Binning Procedure applied to Tara Oceans Dataset using BinSanity (e.g., South Pacific)

Elaina Graham,Benjamin Tully

Abstract

This is the procedure used to cluster MAGs for the *Tara Oceans* dataset assembled by Benjamin Tully and Elaina Graham

Citation: Elaina Graham,Benjamin Tully Binning Procedure applied to Tara Oceans Dataset using BinSanity (e.g., South Pacific). **protocols.io**
https://www.protocols.io/view/binning-procedure-applied-to-tara-oceans-dataset-u-iwgcfbw
Published: 13 Jul 2017

Before start

Before starting be sure you have downloaded Binsanity. Please see the Binsanity github page for full installation instructions and a walkthrough of parameters for each program.

Protocol

Generate a coverage profile using Binsanity-profile

Step 1.

A profile should be generated for all contigs from each seperate metagenome.

An example of the Tara Oceans assembly procedure can be found **here**.

To do this first you will need to generate a list of contig ids that will be binned. This can be done using the **get-ids** command available through **BinSanity**

Following this run **Binsanity-Profile**:

SOFTWARE PACKAGE (N/A -)

BinSanity, 0.2.5.1

Elaina Graham
https://github.com/edgraham/BinSanity

COMMAND

```
Binsanity-profile -i tara_southpacific_SECONDARY_contigs.min14000.fasta -
s directory/to/BAM/files --ids tara_southpacific_SECONDARY_contigs.min14000.ids --
transform scale
```

Run BinSanity Pass1

Step 2.

Now that you have a coverage file you will want to create a directory called "Tara-MAGs" and move the coverage file into this directory.

Once this is done *cd* into the "Tara-MAGs directory and run the first round of Binsanity using a preference of -10 (All other parameters should remain at default).

COMMAND

```
Binsanity -f . -l tara_southpacific_SECONDARY_contigs.min14000.fasta -p -10 -
c tara_southpacific_SECONDARY_min14000.ALLSAMPLES.coverage.x100.lognorm -o PASS1
```

Rename Output Files

Step 3.

Following the commands below you will do the following:

1. First mv the log file into the PASS1 directory.

2. Rename the log file to reflect the clustering step

3. rename the files in PASS1 to reflect the clustering step

COMMAND

```
mv *.txt PASS1
cd PASS1
mv *.txt PASS1-log.txt
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS1-%u" $num).fna"
       let num=$num+1
done
```

Run CheckM

Step 4.

In this next step you will run CheckM and asses the output. The script **checkm_analysis** is a part of

BinSanity's suite of scripts.

The commands below do the following:

1. Run CheckM lineage_wf to assess PASS1 genomes for completion and redundancy estimates

2. Split Genomes into categories considered *high_completion, high_redundnacy,* and *low_completion.*

- High completion: >90% complete with <10% redundancy, greater than 80% with <5% redundancy, or >50% with <2% redundacy
- Low completion: <50% complete with <5%redundancy
- Strain heterogeneity: >90% complete with >90% strain heterogeneity
- High Redundancy: >80% complete with >10% redundacy, or >50% complete >5% redundacy

3. Prior to refinement you'll want to combine all bins in the low_completion category together - contigs from these bins will be used for the next stage of clustering.

4. Add the bins in the strain_heterogeneity directory to high_redundancy

5. Remove the low_completion and strain_heterogeneity directories.

cmd COMMAND

```
checkm lineage_wf -x fna -t 30 . PASS1-checkm> PASS1-checkm_out
checkm_analysis -checkM PASS1-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain_heterogeneity
rm -r low_completion
```

EXPECTED RESULTS

PASS1

Total bins = 2184

High completion = 190

High redundancy = 396

Run Binsanity Refinement on Pass1

Step 5.

Now you can run **Binsanity-refine.**

To do this we will first go into the *high_redundancy* directory.

Once in the *high_redundancy* directory you can run **Binsanity-refine** with a preference of -25 and default parameters.

cmd COMMAND

```
cd high_redundancy
for file in *.fna; do Binsanity-refine -c ../../*lognorm -f . -l "$file" -p -25 -
o ../../PASS1-refine;done
```

Rename output files for PASS1-Refine

Step 6.

Repeat step-3 replacing PASS1 with Pass1-refine

cmd COMMAND

```
mv *.txt ../../PASS1-refine
cd ../../PASS1-refine
mv *.txt PASS1-refine-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS1-refine-%u" $num).fna"
       let num=$num+1
done
```

Run CheckM on PASS1-Refine results

Step 7.

Now you need to run CheckM to check for redundancy, completion, and strain heterogeneity of each bin.

Once CheckM is completed the output file will be `PASS1-refine-checkm_out`

Using the `checkm_analysis` script you can separate bins based on the categories in Step 4 part 2.

The low completion bins will be re-combined and moved to the high_redundancy directory for further refinement.

All bins with high strain heterogeneity will also be moved to the high_redundancy directory.

Then to clean up the files you can remove the strain_heterogeneity and low_completion directories as you have now moved all bins to the high_redundancy directory.

cmd COMMAND

```
checkm lineage_wf -x fna -t 30 . PASS1-refine-checkm> PASS1-refine-checkm_out
checkm_analysis -checkM PASS1-refine-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain*
rm -r low_completion
```

EXPECTED RESULTS

PASS1-refine

Total bins = 1160
High completion = 51
High redundancy = 423

Run Additional Iterations of Binsanity

Step 8.

BinSanity will be run an additional 5 times, with Binsanity-refine being run between each step, except for a modification after iteration 6. Modifications to commands above will generate the correct files.

Preference values used for each PASS

PASS2 -p -5 PASS2-refine -p -25

PASS3 -p -3 PASS3-refine -p -25

PASS4 -p -3 PASS4-refine -p -25

PASS5 -p -3 PASS5-refine -p -25

PASS6 -p -3

EXPECTED RESULTS

PASS2
Total bins = 2362
High completion = 52
High redundancy = 397

PASS2-refine
Total bins = 1175
High completion = 5
High redundancy = 465

PASS3
Total bins = 3053
High completion = 52
High redundancy = 332

PASS3-refine

Total bins = 1183
High completion = 6
High redundancy = 459

PASS4
Total bins = 3012
High completion = 4
High redundancy = 340

PASS4-refine
Total bins = 1176
High completion = 1
High redundancy = 462

PASS5
Total bins = 3031
High completion = 1
High redundancy = 342

PASS5-refine
Total bins = 1175
High completion = 1
High redundancy = 466

Run CheckM on PASS6

Step 9.

cmd COMMAND

```
checkm lineage_wf -x fna -t 30 . PASS6-checkm > PASS6-checkm_out
checkm_analysis_final -checkM PASS6-checkm_out
```

EXPECTED RESULTS

PASS6
Total bins = 3022
High completion = 116
High redundancy = 225

Run Binsanity-refine on PASS6 high redundancy at preference -10

Step 10.

Following CheckM on Binsanity PASS6 is where we will change things up. You will no longer move or alter the low_completion files. Instead you will go directly into the high redundancy file for PASS6 and run **Binsanity-refine** at a preference of -10.

cmd COMMAND

```
cd high_redundancy
for file in *.fna; do Binsanity-refine -f . -l "$file" -p -10 -c ../../*lognorm -
o ../../PASS6-refine-redundant-pref10; done
```

Rename output files for PASS6 refinement of redundancy bins

Step 11.

You can now rename the output files for PASS6 refinement at a preference of -10 for downstream

refinement.

cmd COMMAND

```
mv *.txt ../../PASS6-refine-redundant-pref10
cd ../../PASS6-refine-redundant-pref10
mv *.txt PASS6-refine-redundant-pref10-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS6-refine-redundant-%u" $num).fna"
       let num=$num+1
done
```

Run CheckM on bins generated from PASS6 refinement

Step 12.

Here we will run CheckM on the bins generated using Binsanity-refine on PASS6 high_redundnacy bins at a preference of -10.

cmd COMMAND

```
checkm lineage_wf -x fna -t 30 . PASS6-refine-redundant-checkm > PASS6-refine-redundant-
checkm_out
checkm_analysis_final -checkM PASS6-refine-redundant-checkm_out
```

EXPECTED RESULTS

PASS6-refine-10

Total bins = 225
High completion = 22
High redundancy = 198

Run a second round of refinement on the remaining high redundancy bins at preference -3

Step 13.

Finally we will run another round of refinement on ONLY the high_redundancy bins for PASS6-refine-redundant-pref10. This time Binsanity-refine will be run with a preference of -3.

cmd COMMAND

```
cd high_redundancy
for file in *.fna; do Binsanity-refine -f . -l "$file" -p -3 -c ../../*lognorm -
o ../../PASS6-refine-2-redundant-pref3; done
```

Rename output files from PASS6 refinement round 2

Step 14.

cmd COMMAND

```
mv *.txt ../../PASS6-refine-2-redundant-pref3
cd ../../PASS6-refine-2-redundant-pref3
mv *.txt PASS6-refine-redundant-2-pref3-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS6-refine-redundant-2-%u" $num).fna"
       let num=$num+1
done
```

Run CheckM and evaluate high completion bins for PASS6 refinement round 2

Step 15.

cmd COMMAND

```
checkm lineage_wf -x fna -t 30 . PASS6-refine-redundant-2-checkm > PASS6-refine-
redundant-2-checkm_out
checkm_analysis -checkM PASS6-refine-redundant -checkM PASS6-refine-redundant-2-checkm_out
```

EXPECTED RESULTS

PASS6-refine-3

Total bins = 198
High completion = 36
High redundancy = 124

Make a final bin directory

Step 16.

Now we want to collect all our final bins in one place.

To do this we will organize the high_completion, low_completion, and remaining high_redundancy in one place. Follow the structure below.

Once this is done you can confirm that none of your bins were accidently duplicated by using the `checkm unique` function.

You have now generated your bins for analysis!

cmd COMMAND

```
mkdir FINAL-HIGH-COMPLETION FINAL-LOW-COMPLETION FINAL-HIGH-REDUNDANCY
cp PASS*/high_completion/*.fna FINAL-HIGH-COMPLETION
cp PASS6-refine-redundant-pref10/low_completion FINAL-LOW-COMPLETION
cp PASS6-refine-2-redundant-pref3/low_completion FINAL-LOW-COMPLETION
cp PASS6-refine2-redundancy-pref3/high_redundancy FINAL-HIGH-REDUNDANCY
```

EXPECTED RESULTS

FINAL RESULTS

High completion bins = 537

TOBG Draft genomes = 536

High redundancy = 124

Low completion = 3730

Automate whole process using shell script

Step 17.

The entire process desrcibed above has been automated in a shell script that can be run through all of these steps. Modification of the shell script can increase or decrease the number of BinSanity passes. Shell script assumes that all of the BinSanity scripts are available in your $PATH.

The Tara Oceans project utilized 6 passes of BinSanity, but fewer passes can yield similar outputs.

cmd COMMAND

```
#!/bin/bash
Binsanity -f . -l tara_southpacific_SECONDARY_contigs.min14000.fasta -p -10 -
c tara_southpacific_SECONDARY_min14000.ALLSAMPLES.coverage.x100.lognorm -o PASS1
mv *.txt PASS1
cd PASS1
mv *.txt PASS1-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS1-%u" $num).fna"
       let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS1-checkm> PASS1-checkm_out
checkm_analysis -checkM PASS1-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain*
rm -r low_completion
cd high_redundancy
for file in *.fna; do Binsanity-refine -c ../../*lognorm -f . -l "$file" -p -25 -
o ../../PASS1-refine;done
mv *.txt ../../PASS1-refine
cd ../../PASS1-refine
mv *.txt PASS1-refine-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS1-refine-%u" $num).fna"
       let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS1-refine-checkm> PASS1-refine-checkm_out
checkm_analysis -checkM PASS1-refine-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain*
rm -r low_completion
cd high_redundancy
for file in *.fna; do Binsanity -c ../../*lognorm -f . -l "$file" -p -5 -o ../../PASS2;done
####
mv *.txt ../../PASS2
cd ../../PASS2
mv *.txt PASS2-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS2-%u" $num).fna"
```

```
	let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS2-checkm> PASS2-checkm_out
checkm_analysis -checkM PASS2-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain*
rm -r low_completion
cd high_redundancy
for file in *.fna; do Binsanity-refine -c ../../*lognorm -f . -l "$file" -p -25 -
o ../../PASS2-refine;done
mv *.txt ../../PASS2-refine
cd ../../PASS2-refine
mv *.txt PASS2-refine-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
	mv "$file" "$(printf "PASS2-refine-%u" $num).fna"
	let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS2-refine-checkm> PASS2-refine-checkm_out
checkm_analysis -checkM PASS2-refine-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain*
rm -r low_completion
cd high_redundancy
for file in *.fna; do Binsanity -c ../../*lognorm -f . -l "$file" -p -3 -o ../../PASS3;done
###
mv *.txt ../../PASS3
cd ../../PASS3
mv *.txt PASS3-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
	mv "$file" "$(printf "PASS3-%u" $num).fna"
	let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS3-checkm> PASS3-checkm_out
checkm_analysis -checkM PASS3-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain*
rm -r low_completion
cd high_redundancy
for file in *.fna; do Binsanity-refine -c ../../*lognorm -f . -l "$file" -p -25 -
o ../../PASS3-refine;done
mv *.txt ../../PASS3-refine
cd ../../PASS3-refine
mv *.txt PASS3-refine-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
	mv "$file" "$(printf "PASS3-refine-%u" $num).fna"
	let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS3-refine-checkm> PASS3-refine-checkm_out
checkm_analysis -checkM PASS3-refine-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
```

```
rm -r strain*
rm -r low_completion
cd high_redundancy
for file in *.fna; do Binsanity -c ../../*lognorm -f . -l "$file" -p -3 -o ../../PASS4;done
####
mv *.txt ../../PASS4
cd ../../PASS4
mv *.txt PASS4-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS4-%u" $num).fna"
       let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS4-checkm> PASS4-checkm_out
checkm_analysis -checkM PASS4-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain*
rm -r low_completion
cd high_redundancy
for file in *.fna; do Binsanity-refine -c ../../*lognorm -f . -l "$file" -p -25 -
o ../../PASS4-refine;done
mv *.txt ../../PASS4-refine
cd ../../PASS4-refine
mv *.txt PASS4-refine-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS4-refine-%u" $num).fna"
       let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS4-refine-checkm> PASS4-refine-checkm_out
checkm_analysis -checkM PASS4-refine-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain*
rm -r low_completion
cd high_redundancy
for file in *.fna; do Binsanity -c ../../*lognorm -f . -l "$file" -p -3 -o ../../PASS5;done
###
mv *.txt ../../PASS5
cd ../../PASS5
mv *.txt PASS5-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS5-%u" $num).fna"
       let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS5-checkm> PASS5-checkm_out
checkm_analysis -checkM PASS5-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain*
rm -r low_completion
cd high_redundancy
for file in *.fna; do Binsanity-refine -c ../../*lognorm -f . -l "$file" -p -25 -
o ../../PASS5-refine;done
mv *.txt ../../PASS5-refine
```

```
cd ../../PASS5-refine
mv *.txt PASS5-refine-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS5-refine-%u" $num).fna"
       let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS5-refine-checkm> PASS5-refine-checkm_out
checkm_analysis -checkM PASS5-refine-checkm_out
for file in low_completion/*.fna; do cat "$file" >>high_redundancy/low_completion.fna; done
mv strain*/*.fna high_redundancy
rm -r strain*
rm -r low_completion
cd high_redundancy
for file in *.fna; do Binsanity -c ../../*lognorm -f . -l "$file" -p -3 -o ../../PASS6;done
mv *.txt ../../PASS6
cd ../../PASS6
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS6-%u" $num).fna"
       let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS6-checkm > PASS6-checkm_out
checkm_analysis_final -checkM PASS6-checkm_out
cd high_redundancy
for file in *.fna; do Binsanity-refine -f . -l "$file" -p -10 -c ../../*lognorm -
o ../../PASS6-refine-redundant-pref10; done
mv *.txt ../../PASS6-refine-redundant-pref10
cd ../../PASS6-refine-redundant-pref10
mv *.txt PASS6-refine-redundant-pref10-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS6-refine-redundant-%u" $num).fna"
       let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS6-refine-redundant-checkm > PASS6-refine-redundant-
checkm_out
checkm_analysis_final -checkM PASS6-refine-redundant-checkm_out
cd high_redundancy
for file in *.fna; do Binsanity-refine -f . -l "$file" -p -3 -c ../../*lognorm -
o ../../PASS6-refine-2-redundant-pref3; done
mv *.txt ../../PASS6-refine-2-redundant-pref3
cd ../../PASS6-refine-2-redundant-pref3
mv *.txt PASS6-refine-redundant-2-pref3-log.txt
find . -size 0 -delete
num=1
for file in *.fna; do
       mv "$file" "$(printf "PASS6-refine-redundant-2-%u" $num).fna"
       let num=$num+1
done
checkm lineage_wf -x fna -t 30 . PASS6-refine-redundant-2-checkm > PASS6-refine-
redundant-2-checkm_out
checkm_analysis_final -checkM PASS6-refine-redundant -checkM PASS6-refine-redundant-2-
checkm_out
cd ../
mkdir FINAL-HIGH-COMPLETION FINAL-LOW-COMPLETION FINAL-HIGH-REDUNDANCY
cp PASS*/high_completion/*.fna FINAL-HIGH-COMPLETION
```

```
cp PASS6-refine-redundant-pref10/low_completion FINAL-LOW-COMPLETION
cp PASS6-refine-2-redundant-pref3/low_completion FINAL-LOW-COMPLETION
cp PASS6-refine2-redundancy-pref3/high_redundancy FINAL-HIGH-REDUNDANCY
```