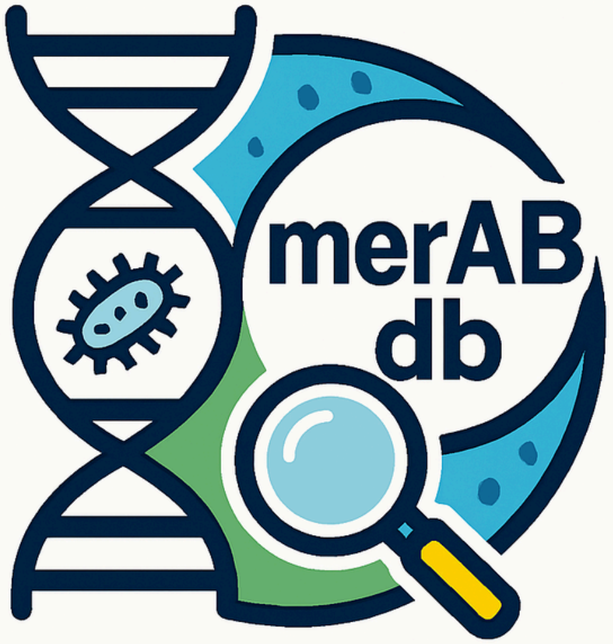


WORKSHOP

Nicola Gambardella

Ph.D student in Biology

ngambardella@ciimar.up.pt

CIIMAR

Interdisciplinary Centre
for Marine and
Environmental Research

University of Porto,
Porto, PT

MICRO-FROST:
Microbial Community Investigation of the
Nitrogen and Mercury Cycle in Permafrost
Environments

- bioinformatics & molecular biology
- co-organizer of the UMD workshop
- developer of markycoco2

ResearchGate

"la Caixa" Foundation

ciimar 25 years new horizons

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Marky Coco

A consensus protocol for
the recovery of mercury methylation genes from metagenomes

MOLECULAR ECOLOGY
RESOURCES

RESOURCE ARTICLE | Open Access

A consensus protocol for the recovery of mercury methylation genes from metagenomes

Eric Capo, Benjamin D. Peterson, Minjae Kim, Daniel S. Jones, Silvia G. Acinas, Marc Amyot, Stefan Bertilsson, Erik Björn, Moritz Buck, Claudia Cosio, Dwayne A. Elias ... See all authors

First published: 15 July 2022 | https://doi.org/10.1111/1755-0998.13687 | Citations: 18

Get it at Umea University Library

Andrea G. Bravo and Caitlin Gionfriddo joint last authors.
Handling Editor: Lucie Zinger

- ✓ hgcA and hgcB homologs
- ✓ merA and merB homologs

- paired end metagenomes
- single end metagenomes
- isolated, SAGs, and MAGs

Installation

INSTALL

```
git clone https://github.com/ericcapo/marky-coco.git
cd marky-coco
conda env create -f environment.yml
```

```
(base) nicola@PC-Nicola:/mnt/c/Users/nikga$ git clone https://github.com/ericcapo/marky-coco.git
Cloning into 'marky-coco'...
remote: Enumerating objects: 551, done.
remote: Counting objects: 100% (191/191), done.
remote: Compressing objects: 100% (124/124), done.
remote: Total 551 (delta 112), reused 119 (delta 67), pack-reused 360 (from 1)
Receiving objects: 100% (551/551), 4.43 MiB | 12.35 MiB/s, done.
Resolving deltas: 100% (320/320), done.
```

Installation

GitHub

```
ls marky-coco/ -l

:01 README.md
:01 db
:01 detect_hgc_from_fna.sh
:01 environment.yml
:01 marky-coco_step-by-step_tutorial_hgcA.html
:01 marky_pe.sh
:01 marky_pe_to_slurm.sh
:01 marky_se.sh
:01 marky_se_to_slurm.sh
:01 tutorial
:01 workflow
```

```
(base) nicola@PC-Nicola:/mnt/c/Users/nikga$ cat marky-coco/environment.yml
name: coco
channels:
- defaults
- conda-forge
- bioconda
dependencies:
- snakemake
- fastp ==0.20.0
- megahit ==1.1.2
- bowtie2 ==2.4.5
- prodigal ==2.6.3
- samtools ==1.9
- subread ==1.5.2
- seqtk
- biopython
- hmmer ==3.2.1
- pplacer ==1.1.alpha19
```

important for GFF recognition

Installation

INSTALL

```
git clone https://github.com/ericcapo/marky-coco.git
cd marky-coco
conda env create -f environment.yml
```

GitHub

conda create -n coco

conda activate coco

conda install -n coco -c bioconda snakemake

conda install -n coco -c bioconda prodigal==2.6.3

...

if creating the env does not work using the .yml file you can always do it “manually”

Installation

```
ls marky-coco/ -l

:01 README.md
:01 db
:01 detect_hgc_from_fna.sh
:01 environment.yml
:01 marky-coco_step-by-step_tutorial_hgcA.html
:01 marky_pe.sh
:01 marky_pe_to_slurm.sh
:01 marky_se.sh
:01 marky_se_to_slurm.sh
:01 tutorial
:01 workflow
```

folder containing the databases used by the pipeline

- Hg MATE (**hgcAB**)
- Christakis 2021 (**merAB**)
- NCBI (**rpoB**)

GitHub

Installation

```
ls marky-coco/ -l

:01 README.md
:01 db
:01 detect_hgc_from_fna.sh
:01 environment.yml
:01 marky-coco_step-by-step_tutorial_hgcA.html
:01 marky_pe.sh
:01 marky_pe_to_slurm.sh
:01 marky_se.sh
:01 marky_se_to_slurm.sh
:01 tutorial
:01 workflow
```

folder containing the databases used by the pipeline

- Hg MATE (**hgcAB**)
- Christakis 2021 (**merAB**)
- NCBI (**rpoB**)

GitHub

folder containing 2 files:

genesearch.sh BASH script

snakefile

snakemake
A framework for reproducible data analysis

SnakeMake is a workflow management system to create reproducible and scalable data analyses.

Installation

```
ls marky-coco/ -l

:01 README.md
:01 db
:01 detect_hgc_from_fna.sh
:01 environment.yml
:01 marky-coco_step-by-step_tutorial_hgcA.html
:01 marky_pe.sh
:01 marky_pe_to_slurm.sh
:01 marky_se.sh
:01 marky_se_to_slurm.sh
:01 tutorial
:01 workflow
```

folder containing the databases used by the pipeline

- Hg MATE (**hgcAB**)
- Christakis 2021 (**merAB**)
- NCBI (**rpoB**)

GitHub

All BASH scripts

What is BASH

BASH is a **shell**

a shell is a **text-based interface** that lets you talk to your computer.

Bourne Again SHell as it is an improved version of **sh** (bourne shell)

folder containing 2 files:

genesearch.sh BASH script

snakefile

snakemake
A framework for reproducible data analysis

SnakeMake is a workflow management system to create reproducible and scalable data analyses.

Installation

```
ls marky-coco/ -l

:01 README.md
:01 db
:01 detect_hgc_from_fna.sh
:01 environment.yml
:01 marky-coco_step_by_step_tutorial_hgcA.html
:01 marky_pe.sh
:01 marky_pe_to_slurm.sh
:01 marky_se.sh
:01 marky_se_to_slurm.sh
:01 tutorial
:01 workflow
```

marky_pe.sh is used for
paired end metagenomes
it calls the snakemake file
& *genesearch.sh*

GitHub

Installation

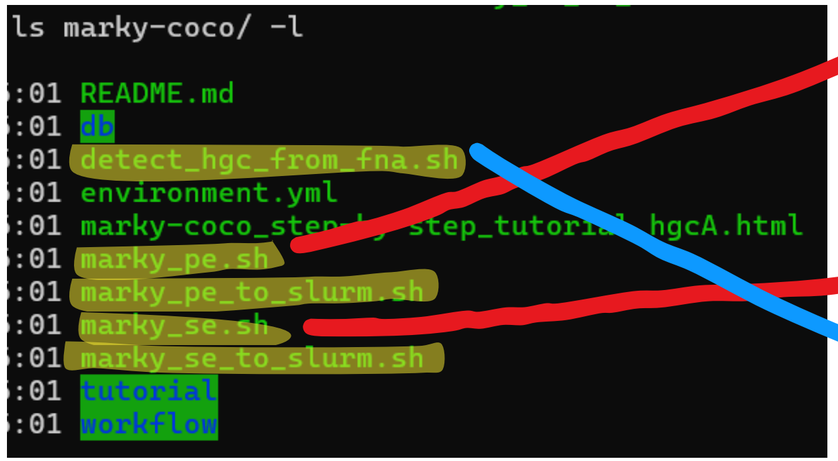
```
ls marky-coco/ -l

:01 README.md
:01 db
:01 detect_hgc_from_fna.sh
:01 environment.yml
:01 marky-coco_step_by_step_tutorial_hgcA.html
:01 marky_pe.sh
:01 marky_pe_to_slurm.sh
:01 marky_se.sh
:01 marky_se_to_slurm.sh
:01 tutorial
:01 workflow
```

marky_pe.sh is used for
paired end metagenomes
it calls the snakemake file
& *genesearch.sh*

marky_se.sh is used for **single end metagenomes**

GitHub

Installation



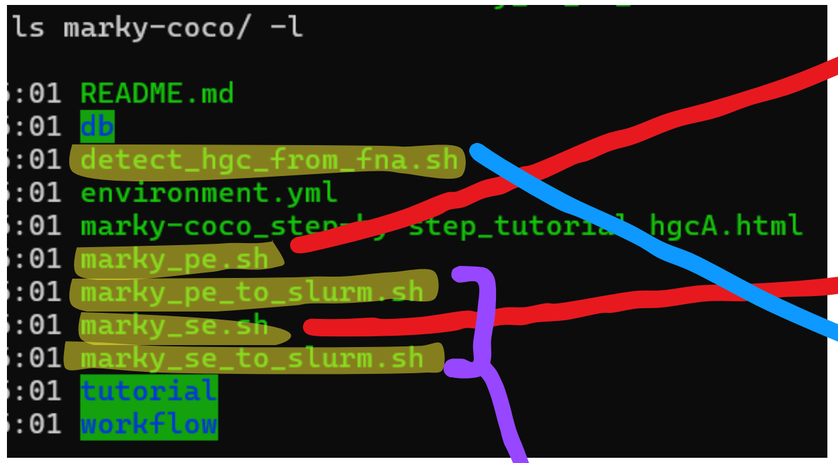
marky_pe.sh is used for
paired end metagenomes
it calls the snakemake file
& *genesearch.sh*

marky_se.sh is used for **single end metagenomes**

detect_hgc_from_fna.sh is used for **isolated**, **SAGs**, and **MAGs**

GitHub

Installation



marky_pe.sh is used for **paired end metagenomes** it calls the snakemake file & *genesearch.sh*

marky_se.sh is used for **single end metagenomes**

detect_hgc_from_fna.sh is used for **isolated**, **SAGs**, and **MAGs**

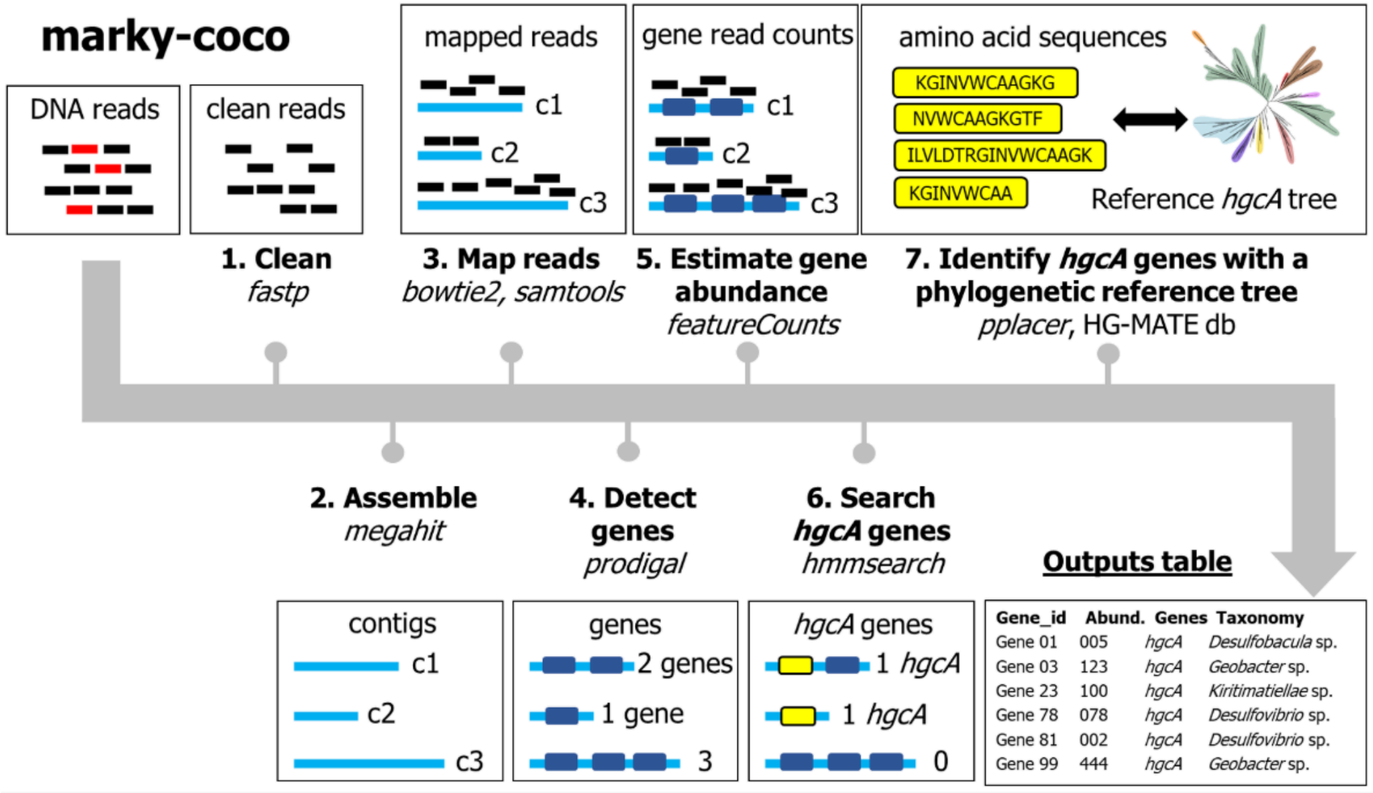
marky_pe_to_slurm.sh and marky_se_to_slurm.sh are both used with **Slurm** either for PE or SE metagenomes

slurm workload manager

job scheduler that is used to allocate, manage and monitor jobs on your cluster

GitHub

How does it work?



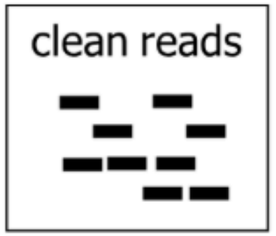
GitHub

marky-coco:

- allow you to obtain **hgcAB homologs**
- it **provides txid** to the sequences
- it calculate gene **coverage**

- allow you to obtain **merAB homologs**
- it calculate gene **coverage**

1 - Cleaning



1. Clean
fastp

fastp -i {input.r1} -I {input.r2} -o {output.out1} -O {output.out2} -h {output.out3} -j {output.out4} -q 30 -l 25 -w 6 --detect_adapter_for_pe --trim_poly_g --trim_poly_x

fastp is a software with multiple utilities for preprocessing and quality control of **short-reads** FastQ data

Shifu Chen, Yanqing Zhou, Yaru Chen, Jia Gu, fastp: an ultra-fast all-in-one FASTQ preprocessor, Bioinformatics, Volume 34, Issue 17, September 2018, Pages i884–i890, https://doi.org/10.1093/bioinformatics/bty560

-q the quality value that a base is qualified.
-l reads shorter than length_required will be discarded
-w worker thread number

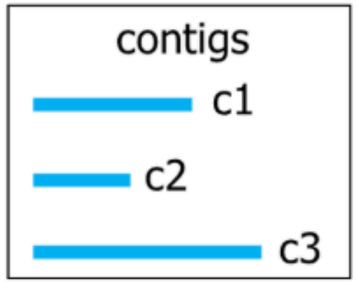
--detect_adapter_for_pe
--trim_poly_g force polyG tail trimming
--trim_poly_x enable polyX trimming in 3' ends.

2 - Assembly

GitHub

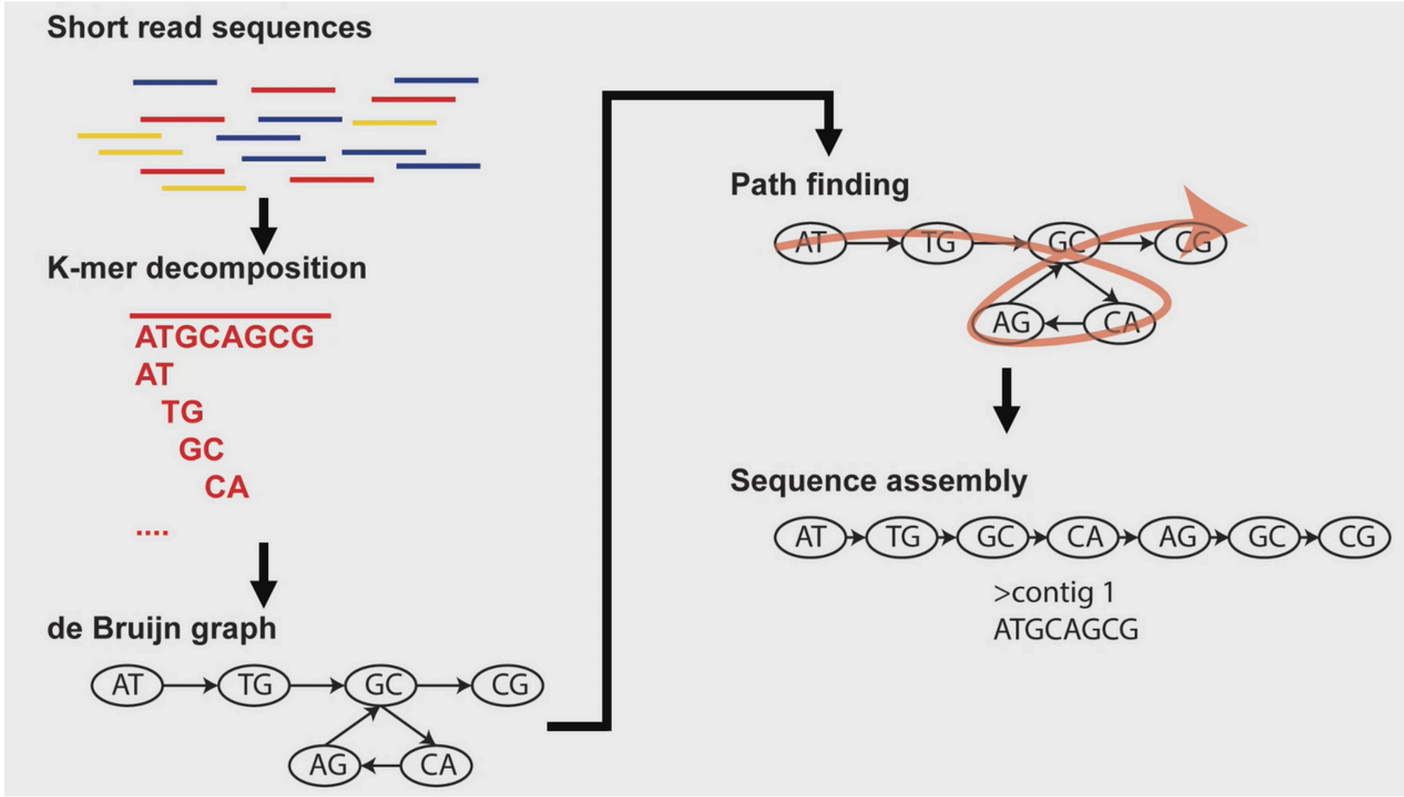
2. Assemble
megahit

megahit -1 {input.r1} -2 {input.r2} -o {output} -t {threads}



megahit is a software to perform assembly on NGS data that is optimized for metagenomes

Li D, Liu CM, Luo R, Sadakane K, Lam TW. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics. 2015 May 15;31(10):1674-6. doi: 10.1093/bioinformatics/btv033. Epub 2015 Jan 20. PMID: 25609793.



Dufault-Thompson, Keith, and Xiaofang Jiang. 2022. “ Applications of de Bruijn Graphs in Microbiome Research.” iMeta 1, e4. https://doi.org/10.1002/imt2.4

3 - Reads mapping

GitHub

3. Map reads
bowtie2, samtools

bowtie2-build {input} {params.basename}

bowtie2 -1 {input.r1} **-2** {input.r2} **-x** {wildcards.sample}_tmp/{wildcards.sample}.index
-p 6 | samtools view -Sb | samtools sort > {output}

bowtie2 is a software for aligning reads to reference sequences

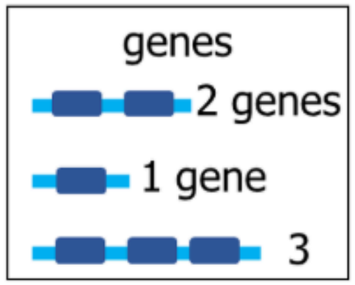
samtools is a suit of tools for interacting and manipulate SAM/BAM/CRAM files

SAM: Sequence Alignment Map
BAM: Binary Alignment Map
CRAM: Compressed Reference-oriented Alignment Map

4 - Gene Detection

GitHub

4. Detect genes
prodigal



prodigal -i {input} -o {output.gff} -f gff -a {output.faa}

prodigal is a software for protein-coding gene prediction in prokaryotic genomes

In this case it is used to output a GFF file

GFF: General Feature Format

Output example

feature

seqname source ...

```
k127_8973513 Prodigal_v2.6.3 CDS  174 401 19.4 +
k127_53841039 Prodigal_v2.6.3 CDS  206 556 50.4 +
k127_46861646 Prodigal_v2.6.3 CDS  1 606 103.0 -
k127_35894032 Prodigal_v2.6.3 CDS  2 715 135.1 -
k127_37888142 Prodigal_v2.6.3 CDS  2 253 51.1 +
k127_9970571 Prodigal_v2.6.3 CDS  1 1500 279.3 -
k127_997063 Prodigal_v2.6.3 CDS  3 329 36.7 -
k127_35894036 Prodigal_v2.6.3 CDS  3 254 39.3 +
k127_39882253 Prodigal_v2.6.3 CDS  1925 2635 54.1 -
k127_39882254 Prodigal_v2.6.3 CDS  3 557 39.3 -
```

Important for the next slide!

Here there should be a columns with several info

5 - Gene Abundance

5. Estimate gene abundance
featureCounts

featureCounts -t CDS -o {output} -g ID -a {input.gff} {input.bam}

The output from this software is used to calculate the gene coverage

This is done in the script *genesearch.sh* where a combination of **sed**, **cat** and **paste** is used to manipulate the output table

featureCounts is a software that counts mapped reads for genomic features

In this case the genomic feature is CDS

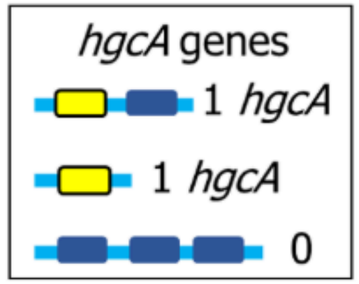
CDS: Coding DNA Sequence

sed transform text using REGEX

cat catenates files

paste perform horizontal merge of tabular files

6 - Gene Search



hmmsearch -o {hgcA_output} **--tblout** {hgcA_hmmer.out}
{Hg-MATE-Db.v1/**HgcA**.hmm}
{sample_prodigal_proteins_output.faa}

hmmsearch -o {hgcB_output} **--tblout** {hgcB_hmmer.out}
{Hg-MATE-Db.v1/**HgcB**.hmm}
{sample_prodigal_proteins_output.faa}

hmmsearch -o {merA_output} **--tblout** {merA_hmmer.out}
{merAB_Christakis2021/**merA**.hmm}
{sample_prodigal_proteins_output.faa}

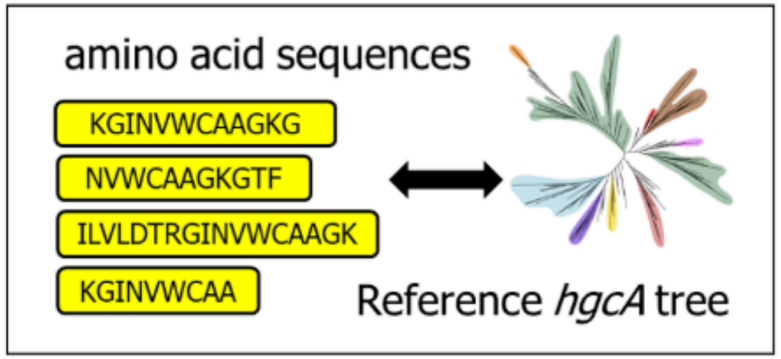
hmmsearch -o {merB_output} **--tblout** {merB_hmmer.out}
{merAB_Christakis2021/**merB**.hmm}
{sample_prodigal_proteins_output.faa}

HMMER is a software to search sequence databases for sequence homologs and for making sequence alignments.

It uses **HMM:** Hidden Markov Models

GitHub

7 - hgcA identification



7. Identify *hgcA* genes with a phylogenetic reference tree
pplacer, HG-MATE db

1 **hmmalign**

Align filtered hgcA sequences to the alignment of reference sequences in the reference package using hmm model producing an alignment of filtered sequences and reference sequences for the next step

2 **pplacer**

Place aligned query sequences onto the HgcA reference tree in the reference package

3 **rppr**

Make a sqlite-formatted database to be used in the next step for classification with guppy

4 **guppy**

Assign taxonomy to query sequences based on the placement on the phylogenetic tree

pplacer, **guppy**, and **rppr** belongs to a suit of programs that reads on a phylogenetic tree, analyzes them, works with reference packages.

GitHub

what about hgcB, merA, and merB ?

currently no taxonomy information is provided for *hgc*B, *mer*A, or *mer*B.

for these gene you have:

- gene id
- gene length
- counts
- coverage
- sequence

GitHub

How does the output look like?

```
PE23_22_bowtie2.log
PE23_22_fastp.html
PE23_22_fastp.json
PE23_22_hgcA_final.txt
PE23_22_hgcA_hmmer.out
PE23_22_hgcA_tree.nwk
PE23_22_hgcB_final.txt
PE23_22_hgcB_hmmer.out
PE23_22_merA_hmmer.out
PE23_22_merA_homologs.txt
PE23_22_merB_hmmer.out
PE23_22_merB_homologs.txt
PE23_22_rpoBa_final.txt
PE23_22_rpoBb_final.txt
```

GitHub

How does the output look like?

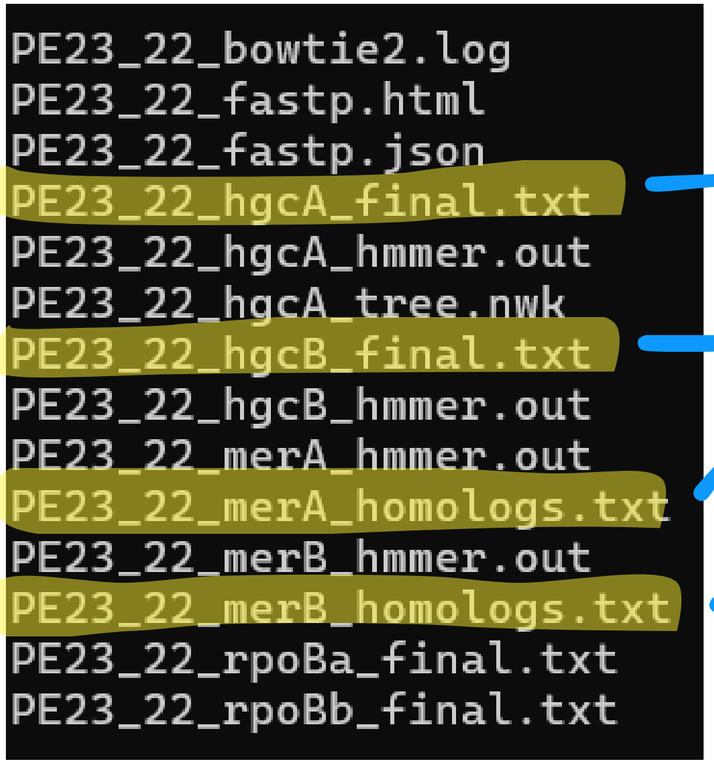
```
PE23_22_bowtie2.log
PE23_22_fastp.html
PE23_22_fastp.json
PE23_22_hgcA_final.txt
PE23_22_hgcA_hmmer.out
PE23_22_hgcA_tree.nwk
PE23_22_hgcB_final.txt
PE23_22_hgcB_hmmer.out
PE23_22_merA_hmmer.out
PE23_22_merA_homologs.txt
PE23_22_merB_hmmer.out
PE23_22_merB_homologs.txt
PE23_22_rpoBa_final.txt
PE23_22_rpoBb_final.txt
```

4 tabular files for 4 genes.

id | length | count | coverage | **txid** | sequence | type

GitHub

How does the output look like?



4 tabular files for 4 genes.

id | length | count | coverage | **txid** | sequence | type

id | length | count | coverage | sequence | type

GitHub

How do I run the program?

BASIC USAGE WITH PAIRED END METAGENOMES

- Copy your fastq files (sample_1.fastq and sample_2.fastq) in the folder marky-coco.

```
cp /remote/folder/sample_1.fastq .
cp /remote/folder/sample_2.fastq .
```

- Activate the conda environment

```
conda activate coco
```

- Run the marky script

```
bash marky_pe.sh sample
```

How do I run the program?

BASIC USAGE WITH SINGLE END METAGENOMES

- Copy your fastq file (sample.fastq) in the folder marky-coco.

```
cp /remote/folder/sample.fastq .
```

- Activate the conda environment

```
conda activate coco
```

- Run the marky script

```
bash marky_se.sh sample
```

How do I run the program? MAGs

RECOVER HGC FROM GENOMES (ISOLATED, SAGS AND MAGS)

"**NEW SINCE 30 JUNE 2023**To detect the presence of hgc genes in your genomes, you only need the script detect_hgc_from_fna.sh, the db folder of marky-coco and a folder with all your genomes in fna format.

- Copy your fastq file (sample.fastq) in the folder marky-coco.

```
cp -r /remote/folder .
```

- Activate the conda environment

```
conda activate coco
```

- Run the script

```
bash detect_hgc_from_fna.sh folder
```

MAGs Output

```
(base) nicolagambardella@connect2oceans-fe:~$ ls GREENLAND/13_markycoco/
outputs_hgcA  outputs_hgcB
```

```
bin.150.fa.3848129  bin.220.fa.3848129
bin.151.fa.3848129  bin.221.fa.3848129
bin.152.fa.3848129  bin.222.fa.3848129
bin.153.fa.3848129  bin.223.fa.3848129
bin.154.fa.3848129  bin.224.fa.3848129
bin.155.fa.3848129  bin.225.fa.3848129
bin.156.fa.3848129  bin.226.fa.3848129
bin.157.fa.3848129  bin.227.fa.3848129
bin.158.fa.3848129  bin.228.fa.3848129
bin.159.fa.3848129  bin.229.fa.3848129
```

contig | sequence | type | bin

Verification & Data Interpretation

The output of marky-coco need to be verified:

- **True hgcA genes** are those with amino acids motifs:
  NVWCAAGK, NVWCASGK, NVWCAGGK, NIWCAAGK, NIWCAGGK or NVWCSAGK
- **True hgcB genes** are those with the amino acids motifs:
  CMECGA or CIECGA **&** colocated with true hgcA genes.

- **True merA and merB gene**: **the reason why we are here**

The coverage calculated need to be normalized:

- To normalize hgc coverage values, sum the coverage values obtained from bacterial and archaeal rpoB genes.

The NCBI txid need to be assigned to the corresponding taxonomy:

- R script provided in GitHub
- manual assignment with db/db_txid_2202220
- check the identity in https://www.ncbi.nlm.nih.gov/taxonomy/

Hands On

Marky Coco

marky coco 2

Comprehensive Tool for Mercury Genes Detection

Marky Coco 2

- ....
- ....
- Automatic validation of hgcA and hgcB
- Automatic validation of merA and merB
- Generation of merA and merB for MAGs, SAGs, and isolates
- Taxonomic assignment of the genes
- Automatic gene coverage normalization with different normalization techniques
- ....
- ....
- ....

Marky Coco 2 → Marky Coco 2 alpha

- ....
- ....
- **Automatic validation of hgcA and hgcB**
- **Automatic validation of merA and merB**
- **Generation of merA and merB for MAGs, SAGs, and isolates**
- Taxonomic assignment of the genes
- Automatic gene coverage normalization with different normalization techniques
- ....
- ....
- ....

Marky Coco 2 alpha

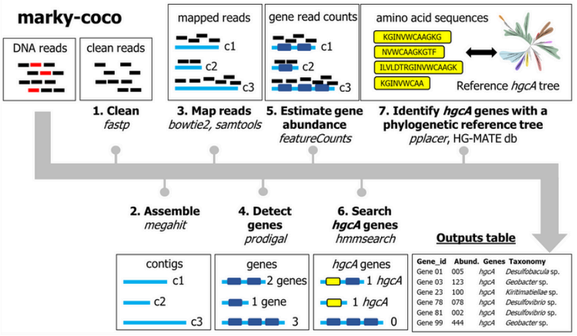
PAIRED END

SINGLE END

GENOMES

gene_search.py

Marky Coco



Conversion to Python

Python will make it more:
maintainable, readable, and robust.
It would eliminate:
temporary files
long chains of awk/sed/grep.
repetition of large blocks of code

hgcA VALIDATION

seq_validate_hgcA.py

hgcB VALIDATION

seq_validate_hgcB.py

Gene Coverage NORMALIZATION

seq_normalize.py

merA VALIDATION

seq_generate_merA.py seq_validate_merA.py

merB VALIDATION

seq_generate_merB.py seq_validate_merB.py

Marky Coco 2
alpha

hgcA and hgcB validation

hgcA VALIDATION

seq_validate_hgcA.py

Search for the identified hgcA motifs in marky-coco output producing a tabular output with the same structure of the marky-coco output

gene_id	length	read	cov	txid	sequence	gene_type

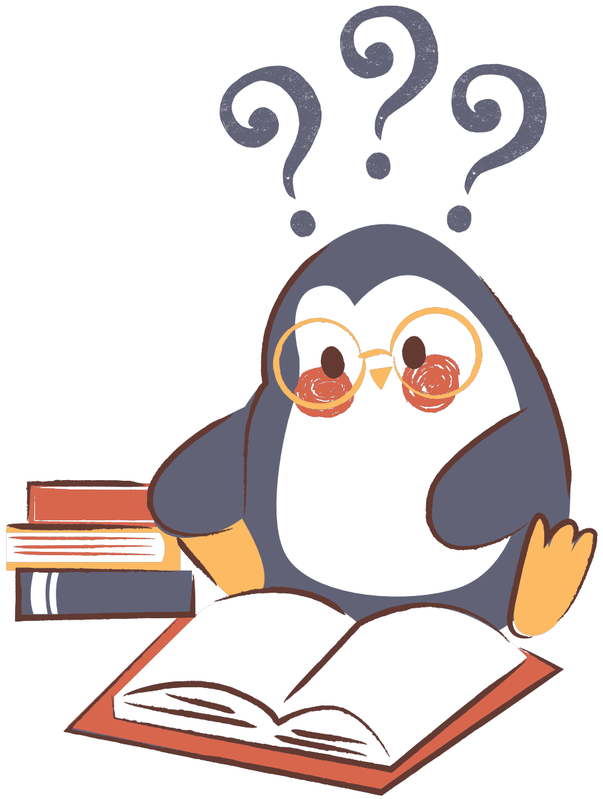
hgcB VALIDATION

seq_validate_hgcB.py

Search for the identified hgcB motifs in marky-coco output and check that those sequences are co-located with the hgcA sequences.

Hands On

Questions



Thank You
For Your Attention