % of the total reads mapped, taxon ID of the organism, how many reads wrer classified as that taxon ID, the reference used in the assembly)

\* runname\_sample\_ref.csv: contains full path to the reference used

\* sample.html : krona plot of results of classification.

Installation:

Please note building the database requires 90GB of storage. Once build most files are removed and the total required space is 2.2GB

In the directory where you want to store the pipeline:

git clone https://github.com/Catanscombe/PCAMMS.git

$cd PCAMMS

$nohup ./setup

this will install the following prerequisites:

\*java version 8

\*fastq-join (https://github.com/brwnj/fastq-join)

\*CLARK CLAssifier based on Reduced K-mers (http://clark.cs.ucr.edu)

\*including building database

\*Krona tools (https://github.com/marbl/Krona/wiki/KronaTools)

\*Nextflow (https://www.nextflow.io)

\*Fastqc (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/)

\*Multiqc (https://multiqc.info)

\*seqtk (https://github.com/lh3/seqtk)

\*bwa (http://bio-bwa.sourceforge.net)

Usage:

uusage: metagenomic\_qc.py [-h] [-nl NEG\_LIST] [-r {yes,no}] [-id SAMPLE\_ID]

[-s SAMPLE\_LIST] [-n1 NEG\_SAMPLE\_R1]

[-n2 NEG\_SAMPLE\_R2] [-t THREADS] [-rh {yes,no}]

input\_dir output\_dir run

positional arguments:

input\_dir directory containing input files

output\_dir output directory

run run description

optional arguments:

-h, --help show this help message and exit

-nl NEG\_LIST, --neg\_list NEG\_LIST

path to csv containig information on negative control

samples, see example for format

-r {yes,no}, --rename {yes,no}

option to rename files when in results directory,

based on sample number and run name, default is yes

-id SAMPLE\_ID, --sample\_ID SAMPLE\_ID

path to csv containing sample number and sample ID,

will rename samples based on sample number and sample

ID, see example for format. If the run contains one or

more negative control, please submit seperatley un

--neg\_list command

-s SAMPLE\_LIST, --sample\_list SAMPLE\_LIST

provide a list of samples to process, default is to

process all files in input directory

-n1 NEG\_SAMPLE\_R1, --neg\_sample\_R1 NEG\_SAMPLE\_R1

path to negtaive control file R1

-n2 NEG\_SAMPLE\_R2, --neg\_sample\_R2 NEG\_SAMPLE\_R2

path to negtaive control file R2

-t THREADS, --threads THREADS

number of threads availble, default is 4

-rh {yes,no}, --remove\_host {yes,no}

option remove host reads at the begiining of analysis,

default yes. Default host is Human, if another host is

needed, place the refernce fasta in host directory.

See README for more detail

usage: metagenomic.py [-h] [-c CONTAMINATION]

[-nc NEG\_CONTROL [NEG\_CONTROL ...]]

[-tc TAXON\_CONTAMINANTS] [-t THREADS]

output\_dir run

positional arguments:

output\_dir output directory

run run description

optional arguments:

-h, --help show this help message and exit

-c CONTAMINATION, --contamination CONTAMINATION

path to fasta containing local contamination library

-nc NEG\_CONTROL [NEG\_CONTROL ...], --neg\_control NEG\_CONTROL [NEG\_CONTROL ...]

list the merged negative control fastq files from

metagenomic.py you wish to be used as contamination

libraries

-tc TAXON\_CONTAMINANTS, --taxon\_contaminants TAXON\_CONTAMINANTS

provide a list of taxon ID you wish to be ignored in

the process

-t THREADS, --threads THREADS

number of threads availble, default is 4

Basic

\*metagenomic\_qc.py\*

python Path/to/PCAMMS/metagenomic\_qc.py directory/containing/raw-illumina/fastqs output/directory runname

outputs:

directory will be made in output/directory/qc\_results

contains:

multiqc\_report.html: a webpage containing basic qc data performed on the merged reads

runname\_host\_map\_info.csv: contains details of the number and % of reads which mapped to the host genome

runname\_read\_info.csv: contains raw read number, merged read number, % reads merged

\*metagenomic.py\*

$ python Path/to/PCAMMS/metagenomic.py output/directory runname

outputs:

directory will be made in output/directory/results

contains:

multiqc\_report.html: a webpage containing basic qc data performed on the merged reads

Example:

$ python PCAMMS/metagenomic\_qc.py PCAMMS/examples/ example\_output example\_run

then

$ python PCAMMS/metagenomic.py example\_output example\_run

Download the directories example\_output/qc\_results and example\_output/results to view output files

With Negative control

$ python Path/to/PCAMMS/metagenomic\_qc.py directory/containing/raw-illumina/fastqs output/directory runname -n1 Path/to/neg\_control\_R1.fastq -n2 Path/to/neg\_control\_R2.fastq