EukCC2 Documentation

Paul Saary

Jul 26, 2022

Contents:

EukCC is a command line too written in python3 to estimate completeness and contamination of novel eukaryotic MAGs.

Note: Version 1 is deprecated

This is the documentation for version 2 of EukCC. This is not compatible with the now deprecated version 1. You can find the documentation for versions 0.2-version1 here: https://github.com/Finn-Lab/EukCC/tree/eukcc1/docs

CHAPTER 1

Please cite

Saary, Paul, Alex L. Mitchell, and Robert D. Finn. "Estimating the quality of eukaryotic genomes recovered from metagenomic analysis with EukCC." Genome biology 21.1 (2020): 1-21.

The project is hosted on GitHub: https://github.com/Finn-Lab/EukCC/

1.1 Quickstart

1.1.1 Setup

EukCC is avaliable to install frommany soruces. We reccomend using our Docker container:

```
docker pull  microbiomeinformatics/eukcc
```

Other options are:

Bioconda (https://anaconda.org/bioconda/eukcc)

```
conda instal -c conda-forge -c bioconda "eukcc>=2"

pip install eukcc
```

You can also fetch the source code from GitHub: https://github.com/Finn-Lab/EukCC/

Database setup

You will need to fetch the database for EukCC once.

Note: The database from version 1 is not compatible. So after upadting to EukCC make sure to upadte the database.

Fetching the database is as simple as:

```
mkdir eukccdb
cd eukccdb
wget http://ftp.ebi.ac.uk/pub/databases/metagenomics/eukcc/eukcc2_db_ver_1.1.tar.gz
tar -xzvf eukcc2_db_ver_1.1.tar.gz
```

If you want to forget about the database location you can add an ENV variable to your *.bash.rc*:

```
export EUKCC2_DB=/path/to/.../eukcc2_db_ver_1.1
```

1.1.2 Running EukCC

EukCC comes with two modes. You can run EukCC on a single bin or on a folder of bins.

Running EukCC on a single bin gives you the most options to tweak the parameters as you see fit. For most metagenomic workflows running EukCC on a folder of bins might be the most simple thing to do.

EukCC on a single MAG

We assume that you did set you `$EUKCC2_DB` to the correct location. If not please use the `--db` flag to pass the database to EukCC.

```
eukcc single --out outfolder --threads 8 bin.fa
```

EukCC will then run on 8 threads. You can pass nucleotide fastas or proteomes to EukCC. It will automatically try to detect if it has to predict proteins or not.

By default it will never use more than a single threads for placing the genomes in the reference tree, to save memory.

EukCC on a folder of bins

```
eukcc folder --out outfolder --threads 8 bins
```

EukCC will assume that the folder contains files with the suffix `.fa`. If that is not the case please adjust the parameter.

In folder mode EukCC will also try to refine bins automatically. To learn more about that please see

1.2 Bin merging

If EukCC is run in `folder` mode, it can try to merge two more more bins to create a refined/merged version of increased completeness.

For this you can and should pass paired read information to EukCC. So only bins linked by at least 100 (default) reads are considered for merging. This greatly improves speed and accuracy.

1.2.1 Preparing your linked reads

If you have paired-end read data you should create a sorted alignment. If you have multiple read files, you can create multiple BAM files.

For this you will need the contigs that were used to create this bins. Alternatively merge all bins into a pseudo-assembly file.

```
cat binfolder/*.fa > pseudo_contigs.fasta
bwa index pseudo_contigs.fasta
bwa mem -t 8 pseudo_contigs.fasta reads_1.fastq.gz reads_2.fastq.gz  |
    samtools view -q 20 -Sb - |
    samtools sort -@ 8 -O bam - -o alignment.bam
samtools index alignment.bam
```

You can then create a bin_linking table by using the EukCC provided script:

```
binlinks.py  --ANI 99 --within 1500 \
    --out linktable.csv binfolder alignment.bam
```

If you have multiple bam files, pass all of them to the script (e.g. **.bam*).

You will obtain a three column file (bin_1,bin_2,links).

1.2.2 Merging bins

You can then launch EukCC on the same binfolder like so:

```
eukcc folder \
    --out outfolder \
    --threads 8  \
    --links linktable.csv \
    binfolder
```

EukCC will fist run on all bins individually. It will then identify medium quality bins that are at least 50% compelete but not yet more than 100-`improve_percent`. It will then identify bins that are linked by at least 100 paired end reads to these medium quality bins. If after merging the quality score goes up this bin will be merged.

Merged bins can be found in the output folder.

Warning: Meging more than two bins. So setting `--n_combine` to anything above 1 is experimental and not yet recommended. We had very good results with merging two bins.

1.2.3 Example Dataset

I created example data to test this based on the Lichen Study ERP123954. Bins were created using CONCOCT but any binner with no prokaryotic bias works.

```
wget ...
gunzip eukcc_example_folder_GT57.zip

# Use at least a couple threads to speed it up
eukcc folder --threads 6 \
    --out output \
    --links eukcc_example_folder_GT57/links.csv \
    --n_combine 1 \
    eukcc_example_folder_GT57/bins
```