# Setting up a portable metagenome assembly pipeline for CBIO - Support #50

## Validate binning using single copy core genes

05/11/2018 11:37 AM - Katie Lennard

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | 05/11/2018 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | **Spent time:** | 0.00 hour |

**Description**

The quality of the binning results from CONCOCT can be examined by looking at single copy core genes (i.e. genes that expected to present across all taxa with only 1 copy - more indicates contamination)
Note that Ulas has his own scripts for validating single copy core genes (described below) but CONCOCT also has options for doing this, and CheckM is another option which might be worth comparing (described here https://concoct.readthedocs.io/en/latest/complete_example.html#validation-using-single-copy-core-genes)

**A) ULAS's pipeline** (NB: after he sent me all these scripts he said the only reason he used these custom scripts was because CheckM wasn't available yet at the time, and he then suggested I use checkM - not sure why he sent me the scripts in the first place). So ignore this pipeline (validation_by_single_copy_core_genes.sh on hex)

~~1. Find genes on the contigs (output from megahit) and functionally annotate these using Prodigal (output is a .faa file)

1. Use hmmer to search a HMM profile file (.hmm extension) against a sequence database (our .faa file from prodigal). The .hmm file can be built from an alignment file using hmmbuild
2. Next we use a series of .R scripts (from Ulas) to format the data and create a table of single copy core genes for each sample >index-fasta_KL.R (input: contigs.fa file + prodigal .faa file -> count number of ORFs/scaffold (he says scaffold but we're using the contigs file; output: orf2faa.RData, scaffolds2norfs.RData, scaffold2fa.RData, scaffold2stats.RData) >readBins_KL.R (input: binning result from concoct + scaffold2stats.RData -> Make scaffold/binning summary including IDs of scaffolds that weren't binned; output: scaffold2bin.RData) >writeScaffoldOrfIds_KL.R (input: scaffold2fa.RData, orf2faa.RData -> Just get the scaffold IDs and orf IDs; output: scaffoldids.RData, orfids.RData) >extractEssenSingleCopy_KL.R (input: scaffoldids.RData, orfids.RData, EssenSingleCopy_domain2gene.txt (provided by Ulas), hits table from hmmer) ~~

**B) checkM** (currently preferred option) - https://github.com/Ecogenomics/CheckM/wiki

1. For checkM I had to format the input so that we had one fasta file for each bin (the output from concoct only provided a mapping between contig IDs and bin IDs, not fasta files). The script I made for this is 'split_bins_fasta.sh' on hex
2. checkM requires prodigal (installed on hex), hmmer (installed on hex) and pplacer (requested install on hex)
3. Recommended checkM workflow: 'lineage_wf' - see https://github.com/Ecogenomics/CheckM/wiki/Workflows > Annotate contigs for each bin using prodigal > Align against HMM single copy genes (hmmer) > Place bins in reference taxonomy tree (pplacer) uses checkm 'tree' command (Note: pplacer apparently needs at least 32GB RAM and will crash if not enough memory - see https://github.com/Ecogenomics/CheckM/issues/41) > Assess phylogenetic markers found in each bin uses checkm 'tree_qa' command (can modify output format e.g. table, newick tree etc.; table summary of genome tree placement indicating the number of unique phylogenetically informative markers found, the number of markers found multiple times, and a taxon string indicating the placement of each bin within the genome tree) > Infer lineage-specific marker sets for each bin uses checkm 'lineage_set' command (uses output from 'tree' command) > Identify marker genes in bins uses the checkm 'analyze' command > Summarize bin qiuality using the checkm 'qa' command (produce different tables summarizing the quality of each genome bin)
4. Several useful functions are available as part of checkM e.g. to check bin uniqueness and do QC plots >Tested bin uniqueness (result written to standard job .o file: "No sequences assigned to multiple bins.")

### History

**#1 - 05/15/2018 10:57 AM - Katie Lennard**

*- Description updated*

**#2 - 05/15/2018 03:42 PM - Katie Lennard**

*- Description updated*

**#3 - 05/15/2018 03:43 PM - Katie Lennard**

*- Description updated*

#### #4 - 05/15/2018 03:43 PM - Katie Lennard

*- Description updated*

#### #5 - 05/15/2018 05:07 PM - Katie Lennard

*- Description updated*

#### #6 - 05/16/2018 02:41 PM - Katie Lennard

checkM successfully run on hex. Example outputs listed under the 'Files' tab (checkm.e1851691.txt and checkm.o1851691.txt)