# 1. Create a conda environment

conda create -n Ametagenomics python=3.6

source activate Ametagenomics

conda install pandas

conda install numpy

conda install -c conda-forge mkl\_random

conda install -c anaconda cython y

conda install -c bioconda bbmap

# 2. Get ars-rqc from the repository and install the editable version. This allows to auto-update the changes made in the repository – so that the used program will be the most recent once.

git clone https://github.com/USDA-ARS-GBRU/ARS-RQC.git

cd ARS-RQC

pip install --editable .

dd

#3. Soft link the input files from the collaborator’s raw-data directory

mkdir data

cd data

ln -vs /project/\*\*/\*\*/\*.fastq.gz .

# 4. Run ***sbatch 01\_arsqc\_interleave.sh*** to create interleaved fastq

#!/bin/bash

#SBATCH --job-name=rqc\_interleave\_Ametagenomics

#SBATCH --output=rqc\_Ametagenomics\_%A\_%a.out

#SBATCH --error=rqc\_Ametagenomics\_%A\_%a.err

#SBATCH --time=01:00:00

#SBATCH --array=1-156

#SBATCH -p short

#SBATCH -N 1

#SBATCH -n 10

module load miniconda

source activate Ametagenomics

module load pigz

module load bbtools/gcc/64/36.86

RUN=${SLURM\_ARRAY\_TASK\_ID}

echo "My Slurm RUN\_ID: '${RUN}'"

echo "My TMPDIR IS: " $TMPDIR

infile1=$(ls data/\*\_R1.fastq.gz | sed -n ${RUN}p)

echo "$infile1"

infile2=$(echo $infile1 | sed 's/R1/R2/')

echo "$infile2"

outbase=$(basename $infile1 \_R1.fastq.gz)

outfile=data/${outbase}\_interleave.fastq.gz

echo "$outfile"

reformat.sh in=$infile1 in2=$infile2 out=$outfile

# 5. Run *02\_arsqc\_filer.sh* to run qc on fastq

#!/bin/bash

#SBATCH --job-name=rqc\_filter\_Ametagenomics

#SBATCH --output=rqc\_Ametagenomics\_%A\_%a.out

#SBATCH --error=rqc\_Ametagenomics\_%A\_%a.err

#SBATCH --time=01:00:00

#SBATCH --array=1-156

#SBATCH -p short

#SBATCH -N 1

#SBATCH -n 10

module load miniconda

source activate Ametagenomics

module load pigz

RUN=${SLURM\_ARRAY\_TASK\_ID}

echo "My Slurm RUN\_ID: '${RUN}'"

echo "My TMPDIR IS: " $TMPDIR

rootdir="/project/"

echo "$rootdir"

itl\_file=$(ls data/\*\_interleave.fastq.gz | sed -n ${RUN}p)

echo "$itl\_file"

time rqcfilter.py --fastq $itl\_file --output "$rootdir"rqc\_data -p

rqcfilter use runs BBtools RQC pipeline.

Adapter-trimming. With BBDuk.

Check for vertebrate read contamination and remove those. If we want to do host genome contamination removal, need to manually. Not included in default bbduk.

# 6. Some sanity check stuff

**# 6.1 ## sort - d gives non-unique**

find rqc\_data/ -type f | cut -d'-' -f1 | sort -d

find logs/ -type f -exec basename "{}" \; | cut -d'-' -f1 | sort -u

find logs/ -type f | cut -d'-' -f1 | sort -u

find logs/ -type f | sort -u

find Ametagenomics/\*\* -type f -exec basename "{}" \; | sort -u

find Ametagenomics/\*\* -type f -exec basename "{}" \; | sort -d

ls rqc\_Ametagenomics\_379050\* | grep -o 'rqc\_Ametagenomics\_379050.\*' | cut -f4- -d\_ | wc -l

**# find the non-unique file and show their path**

find Ametagenomics/\*\* -type f \; | sort -d

**# for unique one run**

find Ametagenomics/\*\* -type f -exec basename "{}" \; | sort -u

**# Fast way of finding lines in one file that are not in another?**

grep -v -f file2 file1

**# Make sure the files name for resulted fastqs are unique and match the expected number of fastqs, which should equal to the number of samples X technical replicates per sample.**

find /project/gbru\_fy18\_apple\_microbiome/BARS\_Project\_Data\_WGS\_Golden/SDroby21869-71827756/\*\*/\*\*/\*.fastq.gz -type f -exec basename "{}" \; | cut -d'-' -f1 | sort -u

find /project/gbru\_fy18\_apple\_microbiome/BARS\_Project\_Data\_WGS\_Golden/SDroby21869-71827756/\*\*/\*\*/\*.fastq.gz -type f -exec basename "{}" \; | sort | uniq -d

# 7. Run 03\_parse\_json.py – crate a summary data-frame using json files obtained from qc. This file is then analyzed in R to create qc-report using R markdown.

import json

import pandas as pd

import os

wdir = os.getcwd()

list = []

for lfile in os.listdir("rqc\_data"):

file = os.path.join(wdir, "rqc\_data", lfile)

if file.endswith('.json'):

bfile = file

with open(file) as js:

js\_dict = json.load(js)

tr = js\_dict['scaffoldStats1.txt']['desc']['TotalReads']

tb = js\_dict['scaffoldStats1.txt']['desc']['TotalBases']

contam = js\_dict['scaffoldStats1.txt']['desc']['ReadsMatched']

pctcontam = js\_dict['scaffoldStats1.txt']['desc']['PctReadsMatched']

list.append((os.path.basename(bfile), tr, tb, contam, pctcontam))

cols = ['SampleID', 'TotalReads', 'TotalBases', 'Contaminants', "Percent\_Contaminants"]

result = pd.DataFrame(list, columns=cols)

result.to\_csv("rqc\_data/parse\_json.csv")

# 8. Create a list of files, open in text editor, then find carry-over and replace with comma, then delete comma at the EOF. This information will be pass as input files for genome assembly in megahit.

find /project/rqc\_data/reads/\*\* -type f > file\_list\_withpath.txt

# 9. Run genome assembly 04\_megahit.sh

#!/bin/bash

#SBATCH --job-name=megahit\_assembly

#SBATCH --output=megahit\_assembly\_%A\_%a.out

#SBATCH --error=megahit\_assembly\_%A\_%a.err

#SBATCH --time=96:00:00

#SBATCH -p mem

#SBATCH -N 1

#SBATCH -n 120

# export PATH=$PATH:/home/ravin.poudel/bbmap

export PATH=$PATH:/home/ravin.poudel/megahit/1.1.1

module load miniconda

#source activate Ametagenomics

module load pigz

module load megahit

echo "My TMPDIR IS: " $TMPDIR

time megahit --k-min 27 --k-max 127 --k-step 10 -m 0.98 -t 120 --out-dir megahit\_output --kmin-1pass --min-contig-len 300 --tmp-dir $TMPDIR --12 /project/rqc\_data/reads/A1\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A1\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A2\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/A3\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C1W\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C2W\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/C3W\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M1\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M2\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/M3\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O1S\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O2S\_run4\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run1\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run2\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run2\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run2\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run2\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run3\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run3\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run3\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run3\_lane4\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run4\_lane1\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run4\_lane2\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run4\_lane3\_interleave.rqc.fq.gz,/project/rqc\_data/reads/O3S\_run4\_lane4\_interleave.rqc.fq.gz

# 11. Run quast analysis

conda install -c bioconda quast

# for running anavio

mkdir mapping

mkdir anvio

# index assembled contig

sbatch 05\_index\_contig.sh

,

# mapping interleved fastq with the index configs.

sbartch 06\_mapping.sh

## after this point may be I need to combine bam files by sample then go to anvio pipeline

sbatch 07\_merge\_index\_bam.sh

######Re-formatting your input FASTA

anvi-script-reformat-fasta contigs.fa -o contigs-fixed.fa -l 0 --simplify-names --report-file contig.report

# mash distance

mash.dist(A1\_finalBamfile.merge.msh, A2\_finalBamfile.merge.msh, A3\_finalBamfile.merge.msh, C1W\_finalBamfile.merge.msh, C2W\_finalBamfile.merge.msh, C3W\_finalBamfile.merge.msh, M1\_finalBamfile.merge.msh, M2\_finalBamfile.merge.msh, M3\_finalBamfile.merge.msh, O1S\_finalBamfile.merge.msh, O2S\_finalBamfile.merge.msh, O3S\_finalBamfile.merge.msh) -p 10 > tab.distance

# create a kaiju db

# create a link from rice microbiome or search for db in basecamp

cd anvio

mkdir kaijudb

ln -vs /project/gbru\_fy18\_rice\_methane/anvio/kaijudb/\*.\* kaijudb/

time anvi-profile \

-i $infile \

-c /project/gbru\_fy18\_apple\_microbiome/rp\_apple\_microbiome/anvio/contigs.db \

--output-dir /project/gbru\_fy18\_apple\_microbiome/rp\_apple\_microbiome/anvio/profiles/"$name" \

--sample-name $name \

--skip-SNV-profiling \

-T 2 \

--write-buffer-size 200

KEGG reformating

sed -e 's/\>/\>genecall\_/g' aa\_gene\_calls.fa > aa\_gene\_calls.fixed.fa

##########after merging anvio profiles

# interactive visulaization of bins

anvi-script-get-collection-info -p SAMPLES-MERGED/PROFILE.db -c contigs.db --list-collections

anvi-interactive -p SAMPLES-MERGED/PROFILE.db -c contigs.db -C CONCOCT

# create a summary files

anvi-summarize -p SAMPLES-MERGED/PROFILE.db -c contigs.db -o SAMPLES-SUMMARY -C CONCOCT

anvi-rename-bins -p SAMPLES-MERGED-5000-Enforceluster/PROFILE.db -c contigs.db --collection-to-read CONCOCT --collection-to-write MAGs --call-MAGs --prefix IGD --size-for-MAG 100 --report-file REPORT

# anvi-export-collection -p SAMPLES-MERGED-5000-Enforceluster/PROFILE.db -C MAGs

# anvi-import-collection collection-MAGs.txt -p SAMPLES-MERGED-5000-Enforceluster/PROFILE.db -c contigs.db -C MAGs

anvi-summarize -p SAMPLES-MERGED-5000-Enforceluster/PROFILE.db -c contigs.db -C MAGs -o SUMMARY\_MAGS --init-gene-coverages --report-aa-seqs-for-gene-calls

###############

selection of fiter bins

anvi-export-collection -p SAMPLES-MERGED-5000-Enforceluster/PROFILE.db -C CONCOCT

# filter the collection file in R.

anvi-import-collection concoct\_selected.txt -p SAMPLES-MERGED-5000-Enforceluster/PROFILE.db -c contigs.db -C CONCOCT

anvi-interactive -p SAMPLES-MERGED-5000-Enforceluster/PROFILE.db -c contigs.db -C CONCOCT

anvi-summarize -p SAMPLES-MERGED-5000-Enforceluster/PROFILE.db -c contigs.db -o SAMPLES-SUMMARY-test-refine -C CONCOCT --init-gene-coverages --report-aa-seqs-for-gene-calls

############# preparing slectted fasta from bins for seraching against KAIJU output

1) cat filter\_bins\_metabat\_checkM/\*.fa > filter\_bin\_concat.fa

2) copy only the header of each seq then remove ">" from the header

grep -e ">" filter\_bin\_concat.fa | awk 'sub(/^>/, "")' > filter\_bin\_concat\_header.txt

#############checkm coverage

checkm coverage -t 40 -x fa final.contigs-fixed.fa.metabat-bins5000/ test.coverage.tsv mapping/\*.bam

## checkm profile coverage.tsv --tab\_table > checkm\_profile.tsv