Shotgun Metagenomics Workflow

Jacob Westaway

Last updated on 2021-03-11

Contents

About.

This workflow is based largely around the MetaWRAP wrapper, however it steps outside of this several times to use updated software. Some chunks can be run directly in the terminal, but others will require submission to the HPC as they are resource intensive. HPC jobs will be denoted with **(qsub)**.

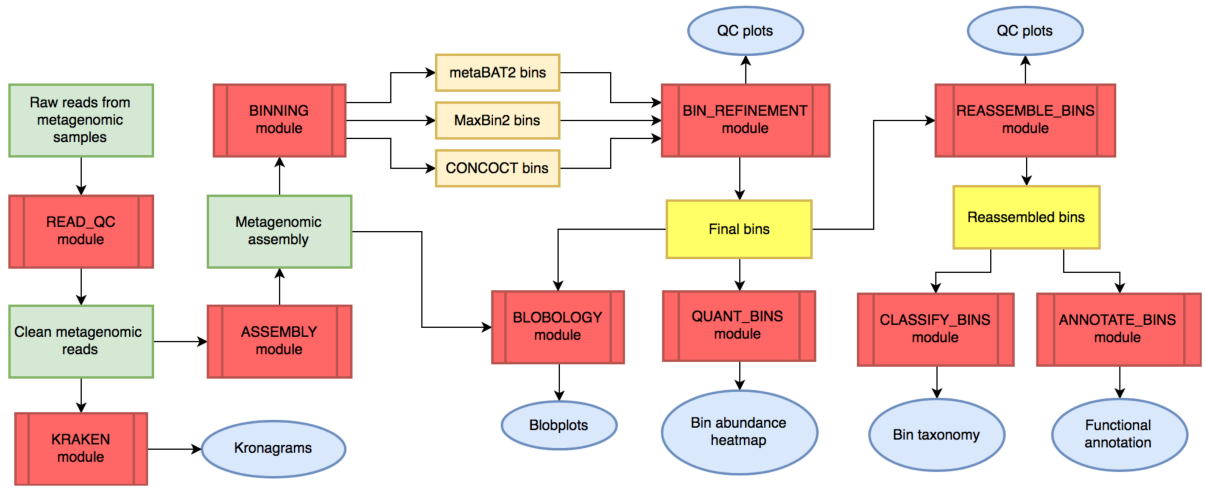

Figure 1: MetaWRAP Workflow.

Set up.

Before jumping into the MetaWRAP modules, we first need to install *metaWRAP* and several databases.

Install MetaWRAP and create environment.

```
# install conda
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
.bash Miniconda3-latest-Linux-x86_64.sh

# activate bash/conda
bash
conda

# create and activate metawrap (and python 2.7)
conda create -y -n metawrap-env python=2.7
conda activate metawrap-env

# add dependencies
conda config --add channels defaults
conda config --add channels conda-forge
conda config --add channels bioconda
conda config --add channels ursky
```

```
# to activate metawrap-env
conda activate metawrap-env

# to deactivate
conda deactivate

# add metawrap to environment/pathway
export PATH="/home/jc490421/minconda3/metawrap-env/bin
```

Creat a directory to store databases.

```
mkdir Databases
```

Install and index the human genome.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N Index_human_Genome
#PBS -l select=1:ncpus=8:mem=100gb
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash and conda"
source ~/.bashrc
source activate conda

echo "make new directories"
mkdir Databases
mkdir Databases/BMTAGGER_INDEX

echo "change directory"
cd ~/BMTAGGER_INDEX

echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "run commands to index human genome"
bmtool -d hg38.fa -o ~/BMTAGGER_INDEX/hg38.bitmask
srprism mkindex -i ~/BMTAGGER_INDEX/hg38.fa -o hg38.srprism
```

Download CheckM database.

```
echo "change directory"
cd ~/Databases

echo "create CheckM directory"
mkdir MY_CHECKM_FOLDER

echo "change to newly created directory"
cd MY_CHECKM_FOLDER

echo "download database"
wget https://data.ace.uq.edu.au/public/CheckM_databases/checkm_data_2015_01_16.tar.gz
tar -xvf *.tar.gz
rm *.gz
```

Download NCBI_nt BLAST database.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N NCBI_nt_DB
#PBS -l select=1:ncpus=10:mem=40gb
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
source ~/.bashrc

echo "change directory"
cd ~/Databases

echo "create directory"
mkdir NCBI_nt

echo "change to newly created directory"
cd NCBI_nt

echo "download database"
wget "ftp://ftp.ncbi.nlm.nih.gov/blast/db/v4/nt_v4.*.tar.gz"
for a in nt_*.tar.gz; do tar xzf $a; done
```

Download NCBI taxonomy.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N NCBI_tax
#PBS -l select=1:ncpus=10:mem=40gb
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
source ~/.bashrc

# NCBI taxonomy
echo "change directory"
cd ~/Databases

echo "create directory"
mkdir -p NCBI_tax

echo "change to newly created directory"
cd NCBI_tax

echo "download database"
wget ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/taxdump.tar.gz

echo "unzip"
tar -xvf taxdump.tar.gz
```

Install KRAKEN2 and build database.

MetaWRAP uses *KRAKEN*, which has several bugs that can be solved using *KRAKEN2* so here we step outside of *MetaWRAP*, install *KRAKEN2* and use this to build our database.

```
conda install -c bioconda kraken2
```

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N KRAKEN2_DB
#PBS -l select=1:ncpus=24:mem=250gb
```

```
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash and conda"
source ~/.bashrc
source activate conda

echo "change directory"
cd ~/Databases

echo "build standard database"
kraken2-build --standard --threads 24 --db MY_KRAKEN_DATABASE

echo "clean database"
kraken2-build --db MY_KRAKEN_DATABASE --clean
```

Install Krona for KRAKEN2 visualisation.

```
conda install -c bioconda krona # install krona
rm -rf ~/miniconda3/opt/krona/taxonomy # delete symbmolic link
mkdir -p ~/Databases/krona/taxonomy # create new directory for taxonomy to be stored
ln -s ~/Databases/krona/taxonomy ~/miniconda3/opt/krona/taxonomy # new symbolic link
ktUpdateTaxonomy.sh # build krona database
```

Update config-metawrap file.

1. Download config-metawrap file from HPC location (location is ~/metawrap-env/bin/).
2. Change paths for *KRAKEN_DB*, *BMTAGGER_BD*, *BLASTDB* and *TAXDUMP*.
3. Move new copy back to HPC location and change permissions (`chmod 777 path`).

Or use the *nano* text editor (`nano path/to/config-metawrap`).

For more information you can click this **link**.

The contents of the config-metawrap file should look like this:

```
# Paths to metaWRAP scripts (dont have to modify)
mw_path=$(which metawrap)
bin_path=${mw_path%/*}
SOFT=${bin_path}/metawrap-scripts
PIPES=${bin_path}/metawrap-modules

# CONFIGURABLE PATHS FOR DATABASES...
# path to kraken standard database
KRAKEN_DB=/home/jc490421/Databases/MY_KRAKEN_DATABASE
```

```
# path to indexed human...
BMTAGGER_DB=/home/jc490421/Databases/BMTAGGER_INDEX

# paths to BLAST databases
BLASTDB=/home/jc490421/Databases/NCBI_nt
TAXDUMP=/home/jc490421/Databases/NCBI_tax
```

Convert samples into correct format for metaWRAP.

Samples can be processed across multiple lanes (as mine were) and may need to be concatenated. Furthermore, *MetaWRAP* does not like **R** in its file name, so its worth removing this from the output names.

```
# concatenate Files

# create new directories for concatinated files (specifcy directory path)
mkdir ~/Microba_metagenomics_analysis/JCU_raw_data_cat/BBF7405_PCG8_LCG6_R123
mkdir ~/Microba_metagenomics_analysis/JCU_raw_data_cat/BBF7406_PCG8_LCG6_R123
mkdir ~/Microba_metagenomics_analysis/JCU_raw_data_cat/BBF7407_PCG8_LCG6_R123
mkdir ~/Microba_metagenomics_analysis/JCU_raw_data_cat/BBF7408_PCG8_LCG6_R123
mkdir ~/Microba_metagenomics_analysis/JCU_raw_data_cat/BBF7409_PCG8_LCG6_R123
mkdir ~/Microba_metagenomics_analysis/JCU_raw_data_cat/BBF7410_PCG8_LCG6_R123

# move files to new directories
cd  ~/Microba_metagenomics_analysis/JCU_raw_data_cat
# 405
mv BBF7405_PCG8_LCG6_R123_EXP_L001/*fastq.gz BBF7405_PCG8_LCG6_R123
mv BBF7405_PCG8_LCG6_R123_EXP_L002/*fastq.gz BBF7405_PCG8_LCG6_R123
# 406
mv BBF7406_PCG8_LCG6_R123_EXP_L001/*fastq.gz BBF7406_PCG8_LCG6_R123
mv BBF7406_PCG8_LCG6_R123_EXP_L002/*fastq.gz BBF7406_PCG8_LCG6_R123
# 407
mv BBF7407_PCG8_LCG6_R123_EXP_L001/*fastq.gz BBF7407_PCG8_LCG6_R123
mv BBF7407_PCG8_LCG6_R123_EXP_L002/*fastq.gz BBF7407_PCG8_LCG6_R123
# 408
mv BBF7408_PCG8_LCG6_R123_EXP_L001/*fastq.gz BBF7408_PCG8_LCG6_R123
mv BBF7408_PCG8_LCG6_R123_EXP_L002/*fastq.gz BBF7408_PCG8_LCG6_R123
# 409
mv BBF7409_PCG8_LCG6_R123_EXP_L001/*fastq.gz BBF7409_PCG8_LCG6_R123
mv BBF7409_PCG8_LCG6_R123_EXP_L002/*fastq.gz BBF7409_PCG8_LCG6_R123
# 410
mv BBF7410_PCG8_LCG6_R123_EXP_L001/*fastq.gz BBF7410_PCG8_LCG6_R123
mv BBF7410_PCG8_LCG6_R123_EXP_L002/*fastq.gz BBF7410_PCG8_LCG6_R123

# organise directory. Create new directories in JCU_raw_data_cat for lanes combined,
# lanes seperate and all the concatenated fastq files
mkdir lanes_combined
mkdir lanes_seperate
mkdir fastq_combined

mv *R123 lanes_combined
mv BBF7* lanes_seperate
```

```
# concatenate L00 files and save in new directory together
# 405
cat lanes_combined/BBF7405_PCG8_LCG6_R123/*R1_001.fastq.gz > fastq_combined/BBF7405_1.fastq.gz
cat lanes_combined/BBF7405_PCG8_LCG6_R123/*R2_001.fastq.gz > fastq_combined/BBF7405_2.fastq.gz
# 406
cat lanes_combined/BBF7406_PCG8_LCG6_R123/*R1_001.fastq.gz > fastq_combined/BBF7406_1.fastq.gz
cat lanes_combined/BBF7406_PCG8_LCG6_R123/*R2_001.fastq.gz > fastq_combined/BBF7406_2.fastq.gz
# 407
cat lanes_combined/BBF7407_PCG8_LCG6_R123/*R1_001.fastq.gz > fastq_combined/BBF7407_1.fastq.gz
cat lanes_combined/BBF7407_PCG8_LCG6_R123/*R2_001.fastq.gz > fastq_combined/BBF7407_2.fastq.gz
# 408
cat lanes_combined/BBF7408_PCG8_LCG6_R123/*R1_001.fastq.gz > fastq_combined/BBF7408_1.fastq.gz
cat lanes_combined/BBF7408_PCG8_LCG6_R123/*R2_001.fastq.gz > fastq_combined/BBF7408_2.fastq.gz
# 409
cat lanes_combined/BBF7409_PCG8_LCG6_R123/*R1_001.fastq.gz > fastq_combined/BBF7409_1.fastq.gz
cat lanes_combined/BBF7409_PCG8_LCG6_R123/*R2_001.fastq.gz > fastq_combined/BBF7409_2.fastq.gz
# 410
cat lanes_combined/BBF7410_PCG8_LCG6_R123/*R1_001.fastq.gz > fastq_combined/BBF7410_1.fastq.gz
cat lanes_combined/BBF7410_PCG8_LCG6_R123/*R2_001.fastq.gz > fastq_combined/BBF7410_2.fastq.gz
```

If you have many samples it would be better to write loops. For example, a loop to remove the **R**'s from the file names (if not done in the previous step).

```
for file in RAW_READS/*_R1.fastq; do
    newfile=$(echo ${file} | sed 's|_R1|_1|')
    ln -s ${file} ${newfile}
    ln -s ${file/_R1/_R2} ${newfile/_1/_2}
done
```

Step 1: QC.

Use *metaWRAP-Read_qc* to trim your reads and remove human contamination. Again, if you have many samples it is worth creating a loop. See the MetaWRAP tutorial for help on this.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N READ_QC
#PBS -l select=1:ncpus=10:mem=40gb
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
source ~/.bashrc
```

```
echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

echo "unzip the fastq files"
gunzip *.gz

echo "create directory for fastq files"
mkdir RAW_READS

echo "save fastq files in new directory"
mv *fastq RAW_READS

echo "Create output directory READ_QC"
mkdir READ_QC

echo "Run metaWRAP-Read_qc to trim reads and remove human contamination"
metawrap read_qc -1 RAW_READS/BBF7405_1.fastq \
  -2 RAW_READS/BBF7405_2.fastq -t 24 -o READ_QC/BBF7405

metawrap read_qc -1 RAW_READS/BBF7406_1.fastq \
  -2 RAW_READS/BBF7406_2.fastq -t 24 -o READ_QC/BBF7406

metawrap read_qc -1 RAW_READS/BBF7407_1.fastq \
  -2 RAW_READS/BBF7407_2.fastq -t 24 -o READ_QC/BBF7407

metawrap read_qc -1 RAW_READS/BBF7408_1.fastq \
  -2 RAW_READS/BBF7408_2.fastq -t 24 -o READ_QC/BBF7408

metawrap read_qc -1 RAW_READS/BBF7409_1.fastq \
  -2 RAW_READS/BBF7409_2.fastq -t 24 -o READ_QC/BBF7409

metawrap read_qc -1 RAW_READS/BBF7410_1.fastq \
  -2 RAW_READS/BBF7410_2.fastq -t 24 -o READ_QC/BBF7410
```

Create a new directory and move *clean reads* here.

```
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

mkdir CLEAN_READS

for i in READ_QC/*; do
    b=${i#*/}
    mv ${i}/final_pure_reads_1.fastq CLEAN_READS/${b}_1.fastq
    mv ${i}/final_pure_reads_2.fastq CLEAN_READS/${b}_2.fastq
done
```

Step 2: Assembly.

MetaWRAP provides two tools for assembly; *megaHIT* and *metaSPAdes*. For comparisons read the paper *Assembling metagenomes, one community at a time*. If considering *metaSPAdes*, it should be noted that it is far more resource intensive than *megaHIT* (for the same data, 20 cpu's and 250gb mem). Furthermore, depending on your goal you may want to assemble the samples seperately. If you want to coassemble and analyze the whole community across samples you need to first concatenate the files.

NB. *MetaSPAde* was run after *MegaHIT*, thus the concatination step is only in the first script.

MegaHIT.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N Assembly
#PBS -l select=1:ncpus=10:mem=40gb
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
source ~/.bashrc

echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

echo "concatenate clean fastq files"
cat CLEAN_READS/BBF*_1.fastq > CLEAN_READS/ALL_READS_1.fastq
cat CLEAN_READS/BBF*_2.fastq > CLEAN_READS/ALL_READS_2.fastq

echo "assemble reads with MegaHit"
metawrap assembly -1 CLEAN_READS/ALL_READS_1.fastq \
  -2 CLEAN_READS/ALL_READS_2.fastq -o ASSEMBLY
```

MetaSPAde.

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
```

```
#PBS -N MetaSPAdes_ASSEMBLY
#PBS -l select=1:ncpus=20:mem=250gb
#PBS -l walltime=48:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "---------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "---------------------------------------"

echo "activate bash"
source ~/.bashrc

echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

echo "assemble reads with MetaSPAde"
metawrap assembly -1 CLEAN_READS/ALL_READS_1.fastq \
  -2 CLEAN_READS/ALL_READS_2.fastq -t 20 --metaspades -o ASSEMBLY
```

QUAST outputs for both assemblies: **MegaHIT** and **MetaSPAde**.

Step 3: KRAKEN(2).

As previously mentioned *KRAKEN* in the *metaWRAP* wrapper has several bugs that can be solved using *KRAKEN2*. The code below has both the assembly and clean fastq data, but KRAKEN2 can be run on either on their own. What you choose depends on your question/objective. The script below also runs on both assemblies.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N KRAKEN2_taxonomy
#PBS -l select=1:ncpus=24:mem=200gb
#PBS -l walltime=48:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "---------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "---------------------------------------"

echo "activate bash and conda"
source ~/.bashrc

echo "cd to Microba_metagenomics_analysis"
```

```
cd ~/Microba_metagenomics_analysis

echo "run KRAKEN on fastq and assembly data"
kraken2 --db ~/Databases/MY_KRAKEN_DATABASE \
  --output KRAKEN2_megahit_OUTPUT \
  --report KRAKEN2_megahit_REPORT \
  --threads 24 \
  JCU_raw_data_cat/fastq_combined/CLEAN_READS/BBF74*fastq \
  JCU_raw_data_cat/fastq_combined/ASSEMBLY_MegaHIT/final_assembly.fasta

echo "run KRAKEN on fastq and assembly data"
kraken2 --db ~/Databases/MY_KRAKEN_DATABASE \
  --output KRAKEN2_metaspade_OUTPUT \
  --report KRAKEN2_metaspade_REPORT \
  --threads 24 \
  JCU_raw_data_cat/fastq_combined/CLEAN_READS/BBF74*fastq \
  JCU_raw_data_cat/fastq_combined/ASSEMBLY_MetaSPAde/final_assembly.fasta
```

Use *krona* to summarise and visualise taxonomy statistics.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N Krona
#PBS -l select=1:ncpus=10:mem=40gb
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash and conda"
source ~/.bashrc

echo "cd to Microba_metagenomics_analysis"
cd ~/Microba_metagenomics_analysis

echo "build 2 column file with read_id<tab>tax_id for input into ktImportTaxonomy"
cat KRAKEN2_megahit_OUTPUT | cut -f 2,3 > KRAKEN2_megahit_OUTPUT.krona

echo "run Krona on new file"
ktImportTaxonomy KRAKEN2_megahit_OUTPUT.krona
```

Step 4: Binning.

metaWRAP supports running three different tools for binning; CONCOCT, MaxBin, and metaBAT. Below I have run all three simultaneously. The downstream bin refinement process can take 3 different bin sets.

This is resource intensive, but with only 6 samples it is manageable. Furthermore, I am running the binning module on two different assemblies(*MegaHIT* and *MetaSPAde*), hence the two seperate commands.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N NCBI_nt_DB
#PBS -l select=1:ncpus=24:mem=200gb
#PBS -l walltime=48:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "---------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "---------------------------------------"

echo "activate bash"
source ~/.bashrc

echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined/

echo "run binning module on megahit assembly"
metawrap binning -o INITIAL_BINNING_megahit -t 24 \
  -a ASSEMBLY/megahit_assembly/final_assembly.fasta \
  --metabat2 --maxbin2 --concoct \
  CLEAN_READS/BBF*fastq

echo "run binning module on metaspade assembly"
metawrap binning -o INITIAL_BINNING_metaspade -t 24 \
  -a ASSEMBLY/metaspade_assembly/final_assembly.fasta \
  --metabat2 --maxbin2 --concoct \
  CLEAN_READS/BBF*fastq
```

Step 5: Bin Refinement.

As we have produced bins with several tools, we can now consolidate them into a single (more accurate) bin set. Minimum completion and maximum contamination can be set with -c and -x respectively. Otherwise, the default values are 70% and 10%. Here we run the refinement module with two different parameter settings on the two assemblies.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
```

```
#PBS -N BIN_REFINEMENT
#PBS -l select=1:ncpus=24:mem=150gb
#PBS -l walltime=48:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
source ~/.bashrc

echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

echo "run bin refinement on megahit assembly with 70_10"
metawrap bin_refinement -o BIN_REFINEMENT_megahit -t 24 \
  -A INITIAL_BINNING_megahit/metabat2_bins/ \
  -B INITIAL_BINNING_megahit/maxbin2_bins/ \
  -C INITIAL_BINNING_megahit/concoct_bins/

echo "run bin refinement on metaspade assembly with 70_10"
metawrap bin_refinement -o BIN_REFINEMENT_metaspade -t 24 \
  -A INITIAL_BINNING_metaspade/metabat2_bins/ \
  -B INITIAL_BINNING_metaspade/maxbin2_bins/ \
  -C INITIAL_BINNING_metaspade/concoct_bins/

echo "run bin refinement on megahit assembly with 90_5"
metawrap bin_refinement -o BIN_REFINEMENT_megahit_905 -t 24 \
  -A INITIAL_BINNING_megahit/metabat2_bins/ \
  -B INITIAL_BINNING_megahit/maxbin2_bins/ \
  -C INITIAL_BINNING_megahit/concoct_bins/ \
  -c 90 -x 5

echo "run bin refinement on metaspade assembly with 90_5"
metawrap bin_refinement -o BIN_REFINEMENT_metaspade_905 -t 24 \
  -A INITIAL_BINNING_metaspade/metabat2_bins/ \
  -B INITIAL_BINNING_metaspade/maxbin2_bins/ \
  -C INITIAL_BINNING_metaspade/concoct_bins/ \
  -c 90 -x 5
```

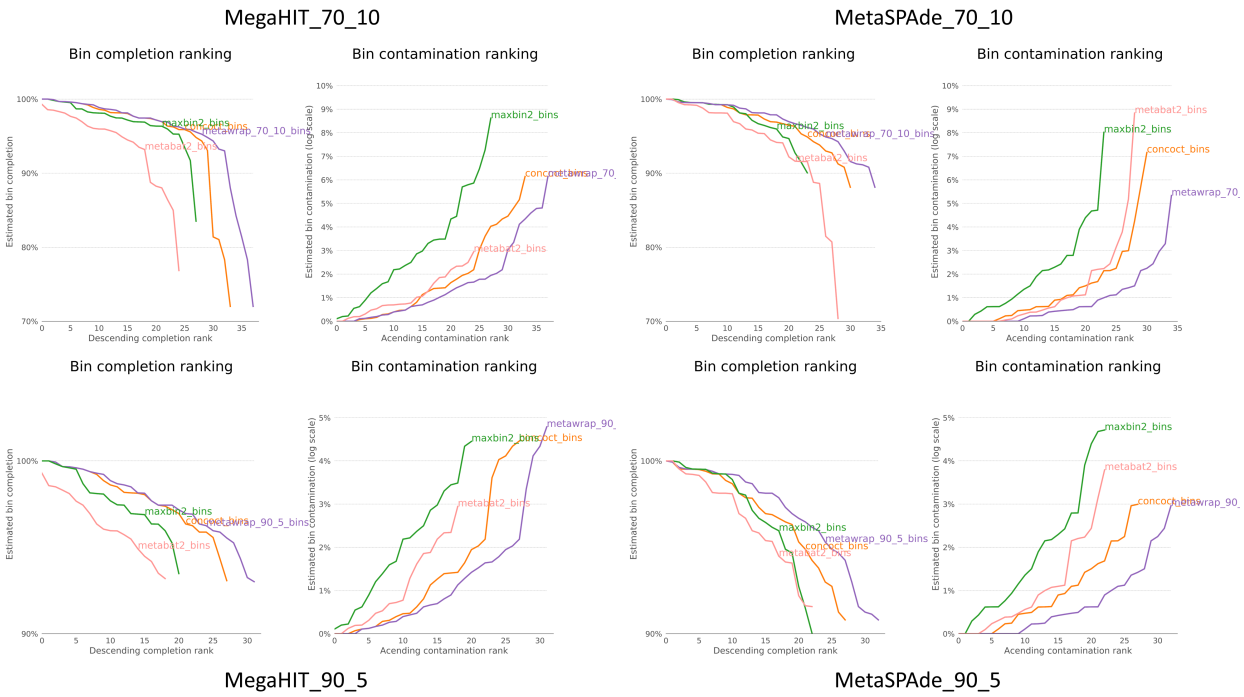

Figure 2: Bining tool comparison.

Evaluate the number of good bins.

Code based on default >70 completion and <10 contamination.The Megahit assembly in combination with the metaWRAP hybrid assembly produced the most bins total (38/36), and based on defined parameters (38/35).

```
cat BIN_REFINEMENT_metaspade/metawrap_70_10_bins.stats | awk '$2>70 && $3<10' | wc -l
cat BIN_REFINEMENT_megahit/metawrap_70_10_bins.stats | awk '$2>70 && $3<10' | wc -l
```

Step 6: Visulaisation of community and extract bins using Blobology module.

Step 6 has been run on both assemblies (*MegaHIT* and *MetaSPAde*) and both bin refinement paramter combinations (*70_10* and *90_5*).This step turns the assembly into a GC vs Abundance plane, with annotations (taxonomy) and bin information, providing information on both microbial communities and binning success.

NB. In order to annotate the blobplot with bins with the –bins flag you MUST NOT USE REASSEMBLED BINS (from **Step 8**), used refined.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
```

```
#PBS -N Blobology
#PBS -l select=1:ncpus=24:mem=100gb
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
source ~/.bashrc


echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

echo "run blobology megahit assembly"
metawrap blobology -a ASSEMBLY/megahit_assembly/final_assembly.fasta -t 24 \
  -o BLOBOLOGY_megahit \
  --bins BIN_REFINEMENT_megahit/metawrap_70_10_bins CLEAN_READS/BBF*fastq

echo "run blobology on metaspade assembly"
metawrap blobology -a ASSEMBLY/metaspade_assembly/final_assembly.fasta -t 24 \
  -o BLOBOLOGY_metaspade \
  --bins BIN_REFINEMENT_metaspade/metawrap_70_10_bins CLEAN_READS/BBF*fastq

echo "run blobology megahit assembly"
metawrap blobology -a ASSEMBLY/megahit_assembly/final_assembly.fasta -t 24 \
  -o BLOBOLOGY_megahit_905 \
  --bins BIN_REFINEMENT_megahit_905/metawrap_90_5_bins CLEAN_READS/BBF*fastq

echo "run blobology on metaspade assembly"
metawrap blobology -a ASSEMBLY/metaspade_assembly/final_assembly.fasta -t 24 \
  -o BLOBOLOGY_metaspade_905 \
  --bins BIN_REFINEMENT_metaspade_905/metawrap_90_5_bins CLEAN_READS/BBF*fastq
```

Step 7: Calculate abundaces of draft bins across samples.

Explore distribution of genomes and abundances aross samples. This modules uses the tool *Salmon*. *Step 7* has also been run on both assemblies (*MegaHIT* and *MetaSPAde*) and with both bin refinement parameter combinations (*70_10* and *90_5*).

NB. Use non-reassembled bins from the bin refinement module for more accurate bin abundances.

(qsub)

```
#!/bin/bash

#PBS -j oe
```

```
#PBS -m ae
#PBS -N Calc_Abund
#PBS -l select=1:ncpus=24:mem=100gb
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
source ~/.bashrc

echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

echo "make output directory"
mkdir -p QUANT_BINS

echo "calculate abundances for megahit"
metawrap quant_bins -b BIN_REFINEMENT/BIN_REFINEMENT_megahit/metawrap_70_10_bins \
  -o QUANT_BINS/QUANT_BINS_megahit_70_10 \
  -a ASSEMBLY/megahit_assembly/final_assembly.fasta CLEAN_READS/BBF*fastq

echo "calculate abundances for megahit_905"
metawrap quant_bins -b BIN_REFINEMENT/BIN_REFINEMENT_megahit_905/metawrap_90_5_bins \
  -o QUANT_BINS/QUANT_BINS_megahit_90_5 \
  -a ASSEMBLY/megahit_assembly/final_assembly.fasta CLEAN_READS/BBF*fastq

echo "calculate abundances for metaspade"
metawrap quant_bins -b BIN_REFINEMENT/BIN_REFINEMENT_metaspade/metawrap_70_10_bins \
  -o QUANT_BINS/QUANT_BINS_metaspade_70_10 \
  -a ASSEMBLY/metaspade_assembly/final_assembly.fasta CLEAN_READS/BBF*fastq

echo "calculate abundances for metaspade_905"
metawrap quant_bins -b BIN_REFINEMENT/BIN_REFINEMENT_metaspade_905/metawrap_90_5_bins \
  -o QUANT_BINS/QUANT_BINS_metaspade_90_5 \
  -a ASSEMBLY/metaspade_assembly/final_assembly.fasta CLEAN_READS/BBF*fastq
```

Step 8: Re-assemble bins.

Improve in the consolidated bins from the bin refinement step. *Step 8* has also been run on both assemblies (*MegaHIT* and *MetaSPAde*) and with both bin refinement parameter combinations (*70_10* and *90_5*), but in seperate submissions due to resource limitations.

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N REASSEMBLY_megahit
#PBS -l select=1:ncpus=24:mem=250gb
#PBS -l walltime=48:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
source ~/.bashrc

echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

echo "make output directory"
mkdir -p BIN_REASSEMBLY

echo "calculate abundances for megahit 70 10"
metawrap reassemble_bins -o BIN_REASSEMBLY/BIN_REASSEMBLY_megahit_70_10 \
  -1 CLEAN_READS/ALL_READS_1.fastq -2 CLEAN_READS/ALL_READS_2.fastq \
  -t 24 -c 70 -x 10 -b BIN_REFINEMENT/BIN_REFINEMENT_megahit/metawrap_70_10_bins

echo "calculate abundances for megahit 90 5"
metawrap reassemble_bins -o BIN_REASSEMBLY/BIN_REASSEMBLY_megahit_90_5 \
  -1 CLEAN_READS/ALL_READS_1.fastq -2 CLEAN_READS/ALL_READS_2.fastq \
  -t 24 -c 90 -x 5 -b BIN_REFINEMENT/BIN_REFINEMENT_megahit_905/metawrap_90_5_bins
```

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N REASSEMBLY_metaspade
#PBS -l select=1:ncpus=24:mem=250gb
#PBS -l walltime=48:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
```

```
source ~/.bashrc

echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

echo "calculate abundances for metaspade 70 10"
metawrap reassemble_bins -o BIN_REASSEMBLY/BIN_REASSEMBLY_metaspade_70_10 \
  -1 CLEAN_READS/ALL_READS_1.fastq -2 CLEAN_READS/ALL_READS_2.fastq \
  -t 24 -c 70 -x 10 -b BIN_REFINEMENT/BIN_REFINEMENT_metaspade/metawrap_70_10_bins

echo "calculate abundances for metaspade 90 5"
metawrap reassemble_bins -o BIN_REASSEMBLY/BIN_REASSEMBLY_metaspade_90_5 \
  -1 CLEAN_READS/ALL_READS_1.fastq -2 CLEAN_READS/ALL_READS_2.fastq \
  -t 24 -c 90 -x 5 -b BIN_REFINEMENT/BIN_REFINEMENT_metaspade_905/metawrap_90_5_bins
```

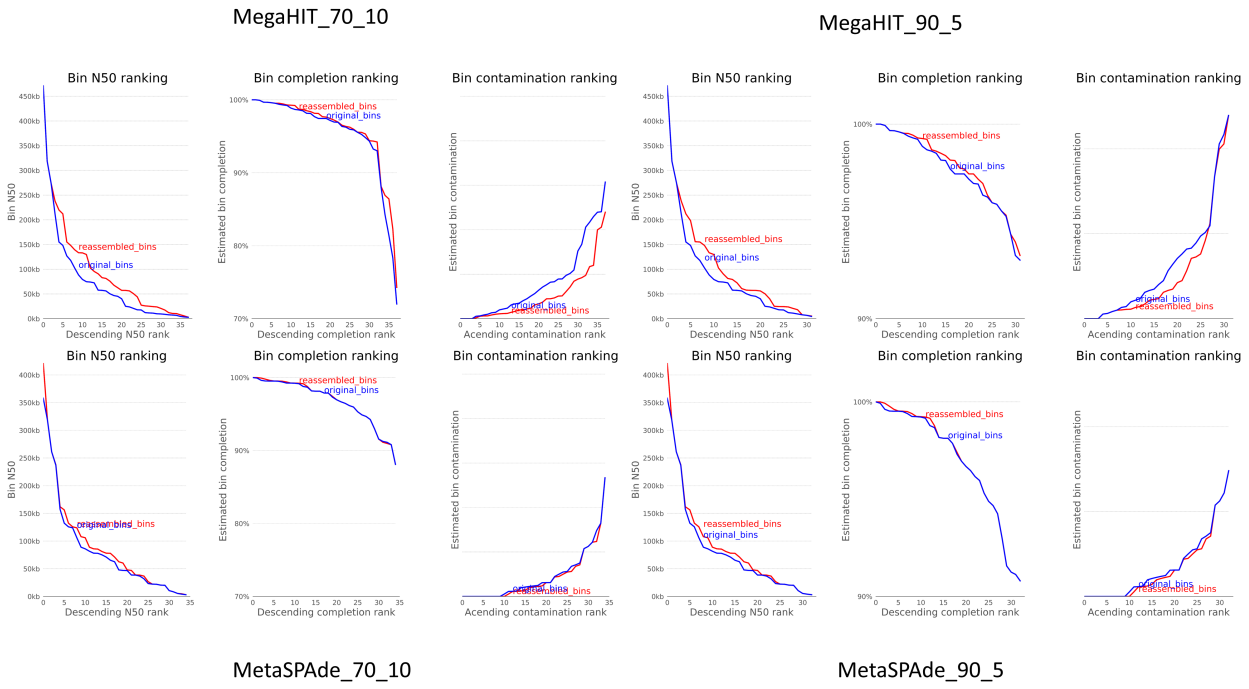

Figure 3: Reassembly Comparison.

Selecting assembler and refinement parameters.

Up until this point the workflow has been run with two assemblers, each with two paramter options. From here we will chose the best option to continue with.

MetaSPAde with 70 and 10 (for completion and contamination respectively) appears to be the best option. *MetaSpade* trumps *MegaHIT* as its quality is slightly better, and 70/10 gives us more bins without compromising much on quality, as across the N50, contamination and completion there doesn't seem to be much difference around bin 33 for both refinement parameters.

Step 9: Assign Taxonomy.

Although the bin refinement and reassembly modules provide approximate taxonomy, the `classify_bins` module uses *Taxator-tk* to conduct more accurate taxonomic assignment assignment. *Taxator-tk* assigns taxonomy to each contig and then consolidates the results to estimate the taxonomy of the entire bin.

NB. If you're using a version of **MetaWRAP** <=1.2.1 you will need to update the module file and a script.

Taxonomy Fix

Replace both *classify_bins.sh* and *prune_blast_hits.py*.

```
cd miniconda3/envs/metawrap-env/bin/metawrap-modules
rm classify_bins.sh
wget https://raw.githubusercontent.com/bxlab/metaWRAP/master/bin/metawrap-modules/classify_bins.sh
chmod 777 classify_bins.sh

cd miniconda3/envs/metawrap-env/bin/metawrap-scripts
rm prune_blast_hits.py
wget https://raw.githubusercontent.com/bxlab/metaWRAP/master/bin/metawrap-scripts/prune_blast_hits.py
chmod 777 prune_blast_hits.py
```

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N Tax_Clas
#PBS -l select=1:ncpus=24:mem=40gb
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
source ~/.bashrc

echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

echo "make output directory"
mkdir -p BIN_CLASSIFICATION

echo "Taxonomic classification"
metawrap classify_bins -b BIN_REASSEMBLY/BIN_REASSEMBLY_metaspade_70_10/reassembled_bins \
  -o BIN_CLASSIFICATION/BIN_CLASSIFICATION_metaspade_70_10 -t 24
```

Step 10: Functional annotation.

Functionally annotate bins with the `annotate_bins` module with *PROKKA*.

NB. If you're using a version of **MetaWRAP** <=1.2.1 you will need to update the module.

Annotation Fix

Replace *annot_bins.sh* module file and force install an older version of *openssl*.

```
cd miniconda3/envs/metawrap-env/bin/metawrap-modules
rm annotate_bins.sh
wget https://raw.githubusercontent.com/bxlab/metaWRAP/master/bin/metawrap-modules/annotate_bins.sh
chmod 777 annotate_bins.sh

conda install openssl=1.0
```

(qsub)

```
#!/bin/bash

#PBS -j oe
#PBS -m ae
#PBS -N Funct_Annot
#PBS -l select=1:ncpus=24:mem=40gb
#PBS -l walltime=24:00:00
#PBS -M jacob.westaway@my.jcu.edu.au

echo "----------------------------------------"
echo "PBS: Job identifier is $PBS_JOBID"
echo "PBS: Job name is $PBS_JOBNAME"
echo "----------------------------------------"

echo "activate bash"
source ~/.bashrc

echo "activate environment"
source activate ~/miniconda3/envs/metawrap-env

echo "change directory"
cd ~/Microba_metagenomics_analysis/JCU_raw_data_cat/fastq_combined

echo "Taxonomic classification"
metawrap annotate_bins -o FUNCT_ANNOT -t 24 \
  -b BIN_REASSEMBLY/BIN_REASSEMBLY_metaspade_70_10/reassembled_bins
```

YOU HAVE NOW COMPLETED THE METAGENOMICS WORKFLOW!